Jiannong Cao
Ivan Stojmenovic
Xiaohua Jia
Sajal K. Das (Eds.)

LNCS 4325

Mobile Ad-hoc and Sensor Networks

Second International Conference, MSN 2006
Hong Kong, China, December 2006
Proceedings

Springer

Lecture Notes in Computer Science 4325

Jiannong Cao   Ivan Stojmenovic
Xiaohua Jia   Sajal K. Das (Eds.)

Mobile Ad-hoc and Sensor Networks

Second International Conference, MSN 2006
Hong Kong, China, December 13-15, 2006
Proceedings

Springer

Volume Editors

Jiannong Cao
Hong Kong Polytechnic University
Department of Computing
Hung Hom, Kowloon, Hong Kong SAR
E-mail: csjcao@comp.polyu.edu.hk

Ivan Stojmenovic
University of Ottawa, SITE
800 King Edward, Ottawa, Ontario K1N 6N5, Canada
E-mail: ivan@site.uottawa.ca

Xiaohua Jia
City University of Hong Kong
Department of Computer Science
Tat Chee Avenue, Kowloon Tong, Hong Kong SAR
E-mail: csjia@cityu.edu.hk

Sajal K. Das
University of Texas at Arlington
Dept of Computer Science and Engineering
Arlington, TX 76019, USA
E-mail: das@cse.uta.edu

Library of Congress Control Number: 2006937265

CR Subject Classification (1998): E.3, C.2, F.2, H.4, D.4.6, K.6.5

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

ISSN 0302-9743
ISBN-10 3-540-49932-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-49932-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11943952 06/3142 5 4 3 2 1 0

Preface

The principal theme of MSN conferences is the development and deployment of protocols, algorithms, systems, and applications for mobile ad-hoc and wireless sensor networks. Following the success of MSN 2005 held during December 13–15 2005 in Wuhan, China, MSN 2006 provided a forum for researchers and practitioners working in related areas to exchange research results and share development experiences.

MSN 2006 attracted 254 submissions. Each paper was reviewed by at least two members of the Program Committee. The final program included 73 papers, which covered a range of different topics, including routing, network protocols, security, etc.

The Organizing Committee would like to thank the Advisory Committee members Keith K.C. Chan, Marco Conti, Mario Gerla, Lionel Ni, and Jie Wu for their support and guidance in the conference organization. We would like to take this opportunity to thank all the authors for their submissions to the conference. Many of them traveled great distances to participate in this symposium and make their valuable contributions. Thanks to all the Program Committee members for their valuable time and effort in reviewing the papers. Without their help and advice this program would not be possible. Special thanks go to the conference PC Vice-Chairs, Eric Fleury, Zhen Jiang, Soung-Chang Liew, Yunhao Liu, Vojislav B. Misic, Amiya Nayak, Pedro M. Ruiz, Paolo Santi, and Jingyuan(Alex) Zhang for their hard work in assembling the international PC and coordinating the review process.

We appreciate the support from the invited speakers, Mario Gerla, Jie Wu, Taieb Znati. Their keynote speeches greatly benefited the audience. Last but not the least, we would like to thank the Local Organization Committee Chair, Allan Wong, and all members for making the arrangements and organizing an attractive social program.

December 2006

Jiannong Cao
Ivan Stojmenovic
Xiaohua Jia
Sajal K. Das

Organization

MSN 2006 was organized mainly by the Department of Computing, Hong Kong Polytechnic University, China.

Executive Committee

Advisory Committee:	Keith K.C. Chan (Hong Kong Polytechnic University, Hong Kong)
	Marco Conti (IIT, CNR, Italy)
	Mario Gerla (UCLA, USA)
	Lionel Ni (Hong Kong University of Sci. and Tech., Hong Kong)
	Jie Wu (Florida Atlantic University, USA)
General Co-chairs:	Sajal K. Das (University of Texas at Arlington, USA)
	Xiaohua Jia (Hong Kong City University, Hong Kong)
Program Co-chairs:	Jiannong Cao (Hong Kong Polytechnic University, Hong Kong)
	Ivan Stojmenovic (University of Ottawa, Canada)
Program Vice Chairs:	Eric Fleury (CITI/ARES, France)
	Zhen Jiang (West Chester University, USA)
	Soung-Chang Liew (Chinese University of Hong Kong, Hong Kong)
	Yunhao Liu (Hong Kong University of Sci. and Tech., Hong Kong)
	Vojislav B. Misic (University of Manitoba, Canada)
	Amiya Nayak (University of Ottawa, Canada)
	Pedro M. Ruiz (University of Murcia, Spain)
	Paolo Santi (IIT, CNR, Italy)
	Jingyuan Zhang (University of Alabama, USA)
Publicity Chairs:	Wei Lou (Hong Kong Polytechnic University, Hong Kong)
	Isabelle Simplot-Ryl (University of Lille 1, France)
	Yu Wang (University of North Carolina at Charlotte, USA)
Local Chair:	Allan Wong (Hong Kong Polytechnic University, Hong Kong)
Publication Chair:	Jianliang Xu (Hong Kong Baptist University, Hong Kong)

Awards Co-chairs:	Stephen Olariu (Old Dominion University, USA) Symeon Papavassiliou (National Technical University of Athens, Greece) Makoto Takizawa (Tokyo Denki University, Japan)
Web Master:	Hui Cheng (Hong Kong Polytechnic University, Hong Kong)

Program Committee

Thomas Kunz	Carleton University, Canada
Miodrag Bolic	University of Ottawa, Canada
Jean Carle	University of Lille, France
Johnson Kuruvila	Extreme Software, Canada
Daniele Miorandi	Create-Net, Italy
Srdjan Krco	Ericsson(BH/LMI), Ireland
Petar Popovski	Aalborg University, Denmark
Costas Constantinou	University of Birmingham, UK
Dong Zhou	Lucent Technologies, USA
Young-Bae Ko	Ajou University, South Korea
Song Guo	University of British Columbia, Canada
Imad Jawhar	United Arab Emirates University, UAE
Thiemo Voigt	SICS, Sweden
Khirsagar Naik	University of Waterloo, Canada
Yang Xiao	University of Memphis, USA
Suprakash Datta	York University, Canada
Chonggang Wang	University of Arkansas, USA
Hannes Frey	University of Trier, Germany
Ling-Jyh Chen	Academia Sinica, Taiwan
Arjan Durresi	Louisiana State University, USA
Wenyue Wang	North Carolina State University, USA
Kui Wu	University of Victoria, Canada
Vincent Wong	University of British Columbia, Canada
Ibrahim Korpeoglu	Bilkent University, Turkey
Xi Zhang	Texas AM University, USA
Jelena Misic	University of Manitoba, Canada
Zhanping Yin	University of British Columbia, Canada
Sachin Deshpande	Sharp Laboratories of America, USA
Tracy Camp	Colorado School of Mines, USA
Thomas Clausen	CNRS, Ecole Polytechnique, France
Mischa Dohler	France Telecom, France
Vasilis Fridericos	King's College London, UK
JJ Garcia-Luna-Aceves	University of California Santa Cruz, USA
Silvia Giordano	University of Applied Science, SUPSI, Switzerland
Antonio F. Gomez-Skarmeta	University of Murcia, Spain

Program Committee (continued)

Miguel A. Labrador	University of South Florida, USA
Pietro Manzoni	University Politecnica Valencia, Spain
Charles E. Perkins	Nokia Research Center, USA
David Simplot-Ryl	University of Lille, France
Shubhranshu Singh	Samsung Adv. Inst. of Technology, Korea
Giovanni Resta	IIT- CNR, Italy
Ossama Younis	University of Arizona, USA
Weifa Liang	Australian National University, Australia
Xiang-Yang Li	Illinois Institute of Technology, USA
Olivier Dousse	Deutsche Telekom Labs, Switzerland
Honghai Zhang	Lucent Technologies Labs, USA
Peng Jun Wan	City University of Hong Kong, China
Chih-Wei Yi	National Chiao Tung University, Taiwan
Santosh Kumar	University of Memphis, USA
Xiao Chen	Texas State University, USA
Laura Galluccio	University of Catania, Italy
Winston Seah	Inst. of Infocomm Research, Singapore
Yuanzhu Peter Chen	Memorial University, Canada
Roger Wattenhofer	ETH, Switzerland
Krish Chakrabarty	Duke University, USA
Yu Wang	University of North Carolina Charlotte, USA
Fei Dai	North Dakota State University, USA
Fabian Kuhn	ETH, Switzerland
Mauro Leoncini	University of Modena and Reggio Emilia, Italy
Wei Chen	Hong Kong University of Sci. and Tech., Hong Kong
Gary Chan	Hong Kong University of Sci. and Tech., Hong Kong
Zhang Qian	Hong Kong University of Sci. and Tech., Hong Kong
Qixiang Pang	University of British Columbia, Canada
Lawrence Yeung	The University of Hong Kong, Hong Kong
DahMing Chiu	City University of Hong Kong, Hong Kong
Brahim Bensaou	Hong Kong University of Sci. and Tech., Hong Kong
Michael Lyu	City University of Hong Kong, Hong Kong
Angela Zhang	City University of Hong Kong, Hong Kong
Chin-Tau Lea	Hong Kong University of Sci. and Tech., Hong Kong
Vincent Lau	Hong Kong University of Sci. and Tech., Hong Kong
Mounir Hamdi	Hong Kong University of Sci. and Tech., Hong Kong

Program Committee (continued)

Fan Pingyi	Tsinghua University, China
Ping Chung Ng	University of Oxford, UK
Libin Jiang	UC Berkeley, USA
Joseph Ng	Hong Kong Baptist University, Hong Kong
Weijia Jia	City University of Hong Kong, Hong Kong
Andrea passarella	CNR, Italy
Anne Fladenmuller	Universite Pierre et Marie Curie, France
Antoine Fraboulet	CITI, France
Artur Ziviani	LNCC, Brazil
Ayalvadi Ganesh	Microsoft, UK
Bartek Blaszczyszyn	ENS, France
Dominique Bartel	France Telecom RD, France
Mischa Dohler	France Telecom RD, France
Gaogang Xie	Institute of Computing Technology, China
Guillaume Chelius	INRIA, France
Hongyi Wu	University of Louisiana at Lafayette, USA
Jiming Chen	Zhejiang University, China
Kui Wu	University of Victoria, Canada
Marin Bertier	IRISA, France
Pietro Michiardi	EURECOM, France
Pilu Crescenzi	University of Florence, Italy
Prudence W.H. Wong	University of Liverpool, UK
Qingfeng Huang	Palo Alto Research Center (PARC) Inc, USA
Suprakash Datta	York University, Canada
Thomas Moscibroda	ETHZ, Switzerland
Vania Conan	THALES, France
Yu Chen	Texas AM University, USA
Stefan Weber	Trinity College Dublin, Ireland
Raffaele Bruno	CNR, Italy
Sotiris Nikoletseas	CTI/University of Patras, Greece
Nael Abu-Ghazaleh	SUNY Binghamton, USA
Saad Biaz	Auburn University, USA
Phillip Bradford	University of Alabama, USA
Mieso K. Denko	University of Guelph, Canada
Guangbin Fan	Intel Research, China
Li Gao	University of Alabama, USA
Xiaoyan Hong	University of Alabama, USA
Anup Kumar	University of Louisville, USA
Keqin Li	SUNY New Paltz, USA
Nidal Nasser	University of Guelph, Canada

Program Committee (continued)

Mohamed Ould-Khaoua	University of Glasgow, UK
Yi Qian	University of Puerto Rico, USA
Huai-Rong Shao	Samsung, USA
Randy Smith	University of Alabama, USA
Yu-Chee Tseng	National Chiao Tung University, Taiwan
Dajin Wang	Montclair State University, USA
Zhijun Wang	Millikin University, USA
Hongyi Wu	University of Louisiana at Lafayette, USA
Chen Zhang	Bryant University, USA
Qing-An Zeng	University of Cincinnati, USA
Jacir L. Bordim	University of Brasilia, Brazil
Koji Nakano	Hiroshima University, Japan
Susan Vrbsky	University of Alabama, USA
Abiola Abimbola	Napier University, UK
Alessandro Acquisti	Carnegie Mellon University, USA
Luciano Burgazzi	ENEA, Italy
Jordi Castella-Roca	Universitat Rovira I Virgili, Spain
Ionut Cardei	Florida Altantic University, USA
Li-pin Chang	National Chiao-Tung University, Taiwan
Xiaowen Chu	Hong Kong Baptist University, Hong Kong
Robert Deng	Singapore Management University, Singapore
Zhenhai Duan	Florida State University, USA
Roberto Di-Pietro	University of Rome “La Sapienza,” Italy
Jordi Forne	Universitat Politecnica de Catalunya, Spain
Yunghsiang Sam Han	National Taipei University, Taiwan
Jaap-Henk Hoepman	Radboud University Nijmegen, Netherlands
Polly Huang	National Taiwan University, Taiwan
Chokchai(Box) Leangsuksun	Louisiana Tech University, USA
Minglu Li	Shanghai Jiaotong University, China
Chae Hoon Lim	Sejong University, Korea
Phone Lin	National Taiwan University, Taiwan
Alex Zhaoyu Liu	University of North Carolina at Charlotte, USA
Jianhua Ma	Hosei University, Japan
Geyong Min	University of Bradford, UK
Yi Mu	University of Wollongong, Australia
Jesper Buus Nielsen	Aarhus University, Denmark
Ai-Chun Pang	National Taiwan University, Taiwan
Christian Rohner	Uppsala Universitet, Sweden
Emilia Rosti	Università degli Studi di Milano, Italy
Jang-ping Sheu	National Central University, Taiwan
Willy Susilo	University of Wollongong, Australia

Program Committee (continued)

Weichao Wang	University of Kansas, USA
Jeong Hyun Yi	Samsung Adv. Inst. of Technology, Korea
Dalu Zhang	Tongji University, China
Chi Zhou	Florida International University, USA
Matt Mutka	Michigan State University, USA
Yan Sun	University of Rhode Island, USA
Chinya Ravishankar	University of California Riverside, USA
Xue Liu	University of Illinois at Urbana-Champaign, USA
Lei Chen	Hong Kong University of Sci. and Tech., Hong Kong
Tao Gu	Institute for Infocomm Research, Singapore
Yan Chen	Northwestern University, USA
Jie Lian	University of Waterloo, Canada
Vicent Oria	New Jersey Institute of CLIPS-IMAG, USA
Guihai Chen	Nanjing University, China
Dan Wu	University of Windsor, Canada
Hongbo Zhou	Slippery Rock University, USA
Baijian Yang	Ball State University, USA
Pei Zheng	Microsoft, USA
Bin, Xu	Tsinghua University, China
Hung-Chang Hsiao	National Cheng Kung University, Taiwan
Abhishek Patil	Kiyon, USA
Tai-Yi Huang	National Tsing Hua University, Taiwan
Yingshu Li	Georgia State University, USA
Zonghua Gu	Hong Kong University of Sci. and Tech., Hong Kong
Frank Feng Zhu	Michigan State University, USA
Kun Tan	Microsoft Research Asia, China
K. S. Lui	The University of Hong Kong, Hong Kong
Ming Jer Tsai	National Tsing-Hua University, Taiwan
Baihua Zheng	Singapore Management University, Singapore
Minghua Chen	UC Berkeley, USA
Guojun Wang	Central South University, China

Sponsoring Institutions

Hong Kong Polytechnic University, Hong Kong
Springer

Table of Contents

Routing

Topology Control Made Practical: Increasing the Performance of Source Routing 1
Nicolas Burri, Pascal von Rickenbach, Roger Wattenhofer, Yves Weber

Routing Transient Traffic in Mobile Ad Hoc Networks 13
Kan Cai, Michael J. Feeley, Norman C. Hutchinson

Comparison of Two Self-organization and Hierarchical Routing Protocols for Ad Hoc Networks 25
Betânia Steffen Abdallah Goncalves, Nathalie Mitton, Isabelle Guérin-Lassous

Location-Based Multicast Routing Protocol for Mobile Ad Hoc Networks 38
Jipeng Zhou, Jianheng Lu, Francis C.M. Lau

Skipping Face Routing with Guaranteed Message Delivery for Wireless Ad Hoc and Sensor Networks 47
Jie Lian, Kshirasagar Naik

An Anti-void Geographic Routing Algorithm for Wireless Sensor Networks 60
Ming Xu, Yingwen Chen, Yi Wu, Wanrong Yu

A Correctness Proof of the DSR Protocol 72
Huabing Yang, Xingyuan Zhang, Yuanyuan Wang

Scalable Proxy Routing in Multi-hop Cellular Networks 84
Nan Yang, Guangbin Fan

An On-Demand Routing Protocol in Ad Hoc Network Using Label Switching 95
Shaohe Lv, Xingming Zhou, Xiaodong Wang, Chi Liu

Source-Based Multiple Gateway Selection Routing Protocol in Ad-Hoc Networks 107
Sang-Jo Yoo, Byung-Jin Lee

Smart Path-Finding with Local Information in a Sensory Field 119
Wenzhe Zhang, Minglu Li, Wei Shu, Min-You Wu

An Efficient Fibonacci Series Based Hierarchical Application-Layer Multicast Protocol 131
Jing Li, Naijie Gu, Weijia Jia

Maximizing the Probability of Delivery of Multipoint Relay Broadcast Protocol in Wireless Ad Hoc Networks with a Realistic Physical Layer ... 143
François Ingelrest, David Simplot-Ryl

On Improving Wireless Broadcast Reliability of Sensor Networks Using Erasure Codes 155
Rajnish Kumar, Arnab Paul, Umakishore Ramachandran, David Kotz

Cost-Aware Route Selection in Wireless Mesh Networks 171
Junmo Yang, Kazuya Sakai, Bonam Kim, Hiromi Okada, Min-Te Sun

Novel Route Metric for High-Throughput in Multi-rate Wireless Ad Hoc Networks 185
Joo-Sang Youn, Chul-Hee Kang

Protocol

Joint Power Control and Proportional Fair Scheduling with Minimum Rate Constraints in Cluster Based MANET 197
Lijun Qian, Xiangfang Li, Dhadesugoor R. Vaman, Zoran Gajic

Network Coding Approach: Intra-cluster Information Exchange in Wireless Sensor Networks 209
Zhiqiang Xiong, Wei Liu, Jiaqing Huang, Wenqing Cheng, Zongkai Yang

FNSCSDP: A Forward Node Selection Based Cross-Layer Service Discovery Protocol for MANETs 220
Zhenguo Gao, Yongtian Yang, Ling Wang, Jianwen Cui, Xiang Li

Service Discovery Protocols for MANETs: A Survey 232
Zhenguo Gao, Yongtian Yang, Jing Zhao, Jianwen Cui, Xiang Li

A BDD-Based Heuristic Algorithm for Design of Reliable Networks with Minimal Cost 244
Gary Hardy, Corinne Lucet, Nikolaos Limnios

Coverage-Enhancing Algorithm for Directional Sensor Networks 256
Dan Tao, Huadong Ma, Liang Liu

Sensor Scheduling for k-Coverage in Wireless Sensor Networks 268
Shan Gao, Chinh T. Vu, Yingshu Li

A New Media Access Control Protocol for Ad-Hoc Wireless Sensor Networks 281
Kai Chen, Fan Jiang, Zongyao Tang

Performance Evaluation of Binary Negative-Exponential Backoff Algorithm in IEEE 802.11 WLAN 294
Hyung Joo Ki, Seung-Hyuk Choi, Min Young Chung, Tae-Jin Lee

An Application-Aware Event-Oriented MAC Protocol in Multimodality Wireless Sensor Networks 304
Junzhao Du, Weisong Shi

Monte-Carlo Localization for Mobile Wireless Sensor Networks 317
Aline Baggio, Koen Langendoen

Novel Sink-Oriented Approach for Efficient Sink Mobility Support in Wireless Sensor Networks 329
Jeongsik In, Jae-Wan Kim, Kwang-il Hwang, Jihoon Lee, Doo-Seop Eom

Reduction of Signaling Cost and Handoff Latency with VMAPs in HMIPv6 341
Jongpil Jeong, Min Young Chung, Hyunseung Choo

Adaptive Geographically Bound Mobile Agents 353
K. Tei, Ch. Sommer, Y. Fukazawa, S. Honiden, P.-L. Garoche

Gradient-Driven Target Acquisition in Mobile Wireless Sensor Networks 365
Qingquan Zhang, Gerald Sobelman, Tian He

Performance Study of Robust Data Transfer Protocol for VANETs 377
Mooi Choo Chuah, Fen Fu

A Multi-layer Approach to Support Multimedia Communication in Mesh Networks with QoS 392
Chungui Liu, Yantai Shu, Lianfang Zhang, Zenghua Zhao, Xiang-Yang Li

Security

Trust Extended Dynamic Security Model and Its Application in Network 404
Xiaofei Zhang, Fang Xu, Yi Liu, Xing Zhang, Changxiang Shen

An Improvement of Remote User Authentication Scheme Using Smart Cards 416
Jun-Cheol Jeon, Byung-Heon Kang, Se-Min Kim, Wan-Soo Lee, Kee-Young Yoo

A Secure Architecture for Mobile Ad Hoc Networks 424
Abderrezak Rachedi, Abderrahim Benslimane

Interlayer Attacks in Mobile Ad Hoc Networks 436
Lei Guang, Chadi Assi, Abderrahim Benslimane

Mutual-Authentication Mechanism for RFID Systems 449
Batbold Toiruul, KyungOh Lee, HyunJun Lee, YoungHan Lee, YoonYoung Park

Achieving Anonymity in Mobile Ad Hoc Networks Using Fuzzy Position Information 461
Xiaoxin Wu, Jun Liu, Xiaoyan Hong, Elisa Bertino

Proxy Signature Without Random Oracles 473
Xinyi Huang, Willy Susilo, Yi Mu, Wei Wu

Building Hierarchical Public Key Infrastructures in Mobile Ad-Hoc Networks 485
Cristina Satizábal, Jordi Forné, Juan Hernández-Serrano, Josep Pegueroles

Key Exchange in 802.15.4 Networks and Its Performance Implications . . . 497
Moazzam Khan, Fereshteh Amini, Jelena Mišić

An Authenticated Key Agreement Protocol for Mobile Ad Hoc Networks 509
Xukai Zou, Amandeep Thukral, Byrav Ramamurthy

Efficient ID-Based Authenticated Group Key Agreement from Bilinear Pairings 521
Lan Zhou, Willy Susilo, Yi Mu

Efficient Augmented Password-Based Encrypted Key Exchange Protocol 533
Shuhua Wu, Yuefei Zhu

Secure Relative Location Determination in Vehicular Network 543
Lei Tang, Xiaoyan Hong, Phillip G. Bradford

Energy Efficiency

A Local-Control Algorithm to Prolong the Lifetime of Wireless Ad Hoc Networks 555
Jacques M. Bahi, Ahmed Mostefaoui, Michel Salomon

Exploiting Local Knowledge to Enhance Energy-Efficient Geographic Routing 567
Juan A. Sanchez, Pedro M. Ruiz

A Graph-Center-Based Scheme for Energy-Efficient Data Collection in Wireless Sensor Networks 579
Dajin Wang

An Energy Efficient Event Processing Algorithm for Wireless Sensor Networks 588
S. Selvakennedy

A Novel Energy-Efficient Backbone for Sensor Networks 600
Hekang Chen, Shuigeng Zhou, Bin Xiao, Jihong Guan, Bo Huang

An Energy-Balanced Strategy for Clustering Protocols in Wireless Sensor Networks 612
Feng Zhao, Feng Xi, Zhong Liu

QoS Topology Control with Minimal Total Energy Cost in Ad Hoc Wireless Networks 622
Hai Liu, Deying Li, Xiaohua Jia

An Energy Efficient TDMA Protocol for Event Driven Applications in Wireless Sensor Networks 638
Haigang Gong, Ming Liu, Xiaomin Wang, Li Xie

Data Processing

Self-organization Data Gathering for Wireless Sensor Networks 650
Hongli Xu, Liusheng Huang, Junmin Wu, Yang Wang, Jichun Wang, Xu Wang

Continuous Monitoring of *k*NN Queries in Wireless Sensor Networks 662
Yuxia Yao, Xueyan Tang, Ee-Peng Lim

History-Sensitive Based Approach to Optimizing Top-k Queries in Sensor Networks 674
Qunhua Pan, Minglu Li, Min-You Wu

Data Replication in Mobile Ad Hoc Networks 685
Samira Moussaoui, Mohamed Guerroumi, Nadjib Badache

Zone-Based Replication Scheme for Mobile Ad Hoc Networks Using Cross-Layer Design 698
Ke Shi, Rong Chen, Hai Jin

Latency of Event Reporting in Duty-Cycled Wireless Sensor Networks 711
Yu Chen, Eric Fleury

Deployment

Improving Bluetooth EDR Data Throughput Using FEC and Interleaving 724
Ling-Jyh Chen, Tony Sun, Yung-Chih Chen

Information-Accuracy-Aware Jointly Sensing Nodes Selection in Wireless Sensor Networks 736
Huifang Li, Shengming Jiang, Gang Wei

Supporting Application-Oriented Kernel Functionality for Resource Constrained Wireless Sensor Nodes 748
Hyojeong Shin, Hojung Cha

XMAS: An eXtraordinary Memory Allocation Scheme for Resource-Constrained Sensor Operating Systems 760
Sangho Yi, Hong Min, Junyoung Heo, Boncheol Gu, Yookun Cho, Jiman Hong, Hyukjun Oh, Byunghun Song

An Adaptive Distributed Resource Allocation Scheme for Sensor Networks 770
Hock Beng Lim, Vinh The Lam, Mao Ching Foo, Yulian Zeng

Sequential Approach for Type-Based Detection in Wireless Sensor Networks 782
Dmitry Kramarev, Insoo Koo, Kiseon Kim

System Support for Cross-Layering in Sensor Network Stack 793
Rajnish Kumar, Santashil PalChaudhuri, Charles Reiss, Umakishore Ramachandran

A Topology Controllable Testing Environment for Mobile Ad Hoc Network Software 808
Atsushi Kawano, Daisuke Oka, Yasunori Kubo, Shinji Yamashita, Kaori Maeda, Tomoyuki Ohta, Kenji Ishida, Yoshiaki Kakuda

Microcosm: A Low Cost 3-D Wireless Sensor Test-Bed Within a Controllable Environment 820
David Marsh, Richard Tynan, Stephen Beirne, Roderick Shepherd, Gregory O'Hare, Dermot Diamond, Brian Corcoran

System-Level WSN Application Software Test Using Multi-platform Hardware Abstraction Layers 835
Jochen Koberstein, Norbert Luttenberger

An Integrated Self-deployment and Coverage Maintenance Scheme for Mobile Sensor Networks 847
Xu Li, Nicola Santoro

A Power-Aware Peer-to-Peer System for Ad-Hoc Networks 861
Hyun-Duk Choi, Miae Woo

Load Balancing by Distributed Optimisation in Ad Hoc Networks 873
André Schumacher, Harri Haanpää, Satu Elisa Schaeffer, Pekka Orponen

Author Index 885

Topology Control Made Practical: Increasing the Performance of Source Routing*

Nicolas Burri, Pascal von Rickenbach, Roger Wattenhofer, and Yves Weber

Computer Engineering and Networks Lab, ETH Zurich
CH-8092 Zurich, Switzerland
{nburri, pascalv, wattenhofer, webery}@tik.ee.ethz.ch

Abstract. Wireless ad hoc and sensor networks need to deal with unstable links. In practice the link quality between neighboring nodes fluctuates significantly over time. In this paper we evaluate the impact of topology control on routing performance. We propose a dynamic version of the XTC topology control algorithm. This simple and strictly local protocol removes unreliable and redundant links from the network. By means of physical experiments on an indoor mica2 testbed we study the beneficial effects of topology control on source routing, one of the most common routing schemes for ad hoc and sensor networks. In particular we compare the performance of source routing with and without topology control. Our results show that topology control reduces route failures, increases network throughput, and diminishes average packet delay.

1 Introduction

Sensor networks ask for highly optimized protocols. In order to meet the demands, we witness that more and more researchers acquit themselves of the orthodox layering hierarchy, pushing the envelope of their protocols with cross-layer design. Abandoning layering (which after all is one of the most well-accepted principles in networking) however comes at the cost of reusability. Moreover, it is not always clear that an integrated cross-layer design has advantages over a well-defined layered interface. In this paper we consider the effect of eliminating unreliable connections at the link layer (a technique usually known as topology control) on higher communication layers. Nowadays this task is often integrated into the network layer, leading to complex routing protocols. We limit our study to non-mobile wireless networks. Such networks are sometimes called mesh or rooftop networks. In spite of being static, link qualities vary over several different time scales, from seconds to hours, due to interference or mobile obstacles. This leads to frequent network topology changes. Hence, if it comes down to implementing a real system for wireless networks the network stack has to deal with the problems arising

* The work presented in this paper was partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 1–12, 2006.

from unreliable communication links. In literature, many studies have been published under the name of topology control trying to mitigate the effects of unstable network connections [1,2,3,4,5]. So far, topology control has only been considered from a theoretical point of view. Researchers have devised algorithms establishing a subgraph of the initial network by dropping specific connections such that the resulting topology features a variety of desired properties. However most topology control algorithms are based on assumptions that are questionable in practice. Besides too simplistic network models it is also often assumed that the nodes have detailed information about their neighbors. Furthermore, the communication graph is always supposed to be static; consequently, all proposed algorithms compute their resulting topology only once. In practice there is no such thing as a perfect topology since the underlying network graph changes over time. All these *one-shot* solutions would need major adaptations in order to deal with dynamic networks. Thus, despite the considerable body of research devoted to topology control and the theoretical and simulation-based evidence of its effectiveness, to date there is little *experimental* evidence that topology control can actually be used to circumvent the problem of unreliable links in wireless networks. Apart from the efforts made in the field of topology control there have been attempts to cope with unreliable communication links directly on the network layer. Many routing protocols for wireless networks were proposed trying to predict link stability based on signal strength and to choose the "best" route using these predictions [6,7,8,9,10]. These protocols have in common that their decisions are threshold based; that is, if the link quality is above a fixed threshold the link is incorporated in a route. This can result in failed routing attempts even if working paths exist. Sometimes unreliable links need to be chosen in order to prevent the network from being disconnected.

In this paper we provide an implementation of the XTC algorithm [11] which guarantees connectivity of the network while discarding unreliable communication links. We have extended the algorithm proposed in [11] such that it now also copes with dynamic networks comprising fluctuating links. If we refer to XTC throughout the rest of the paper we always refer to this extended version. The algorithm is implemented on the mica2 sensor node platform which facilitates rapid prototyping for wireless ad hoc networks. In order to examine the benefits of XTC we evaluate the performance of source routing as a generic sample application in wireless networks with and without XTC. The algorithm manages to identify stable connections and thus preserves source routing from taking unreliable links. To be more specific, XTC decreases the number of route failures and retransmission attempts and is therefore able to increase network throughput and to lower average packet delay. Moreover, the omission of fluctuating links (if not needed) by XTC only modestly increases average route lengths.

The remainder of the paper is organized a follows: Section 2 compares our contributions with previous related work. In Section 3 an extended version of the XTC algorithm is described that also copes with a dynamic environment. The algorithm's behavior in practical networks is the subject of Section 4. Section 5 concludes the paper.

2 Related Work

The characterization of wireless links in different environments is analyzed in [12,13]. Both papers identify the existence of "gray links", links that are highly variant and unreliable. One of the goals of topology control is to shield the upper layers from the problems arising from these unstable links. However, modern topology control algorithms offer a multitude of other goals such as low node degree, planarity, or reduced power consumption (nodes adjust their transmission power level in order to save energy; a mechanism also known as *power control*). In this work, we concentrate on unreliable links. Power control is orthogonal to our solution and could be incorporated.

Most proposed topology control algorithms require hardware technology that is not available today. The algorithms presented in [4,3,5,14] require knowledge of exact node locations. Other work assumes that relative distance and directional information is available [2]. The protocol described in [15] needs the node distribution to be uniform-random. All mentioned protocols operate on idealized radio models such as unit disk graphs. The simple XTC algorithm [11] extended in Section 3 always maintains connectivity given a general weighted network graph. In [16] a generalized version of XTC called k-XTC is presented; this algorithms drops a communication link only if at least k alternative paths exist. In [17] the RTC algorithm is proposed that slightly changes XTC such that link qualities are determined randomly. In [18] the authors propose S-XTC, an extended version of [11] implemented on Bluetooth enabled sensor nodes.

To the best of our knowledge, there exists no practical evaluation of the influence of topology control to wireless networks – with one notable exception [19]. In [19] the authors provide an experimental study of the impact of variable transmission power levels on link quality. Their protocol uses power control and blacklisting to eliminate unreliable links but (in contrast to our work) does not give any connectivity guarantees of the resulting topology.

In the domain of routing protocols several previous papers improve the proposed protocols by predicting link qualities to enhance their performance. In [8] and [20] preemptive route maintenance algorithms based on signal strength are presented; they proactively initiate a new route discovery if a link on an active route becomes worse that a given threshold. In [21] a metric is presented to identify high-throughput routes when different links can run at different bitrates. However, their metric does not consider packet losses and is thus complementary to our work. There exists a number of wireless routing algorithms collecting per-link signal strength information and apply a threshold to avoid connections with high loss ratios [6,7,8,9]. In contrast to our work this approach may eliminate links that are necessary for connectivity (if the threshold is too high), or keep unnecessary bad-quality links (if the threshold is too low); both of these are likely to be issues in networks with many "gray" links.

The authors in [22] propose the ETX metric predicting the number of retransmissions required using per-link measurements of packet-loss ratios. The effectiveness of this approach is demonstrated by showing that the metric improves Dynamic Source Routing (DSR) in an experimental testbed using WLAN

XTC Algorithm

```
Update order $\prec_u$ over $u$'s neighbors
Request current orders from neighbors
Select topology control neighbors:
    $N_u := \{\}$; $\widetilde{N}_u := \{\}$
    while ($\prec_u$ contains unprocessed neighbors) {
        $v :=$ least unprocessed neighbor in $\prec_u$
        if ($\exists\, w \in N_u \cup \widetilde{N}_u : w \prec_v u$)
            $\widetilde{N}_u := \widetilde{N}_u \cup \{v\}$
        else
            $N_u := N_u \cup \{v\}$
    }
```

technology. A major drawback of their solution is that the flooding initiated during route discovery can generate a large amount of network traffic since intermediate nodes are required to retransmit an already forwarded route request in case of a potentially better path. Consequently, network performance may degrade drastically since a single route search can trigger multiple (in theory, even exponentially many!) floodings—we have a broadcast storm.

3 XTC in Dynamic Networks

In [11] the XTC topology control algorithm was introduced exhibiting several desirable properties. However, the algorithm assumes the underlying network to be static. In this section we adapt the original algorithm to deal with dynamic networks comprising unreliable links. The algorithm still consists of three main steps: Neighbor ordering, neighbor order exchange, and link selection. Each node repeats these three steps periodically in order to cope with changing topologies. Although XTC is executed at all nodes, the following description assumes the point of view of a network node u.

In the first step a node u updates its total order $\prec_u$ over all neighbors in the network. From an abstract point of view, this order is supposed to reflect the quality of the links to the neighboring nodes. The neighbors of u are thereby arranged in $\prec_u$ with respect to decreasing link qualities. From an implementation point of view, node u first has to update the link qualities according to a particular metric. In our experiments the applied metric is based on the packet loss ratio of past transmissions. The XTC algorithm in [11] assumes that both endpoints of a link agree on the quality of their connection. In reality this does not have to be the case. Both endpoints can only judge the link quality from their own perspective and thus may come to different results. Consequently, the concerned nodes have to negotiate and settle on the same link quality value.

In the second step the neighbor order information is exchanged among all neighbors. To limit communication overhead, node u only requests $\prec_v$ if it has outdated order information from neighbor v.

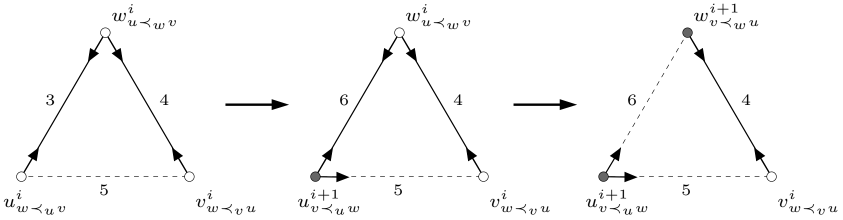


Fig. 1. On the left, the initial state of a sample network that consists of three nodes u, v, and w is depicted including links picked by XTC (solid). The quality of link (u, w) deteriorates from 5 to 2. The topology graph after an iteration of XTC at node u is shown in the middle. On the right, it can be seen that the topology becomes disconnected after w's update cycle.

During the third step, node u locally selects the neighboring nodes for the next iteration of the algorithm. For this purpose node u traverses $\prec_u$ with decreasing link quality: "Good" neighbors are considered first, "worse" ones later. Node u only builds a direct communication link to a neighboring node v if u has no "better" neighbor w that can be reached more easily from v than u itself. For a more detailed description of the third step we refer the interested reader to [11].

Besides the above mentioned issues related to the XTC algorithm itself, difficulties arise from the fact that neighboring nodes are not attuned to one another. In particular, we have to ensure that the properties of XTC as shown in [11] are still valid in the presence of partially updated order information. In case of changing link qualities it is necessary that a node detecting a change, triggers other neighbors to start another iteration of the algorithm to avoid temporary network partition. The problem is illustrated by means of a simple network consisting of three nodes u, v and w in Figure 1. The link quality values are depicted next to all possible connections. Higher values thereby indicate better link quality. We assume that all nodes have already executed i iterations of the algorithm at the beginning of our examination. Furthermore, let $u^i_{w \prec_u v}$ denote node u after the i-th iteration of XTC with the neighbor order $w \prec_u v$. On the left hand side of Figure 1 the network is in a consistent state where all nodes have finished iteration i. The arrows pointing out of a node indicate the links selected by XTC and solid lines show the resulting topology graph. In the middle, a snapshot of the network is shown after node u finishes its $(i+1)$-th update cycle. Note that the link (u, w) got worse. Consequently, node u selected (u, v) in iteration $i+1$ to be in the topology control graph. However, the connection between u and v is not yet established since v has not executed iteration $i+1$. On the right, Figure 1 depicts the situation after node w successfully completed the $(i+1)$-th iteration of XTC resulting in an order $v \prec_w u$. Using this order w drops the connection link (u, w) which results in a temporary partition of the network. However, when v has executed XTC as part of the completion of iteration $i+1$ the topology control graph will be connected again.

In order to obviate the above mentioned problem, a node detecting a change in link quality of a particular connection informs its neighbors. If a node receiving such a trigger message is adjacent to the affected link or contains both endpoints of the connection in its neighborhood it instantly executes XTC in order to minimize the duration of potential network partition.

4 Experiments

For empirical study the XTC algorithm described in Section 3 was implemented on the mica2 sensor node platform. On an office floor we set up networks of different sizes and ran various experiments evaluating the practical impact of topology control on source routing in wireless ad hoc and sensor networks.

4.1 Link Quality Metric

Defining a reasonable link quality metric is a fundamental requirement for a physical implementation of the XTC algorithm. To specify such a metric it is necessary to contemplate link characteristics of mica2 networks. Therefore, several experiments were made on various networks evaluating the behavior of links in different settings. It is a surprising result that even in segregated environments links do not have constant error probabilities over time. To exemplify this, a setup consisting of two nodes exchanging 1000 messages in 30 minutes is used of which 537 arrived. As can be seen in Figure 2 the link reliability worsened in the course of the experiment. After transmitting approximately 620 packets (of which 360 arrived at the receiver) packet loss increased drastically and the link started to fail for up to 50 consecutive packets. Such breakdowns could be observed in nearly all of our experiments.

Packet loss is a property which can be used to characterize the quality of a wireless link[1]. Figure 3 shows a link quality indicator for this experiment based on packet loss. This metric attempts to predict future packet loss based on past transmission failures. To achieve a reasonably stable link quality indicator we apply a moving average function on measured consecutive packet loss. Figure 3 shows that a moving average using 0.1 as weight for the current value leads to a fluctuating link quality indicator (dotted). Using a weight of 0.01 results in a more stable curve (solid). With this metric the lower link reliability during the last third of the experiment leads to a distinct decrease of the link quality indicator. Ultimately, the goal of the XTC algorithm is to avoid unreliable links. Ordering neighbors according to a metric based on packet loss allows to achieve this. Therefore, we decided to use this metric for all further experiments.

4.2 Source Routing and XTC

To evaluate the fitness of topologies created by XTC for real life purposes we implemented a basic source routing protocol. This implementation was designed

[1] We have also evaluated the Received Signal Strength Indicator (RSSI) as a potential link metric. However it turned out that RSSI is no adequate link quality indicator for our purposes. Details can be found in [23].

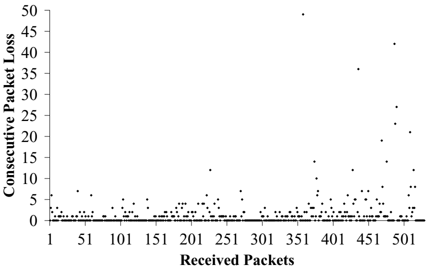


Fig. 2. Consecutive packet loss while sending 1000 messages over a link

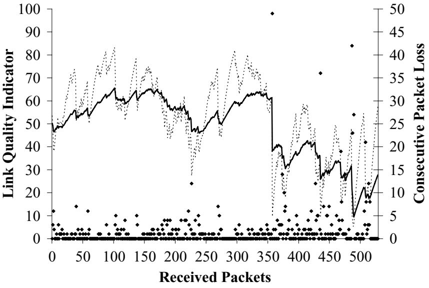


Fig. 3. Link quality indicator with weight 0.1 (dotted) and weight 0.01 (solid)

to incorporate knowledge gained from the topology control algorithm. That is, for route discovery only edges which are part of the current XTC graph are used. Discovered routes are cached and reused until a route error occurs, independent of whether all used links are still part of the XTC graph. This proceeding is reasonable since the XTC graph changes over time. It always contains the currently most reliable links and thus it may occur that good links in the graph are replaced by even better ones. Existing routes using the replaced edges may still work perfectly well and should therefore not be discarded until message transmissions start to fail.

4.3 Small Testbed

The first experiment with our implementation of XTC was run on a small network of seven nodes placed in a zig-zag like topology shown in Figure 4. The transmission power of the nodes was adjusted such that the topology featured stable short links (solid) and unreliable longer links (dashed). It was set up in an empty corridor where it was possible to minimize external interference. Over a period of eight hours node 1 sent multi-hop ping messages to all other nodes in the network. To evaluate the impact of topology control on this experiment we measured the successful number of transmissions, route lengths, and necessary route searches with and without XTC, respectively. Figure 5 shows the results of this experiment. Since the network was designed to be reasonably stable,

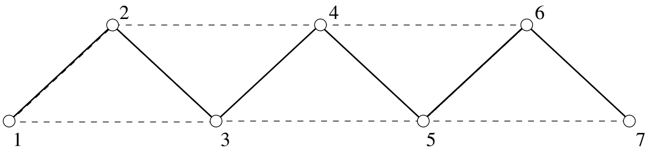
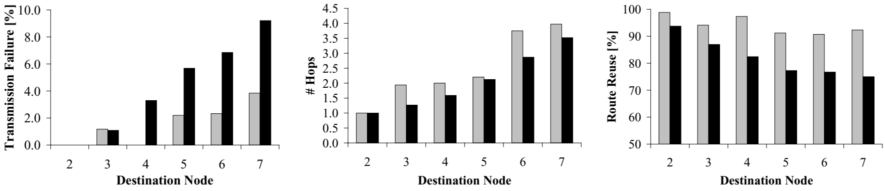


Fig. 4. Test network consisting of seven nodes. Node 1 is the initiator of all communication in the network.



(a) Transmission failures (b) Average route length (c) Route reuse

Fig. 5. Evaluation of the small testbed with (gray) and without (black) XTC

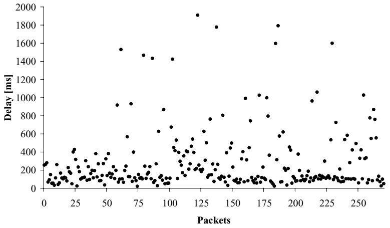


Fig. 6. Delay for successful message transmission across the small testbed *without* XTC

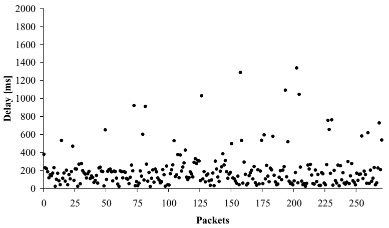


Fig. 7. Delay for successful message transmission across the small testbed *with* XTC

the number of successful transmissions is above 90% independently of whether topology control was enabled or not. However, XTC manages to reduce transmission failures to nearly all receivers. Especially the route stability to distant nodes was improved as can be seen in Figure 5(a). Since source routing always communicates over the first connection replying to a route request the risk of having unstable, long links on the used path increases with the number of hops; thus also the packet loss probability increases with increasing route length. XTC does not hinder communication over fast edges but prevents source routing from choosing unreliable links. Consequently, source routing on XTC uses the path with minimal delay consisting of stable links. This is often achieved by replacing one unreliable link with multiple high-quality links. Hence, the measured route lengths on the XTC graph are generally longer than the ones chosen by pure source routing (see Figure 5(b)). However, for most receivers the chosen paths have similar lengths for pure source routing and the topology control assisted version. The maximum average route length difference is 0.9 hops for the path to node 6. For all other nodes the increase is less than 0.5 hops.

This small concession made to improve route stability pays off if the percentage of reusable routes is considered. With XTC route reuse is above 94% (cf. Figure 5(c)). Pure source routing reaches a reuse rate of 82%. This increased route stability implies that less of the expensive route discoveries were needed which in turn leads to an improved throughput. To evaluate this expected throughput increase we performed an additional experiment. Node 1 sent 300 messages

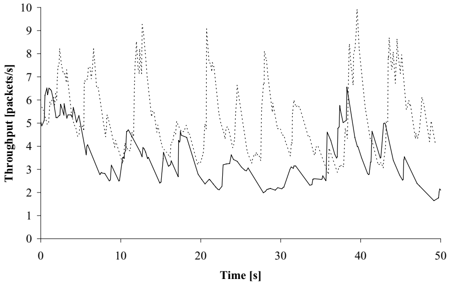


Fig. 8. Message throughput across the small testbed with (dotted) and without (solid) XTC

as fast as possible—using end-to-end acknowledgments—to node 7 at the other end of the network. We allowed three retransmissions per packet followed by an unlimited number of route searches until node 7 could be reached again. We measured the total transmission time for each data packet including time spent on retransmissions and route discoveries.

Figures 6 and 7 show the measured delays for all packets with and without XTC, respecively. In both scenarios the majority of all packets could be sent using the cached route. These packets had a delay of 50 to 250 milliseconds depending on the number of necessary retransmissions. For some packets the cached route failed and thus, a new route discovery was initiated. Due to the various timeouts of the routing protocol their delay increased up to two seconds. Consequently, the average delay dropped from 290 ms to 200 ms if XTC was active. The measured throughput as depicted in Figure 8 was also generally higher if XTC was used for route discovery. On average it increased from 3.45 to 4.98 packets per second which corresponds to an improvement of more than 44%.

4.4 Office Floor Network

The promising results gained from the small testbed encouraged us to evaluate XTC in a larger network. We therefore distributed 33 nodes on an office floor (see Figure 9) and performed the same experiments as in the small testbed considered in Section 4.3. Since these experiments were executed during day time with numerous people working on the floor various real world effects such as moving obstacles and temporary interference with other wireless devices occurred. Such perturbations seem to have existed for example in the region of nodes 9, 10, and 28. We assume that the sources of this observed interference lie in the nearby elevators and an adjacent student lab.

Figure 10 shows that XTC manages to decrease transmission failures. On average the relative improvement is 11%. For the few nodes, such as 22, where XTC results in a decreased routing performance the additional penalty is below 5%. Similarly, route reuse increased if XTC was active (cf. Figure 12). Figure 11 shows that the route length did not increase drastically with XTC. That is, the

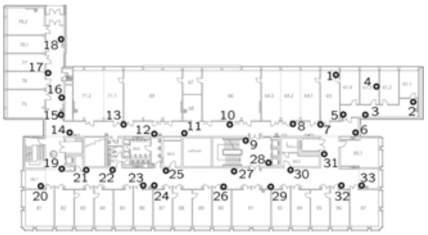


Fig. 9. Large scale experiment consisting of 40 nodes spread out on an office floor. Node 1 is the initiator of all data communication.

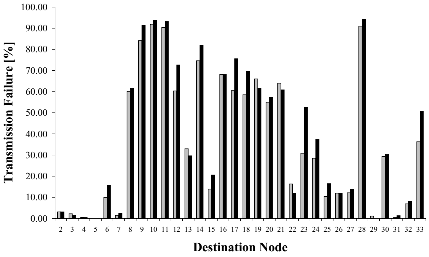


Fig. 10. Transmission failures in percent with (gray) and without (black) XTC

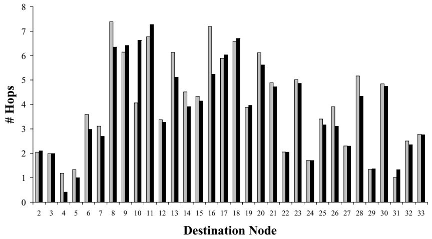


Fig. 11. Average route length with (gray) and without (black) XTC

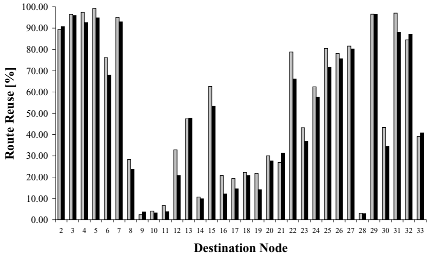


Fig. 12. Percentage of route reuse for communication with (gray) and without (black) XTC

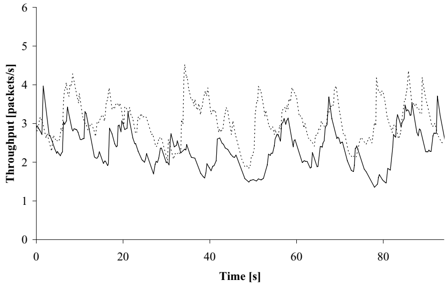


Fig. 13. Message throughput between node 1 and 23 in the office floor testbed with (dotted) and without (solid) XTC

average hop count increases from 3.77 to 3.96. Analogous to the small testbed experiment we also evaluated throughput gains induced by XTC. Figure 13 exemplifies the benefit of XTC by showing the performance measurements from node 1 to 23. Without XTC the average throughput was 2.29 packets per second and increases to 3.02 with XTC. This is a relative improvement of roughly 32%. Summarized, the positive impact of XTC on source routing as seen in the small testbed also applies to this experiment. Especially, in the gray regions XTC proved its usefulness and lead to improved throughput.

5 Conclusions and Future Work

In this paper we have presented a practical implementation of an extended version of the XTC topology control algorithm which dynamically adapts to network changes. Using a packet loss based metric unreliable links are identified and excluded while connectivity is maintained. The beneficial effect of XTC on source routing—a common application in wireless networks—was first shown in a secluded environment using a small number of nodes. We then verified the obtained results in a real world scenario. Also in this environment XTC performed well reducing packet loss and delay and thus leading to improved network throughput.

So far, the usefulness of XTC is only evaluated in non-mobile wireless networks. We believe XTC will also show its advantages in networks with mobile nodes. In such environments link qualities change more frequently and thus it is important to select reliable and long-living connections. Adapting XTC's link quality metric to incorporate link stability may be required. Another interesting field of application are networks with high density. Due to the large amount of potential paths from one node to any other there is a high probability of choosing a route containing an unreliable link. Consequently, the ability of XTC to exclude such links is important. Finally, the impact of XTC on other applications than source routing is worth studying. For example, alarm systems with a low tolerance toward communication failures or other data gathering applications may also benefit from the XTC protocol.

References

1. Hu, L.: Topology Control for Multihop Packet Radio Networks. IEEE Trans. on Communications **41**(10) (1993)
2. Wattenhofer, R., Li, L., Bahl, P., Wang, Y.M.: Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In: Proc. of the *20*th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2001)
3. Li, X.Y., Calinescu, G., Wan, P.J.: Distributed Construction of Planar Spanner and Routing for Ad Hoc Wireless Networks. In: Proc. of the *21*st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2002)
4. Li, N., Hou, C.J., Sha, L.: Design and Analysis of an MST-Based Topology Control Algorithm. In: Proc. of the *22*nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2003)
5. Li, X.Y., Song, W.Z., Wang, W.: A Unified Energy Efficient Topology for Unicast and Broadcast. In: Proc. of the *11*th Annual International Conference on Mobile Computing and Networking (MOBICOM). (2005)
6. Chin, K.W., Judge, J., Williams, A., Kermode, R.: Implementation experience with manet routing protocols. ACM SIGCOMM Computer Communications Review **32**(5) (2002) 49–59

7. Dube, R., Rais, C.D., Wang, K.Y., Tripathi, S.K.: Signal Stability-Based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks. IEEE Personal Communications (1997) 36–45
8. Goff, T., Abu-Ghazaleh, N.B., Phatak, D.S., Kahvecioglu, R.: Preemptive Routing in Ad Hoc Networks. In: Proc. of the *7th* Annual International Conference on Mobile Computing and Networking (MOBICOM). (2001)
9. Hu, Y., Johnson, D.: Design and Demonstration of Live Audio and Video over Multihop Wireless Ad Hoc Networks. In: Proc. of the Military Communication Conference (MILCOM). (2002)
10. Qin, L., Kunz, T.: Increasing Packet Delivery Ratio in DSR by Link Prediction. In: Proc. of the *36th* Annual Hawaii International Conference on System Sciences (HICSS). (2003)
11. Wattenhofer, R., Zollinger, A.: XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks. In: Proc. of the *4th* International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN). (2004)
12. Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., Wicker, S.: Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor. Technical Report 02-0013, UCLA CS (2002)
13. Zhao, J., Govindan, R.: Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. In: Proc. of the *1st* ACM Confernce on Embedded Networked Sensor Systems (SenSys). (2003)
14. Moaveni-Nejad, K., Li, X.Y.: Low-Interference Topology Control for Wireless Ad Hoc Networks. In: Proc. of the *2nd* Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California, USA (2005)
15. Blough, D., Leoncini, M., Resta, G., Santi, P.: The K-Neigh Protocol for Symmetric Topology in Ad Hoc Networks. In: Proc. of the *4th* ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC). (2003)
16. Ghosh, S., Lillis, K., Pandit, S., Pemmaraju, S.V.: Robust Topology Control Protocols. In: Proc. of the *8th* International Conference on Principles of Distributed Systems (OPODIS). (2004)
17. Lillis, K., Pemmaraju, S.V.: Topology Control with Limited Geometric Information. In: Proc. of the *9th* International Conference on Principles of Distributed Systems (OPODIS). (2005)
18. Dyer, M.: S-XTC: A Signal-Strength Based Topology Control Algorithm. TIK Report 235, ETH Zurich, Switzerland (2005)
19. Son, D., Krishnamachari, B., Heidemann, J.: Experimental study of the effects of Transmission Power Control and Blacklisting in Wireless Sensor Networks. In: Proc. of the *1st* IEEE Conference on Sensor and Ad Hoc Communication and Networks (SECON). (2004)
20. Qin, L., Kunz, T.: Pro-active route maintenance in dsr. ACM SIGMOBILE Mobile Computing Communications Review **6**(3) (2002) 79–89
21. Awerbuch, B., Holmer, D., Rubens, H.: High Throughput Route Selection in Multi-Rate Ad Hoc Wireless Networks. In: Proc. of the *1st* Annual Conference on Wireless On-Demand Network Systems (WONS). (2003)
22. Couto, D.S.J.D., Aguayo, D., Bicket, J., Morris, R.: A High-Throughput Path Metric for Multi-Hop Wireless Routing. In: Proc. of the *9th* Annual International Conference on Mobile Computing and Networking (MOBICOM). (2003)
23. Burri, N., von Rickenbach, P., Wattenhofer, R., Weber, Y.: A Practical Evaluation of the XTC Algorithm. TIK Report 263, ETH Zurich, Switzerland (2006)

Routing Transient Traffic in Mobile Ad Hoc Networks

Kan Cai, Michael J. Feeley, and Norman C. Hutchinson

Department of Computer Science, University of British Columbia
{kcai, feeley, norm}@cs.ubc.ca

Abstract. Recent research shows that the traffic in public wireless networks is mostly transient and bursty. There is good reason to believe that ad-hoc traffic will follow the same pattern as its popularity grows. Unfortunately transient traffic generates route discoveries much more frequently than the long-term, constant-bit-rate traffic, causing network congestion problems for existing routing protocols. This paper describes the design of a new routing algorithm, called ECBR, that uses hybrid backbone routing in a manner that is well suited to workloads that include transient traffic. We explain three key features of our algorithm and demonstrate their roles in greatly improving the performance compared to existing reactive and backbone routing techniques.

Keywords: Transient Traffic, Mobile Ad Hoc Networks, Routing.

1 Introduction

It has been standard practice to evaluate mobile ad-hoc routing algorithms using long-term, constant-bit-rate traffic [1,2,3]. A key characteristic of this workload is that route discovery is infrequent with its cost amortized over a large number of data packets sent using each route. If some connections are more transient or traffic is more bursty, however, route discovery becomes more frequent and the performance of existing algorithms degrades to an extent not anticipated by previous constant-bit-rate studies.

In fact, there is good reason to believe that real ad-hoc networks will indeed include transient and bursty traffic. Previous analyses of public wireless networks in both academic and corporate environments [6,7] have shown that users are often passive and network traffic is bursty. In these studies, wireless sessions were usually short-lived and the long-term connections that did exist were idle much of the time. Similar studies of PDA users [8,9] have shown that median user-session duration is 5-6 minutes and that users switch access points every 1.8 minutes. Furthermore, applications such as web browsers and instant messaging are inherently bursty. Message bursts from these applications are typically as short as a few seconds and are often separated by long periods of inactivity.

We show in this paper that backbone approaches are better suited to transient traffic than flat, reactive algorithms such as DSR. The reason is that when route

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 13–24, 2006.

discovery is frequent, reactive algorithms flood the network with too many route-discovery messages, triggering broadcast storms that render the entire network temporarily useless. Backbone algorithms, on the other hand, confine routing to the backbone, which is comprised of only a small subset of the network. As a result, fewer messages are required to discover new routes, because route caching is more effective and route-discovery broadcasts are forwarded by fewer nodes.

This paper describes a hybrid, backbone routing algorithm we have developed called ECBR (end-point cache backbone routing). ECBR is similar to other backbone protocols such as DSRCEDAR and CBRP, but with three key differences related to how it acquires and caches routing information. First, ECBR piggybacks additional information on each route-discovery reply message to prefetch routes for multiple nodes. Second, ECBR uses timestamps on route-cache entries to keep cached routing information current. Third, ECBR uses proactive, local route adaptation and recovery to dynamically change inter-backbone-node links without changing globally cached routes or dropping packets.

Our evaluation shows that ECBR can robustly handle a wide range of traffic patterns, including both short-term and long-term connections. For transient traffic, ECBR significantly outperforms DSR and a DSRCEDAR-like algorithm we implemented for comparison purposes, called DSRX.

2 Related Work

In contrast to *flat* routing protocols such as DSR [4] and AODV [5], *hierarchical* routing protocols minimize overhead by aggregating routing information and reducing the range and frequency of route-discovery broadcasts. They differ in the way they use the backbone to route packets. Some closely integrate routing with the backbone construction algorithm itself, using proactivity in aspects of both, while others layer a reactive algorithm such as DSR on top of the backbone in a more modular fashion.

Hierarchical link-state protocols such as HSR [10] and HierLS [11] follow the more integrated approach. They build multi-level and multi-hop clusters, and then proactively maintain routes to all the other nodes in the network. This integrated approach is complex and can thus be hard to implement. Another set of integrated approaches are the spine-based algorithms from R. Sivakumar et al. [12,13]. It uses only two layers in the routing hierarchy and assumes that the radio network provides a reliable broadcast scheme.

In any case, the main potential drawback of integrated approaches such as these is that by making the routing protocol proactive there is a high cost to maintaining routing structures, particularly when nodes fail or move. The alternative to the integrated approach is to layer a reactive algorithm on top of the backbone; this is the approach that ECBR and three other algorithms follow.

First, K. Xu et al. [14] propose a two-level, hierarchical, clustering algorithm, called *mobile backbone*. It uses distinguished backbone nodes with more-powerful radios that communicate directly with each other. This direct communication simplifies the routing problem greatly compared to the environment we target.

Second, M. Jiang et al. [15] describe a DSR-like routing algorithm on top of a backbone, called CBRP. All nodes send periodic, heartbeat messages, which are used to elect cluster heads and maintain cluster-membership lists. Cluster heads also use these messages to build inter-cluster connections using gateway nodes. CBRP uses a complex variant of DSR to route packets on this backbone. If a destination is unknown or a route is broken, the cluster heads have to flood the backbone using a combination of broadcast and unicast messages.

Third, P. Sinha et al. [16] propose a core-based routing algorithm, called CEDAR, which includes various features designed to support QoS routing. Each core node establishes tunnels with its neighbouring core nodes via three-hop broadcast messages. CEDAR requires to modify the MAC layer and uses promiscuous mode to limit core-broadcast message propagation. P. Sinha et al. subsequently improved CEDAR by reducing its core broadcast overhead [17], in which routing is performed by a standard, flat protocol layered on top of the core. Their results show that the addition of the backbone improves the performance of standard, flat algorithms such as DSR and AODV.

In many ways ECBR builds on earlier systems such as CBRP and DSRCEDAR. What makes ECBR unique is the way it uses the backbone to improve the route-cache effectiveness and thus reduce the frequency of route discovery and better handle transient communication. It does this in three main ways. First, it uses piggybacking to prefetch routing information into caches. Second, it uses timestamps to flush out-of-date routes from caches. Third, it uses proactively maintained neighbourhood information to adapt to changes in inter-backbone-node connectivity without invalidating globally cached routes or dropping packets.

3 ECBR Routing Algorithm

ECBR is partly proactive and partly reactive. Its proactive component acts to maintain a single backbone for the network. Routing, on the other hand, is performed reactively in a manner similar to DSR, but where routes are confined to follow the backbone.

ECBR selects certain nodes to act as cluster heads, called *dominators*; all other nodes are called *dominatees*. The dominators cover the entire network and no two dominators are in range of each other. A dominator uses its dominatees to connect its cluster to the nearby clusters via either two- or three-hop paths.

Periodic heartbeat messages are used to maintain the clusters, their cluster heads and inter-cluster links. Dominators store a list of in-radio-range dominatees in the *dominatee ownership table* (DOT). Dominators also store a *connectivity list* containing paths to other dominators that are two or three hops away. Similarly, dominatees store a list of in-range dominators and a connectivity list containing paths to dominators that are at most two hops away.

The backbone is constructed using a variant of the message-optimal connected dominating set algorithm [18]. Due to space limitation, please refer to our technique report [19] for the details of proactive backbone construction. The rest of this section is to describe the reactive routing protocol that delivers payload packets from source nodes to their targets over an existing bakcbone.

3.1 Routing Caches

The two routing caches stored on dominator nodes are the *dominatee routing table*, DRT, which caches dominatee-dominator pairings, and the *backbone routing table*, BRT, which caches backbone-topology in the form of a list of connected dominators. Dominators maintain their DRT and BRT caches reactively using the content of messages they receive.

The DRT is updated by route-discovery reply messages, which typically include multiple dominatee-dominator pairings, due to the piggybacking feature described in Section 3.3. The BRT is updated by every packet a dominator receives. Typically, the BRT accurately captures backbone topology and rapidly adapts to backbone changes. These desirable properties derive from the fact that backbone uses source routing for packet delivery.

3.2 Base Routing Scheme

ECBR uses a DSR-like routing protocol to discover and maintain backbone routes using three types of control messages: *route discovery*, *route reply* and *route nack*.

A dominator triggers the route discovery procedure by unicasting a route discovery message to each of its backbone neighbours when it cannot resolve a path to a destination node requested by one of its dominatees. When another dominator receives a discovery message, it first checks its DOT to determine whether it has the target node in radio range. If not, it checks its DRT and BRT to determine whether it caches a route to the target. If all of these checks fail, it adds itself to the *path-traveled* list in the message's header and forwards the message to all of its backbone neighbours that are not yet listed in the path-traveled list. When a dominator locates the target, it sends a reply message by reversing the path-traveled list in the request message.

One optimization of ECBR is that a newly-initiated route-discovery message is first propagated on the backbone following a *spanning tree* rooted at the requesting dominator, as an attempt to reduce the number of redundant route-discovery messages. If this fails, the subsequent route-discovery retries use the above backbone broadcast scheme.

Whenever a dominator is unable to forward a packet to a neighbouring dominator, it first deletes the corresponding backbone link from its BRT. Then, it sends a *route-nack* message over the reverse path-traveled route back to packet's source dominator and each dominator that receives the route-nack message removes the failed link from its BRT.

3.3 Three Optimizations

Cache Timestamping. A common problem with reactive caches is detecting routes that become invalid when nodes move or fail. Using an outdated route for packet delivery may cause packets to be dropped. In ECBR this problem primarily affects the accuracy of the DRT caches that record node location by pairing nodes with their nearby dominators.

The DRT attacks this problem by using timestamps to estimate the freshness of cached pairings and then using this freshness as a heuristic to predict accuracy. ECBR requires the dominatees to timestamp their heartbeat messages and the dominators to record this information when receiving these messages. These timestamps will be later propogated along the backbone using route-reply messages and kept in dominators' DRT tables.

When a remote dominator compares two possible dominator pairings for a dominatee, the pairing with the most recent timestamp is a good estimate of the current location of the dominatee. Each forwarding dominator also uses the DRT timestamps to check whether it has a *newer* entry for the target than reflected in the current route. If so, it updates the packet's route accordingly. Since these timestamps are orginated at the dominatees, there is no need to use any global time synchronization algorithm in ECBR.

Timestamps are also used by the lower-level routing mechanism that connects neighbouring dominators to each other. Often a pair of dominators have many two- and three-hop paths that connect them. Connectivity lists that describe these paths are timestamped by heartbeat messages. A dominator can thus pick the connector with the most recent timestamp for packet delivery.

Piggybacked Cache Prefetching. The second key optimization in ECBR is that route-discovery reply messages are padded to piggyback route information for multiple targets. As a result, a single reply can resolve routes to many distinct nodes, at the cost of a small marginal increase in reply-packet size, and thus many subsequent costly route-discovery messages are avoided.

To implement piggybacking, every dominator maintains a vector timestamp that records the last DOT update it received from every other dominator. Dominators include this vector timestamp in route-discovery messages they originates. When a target dominator receives a route request, it checks its entry in this vector to determine which of its DOT entries are not cached at the source. The target then sends a reply that includes the vector timestamp and as many of these entries as will fit in the message. Every node on the reply path repeats this process using their own entry in the message's vector timestamp until the packet reaches its IP MTU limit. Each also extracts entries from the packet to add to its own DRT.

Route Adaptation and Recovery. The final feature of ECBR routing is the way it corrects for local failures. While the routing layer treats backbone links between dominators as single paths when constructing routes, in reality each link is a multi-path connection involving one or two connector nodes. Because there are multiple low-level connections that can instantiate a upper-level path, the low-level routing protocol is afforded flexibility when dealing with link failures.

The basic backbone routing between two dominators works as follows. An upstream dominator checks its connection list for connections to the next dominator. If two-hop connections exist, the dominator chooses the connector with the most recent timestamp, otherwise it chooses the connector of the most recent three-hop link. The dominator then sends a unicast message containing the payload to the connector. A two-hop connector directly sends the packet to the target dominator, while a three-hop connector repeats the process to select a second connector.

If a dominator or connector is unable to send the packet, it receives a MAC-level error; connectors forward the error to their upstream dominator. The upstream dominator, which buffers recently sent packets, selects another connection and tries again. Only when a statically-defined retry limit is reached or when no connections is available is the error reported to the upper routing-level protocol. This error initiates a route nack packet that is sent back to the source dominator, and triggers an attempt to salvage the packet at that level.

4 Evaluations

4.1 The Scenarios

Our simulations use Glomosim [20] and the parameters are set as follows: bandwidth is 2 Mb/s, radio frequency is 2.4 GHz and transmission range is 250 m.

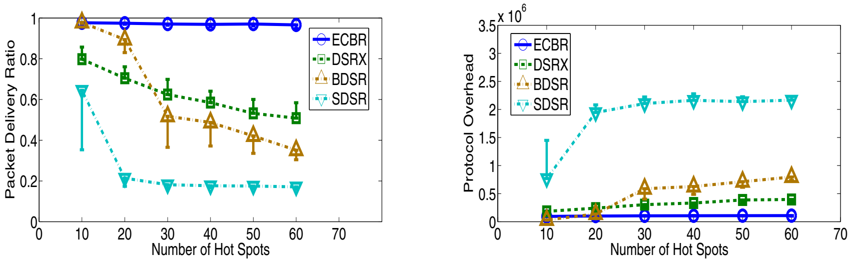
Most of the evaluations are conducted with a total of 200 nodes randomly distributed in an area of $1500m$ x $750m$. We choose an area that is three times larger than previous work [1,2] to avoid the formation of chain-like backbones, which would tend to favour backbone approaches.

Each simulation lasts 910 seconds. We avoid startup and shutdown effects by waiting 50 seconds before opening the first CBR connection and by stopping measurements 10 seconds before the end of the simulation. We use Random Waypoint [4] to model mobility. The minimum speed is set to 1.0 m/s and the pause period is set to 60 seconds. The maximum speed is set to 5.0 m/s in most of the simulations to investigate performance in scenarios with human mobility. However, in Sections 4.3 and 4.4, we examine ECBR performance for a variety of faster velocities.

We adopt a multi-destination, varying-duration CBR traffic pattern, in which a set of destinations are randomly selected to act as communication hot spots. Source nodes are chosen randomly from the network and destinations are chosen randomly from the list of hot spots. The duration of each CBR connection is a parameter that we vary. We are thus able to simulate a wireless environments where transient traffic exists. The size of each CBR packet is 256 bytes and packets are generated at a fixed rate of 1 pkt/sec. We vary network load by changing the number of concurrent CBR connections.

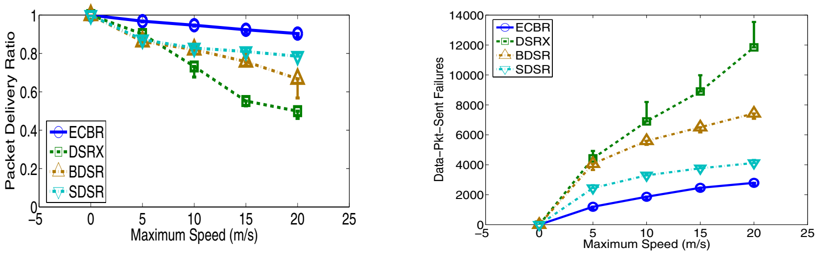
4.2 The Protocols

We compare ECBR to three other protocols, two versions of DSR that we call SDSR and BDSR and a hierarchical protocol we built called DSRX. SDSR is the



(a) Packet Delivery Ratio (b) Protocol Overhead

Fig. 1. Increasing the Number of Hot Spots



(a) PDR (20 CBRs, 40 Hotspots) (b) Stale Caches (10 CBRs, 10 Hotspots)

Fig. 2. Increasing Mobility

standard version of DSR while BDSR uses a bigger routing cache: 200 entries instead of 64. We implemented DSRX as a model for a class of hierarchical protocols that are similar to DSRCEDAR. It uses the same backbone algorithm as ECBR but with a standard implementation of DSR built on top of it.

We compared DSRX to the reported performance of DSRCEDAR to determine the extent to which our DSRX results might generalize. Our results show that DSRX performance is similar to DSRCEDAR published results but in some cases up to 5% worse. This difference is due to the fact that DSRCEDAR modifies the MAC layer control messages and uses promiscuous mode to improve its core broadcast efficiency.

Finally, we fix the DOMINATOR heartbeat interval at 0.5 s and DOMINATEE heartbeat interval 5 s. In fact, the optimal setting for heartbeat intervals is 0.25 s for dominators and 2.5 s for dominatees in most of the cases we simulated. We use a set of more conservative settings to avoid any biased results with values carefully tuned to characteristics of our particular workload.

4.3 The Impact of Transient Traffic

This section varies three parameters: the number of hot spots, CBR duration, and mobility, to examine the impact of transient traffic. All of the data reported is the mean of ten trials; the standard deviations are also presented. We only show half of the symmetric deviation to avoid cluttering the figures.

Varying the Number of Hot Spots. Figure 1 shows the impact of varying the number of hot spots from 10 to 60 (i.e., 5% to 30% of the nodes); the number of concurrent connections is fixed at 40, connection duration at 10 s and mobility at 1–5 m/s. ECBR outperforms the other algorithms in virtually every situation; BDSR matches the performance of ECBR in the case of 10 hot spots. The performance gap between ECBR and the other algorithms widens as the number of hot spots increases.

Increasing Connection Lifetime. We also vary connection lifetime from 5 s to 50 s and fix the number of hot spots is fixed at 40. In this scenario, ECBR again outperforms all the other protocols in every case. While ECBR's performance advantage narrows as connection duration increases, it provides the best packet delivery ratio of 97% even at 50 s; the best of the others, BDSR performs at 94%. With a more realistic scenario in which 25% of the connections are short-lived, at 5 s, and the remainder long-lived, at 850 s, ECBR delivers 97% of packets (std. dev.: 1.1%), BDSR 76% (std. dev.: 20%), SDSR 27% (std. dev.: 8%) and DSRX 68% (std. dev.: 5%).

Increasing Mobility. We evaluate how these protocols handle transient traffic under different mobility settings and show the results in Figure 2(a). We vary the maximum node mobility between 0 m/s (no mobility) and 20 m/s. The number of hot spots is fixed at 40, the number of concurrent connections at 20, CBR lifetime at 10 s. ECBR again dominates the other algorithms by a substantial margin, which suggests that ECBR is best able to keep track of topology changes. We can also see that, with high mobility of 1-20 m/s, BDSR's large cache can harm its performance compared to SDSR.

4.4 Analysis of DSR's Performance

The fundamental reason that causes both BDSR and SDSR to perform poorly is that transient connections cause more route-cache misses than long-term ones do, and thus generate more route-discovery messages that results in severe network congestion.

Cache Misses. We count those *real* cache misses that indeed initiate network-wide route discoveries. Our results show that the number of cache misses of both DSRs increases significantly when either the connection duration decreases or the number of hot spots increases. This is because, given a fixed network load, reducing the CBR lifetime increases the number of connections in the network. Cache misses are thus more likely to happen. Also, with more hot spots more unique routes are used and thus route-caching is less effective. For example, BDSR's cache misses jump from 2152 to 6832 as the number of hot spots increases from 20 to 30; SDSR's increases to 8610 when there are only 20 hot spots in the network.

Mobility. Transient communication can worsen the well-known stale-cache problem of DSR. This is because a cached route might become inaccurate next time when the node uses it or provides it to other route discoveries due to topology changes, especially when node mobility is high. We compare the algorithms when node mobility varies from no-mobility to mobility chosen randomly between 1 and 20 m/s. The other parameters are the same as in Section 4.3 except that we use only 10 hot spots and 10 concurrent CBRs so that none of the four algorithms experience significant congestion.

Figure 2(b) present the number of times each algorithm attempts to use a broken link between two nodes. When mobility is high, both BDSR and DSRX are impacted by stale routes much more than ECBR or SDSR. The reason that SDSR is better is that its caches are smaller than BDSR. High-mobility is a case where big caches do not improve the performance of DSR.

4.5 Analysis of DSRX's Performance

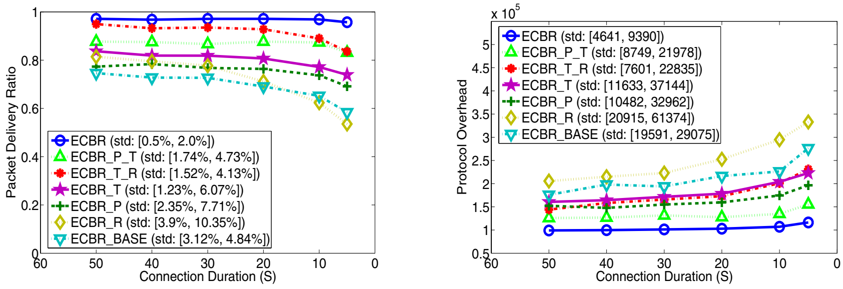
Our simulation results show that even the naive hierarchical algorithm, DSRX, can consistently better handle transient traffic in high-congestion and low-mobility scenarios. However, DSRX's performance is always substantially worse than ECBR's. This section discusses the limitations of DSRX in details.

The Bottleneck Backbone. Even though the backbone is able to reduce the range and frequency of broadcast route discovery messages, it confines both data and control packets to the backbone. This not only results in less spatial reuse but also makes the backbone the potential bottleneck for performance improvement. For example, in the experiments where we vary the number of hot spots and CBR duration, DSRX generates at least twice as much overhead as ECBR. This amount of control packets together with data packets has already caused enough congestion problem to break the backbone apart.

Stale Caches. We can also see from Figure 2(b) that DSRX suffers the most from stale caches. This is because it simply lays DSR on top of the backbone and thus drops as many data packets due to stale caches as BDSR at 5 m/s maximum mobility. It is worse than BDSR as node mobility increases because DSRX does not use the optimizations deployed in DSR such as gratuitous error messages and promiscuous mode to alleviate the impact of outdated routes.

4.6 Detailed Evaluation of ECBR

ECBR outperforms both DSRs and DSRX by a substantial margin in all the scenarios, regardless of the degree of contention and mobility. Figure 3 shows how ECBR's performance gain can be attributed to each of the three features that it has adopted. It evaluates seven versions of ECBR under the same conditions as the connection lifetime experiment in Section 4.3; the label indicates which of the three features is enabled. If a feature is not listed it is disabled; ECBR_BASE



(a) PDR Gain

(b) Overhead Gain

Fig. 3. Break-down of ECBR's Performance Gain

excludes all three features. The labels for the features are P for piggybacked prefetching, T for timestamped caches and R for proactive route adaptation and recovery.

When each of the three features is examined in isolation, timestamped caches provide the biggest PDR benefit, while prefetching alone reduces most of the overhead. It results in more benefit, however, when piggybacking and timestamping are combined. This behaviour is not surprising, as the features are largely complementary. Prefetching puts more routes in caches and timestamping flushes stale routes from them. Finally, using proactive route adaptation and recovery alone always generates more protocol overhead because it keeps reporting route-error messages back to the upstreaming dominator whenever a backbone link breaks. However, when added to the other two features, we see that it not only significantly increases PDR but also reduces the overall overhead. This behaviour shows that proactive, local route information allows ECBR to re-route many packets that would otherwise be dropped. The danger with any such scheme is that when delivery fails due to congestion not mobility or failure, salvaging can increase overhead without improving PDR.

4.7 Discussion of ECBR Limitations

Scalability of Piggybacked Prefetching. The effectiveness of piggybacked prefetching relies on the assumption that the marginal cost of increasing message size is small compared to the benefit of having more routes cached. Therefore, to evaluate the limits of piggybacking scalability, we simulate a high-mobility (20 m/s), high-contention (40 hot spots) scenario while artificially constraining the number of dominator-dominatee pairings that can be piggybacked. Due to the IP MTU limit, the maximum number of piggyback entries that fit in a reply packet is 69 and we vary this limit between 10 and 69 in this experiment. The simulation results suggest that a network that was either twice as dense or in which routes were twice as long might see PDR drop from 90% to around 88% and see overhead increase by about 5%.

Sending Rate (Bottleneck Issue). As a backbone approach, ECBR also suffers from the bottleneck problem. Here, we investigates this problem further by varying the sending rate from 1 pkts/s to 7 pkts/s. We use a static network and only 20 long-term CBR connections, each of which lasts 850 seconds. This setting makes the piggybacking and timestamping features of ECBR useless. The results show that ECBR still shares the bottleneck problem with all the other backbone approaches. Its performance starts to drop below 90% when sending rate is at 4 pkts/s. At the sending rate of 6 pkts/s, it fails to deliver over 40% of the data packets. This suggests that backbone approaches are *not* suitable for long-term heavy-load network traffic. The source nodes should choose proper ad-hoc routing algorithms adaptively according to its traffic type.

Network Area. Network area is another parameter that we held constant in the previous experiments. We vary it between 1500 m x 250 m and 1500 m x 1500 m in this section. The other parameters are set to the same as the hot-spot scenario in Section 4.3 except that the number of hot spots is kept at 60. The simulation result show that ECBR's performance gradually degrades from 99.3% to 90.5% as we increase the network area by six times. This is because the backbone complexity grows as the network area increases; the number of dominator nodes and backbone links increases proportionally to the increase of network area. Therefore, the congestion impact of one route discovery packet worsens in a bigger network since it is broadcast on the backbone. There are two ways to alleviate this problem. One is to use a spanning tree for route discovery on the backbone instead of broadcast, as ECBR does. Another is to make the backbone small, which has been the research focus of MCDS in recent years.

5 Conclusion

This paper describes the design of a backbone-based, hybrid routing protocol called ECBR that works well with transient traffic. The protocol has three novel features. First, route information is prefetched into caches by piggybacking multiple routes in a single route-discovery reply message. Second, cache entries are timestamped, providing a heuristic for discarding redundant, out-of-date routes from caches. Third, cached routing information specifies only the backbone nodes on the path to the target, leaving each backbone node free to select any two- or three-hop path to the next backbone node on a route. Our algorithm significantly out performs the others when connections are short, the network is congested or mobility is high.

References

1. Broch, J., Maltz, D., Johnson, D., Hu, Y.-C., Jetcheva, J.: A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols. Mobicom '98
2. Das S., Perkins, C., Royer, E.: Performance Comparison on Two On-Demand Routing Protocols for Ad Hoc Networks. Inforcom '00

3. Johansson, P., Larsson, T., Hedman, N., Mielczarek, B.: Routing Protocols for Mobile Ad Hoc Networks - a Comparative Performance Analysis. MOBICOM '99
4. Johnson, D., Maltz, D.: Dynamic Source Routing in Ad Hoc Wireless Networks. Chapter 5, Mobile Computing. Kluwer Academic Publishers. 1996
5. Perkins, C., Royer, E.: Ad-Hoc On-Demand Distance Vector Routing. Second IEEE Workshop on Mobile Computing Systems and Applications Feb., 1999
6. Balazinska, M., Castro, P.: Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. MobiSys '03. May., 2003
7. Balachandran, A., Voelker, G., Bahl, P., Rangan P.: Characterizing User Behavior and Network Performance in a Public Wireless LAN. SIGMETRICS '02. Jun., 2002
8. Henderson, T., Kotz, D., Abyzov, I.: The Changing Usage of a Mature Campus-Wide Wireless Network. MobiCom '04. Sep., 2004
9. McNett, M., Voelker, G.: Access and Mobility of Wireless PDA Users. Technical Report CS2004-0728, Department of Computer Science and Engineering, University of California, San Diego. Feb., 2004
10. Iwata, A., Chiang, C.-C., Pei, G., Gerla, M., Chen T.-W.: Scalable Routing Strategies for Ad Hoc Wireless Networks. IEEE Journal on Selected Areas in Communications. Aug., 1999
11. Ramanathan, S., Steenstrup, M.: Hierarchically-Organized, Multihop Mobile Networks for Multimedia Support. Mobile Networks and Applications. 1999
12. Das, B., Bharghavan, V.: Routing in Ad-hoc Networks Using Minimum Connected Dominating Sets. ICC '97. Jun., 1997
13. Das, B., Sivakumar, R., Bharghavan V.: Routing in Ad-hoc Networks Using a Spine. Proceedings of the IEEE International Conference on Computers and Communications Networks. Sep., 1997
14. Xu, K., Hong, X., Gerla, M.: An Ad Hoc Network with Mobile Backbones. ICC '02. Apr., 2002
15. Jiang, M., Li, J., Tay Y.-C.: Cluster Based Routing Protocol (CBRP) Internet-Draft, draft-ietf-manet-cbrp-spec-01.txt, 1999, Work in progress
16. Sinha, P., Sivakumar, R., Bharghavan, V.: CEDAR: a Core-Extraction Distributed Ad-hoc Routing Algorithm. IEEE Journal on Selected Areas in Communications. Aug., 1999
17. Sivakumar, R., Sinha, P., Bharghavan, V.: Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures. INFOCOM '99. Aug., 1999
18. Alzoubi, K., Wan, P.-J., Frieder, O.: Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks. MobiHoc '02. Jun., 2002
19. Cai, K., Feeley, M. J., Hutchinson, N. C.: Routing Transient Traffic in Mobile Ad Hoc Networks. TR-2006-19, Department of Computer Science, University of British Columbia. Sep., 2006
20. Zeng, X., Bagrodia, R., Gerla, M.: GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. PADS '98. May, 1998

Comparison of Two Self-organization and Hierarchical Routing Protocols for Ad Hoc Networks

Betânia Steffen Abdallah Goncalves[1], Nathalie Mitton[2], and Isabelle Guérin-Lassous[3]

[1] INSA de Lyon
[2] POPS USTL - INRIA - CNRS
Nathalie.Mitton@lifl.fr
[3] LIP - UCBL
France
Isabelle.Guerin-Lassous@ens-lyon.fr

Abstract. In this article, we compare two self-organization and hierarchical routing protocols for *ad hoc* networks. These two protocols apply the reverse approach from the classical one, since they use a reactive routing protocol inside the clusters and a proactive routing protocol between the clusters. We compare them regarding the cluster organization they provide and the routing that is then performed over it. This study gives an idea of the impact of the use of recursiveness and of the partition of the DHT on self-organization and hierarchical routing in ad hoc networks.

Keywords: *ad hoc* networks, hierarchical routing, DHT, comparison, simulation.

1 Introduction

Ad hoc networks are composed of independent terminals that communicate via wireless interfaces. Every mobile node can move everywhere, disappear or appear in the network at any time. In order to allow communications between any pair of nodes that are not within communication range, intermediate nodes need to relay the messages. A routing protocol is thus required to provide routes between any pair of nodes in the network. *Ad hoc* networks have been becoming very popular these last years due to their easiness of use and deployment (no infrastructure needed). Their applications range from the network extension to spontaneous networks in case of natural disaster where the infrastructure has been totally destroyed, to the monitoring and the gathering of data with wireless sensor networks. Due to the dynamics of wireless networks and the terminal specificities (limited memory size and computing capacities), the routing protocols for fixed networks are not adapted. *Ad hoc* routing protocols proposed in the MANET working group at IETF[1] are all flat routing protocols, with no hierarchy. If flat routing protocols (proactive[2] and reactive[3] routing protocols) are quite effective on small and medium size networks, they are not suitable for large scale or very dense networks because of bandwidth and processing overheads they generate [18,8]. A common solution to this scalability problem is to introduce a hierarchical routing.

[1] http://www.ietf.org/html.charters/manet-charter.html

[2] Nodes permanently keep a view of the topology. All routes are available as soon as needed.

[3] Routes are searched on-demand. Only active routes are maintained.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 25–37, 2006.

A hierarchical routing relies on a self-organization of the network in a specific partition, called *clustering*: the terminals are gathered into clusters according to some criteria, each cluster being identified by a special node called *cluster-head*. In this way, nodes store full information concerning nodes in their cluster and only partial information about other nodes. In addition to its scalable feature, such an organization also presents numerous advantages as to synchronize mobile nodes in a cluster or to attribute new service zones. Based on this partition, different routing policies are used in and between clusters:

(i) either proactive routing in the clusters and reactive routing between the clusters, which is the most common approach in the literature [5,7,16],
(ii) or reactive routing in the clusters and proactive routing between the clusters [17,11].

In this paper, we study the second approach of the hierarchical routing, *i.e.* using a reactive routing in the clusters and a proactive routing between the clusters. Such a hierarchical routing implies an indirect routing, *i.e.* the routing is performed in two steps: the look-up step that locates the destination node and the routing step to directly join it. Such an approach for hierarchical routing seems to us more scalable and more promising than the first one. Indeed, most of the clustering algorithms found in the literature provide a constant number of clusters when the intensity of nodes increases [4,9,15]. Thus, when the node density increases, there are still $O(n)$ nodes per cluster and using a proactive routing scheme in each cluster implies that each node still stores $O(n)$ routes, which is not more scalable than flat routing.

As far as we know, nowadays, in the literature, only two works propose this reverse hierarchical routing approach. They mainly differ in the self-organized structure they provide. The first one is called the density-based protocol [11] and uses a simple clustering structure. The second one is SAFARI [17] and uses a recursive hierarchical clustering structure. Both protocols use a DHT to perform the indirect routing.

In this paper, we compare SAFARI and the density-based protocol to analyze the impact of the use of recursiveness in the self-organization. Comparisons are lead regarding to the clustering structure provided and the quality of each indirect routing step (look-up and final routing). We will see that the main differences concern the stabilization time and the way the DHT has to be implemented over the resulting clustering structure. The remaining of this paper is organized as follows. Section 2 summarizes the indirect routing and the DHT principles. Section 3 briefly describes SAFARI and the "density-based" algorithm. Section 4 presents the simulation model. The cluster organizations are analyzed and compared in Section 5 while Section 6 provides a comparison of both routing steps and DHT utilization. Finally, in Section 7, we discuss some improvements and future works.

2 Indirect Routing and DHT

The goal of this paper is to compare two ways of applying a proactive routing protocol between clusters and a reactive routing protocol inside clusters. For such a hierarchical routing, an indirect routing is required. Indeed, a proactive routing protocol between clusters means that, knowing the cluster of the target node, the source node is able to route toward this cluster without any extra information. Then, once the message has reached the destination cluster, it can reach the target node thanks to a reactive routing protocol inside this cluster. Nevertheless, to be able to proceed like this, the source node has to know to which cluster the target belongs to. This routing process thus needs a preliminary step which allows the source node to learn the location of the target node. Routing is thus performed in two steps: a first step called *look-up* which allows to learn the location of the destination and a second step which sends the message toward this location.

The most common tool used to performed an indirect routing is the Distributed Hash Table (DHT). It allows to share routing information over the nodes of the network. In this way, the required memory size on each node is minimized, which allows network scalability. A DHT uses a virtual addressing space $\mathcal{V}$. Each node u of the network is assigned a "virtual address" $VId_u \in \mathcal{V}$ as well as a partition of this virtual space. Let denote by I_u the partition of $\mathcal{V}$ assigned to node u. A $hash$ function associates the identifier of every node v to a virtual $key_v \in \mathcal{V}$ (one says that the id of v is *hashed* into the $key_v \in \mathcal{V}$). v will then register its location at node y such that $key_v \in I_y$. This allows to identify some rendezvous points: y is a rendezvous point for node v. The $hash$ function is known by every node of the network and may then be used by a source to identify the rendezvous point that stores the location of the target node.

The routing process used by the source to reach the rendezvous point and then the final node is either dependent or independent of the virtual space of the DHT. In DHT-independent routing schemes, the virtual address is not used for the routing operation. The nodes generally know their geographical coordinates, either absolute (by using a GPS for example) or relative, that is the location information they associate to the key. By performing $hash$(target), a node u gets the geographical coordinates of a rendezvous area $\mathcal{A}$. u then applies a geographical routing protocol to join a node v laying in $\mathcal{A}$ and that is aware of the geographical coordinates of the target node. From it, node u is able to reach the destination by performing a geographical routing once more [1,2,14].

In DHT-dependent routing schemes, the virtual space of the DHT is used not only for locating but also for routing toward the destination. The virtual address is dependent of the location. In this way, the consistency of the routing protocols rely on the consistent sharing of the virtual addressing space among all nodes in the network. The routing is performed over the virtual structure. In such scenarii, a node u performing $hash(w)$ gets the virtual address of the rendezvous point v. From it, u routes in the virtual space to v and obtains the virtual address of w. Thus u is able to reach w by routing in the virtual space again. The routing scheme used is generally a greedy routing: "Forward to the neighbor in the virtual space which virtual address matches the best the virtual address of the destination". This is for instance the case of Tribe [20] or L+ [3]. The main challenge here is to disseminate the partitions of the virtual space in such a manner that the paths in the virtual space are not much longer than the physical routes.

In most of the proposals, both indirect routing phases (look-up and final routing) are performed in the same way, either in the physical network (for DHT-independent routings) or in the virtual one (for DHT-dependent routings). SAFARI uses a DHT-dependent routing scheme. In the density-based approach, each routing step is performed in a different manner. The look-up is routed by using the virtual address of the rendezvous point (DHT-dependent) whereas the routing toward the final destination is performed over the physical network (DHT-independent).

There exist many ways to distribute the virtual addressing space over the nodes of the network, as it is strongly linked to the way the routing is attended to be performed. Each proposal introduces its own way to distribute the virtual space. As we will see in Section 3, SAFARI and the density-based algorithm strongly differ in the way they distribute the partitions of the DHT addressing space. In SAFARI, the self-organization of the network and the distribution of the DHT intervals are performed simultaneously whereas in the density-based algorithm, both steps are distinct.

3 SAFARI and the Density-Based Algorithm

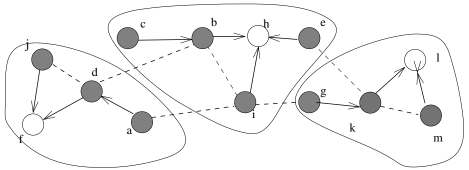
In this section, we describe the two protocols we have considered: the SAFARI protocol [17] and the density-based protocol [11]. We only give the main ideas and basis of the clustering, locating and routing steps of the two protocols. For more details, please refer to their respective references. Both protocols pursue the same goal to offer a network organization allowing a scalable routing. We will see that the two protocols strongly differ in the way they self-organize the network and in the DHT implementation. The density-based algorithm computes an organization with only one level and distributes the DHT addressing space afterward. The SAFARI heuristic proposes a recursive cluster hierarchy: nodes are grouped into clusters (cells or level-1 clusters), clusters into super-clusters (super-cells or level-2 clusters) and so on. This hierarchical structure is built simultaneously with the DHT implementation.

3.1 Density-Based Heuristic

Cluster formation: The cluster formation is based on a metric called ”density”, previously introduced in [10]. The density of a node u is the ratio between the number of links[4] in its neighborhood (links between u and its neighbors[5] and links between two neighbors of u) and the number of its neighbors. To compute clusters, each node locally computes its density value and periodically broadcasts it locally to its neighbors. Each node is thus able to compare its own density value to its neighbors' density values and decides by itself whether it joins one of them (the one with the highest density value) or it wins and elects itself as a cluster-head. Figure 1 illustrates the cluster formation. Node i has elected node h as its parent. In case of ties, the node with the lowest identity (denoted Id henceforth) wins. This is the case in Figure 1 for instance for nodes j and f which both have the same density value but as $Id(f) < Id(j)$, node j joins node f. A node's parent can also have joined another node and so on (node c joins node b which joins node h). A cluster extends itself until it reaches another cluster and the cluster radius is thus not defined *a priori*. Clusters are then identified by the Id of the cluster-head. By performing this joining process, we actually build a directed acyclic graph (DAG). The clustering process builds clusters by building a spanning forest of the network. One-level hierarchy is built.

[4] There is a link as soon as two nodes are within transmission range.

[5] Two nodes are neighbors if there exists a link between them.



Node	a	b	c	d	e	f	g	h	i	j	k	l	m
Density	1	1.25	1	1.25	1	1.5	1	1.33	1.25	1.5	1.25	1.5	1.5

Fig. 1. Example of trees and clusters built by the density-based algorithm. Dashed links represent the wireless links which do not belong to the DAG, arrows represent a link in a tree directed from a node to its parent, cluster-heads appears in white.

Distributing the DHT addressing space [11]: In the density-based algorithm, the virtual space $\mathcal{V}$ is shared in each cluster among the branches of the trees. $\mathcal{V}$ is first shared by the cluster-heads between themselves and their children, proportionally to the size of the subtree of each of them. Then, each internal node recursively shares, between itself and its children, the partition given by its parent, and so one, till reaching the leaves of the trees. This step has a time complexity in $O(tree\_length)$. The tree length has been proved to be bounded by a low constant [13].

Indirect routing [11]: In the density-based protocol, the two steps of the indirect routing are not performed in the same way. The first step (look-up and registration) is routed in the DHT space, whereas the second step is performed in the physical space. The location of nodes is the identifier of their cluster. The $hash$ function returns one virtual address which exists in each cluster. Each node registers its location in each cluster, at nodes identified by the $hash$ function. In this way, the location of every node in the network is contained in each cluster. The look-up can thus be performed locally in a cluster. Routing in the DHT logical space is done by using an interval routing over the clustering tree. Interval routing allows to minimize information routing stored at each node. As the partition of the DHT in each tree of the DAG, the interval routing provides the shortest paths in the tree [19]. The second step of the indirect routing is performed in the physical space: if the source and the target node are in the same cluster, a reactive routing is performed by using an energy-efficient broadcasting operation described in [12]. Otherwise, the target cluster is reached by using the path of clusters returned by the inter-cluster proactive routing.

3.2 The SAFARI Project

Cluster formation: SAFARI provides a l-level hierarchy of cells. It is built recursively, based on an automatic self-selection of nodes as cluster-heads (also called *drums*). At the initializing step, nodes have to wait during a random time before deciding to self-declare themselves as level-1 drums or not. A level-i drum may decide to up or down its level according to how far it is from other level-$(i + 1)$ and level-(i) drums. If a level-i drum does not hear from any level-$(i+1)$ drum within a distance lower than a threshold depending on i, it self-declares itself as a level-$(i + 1)$ drum. If two level-i drums are within a distance lower than another threshold also depending on i, only the greatest

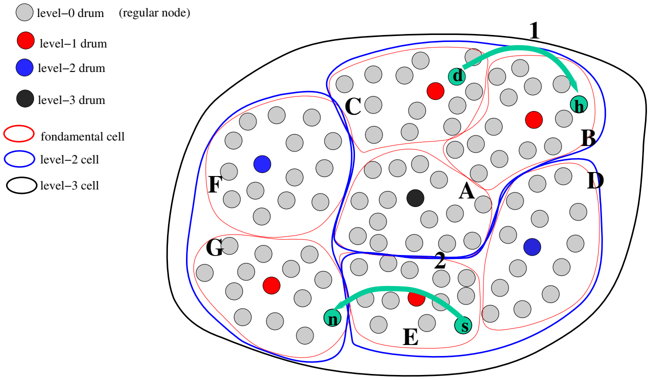


Fig. 2. Example of a SAFARI cluster organization. Fundamental cells have the following coordinates : A=[256 387 966], B=[102 387 966], C=[071 387 966], D=[308 659 966], E=[285 659 966], F=[003 741 966], G=[593 741 966]. Nodes S and D thus have the following DART: S =([256 387 966];[308 659 966];[285 659 966];[003 741 966]);D =([256 387 966];[102 387 966];[071 387 966];[308 659 966];[003 741 966]).

Id remains a level-i drum, the other one becoming a level-$(i-1)$ drum back. A level-i drum is also a level-j drum for all $0 \leq j \leq i$. With such a process, level-i cells are gathered into level-$(i+1)$ cells and so on. Plain nodes are seen as level-0 cells and the unique highest level cell (level-l cell) gathers all the nodes of the network. Each level-i drum joins a level-$(i+1)$ drum. All level-i drums joining the same level-$(i+1)$ drum belong to the same level-$(i+1)$ cell. The radius of each cell is bounded, according to its level to D_i hops, i being the cell level. Each level-i drum periodically emits a packet called *beacon* every T_i, T_i depending on the level i. The higher the drum level, the longer the beacon emission period. These beacons are forwarded by all nodes within $h \times D_i$ hops, h being a constant. All nodes store all the beacons they forward in a Drum Ad Hoc Routing Table (DART). This hierarchical algorithm gives a unique ancestry for each node. Figure 2 gives a SAFARI cluster organization that has a recursive structure.

Distributing the DHT addressing space: Contrariwise of the density-based algorithm, SAFARI distributes the DHT virtual addressing space when building the clustering structure. Indeed, each node is assigned a coordinate based on the drum structure. If $COORD(d_i)$ is the coordinate of a level-i drum d_i and $PARENT(d_i)$ is the level-$(i+1)$ drum joined by d_i, we have: $COORD(d_i) = COORD(PARENT(d_i)).Rand$ where $Rand$ is a uniform random number. Level-0 drums (regular nodes) are leaves in the hierarchy and their coordinate is $COORD(d_0) = COORD(PARENT(d_0))$. Thus, all nodes in a fundamental cell have the same coordinate. The coordinates of the nodes form the logical Id space $\mathcal{V}$ of the DHT. $\mathcal{V}$ is shared over every node of the network while in the density-based, $\mathcal{V}$ is shared as many times as there are clusters.

Indirect routing: In SAFARI, both steps of the indirect routing, *i.e.* the look-up and the routing toward the final target, are performed the same way in the DHT space. The underlying idea of the look-up process is that generally, nodes communicate more with other nodes that are close to them. When performing $hash(v)$, the $hash$ function returns k different coordinates for each level i $(2 < i \leq l)$, that is $k * (l - 1)$ rendezvous points. These coordinates will be used by v to identify the nodes at which it has to register its location and by other nodes looking for node v. Node v will register its coordinate k times in each level-i cell it belongs to, for i ranging from 2 to l (as every node has the same coordinate in the same level-1 cell, nodes do not register in their level-1 cell). As node coordinates are randomly chosen by the nodes themselves, the coordinates returned by the hash function are not necessary hold by a node. Therefore, nodes look for nodes in their DART which coordinate is the closest to the one returned by the $hash$ function. A node x looking for node v first sends a look-up request in the level-1 cells that belong to the same level-2 cell than itself by selecting the closest coordinate in its DART (let's say node u). If it does not find any right rendezvous point, it looks in level-2 cells, and so on, upping the level after each look-up failure. Otherwise, u does the same to forward the request of node x till reaching the fundamental cell of the rendezvous node r. There, the location is stored either by the drum of this cell, or by a node in the cell, reached thanks to a reactive routing inside the fundamental cell (it is not clear in SAFARI). If r knows the coordinate requested, it returns them to u which will then be able to reach x by the same way it had reached r. If r does not know the coordinate, x reiterates the look-up process at an upper level and so on.

4 Simulation Model

We use a simulator we developed in C language that assumes an ideal MAC layer, *i.e.* it does not consider interferences and packet collisions occurring at the MAC layer. Voluntarily, we did not use a network simulator which simulates a realist MAC layer because, as we wish to compare two network layer protocols, we do not want to be mistaken about their performances if any problem occurs at some lower layer. Nodes are randomly deployed in a 1×1 square using a Poisson Point Process (node positions are independent) with various levels of intensity λ (in such processes, λ represents the mean number of nodes per surface unit). Every node has the same transmission range R. There is an edge between two nodes if and only if their Euclidean distance is at most R (derived from the Unit Disk Graph model [6]). All the given results have a 95% - confidence interval. Both algorithms are compared over the same samples of node distribution.

Simulation parameters: In order to fairly compare both protocols, parameters have been tuned similarly for SAFARI and the density-based algorithms. In SAFARI, the cell radius D_i and the periods of beacon transmission T_i need to be determined for the lowest hierarchical level $i = 1$ and then, upper levels parameters are computed from it. Previous simulations of the density-based algorithm have shown that it provides a cluster radius between 3 and 4 hops [10]. Thus, in our simulations, we have fixed $D_1 = 3$ in SAFARI in order to get similar level-1 clusters in both protocols. Note that $D_1 = 3$ is also the value set by the authors of SAFARI in [17]. Still in order to compare both protocols, we assume that the packets exchanged in the density-based protocol are emitted every T_1 time units (period of level-1 beacons transmission in SAFARI). In SAFARI, at the initializing step, nodes have to wait during a random time drawn in $[0..X]s$ before deciding to self-declare themselves as a drum or not. For all these parameters, we used the same values as the authors of SAFARI, *i.e.* $T_1 = 2s$, $X = 5s$ and a SAFARI node registers its location $k = 3$ times at each level.

5 Cluster Formation

In this section, we provide an analysis of the differences and similarities of both protocols regarding the cluster formation. As already mentioned, the main difference between both protocols is that the density-based algorithm provides a 1-level hierarchy whereas SAFARI builds a recursive hierarchy of l levels. Nodes in the density-based algorithm only use two-hop-away information (to compute their density value and then to elect their parent) while level-i drums in SAFARI need information in their DART, collected up to D_{i+1} hops. Thus, to build fundamental (level-1) clusters, nodes need to collect information up to D_1 hops. In order to fit different kinds of topologies and environments, the radius of clusters in the density-based algorithm is not set *a priori* whereas in SAFARI, the maximum radius D_i of level-i clusters has to be previously fixed ($1 \leq i \leq l$).

In the density-based algorithm, each new node entering the network checks its neighborhood, computes its density value and elects its parent. The algorithm stabilizes pretty quickly in a time proportional to the cluster radius [13]. In SAFARI, at the initializing step, nodes have to wait during a random time before deciding to self-declare themselves as drums or not. Moreover, the drum selection decision is based on the DART and thus on the beacons emitted by the drums. The stabilization time is thus linked to the initializing random back-off and also to the beacon frequency T_i (thus T_1 for the fundamental cells).

We have compared by simulation the clusters built by both protocols. The simulation model is described in Section 4. Note that, in both cases, because of the clustering algorithms, only the node degree impacts the cluster/drum characteristics of the clustering structures[6]. The network expansion only impacts the number of hierarchical levels built by SAFARI: between 3 and 4 levels for a 500-node topology with radius node set higher than 0.1 and between 2 and 3 levels otherwise. We will see later that this number of levels strongly impacts the stabilization time and the look-up performances in SAFARI.

Table 1 shows the different cluster characteristics we computed for different values of R that correspond to different values for the mean degree. In order to fairly compare these two protocols, all these data concern only the features of the level-1 clusters for SAFARI. The diameter of a cluster is the maximum number of hops between any pair of nodes of this cluster. Results show that clusters present similar average characteristics whatever the node degree (similar amount of clusters and diameters). However, even if the average values are similar, we can note that the density-based algorithm is much more stable as the clustering stabilization time and its standard deviation σ shows. Note that SAFARI does not stabilize at every time. This is due to the fact that, at the initialization step, the first drum that appears in the network is the one which random waiting time has expired the first. The cells are henceforth built according to the order of the waiting time periods of the nodes. As this waiting period is random, the

[6] Therefore, in topologies like grid or chain, clusters features remain the same.

Table 1. Some cluster characteristics for both metrics over a 500-node Poisson distribution and for different values of R (which gives different values of $\overline{\delta}$)

$\overline{\delta}$	15.7		18.8		22.0	
	Density	SAFARI	Density	SAFARI	Density	SAFARI
# clusters	11.70	16.2	10.08	12.6	8.06	11.4
Diameter	4.99	4.67	5.52	4.62	5.50	4.76
Clustering stabilization time	5.27	107.67	5.34	113.41	5.33	91.95
$\sigma(Clustering\,stabilization\,time)$	0.63	132.41	0.74	135.56	0.85	123.69
DHT stabilization time	11.29	107.67	11.52	113.41	12.07	91.95
$\overline{\delta}$	25.1		28.3		31.4	
	Density	SAFARI	Density	SAFARI	Density	SAFARI
# clusters	7.03	9.10	6.15	8.10	5.57	7.40
Diameter	5.65	4.83	6.34	4.77	6.1	4.73
Clustering stabilization time	5.34	90.55	5.43	60.61	5.51	61.97
$\sigma(Clustering\,stabilization\,time)$	0.99	111.18	1.21	115.58	1.44	118.69
DHT stabilization time	12.02	90.55	12.29	60.61	12.51	61.97

clusters formed in the network may not be distributed in a good way, *i.e.* it is possible that nodes have to up and down their levels many times before stabilizing, which may take a long time. It is also possible that a node oscillates between two different levels trying to respect all conditions of the level selection algorithm. As long as a node oscillates, the network never stabilizes. The clustering stabilization time is given in time units in Table 1. It represents the number of steps required before the cluster formation has stabilized. For SAFARI, results have been considered into computations only when the algorithm converges. We can notice that, for the density-based algorithm, the node intensity in the network does not impact the stabilization time. SAFARI is much longer to stabilize than the density-based heuristic (between 12 and 20 times longer) and that the clustering stabilization time of SAFARI fluctuates a lot as the standard deviation $\sigma(clustering\,stabilization\,time)$ shows.

The DHT stabilization time, also given in Table 1, represents the number of steps required before both the clustering structure has stabilized and the DHT addressing space has been shared between nodes. Note that for SAFARI, this corresponds to the clustering stabilization time as both operations are performed simultaneously. Contrarily, the density-based algorithm needs some more steps to distribute the virtual addressing space over each cluster. Nevertheless, note that this number of additional steps remains low and constant as it is equal to twice the tree depth which is bounded by a constant.

6 Look-Up and Routing

In this section, we provide analysis and comparisons of the lookup and routing steps of both algorithms. In the density-based heuristic, the look-up is performed inside a cluster by performing an interval routing over the different branches of the clustering tree [11]. As the cluster diameter is generally low (as seen in Section 5), the look-up

Table 2. Comparison of SAFARI and the density-based algorithm for the routing steps

$\bar{\delta}$	15.7		18.8		22.0	
	Density	SAFARI	Density	SAFARI	Density	SAFARI
# look-up Requests	1	1.71	1	1.82	1	1.78
Look-up Success	100%	95.70%	100%	92.20%	100%	90.90%
Look-up length	2.96	14.94	3.07	12.56	3.15	10.68
Route Length	5.69	7.28	6.67	5.87	6.37	6.17
Global Route Length	11.61	37.16	12.81	30.99	12.67	27.53
$\bar{\delta}$	25.1		28.3		31.4	
	Density	SAFARI	Density	SAFARI	Density	SAFARI
# look-up Requests	1	1.79	1	1.58	1	1.54
Look-up Success	100%	85.80%	100%	90.50%	100%	91.00%
Look-up length	3.16	10.36	3.21	8.63	3.24	5.04
Route length	6.75	5.88	6.61	5.73	6.66	5.09
Global Route Length	13.07	26.60	13.03	22.99	13.14	15.17

step is expected to need only a few hops to reach the rendezvous point, unlike SAFARI in which a rendezvous point may be everywhere in the network.

Moreover, if we assume a static network, the look-up in the density-based algorithm always succeeds, which may not be the case in SAFARI. This is due to the fact that level-i drums send their beacons to nodes in the level-i cells in the same level-$(i+1)$ cell than themselves. Thus, in a hierarchy with 3 or more levels, all nodes do not receive all beacons from all drums and do not have the same information in their DART. When a node d (see Figure 2) wants to register, it hashes its own identifier and sends its registration request to the node u which coordinate is the closest to the one returned by the DHT. u will do the same to forward the request of node d till reaching the fundamental cell of a rendezvous node h. Thus, node h is the node reached from the DART of node d. But as nodes have different information in their DART, when a node s locating in another level-2 cell wants to find node d, it will reach an eventual rendezvous node n from its own DART which would not have been previously contacted by node d. Thus the look-up fails. The bigger the number of levels of the network, the greater the chances that a source does never find any node responsible for storing the coordinate of the destination node.

Table 2 gives several features of look-up and routing steps for both algorithms. As in SAFARI, if a look-up request fails, a node reiterates the look-up operation toward another potential rendezvous node at the upper level, we give the mean number of tries before a success (*i.e.* the average level a query has to visit before succeeding). This number is given only for the look-up steps that finally find the coordinate of the destination node. The mean number of needed look-ups and the look-up success ratio in SAFARI depend on the number of hierarchical levels SAFARI built, as explained before. If 2 levels are built, every look-up query succeeds, as every node has the same DART. The look-up route length is the mean number of hops a query has to do before reaching the rendezvous node. As the look-up queries are routed inside a cluster in the density-based algorithm and in the whole network in SAFARI, the look-up paths are

Table 3. Comparison of SAFARI and the density-based algorithm

	Density-based	SAFARI
Hierarchy	Simple	Recursive
Metric	density + ID	Random time + ID
Cluster radius	automatic	fixed
Level	fixed ($= 1$ for routing)	l levels (automatic)
Convergence	fast and ensured	variable and not ensured
Registration	Once in each cluster	$k * (l - 1)$ times in the network
Lookup Success Ratio	100%	Depends on the number of levels

obviously shorter in the density-based approach. The routes length is the path length of the message from the source to the destination in the second step of the indirect routing. The global route length is the total number of hops that a message from the source have to do before reaching the final destination. It is the sum of twice the look-up path length and the route length. Indeed, the query has to make a round trip before the source be able to route toward the destination. We can also note that the look-up paths in SAFARI are much longer than the paths in the second step of the indirect routing, which is not the case in our approach. Note that the route length also depends on the network expansion. The more expanded the network, the higher number of hops in average between the source and the destination. For the same reasons, the look-up path in SAFARI also depends on the network diameter, but, as the clusters are based only on local information, they do not depend on the network diameter. Since the density-based algorithm routes look-up queries inside a cluster, the look-up paths in the density-based algorithm are the same whatever the network expansion. Thus, even if the look-up path is about half of the global route, this will not be the case when the network will grow as the look-up path length is constant. Then, in a very expanded network, the look-up path length will become negligible before the routing path length whereas in SAFARI, the look-up path length will remain important before the global route length. Finally, because of SAFARI stabilization problem, we did not run simulations for more expanded networks to verify this feature.

7 Conclusion

In this article, we have compared the two only protocols proposing a hierarchical routing protocol for *ad hoc* networks based on a proactive routing between clusters and a reactive routing in clusters. One of them presents a recursive hierarchical organization whereas the other one presents a single level hierarchy. These two algorithms have different features that greatly impact the performances of the built structure and of the routing. The recursive structure and the partition of the DHT in SAFARI seem to make the routing more complex and lead to low performances compared to a simple cluster organization and to a natural distribution of the hash table. We summarize in Table 3 the main differences between both protocols. As we noticed, both algorithms build equivalent clusters (level-1 clusters for SAFARI) in terms of diameter and size. Nevertheless, the SAFARI algorithm has shown stabilization failures as it can be very slow to converge because of oscillation of the levels of the drums. Moreover, the look-up proposed by SAFARI does not guarantee to find the coordinate of the destination node and, even when it succeeds, the look-up path is much longer than the physical path from the source to the destination, which implies latency, weakness and useless energy spent and may lead to scalability problems. At the opposite, the density-based algorithm presents a good behavior compared to SAFARI since it has been proved to stabilize in a low and constant time. Moreover, the look-up queries always succeed and the look-up path length remains low as queries are routed locally.

For future works, we intend to pursue the comparisons *i)* in term of message complexity *(ii)* by evaluating the performances of this hierarchical routing in an *ad hoc* environment based on a more realistic MAC layer (*i.e.* that takes collision into account) and *iii)* by studying the behavior of each of the algorithm in the context of mobility. Nevertheless, since even in a static and ideal MAC environment, SAFARI presents several failures, we are not very optimistic. Finally, we also intend to compare the performances of this kind of hierarchical routing with a classical *ad hoc* flat routing protocols (like OLSR or AODV) and with a hierarchical routing using a proactive routing inside clusters and a reactive routing between clusters.

References

1. F. Araujo, L. Rodrigues, J. Kaiser, L. Changling, and C. Mitidieri. CHR: A Distributed Hash Table for Wireless Ad Hoc Networks. In *DEBS'05*, Columbus, Ohio, USA, June 2005.
2. L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self-organized Terminode routing. *Journal of Cluster Computing*, 5(2), April 2002.
3. B. Chen and R. Morris. L+: Scalable landmark routing and address lookup for multi-hop wireless networks. Mit lcs technical report 837, MIT, March 2002.
4. G. Chen, F. Garcia, J. Solano, and I. Stojmenovic. Connectivity-based k-hop clustering in wireless networks. In *HICSS'02*, Hawaii, USA, January 2002.
5. Y. P. Chen, A. L. Liestman, and J. Liu. Clustering algorithms for *Ad Hoc* wireless networks. *Ad Hoc and Sensor Networks*, 2004.
6. B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.
7. P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster based approach for routing in dynamic networks. In *ACM SIGCOMM*, pages 49–65. ACM, April 1997.
8. B.-J. Kwak, N.-O. Song, and L. Miller. On the scalability of *ad hoc* networks. *Communications Letters, IEEE*, 8:503– 505, 2004.
9. C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.
10. N. Mitton, A. Busson, and E. Fleury. Self-organization in large scale ad hoc networks. In *MED-HOC-NET 04*, Bodrum, Turkey, June 2004.
11. N. Mitton and E. Fleury. Distributed node location in clustered multi-hop wireless networks. In *AINTEC'05*, Bangkok, Thailand, December 2005.
12. N. Mitton and E. Fleury. Efficient broadcasting in self-organizing multi-hop wireless network. In *Ad Hoc Now'05*, Cancun, Mexico, October 2005.
13. N. Mitton, E. Fleury, I. Guérin-Lassous, and S. Tixeuil. Self-stabilization in self-organized multihop wireless networks. In *WWAN'05*, Columbus, Ohio, USA, June 2005.

14. D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of GLOBE-COM'01*, November 2001.
15. N. Nikaein, H. Labiod, and C. Bonnet. DDR-distributed dynamic routing algorithm for mobile ad hoc networks. In *Mobhihoc'00*, Boston, MA, USA, November, 20th 2000. ACM.
16. C. Perkins. *Ad hoc networking.* Addison-Wesley, 2001.
17. R. Riedi, P. Druschel, Y. C. Hu, D. B. Johnson, and R. Baraniuk. SAFARI: A self-organizing hierarchical architecture for scalable ad hoc networking networking. Research report TR04-433, Rice University, February 2005.
18. C. Santivanez, B. McDonald, I. Stavrakakis, and R. R. Ramanathan. On the scalability of ad hoc routing protocols. In *INFOCOM*, New-York, USA, June 2002.
19. J. Van Leeuven and R. Tan. Interval routing. *The computer Journal*, 30:298–307, 1987.
20. A. C. Viana, M. Dias de Armorim, S. Fdida, and J. Ferreira de Rezende. Self-organization in spontaneous networks: the approach of DHT-based routing protocols. *Ad Hoc Networks Journal*, 2005.

Location-Based Multicast Routing Protocol for Mobile Ad Hoc Networks*

Jipeng Zhou[1], Jianheng Lu[1], and Francis C.M. Lau[2]

[1] Department of Computer Science, Jinan University
Guang Zhou 510632, P.R. China
tjpzhou@jnu.edu.cn
[2] Department of Computer Science
The University of Hong Kong, Pokfulam, Hong Kong, P.R. China
fcmlau@cs.hku.hk

Abstract. Location-based multicast protocol for mobile ad hoc networks is proposed in this paper. A network is divided into grids according to geographical location information and the grid network is divided into a high-channel subnetwork and a low-channel subnetwork according to labels of grids, where the destination set is divided into subsets according to its source node. Then destination nodes are partitioned into groups by using location information, the multicast routing is done in label order for each group. The proposed protocol does not require the maintenance of a distribution structure(e.g., a tree or a mesh). A forwarding node only uses information about positions of its destinations and its own neighbors to determine next hops that packet should be forwarded to and is thus very well suited for highly dynamic networks. Proposed protocol is scalable.

1 Introduction

A mobile ad hoc network(MANET) is self-organizing, dynamic topology network formed by a collection of mobile nodes through radio links. Many applications of mobile ad hoc networks rely on group communication. Communication during disaster relief, networked games, and emergency warning in vehicular networks is common example for these applications. As a consequence, multicast plays an important role in mobile ad hoc networks and has received significant attention in recent years. A number of multicast protocols for ad hoc networks have been proposed, most of them maintain some forms of distribution structure for the delivery of multicast. They can be broadly classified into tree-based protocols and mesh-based protocols. The tree-based protocols, such as AMRoute [13], MZR[3], ADMR[6], and DRMR [15], only provide one path between a pair of source and receiver. The union of paths to all receivers forms a multicast tree. In mesh-based approaches, there may be multiple paths between a sender and a receiver. This redundancy provides increased protection against topological changes. Examples of mesh-based multicast routing protocols for mobile ad hoc networks

* This work is supported by GDNSF(04300769), the Ministry of Education Study Abroad Returnees Startup Fund and JNU Grant(640581).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 38–46, 2006.

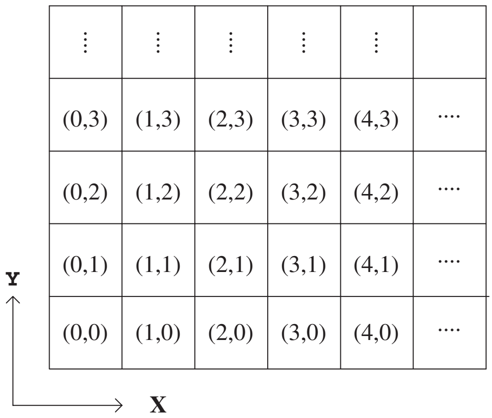
are DCMP[2], CAMP [4], NSMP[8], and ODMRP [9]. Performance comparison study demonstrates that tree-based protocols have lower data packet delivery ratio and worse composite performance than mesh-based ones in a mobile scenario [11]. Knowledge about geographical positions of nodes has been used in [12] to improve ODMRP[9] with mobility prediction and in [7] to limit flooding when a multicast group member resides in one specific area. In dynamic source multicast(DSM), each node floods the network with information about its own position, thus each node knows positions of all other nodes in an ad hoc network. A sender of a multicast packet then constructs a multicast tree from position information of all receivers. This tree is efficiently encoded in the header of a packet. Position-Based Multicast(PBM) [10] is a generalization of existing unicast routing algorithms, which use geographic positions of participating nodes for the forwarding of packets. The key advantage of PBM is rules for splitting of multicast packets and repair strategy for situation where there exists no direct neighbor that makes progress toward one or more destinations. PBM only includes all addresses of destinations in the header of multicast packets, it does not give the strategy of organization of all destinations and the selection of neighbors only depends on the distance, scalability remains open problems. A scalable team multicast is proposed in [14].

In this paper, a geographic position-based multicast protocol is proposed, a network is divided into grids, the proposed protocol utilizes location information to classify a destination set and to select a host in each grid. The grid structure is successfully used to sort a destination node set and to direct transmission of multicast packets.

This paper is organized as follows. Section 2 presents the scheme of construction and label assignment of 2D grid; Section 3 gives a position-based unicast routing protocol; In section 4, we present a multicast routing protocol; Section 5 concludes the paper and proposes further works.

2 Construction and Label Assignment of 2D Grid

It is assumed that each node knows its own physical location, i.e., its precise geographic coordinates, which may be obtained by using the Global Positioning System(GPS)[5] and whenever a node receives its own location by deploying GPS in user terminals, it will flood its location information to all other nodes. The location service technology will be discussed in a special paper. The geographic area of a MANET is partitioned into 2D logical grids as shown in Fig. 1, in which each square is called a grid zone. Each grid is a square of size $d \times d$, where d is side length of grids. Let r be the transmission of a radio signal. The smaller value of d means more number of gateways in the network, which will in turn imply a higher overhead of delivering packet and more broadcast storm. If d is too large, the radio signal of a gateway host will have difficulty in reaching places outside of the grid, and thus a gateway-to-gateway communication is unlikely to succeed. Selection of d is related to r and routing protocol. In this paper, we define the relation of d and r as $r = 2\sqrt{2}d$ as shown in Fig. 2, which guarantees



⋮	⋮	⋮	⋮	⋮	
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)	....
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	....
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	....
(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	....

Fig. 1. Logical grid to partition a physical area

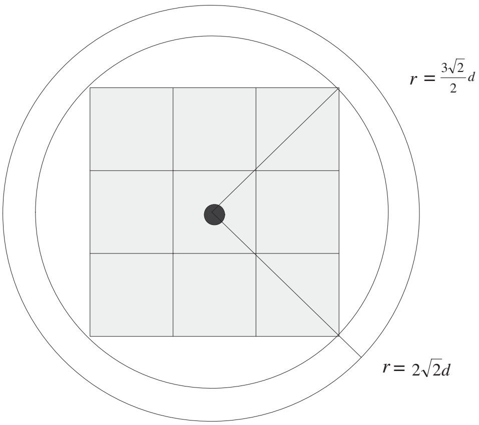


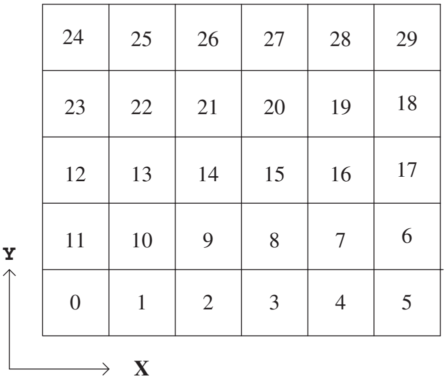
Fig. 2. The relation of transmission of radio signal r with side length d of a grid

a host in a grid to reach any host in its eight neighbor grids. This is convenient for selection of the gateway in a grid.

In this section, we will propose a label assignment scheme for grid topology and prove that this assignment scheme will provide the shortest unicast routing path in grid hops for any given pair of source and destination nodes. Suppose the address of a grid in 2D region is represented by its integer coordinate (x, y), where the lower left grid is $(0, 0)$. Each grid u is assigned a label $l(u)$. The label assignment function l for an $m \times n$ grid region can be expressed in terms of the $x-$ and $y-$coordinates of grids as follows:

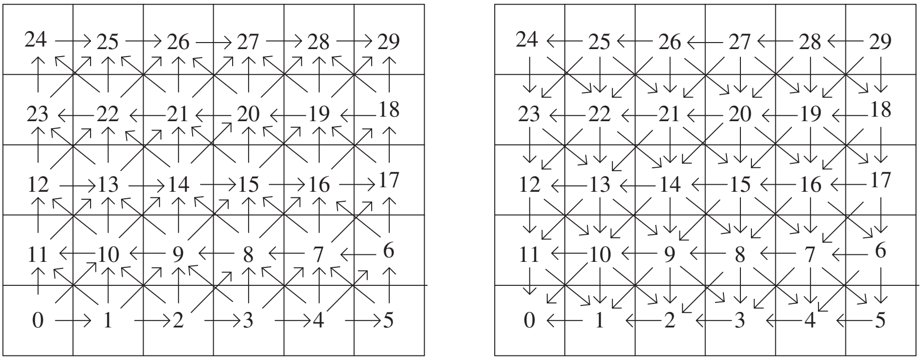
$$l(x, y) = \begin{cases} y * n + x & \text{if } y \text{ is even,} \\ y * n + n - x - 1 & \text{if } y \text{ is odd} \end{cases}$$

Each logical grid u in Fig. 1 is assigned a label $l(u)$. Fig. 3 shows an example in an 6×5 grid region, in which each grid is represented by an integer. We assume that every grid can communicate with its eight neighbors directly, labels effectively divide a grid network into two kinds of sub-networks. A *high-channel subnetwork* can be used to communicate from lower-labeled grids(gateways) to higher-labeled grids (gateways), for an example in Fig. 4(a); A *low-channel subnetwork* can be used to pass messages from higher-labeled grids(gateways) to lower-labeled grids(gateways), such as in Fig. 4(b). Multicast communication will use labels for message routing. If the label of a destination zone is greater than the label of its source zone, the multicast routing always takes place in the high-channel subnetwork, otherwise, it will take the low-channel subnetwork.



24	25	26	27	28	29
23	22	21	20	19	18
12	13	14	15	16	17
11	10	9	8	7	6
0	1	2	3	4	5

Fig. 3. 6×5 grid network with label



(a)High-channel subnetwork

(b)Low-channel subnetwork

Fig. 4. The high-channel and low-channel subnetwork in 6×5 grid network

3 Unicast Routing Protocol

Three major issues should be considered in designing a routing protocol, that is route discovery, packet relay, and route maintenance. In route discovery, location information is used to determine the quality of a route. A node in an ad hoc network obtains its location from a system such as the Global Positioning System(GPS). In location-unaware protocols, the route discovery is done by a blind flooding, it easily leads to cause broadcast storm problem. Location-based multicast schemes in [7] use forwarding zones to avoid network-wide flooding, since its forwarding zones are too large, there may still exist a lot of unnecessary flooding packets within a forwarding zone and it does not give a solution how to select a relay host, when a source cannot reach its destination. This problem will be solved in this paper. We assume that a source node knows locations of all its destinations. Location service will be discussed specially in another paper. A source node will forward a packet to a neighbor node that is closest to its destination node. Locations of a source and its destination are used to confine the forwarding range. The same procedure is repeated until a destination node is reached.

In unicast routing protocol, routing is performed in a grid-by-grid manner through grid gateways. If a gateway leaves its original grid, a behavior similar to the 'hand off' procedure in cellular systems will take place. In this case, a gateway passes its routing information to the next gateway. Each gateway keeps the information of nodes in its grid zone, and has a rule to determine a node in which grid to forward packet to. Only local information, instead of global information, is used to forward a packet. We will consider two issues in our protocol design, one is that as less as possible nodes are searched in each step of message routing. Another is that a route path is as short as possible.

We assume that each source node can get the position of its destination by location service mechanism. So a route to a forwarding zone is determined by the location of a sender and coordinates of its destination. We define the distance between two grids $u = (x_1, y_1)$ and $v = (x_2, y_2)$ as $d(u, v) = max\{|x_1 - x_2|, |y_1 - y_2|\}$, a forwarding grid is selected by computing distances between neighbors of current grid and the grid with destination node in, then select a grid with the minimum distance to the destination as a forwarding grid.

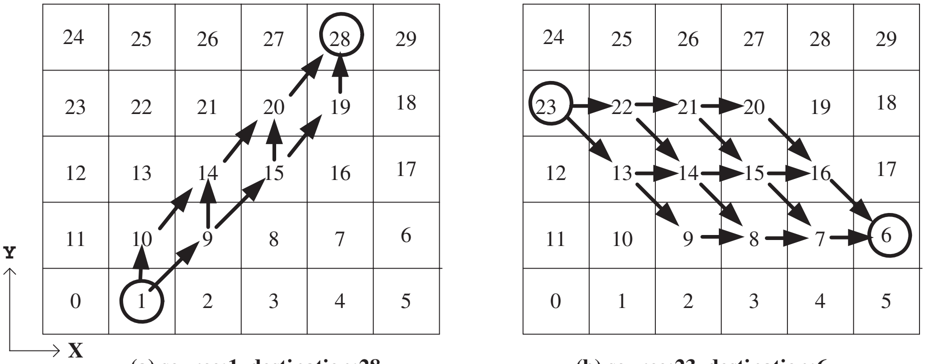
Let V be a set of all grids in a network. Finding a deadlock-free unicast algorithm for 2D grid is to define a routing function $R_1 : V \times V \rightarrow V$ that uses two subnetworks in such a way to avoid cycle routing. Here two grids (i_0, j_0) and (i_1, j_1) are neighbors(8-neighbors), i.e., $max\{|i_0 - i_1|, |j_0 - j_1|\} = 1$. One such routing function, for a source node in grid u and a destination node in grid v, is defined as $R_1(u, v) = w$, such that w is a neighbor grid of u, and if $l(u) < l(v)$, then we have the following equation:

$$d(w, v) = min\{d(z, v) : l(z) \leq l(v) \text{ and z is a neighboring grid of u}\},$$

or if $l(u) > l(v)$, then we have the following equation:

$$d(w, v) = min\{d(z, v) : l(z) \geq l(v) \text{ and z is a neighboring grid of u}\}.$$

If more hosts satisfy the condition, we only select anyone of them.



(a) source:1, destination:28

(b) source:23, destination:6

Fig. 5. Unicast routing in the high-channel and low-channel subnetwork of 6 × 5 grid

Unicast communication uses labels for routing. Examples of unicast routing in high-channel subnetwork and low-channel subnetwork are shown in Fig. 5. All possible routes from a source host in grid s ($l(s) = 1$) to a destination host in grid $d(l(d) = 28)$ in high-channel subnetwork are shown in Fig. 5(a), where all routes are only from hosts in a low label grid to hosts in a high label grid; All possible routes from a source host in grid s ($l(s) = 23$) to a destination host in grid $d(l(d) = 6)$ in low-channel subnetwork are shown in Fig.5(b), where all routes are only from hosts in high label grid to hosts in low label grid. The packet is forwarded one hop closer to its destination at each step and the route is along the shortest path between a source and its destination in grid level.

4 Multicast Routing Protocol

For multicast routing from source node S to destination node set D, the algorithm has two parts: message preparation and routing. The first part splits the destination set for a message into more subsets in two steps. In the first step, the destination set D for a message generated at node S is divided into two subsets D_H and D_L such that D_H contains all destination nodes with grid value l higher than the grid with S and D_L nodes with grid value l lower than the grid with S. Destination nodes in D_H are sorted in ascending order by using their grid label l as a key. The same is done for D_L, but in descending order. In the second step, destination set D_H and D_L are divided into more groups respectively, which depend on the distance of grids with corresponding destination nodes in D_H and D_L. The second part determines the path followed by the message until all destination nodes in each group are reached. The routing is performed by using unicast routing protocol in the order of labels in each group. For simplicity of presentation, a grid and the nodes in it are denoted by the same label, the distance of two nodes means the distance of two grids with two nodes respectively in what follows.

Algorithm 1. *Multicast routing from S to destination set D_H(or D_L)*

```
1. Let node $d_n, n = 1, 2, \ldots,$ be nth node in $D_H$ in ascending order(or $D_L$ in
   descending order). If $d(d_1, d_2) > d(S, d_2)$, then $d_1$ and $d_2$ are in different
   groups, else $d_1$ and $d_2$ are in the same group.
2. Assume that the first n nodes in $D_H$(or $D_L$) have been classified into different
   groups $G_i, 1 \leq i \leq k$, and let $g_i$ be the node with the maximum sequence
   number(the minimum sequence number) in $G_i$, if $d(d_{n+1}, g_i) > d(d_{n+1}, S)$
   for $1 \leq i \leq k$ , then $d_{n+1}$ is in new group $G_{k+1}$ and $k + 1 \Rightarrow k$, else $d_{n+1}$
   belongs to $G_m$ , where $d(d_{n+1}, g_m) = min\{d(d_{n+1}, g_i) | 1 \leq i \leq k\}$ , and
   $n + 1 \Rightarrow n$.
3. If $n < |D_H|$(or $n < |D_L|$ ), then go to 2.
4. S send the packet to the first node in $G_i$ with unicast routing algorithm,
   $1 \leq i \leq k$.
5. For each $G_i$ in parallel, let $|G_i| = k_i, g_j \in G_i, 1 \leq j \leq k_i$,
   For $j = 1$ to $k_i - 1$
   send packets from $g_j$ to $g_{j+1}$ with unicast routing algorithm.
```

Here we will discuss time complexity of algorithm 1, we have $|D_H|$ destination nodes. In step 1, we needs constant time $O(1)$; the time complexity of step 2 is k, where k is the number of groups and $k \leq |D_H|$. Step 2 and step 3 will repeat at most D_H times. So if $|D_H| = n_1$, the time complexity of step 2 and step 3 in algorithm 1 is $n_1 + (n_1 - 1) + (n_1 - 2) + \ldots + 1 = \frac{n_1 \times (n_1+1)}{2}$. Time complexity of sorting D_H is $O({n_1}^2)$. Total time complexity of partitioning destination nodes D_H is $O({n_1}^2)$. The hops of any unicast routing in step 4 and step 5 are no more than the diameter d of the network, so time complexity of step 4 and step 5 is $O(dn_1)$. The total time complexity of algorithm 1 to destination D_H is $O(n_1(d + n_1))$. If $|D_L| = n_2$, then the time complexity of algorithm 1 to destination D_L is $O(n_2(d + n_2))$.

The performance of proposed multicast routing algorithm is dependent on location distribution of destination nodes. In order to reduce its time cost in 2D grid, destination set D_H and D_L can be further partitioned. The set D_H can be divided into two sets, one contains nodes whose x coordinates are greater than or equal to that of source node S and another contains remaining nodes in D_H. D_L is partitioned in the similar manner. The algorithm 1 is used to four subset simultaneously, the time complexity can be reduced.

Consider an example as shown in Fig. 6, where source node S is 15 and there are 11 destination nodes in destination set D, where $D = \{0, 12, 35, 22, 9, 26, 31, 5, 6, 18, 29\}$. D_H is represented as $\{18, 22, 26, 29, 31, 35\}$ in ascending order, D_L as $\{12, 9, 6, 5, 0\}$ in descending order. We use algorithm 1 to D_H, in step 1, since $d(18, 22) > d(15, 22)$, 18 and 22 belong to different groups, let $G_1 = \{18\}$ and $G_2 = \{22\}$; In step 2, let $d_{n+1} = 26$, since $d(22, 26) < d(15, 26)$, and $d(18, 26) > d(18, 15)$, so 26 belongs to G_2, that is $G_2 = \{22, 26\}$, repeat step 2 to group all destination nodes into $G_1 = \{18, 29, 31\}$ and $G_2 = \{22, 26, 35\}$. D_L can be grouped into $G_1 = \{12\}$,$G_2 = \{0, 9\}$ and $G_3 = \{5, 6\}$.

Destination nodes are in ascending order for each group in destination set D_H and in descending order for each group in destination set D_L, the message

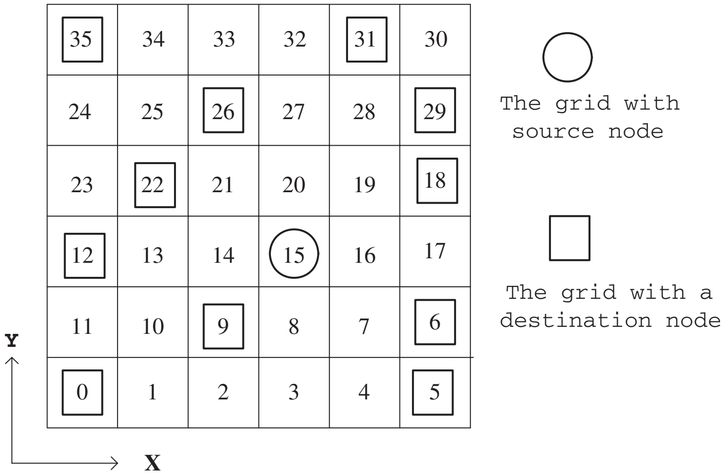


Fig. 6. An example of partitioning destination set into different groups in 6 × 5 grid

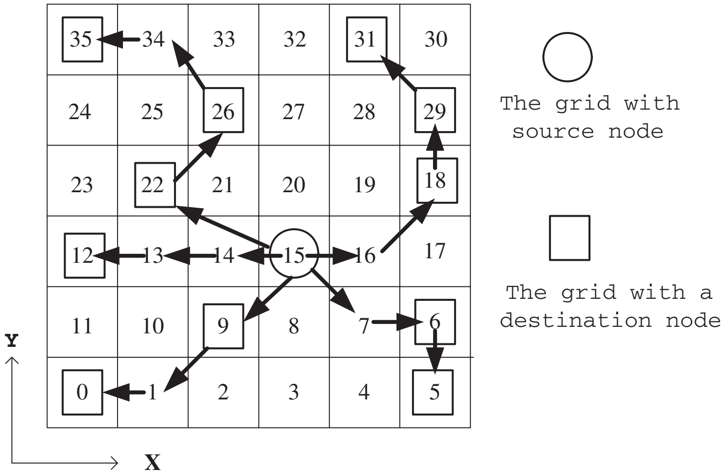


Fig. 7. Multicast routing in different groups in 6 × 5 grid

is routed by using unicast routing protocol for each pair of successive destination nodes in each group. Examples of multicast routing paths are shown in Fig. 7.

5 Conclusion

In this paper, we propose a multicast routing paradigm for mobile ad hoc networks. The new routing scheme is based on geographical location information of hosts, first, a network is divided into grids, and the grid network is divided into high-channel subnetwork and low-channel subnetwork according to labels of grids. Second, a destination set is partitioned into two subsets by its source node, then destination nodes in each set are classified into groups; After that, the multicast routing for each group is done in label order for each group. The proposed protocol does not require the maintenance of a distribution structure(e.g., a tree or a mesh). A forwarding node only uses information about positions of destinations and its own neighbors to determine next hops that a packet should be forwarded to and is thus very well suited for highly dynamic networks. The proposed protocol is scalable. The performance simulation of proposed protocol is under way by using NS-2 simulator. We will further investigate the robustness of the multicast routing.

References

1. M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Network ,* 2(2004), pp.1-22.
2. S. Das, B. Manoj, and C. Murthy. A dynamic core based multicast routing protocol for ad hoc wireless networks. Proceedings of ACM Mobihoc 2002, Lausanne, Switzerland, June 2002, pp. 24-35.
3. Vijay Devarapalli and Deepinder Sidhu. MZR: A multicast protocol for mobile ad hoc networks. In proceedings of IEEE ICC'2001, Helsinki, Finland, June 2001, pp.886-891.
4. J. Garcia-Luna-Aceves and E. Madruga. The core-assisted mesh protocol. IEEE Journal on Selected Areas in Communications, Vol.17, No.8, pp.1380-1994, August 1999.
5. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. Global Positioning System: Theory and Practice, 4th ed. (Springer-Verlag, New York, 1997)
6. G. J. Jorjeta and B. J. David. Adaptive demand-driven multicast routing in multi hop wireless ad hoc networks. In Proceedings of Mobihoc 2001, Long Beach, CA, USA, Oct. 2001, pp.33-44.
7. Y. Ko and N. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In Proceedings of IEEE WMCSA, February 1999.
8. S. Lee and C. Kim. Neighbor supporting ad hoc multicast routing protocol. Proc. of ACM Mobihoc 2000, Boston, Massachusetts, USA, August 2000, pp.37-50.
9. S. J. Lee, M. Gerla and C.C. Chiang. On-demand multicast routing protocol in multihop wireless mobiles networks. Mobile Networks and Applications, 2002, 7(6):441-453.
10. M. Mauve, H. Fubler, J. Widmer and T. Lang. Position-based multicast routing for mobile ad hoc networks. Technical Report TR-03-004, Institute for mathematics andComputer Science, University of Mannheim, Germany, 2003.
11. S. Lee, W.Su, J. Hsu, M. Gerla, and R. Bagrodia. A performance comparison study of ad hoc wireless multicast protocols. In proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000, pp.565-574.
12. S. Su and M. Gerla. Wireless ad hoc multicast routing with mobility prediction. Mobile Networks and Applications, Vol.6, No.4, pp.351-360, 2001.
13. J. Xie, R.R. Talpade, A. Mcauley and M. Liu. AMRoute: ad hoc multicast routing protocol. Mobile networks and Applications, 2002, 7(6):429-439.
14. Yunjung Yi, Mario Gerla, and Katia Obraczka. Scalable team multicast in wireless ad hoc networks exploiting coordinated motion. *Ad Hoc Networks* 2(2004) 171-184.
15. Yuan Zhou, Guang Sheng Li, Yong-Zhao Zhan, and etc. DRMR: Dynamic-ring-based multicast routing protocol for ad hoc networks. Journal of Computer Science & Technology, Nov. 2004, Vol.19, No.6, pp.909-919.

Skipping Face Routing with Guaranteed Message Delivery for Wireless Ad Hoc and Sensor Networks

Jie Lian and Kshirasagar Naik

Department of Electrical and Computer Engineering,
University of Waterloo, Ontario, Canada
jlian@swen.uwaterloo.ca, knaik@swen.uwaterloo.ca

Abstract. Location-based routing techniques, greedy routing and face routing, route data by using the location information of wireless nodes. Greedy routing efficiently routes data in dense networks by giving short hop paths, but it does not guarantee message delivery. Face routing has been designed and combined with greedy routing to achieve both transmission efficiency and guaranteed message delivery. The existing face routing algorithms mainly works on three types of planar graphs: Gabriel graph, relative neighborhood graph, and Delaunay triangulation. One major observation is that each transmission in face routing only can pass message over a short distance, resulting in that the existing face routing traverses long hop paths to destinations. In this paper, we present a Skip Face Routing (SFR) to reduce the face traversal cost incurred in the existing approaches. By using simulation studies, we show that SFR significantly increases routing performance.

1 Introduction

In location-based routing, geographical locations of networked nodes are used to guide data forwarding. Location-based routing and location-awareness techniques have been extensively studied in the literature for sensor and ad hoc networks [2-9, 11-13, 21-25], and a summary can be found in [10, 20]. One desired property of these algorithms is to deliver message through short routing paths. Unit disk graph (UDG) is a generally accepted model for wireless ad hoc networks (or sensor networks) [5, 15], in which all nodes have the same maximum transmission distance and this distance is normalized as one unit. Consequently, a network with the node set V is represented by a unit disk graph $UDG(V)$, in which an edge e_{uv} exists if the Euclidean distance $d(u, v)$ between u and v is not greater than one unit.

One type of location-based routing algorithms is based on the greedy principle [2], that is, when a source sends data to a destination, the source forwards the data to its neighbor which has the shortest distance to the destination. In greedy routing, each transmission shortens the distance from the node holding the message to a destination, and the reduced distance after one message transmission is known as *progress* [1]. Let *MaxP* denote the average maximum shortened distance towards destinations for all transmissions. Two simple strategies, called *most forward within radius* (MFR) and *random neighbor forwarding* (RNF) [11, 12, 13], use the concept of progress, but without detailed descriptions at the system level. The first location-based routing,

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 47 – 59, 2006.

Cartesian routing, was proposed in [2], and based on the greedy principle. Instead of using progress, *compass routing* [3] uses the direction angle to guide data forwarding. Greedy routing is proved to be an efficient algorithm for dense networks in terms of hop's lengths of routing paths. However, greedy routing fails if a forwarding node encounters a *void* (a large area without node) on its routing direction, which may occur frequently in sparse networks and in environments with obstacles.

To guarantee message delivery, *face routing* [3] has been proposed. Face routing partitions the graph into a set of faces along the line connecting the source node and the destination, and delivers messages along these faces. Face routing was later combined with greedy routing to improve the average-case performance (e.g., average path length) in the two algorithms, *greedy-face-greedy* (GFG) [4, 5] and *greedy perimeter stateless routing* (GPSR) [6]. For dense networks, the paths found by GFG and GPSR are close to the shortest hop paths. However, both the algorithms are not worst-case efficient. Adaptive face routing (AFR) [7] is known as the first algorithm that combined face routing and the greedy algorithm with a worst case guarantee. In the two follow up papers [8, 9], two algorithms, GOAFR and GOAFR$^+$, were proposed with both asymptotically worst case optimal and average case efficient.

Internal and shortcut-based routing (ISR) [22] differs from the preceding algorithms in the way of which underlying topology graphs are used in routing. In ISR, the nodes in the connected dominating set [26, 27] of a network topology graph are used to route data. ISR also introduced a skipping technique by which a node can skip some nodes during face traversal. However, this technique requires two-hop neighbor information for each node, resulting in a high transmission overhead. In [24], the authors proposed a Morelia test to generate a planar graph based on the Gabriel graph of a network with a longer average edge length. By using the graph obtained from the Morelia test, the average path length in face routing is reduced by 10%. However, similar to ISR, this test needs two-hop neighbor information of each node.

Face routing only works on a planar graph which is defined as a graph with no two edges crossing one another. For a given *UDG*(*V*), three planar graphs (sub-graphs of *UDG*(*V*)) are commonly used: Gabriel graph *GG*(*V*), relative neighborhood graph *RNG*(*V*), and unit Delaunay triangulation *UDel*(*V*). If two nodes are connected by an edge on a planar graph, they are called *planar neighbors*. In the existing face routing, one constraint is that each face traversal message at a node can only be forwarded to a planar neighbor of the node. This constraint has a major drawback as follows.

As illustrated in Fig. 1, a source s sends data to a destination d. When v receives the forwarding message from s, v performs face traversal along the path $v \rightarrow w \rightarrow x \rightarrow y \rightarrow \ldots$ to bypass the void. In the existing approaches, even though x is a direct neighbor of v (the dotted circle denotes the transmission radius of v in Fig. 1), the message is sent from v to w, and then from w to x, which uses an extra transmission from w to x. For networks with voids, one intuition is that these additional transmissions may be significant comparing with the total number of transmissions along routing paths. This intuition is enhanced in Fig. 2, which shows the average lengths of all edges in *GG*(*V*), *RNG*(*V*), *UDel*(*V*), and *UDG*(*V*), and the average maximum progress (MaxP) obtained from networks with different densities (specified by the average degree). The average edge lengths in Fig. 2 implicitly indicates the average distance achieved by each transmission using the existing face routing on

GG(*V*), *RNG*(*V*), and *UDel*(*V*). However, it can also be observed that the average lengths of edges in *UDG*(*V*) and MaxP are much larger than those in *GG*(*V*), *RNG*(*V*), *UDel*(*V*). This observation implies that for a face with fixed boundary length, the total number of transmissions required to traverse the face by using the existing approaches is much larger than the optimal transmission number.

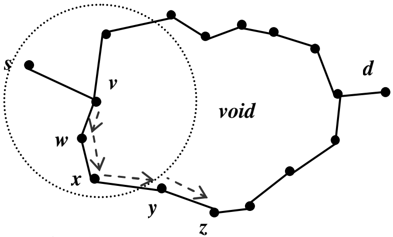


Fig. 1. Face partition and traversal

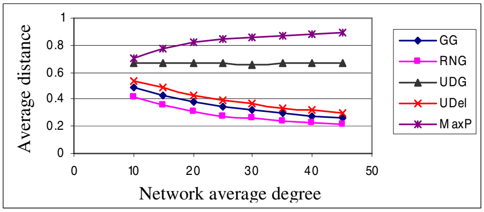


Fig. 2. Average transmission distance

In this paper, we present a Skipping Face Routing (SFR) algorithm working on GG to reduce the total number of transmissions associated to face traversal. The basic idea of SFR is to define a set of conditions by which a node can skip some intermediate nodes during face traversal. The major contribution of this paper is as follows:

- In SFR, we define a set of sufficient conditions by which a node decides if it can skip one or more intermediate nodes in face traversal. One of desired properties of SFR is that each node in SFR only needs to know locations of its one-hop neighbors, and therefore, no additional communication overhead is required.
- SFR can be embedded in GFG [4, 5] with minor modifications.
- The simulation studies show that SFR can save up to 40% of the total number of transmissions required by the existing pure face routing algorithms. The simulation studies also show that GFG embedded with SFR can save up to 30% of the total number of transmissions required by the original GFG.

The rest of the paper has been organized as follows. We introduce some basic terms and derive skipping conditions in Section 2. The detailed descriptions of SFR are given in Section 3. Simulation studies are presented in Section 4. Finally, we give some concluding remarks in Section 5.

2 Terminology and Skipping Conditions in Face Traversal

Planar Graphs: Face routing can only be applied on a planar graph. Let *V* denote a set of nodes in a network, and *UDG*(*V*) represent the network. To planarize a *UDG*(*V*), a deduced sub-graph of *UDG*(*V*), called Gabriel graph (GG), is employed. The Gabriel graph on a *UDG*(*V*) is defined as a graph *GG*(*V*) so that for each edge e_{uv} in *UDG*(*V*), e_{uv} is in *GG*(*V*) *iff* the circle with e_{uv} as a diameter does not contain any node other than *u* and *v*. For an edge e_{uv} in *GG*(*V*), nodes *u* and *v* are called *Gabriel neighbors*. Two additional planar graphs, relative neighborhood graph (RNG) and unit Delaunay triangulation (UDel), are also used. An RNG on *UDG*(*V*) is defined as a graph *RNG*(*V*) so that for each edge e_{uv} in *UDG*(*V*), e_{uv} is in *RNG*(*V*) *iff* there is no node $w \in V$ such that $d(u, w) < d(u, v)$ and $d(w, v) < d(u, v)$. A triangulation of a given node set V is a Delaunay triangulation if the circum-circle of each of its triangles does not contain any node of V in its interior [18]. Given a Delaunay triangulation of V, *UDel*(V) is the graph obtained by removing all edges of the Delaunay triangulation that are not in *UDG*(V).

Faces in Planar Graphs: The edges in a planar graph partition the network area into a set of faces [2, 3]. There are two types of faces: *interior face* and *exterior face*. The former is the continuous area bounded by one or more closed edge paths. The latter is the unbounded area outside the network graph. In Fig. 3, the network area is partitioned into three faces, F_1, F_2, and, F_3, where F_3 is an exterior face.

Face Traversing: We employ *Right-Hand Rule* or *Left-Hand Rule* to traverse a face. In Right-Hand Rule, a person explores a face by keeping her right hand on the walls (edges) and she will eventually visit all edges on the face. We define a face traversing method as $trav(v_s, v_r, rule, v_i)$, where v_s is the message sender, v_r is the recipient, *rule* is Right- or Left-Hand Rule, and v_i is the node initiating the face traversal. For example in Fig. 3, starting from u, to traverse F_1 by Right-Hand Rule, u sends v a message $trav(u, v, Right, u)$, and v is called next visited node with respect to u by Right-Hand rule. When v receives this message, v sends a message $trav(v, w, Right, u)$ to the node w. Repeating this step, F_1 can be traversed counterclockwise.

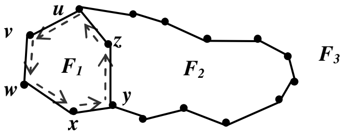


Fig. 3. Face partition and traversal

Single Node Skipping Condition

We discuss conditions in which a node can skip some intermediate nodes during face traversal. All the results in this section are obtained by using Right-Hand rule during face traversal and the discussion is based on GG. Similar results hold for Left-Hand rule too. Due to the space limitation, the proofs of all Lemmas are ignored.

Assume that node x knows the locations of its one-hop neighbors, and a node u is the next traversed node of x during face traversal. From viewpoint of x, if x can skip u and sends the traversal message to node v depends on the two conditions as follows:

- C1: by which x can determine if u and v are Gabriel neighbors of each other.
- C2: by which x can determine if v is the next traversed node with respect of u.

Let $\odot(u, v)$ denote the circle with the line segment connecting u and v as its diameter, and $c(u, v)$ the center point of $\odot(u, v)$. We have Lemma 1 to answer Condition C1.

Lemma 1: Let u and v be two neighbors of a node x. If $d(x, c(u, v)) + d(u, v)/2 \leq 1$, x can determine whether u and v are Gabriel neighbors of each other.

To derive Condition C2, we define some terms as follows. For an edge e_{uv} in $GG(V)$, let $\perp(e_{uv}, v)$ denote the perpendicular line of e_{uv} through v. $\perp(e_{uv}, v)$ partitions the network plane into two half planes (Fig. 4). Let $\perp^{+}(e_{uv}, v)$ denote the half plane (shaded area in Fig. 4) not containing u. We have the following Lemma.

Lemma 2: For two nodes u and v which are Gabriel neighbors of each other, u can not have a Gabriel neighbor located in the half plane $\perp^{+}(e_{uv}, v)$.

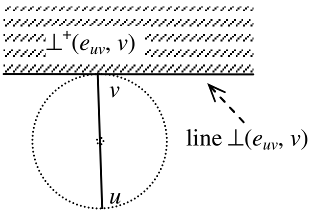
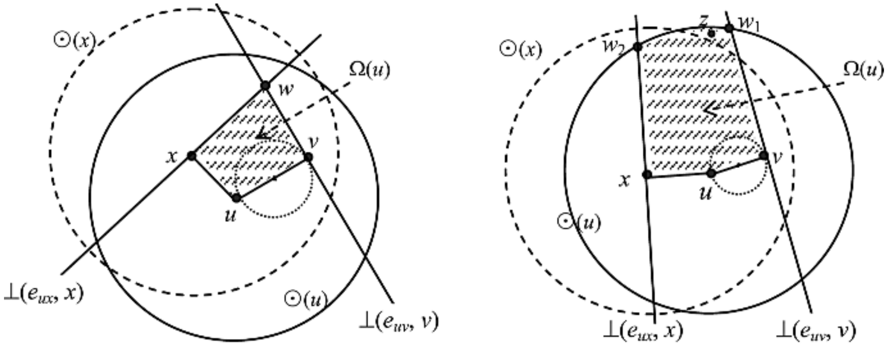


Fig. 4. Half plane of line $\perp(e_{uv}, v)$ in the derivation of skipping conditions



(a) Case 1

(b) Case 2

Fig. 5. Two cases of the searching space (shaded areas) of u with respect to x

Then we derive the skipping condition (Condition C2). Let $\odot(u)$ denote the unit disk centered at u, and $\overrightarrow{ux}$ denote a ray starting at u through x. Considering Fig. 5(a), assume that u is the next visited node with respect to x. Node x scans its covered area by rotating $\overrightarrow{ux}$ clockwise (keeping u stationary) until find the first encountered node v which is a Gabriel neighbor of u. This step is called *Gabriel neighbor scan process* of $\overrightarrow{ux}$ performed by x (denoted by $scan(\overrightarrow{ux}, x)$), and the angle $\angle xuv$ is called *scan angle*. Then we need to find a condition by which x can determine if v is the next visited node with respect to u based solely on x's local knowledge. Assume that v is the node obtained by using $scan(\overrightarrow{ux}, x)$ in Fig. 5(a). We draw two lines $\perp(e_{uv}, v)$ and $\perp(e_{ux}, x)$, and then define *decision region of u* (denoted by $\Omega(u)$) as follows.

- Case 1: If the two lines intersect at a point w within the scan angle and located in $\odot(u)$, $\Omega(u)$ is defined as the trapezium $wxuv$ (the shaded area in Fig. 5(a)).
- Case 2: If not the case 1, let w_1 denote the intersected point of $\perp(e_{uv}, v)$ and $\odot(u)$ located in the scan angle $\angle xuv$, and let w_2 denote the intersected point of $\perp(e_{ux}, x)$

and $\odot(u)$ located in $\angle xuv$. $\Omega(u)$ is defined as the area enclosed by line segments w_1v, vu, ux, xw_2, and arc w_2w_1 (the shaded area in Fig. 5(b)).

Then we give the single node skipping condition in Lemma 3 as follows.

Lemma 3: For the nodes x, u and v shown in Fig. 5, u is the next visited node with respect to x, and v is the node found by the scan process $scan(\overrightarrow{ux}, x)$. If decision region $\Omega(u)$ is fully contained by $\odot(x)$ and there is no node in $\Omega(u)$, x can determine by using its local knowledge that v is the next visited node with respect to u.

Multiple-node Skipping Condition

The preceding section gives the condition in which a traversal message skips one intermediate node. In many applications, nodes are densely deployed so that it is possible to skip over multiple nodes. Hence, we derive the multiple-node skipping condition for nodes with one-hop neighbor information. First, assume that a node s, which holds a traversal message, determines that s can skip $(k - 1)$ nodes, and the traversing sequence without skipping is $s \rightarrow s_1 \rightarrow \dots \rightarrow s_{k\text{-}2} \rightarrow x \rightarrow u$. Then we obtain:

Lemma 4: For the sequence of nodes $s \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_{k\text{-}2} \rightarrow x \rightarrow u$ given above, let v be the node found by the scan process $scan(\overrightarrow{ux}, s)$. Then if the decision region $\Omega(u)$ is fully located in $\odot(s)$ and there is no node located in $\Omega(u)$, s can determine by its local knowledge that v is the next visited node with respect to u.

From Lemma 3, a node s can find the next skipped node s_1. By using Lemma 4 repeatedly, the node s can find a sequence of skipped nodes satisfying Lemma 4. This process ends if no node can be found. Then, s sends the message to the last node in the skipping sequence. Let SKIP denote the multiple-nodes skipping algorithm for a node with one-hop neighbor information and its pseudo-code is given in Algorithm 1.

Algorithm 1. Multiple-nodes skipping algorithm (SKIP) for nodes with one-hop neighbor information

```
Input:  a node $s$ which holds the current traversing message
Output: a list $L$ of nodes that will be traversed from $s$ in the visited order
BEGIN
  $u \leftarrow$ the next visited node of $s$;        $x \leftarrow s$;
  for (true)
    append $u$ at the end of to the list $L$;
    $v \leftarrow$ the node found by the scan process $scan(\overrightarrow{ux}, s)$;
    if( ($\Omega(u) \subset \odot(s)$) && ($\Omega(u)$ contains no node))  then { $x \leftarrow u$;        $u \leftarrow v$;}  // Lemma 4
    else          break;
    end if
  end for
END
```

Avoidance of Infinite Loop

SKIP may return a list containing infinite number of nodes in a special situation illustrated in Fig. 6. In the figure, the face is bounded by a node sequence $L' = x \rightarrow u \rightarrow v \rightarrow w$. However, starting at x and using SKIP, the returned sequence L will repeat L' infinite number of times. One solution to avoid an infinite sequence is that the sequence L stops adding a newly found node r when r is in L. However, this solution will lose a chance in which x can skip more nodes. As illustrated in Fig. 7, a true skipping sequence is $x \rightarrow u \rightarrow w \rightarrow u \rightarrow v$. If using the preceding solution, x transmits its message to w, instead of to v.

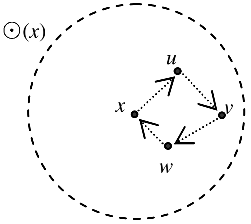


Fig. 6. Infinite skipping sequence

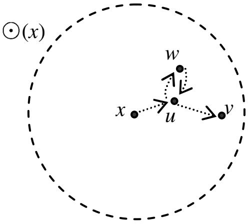


Fig. 7. Missing node during skipping

To avoid infinite sequences and not miss skipped nodes, a *skip halting rule* is given as follows. Let L denote the sequence found by SKIP so far, and $L = s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_{k-1}$. For any two consecutive traversing node s_k and s_{k+1} obtained by SKIP, if $s_k = s_i$ and $s_{k+1} = s_{i+1}$ for some $i < k$, then s_k is appended at the end of L and SKIP is terminated.

3 Skipping Face Routing (SFR)

Double-Direction Face Traversal

There are two methods to traverse a face: *single-direction traversal* and *double-direction traversal.* In the former, Right-Hand rule or Left-Hand rule is used. The existing approaches use single-direction traversal (Fig. 8). On the other hand, the latter applies both Right-Hand rule and Left-Hand rule concurrently (Fig. 9). Single-direction traversal has one drawback: longer face traversal time.

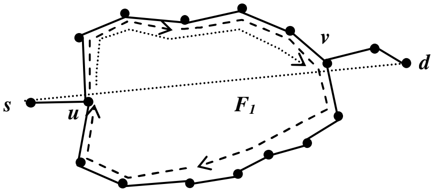


Fig. 8. Single direction face traversal

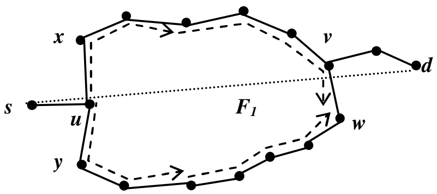


Fig. 9. Double direction face traversal

Termination Condition of SFR for Double-Direction Face Traversal

In SFR, a node involved in a face traversal uses a traversing message MSG(*source*, *destination*, *trav*(…)), where the *source* is the node trying to communicate to the *destination*, and *trav*(…) is the traversing method defined in Section 2. For example in Fig. 9, a traversing message sent by u is MSG(s, d, $trav$(u, x, Left, u)). To avoid traversing a face many times, each node receiving MSGs must check a *termination condition*, by which a node decides if the received MSGs can be discarded.

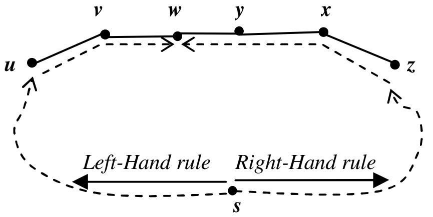


Fig. 10. Termination condition of face traversal

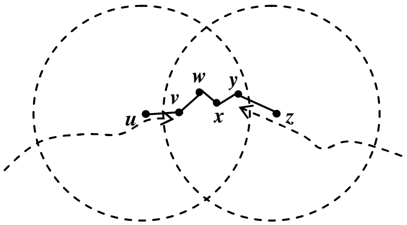


Fig. 11. Double-direction traversal in SKIP

The termination condition can be stated as follows. For two Gabriel neighbors *u* and *v* receiving two traversing messages, if the message at *u* will traverse *v* next and the message at *v* will traverse *u* next, these two messages are discarded by *u* or *v* depending on which node transmits its traversing message first. If *u* transmits its traversing message to *v* first, then *v* discards both the messages and stops its transmission to *u*. If *v* transmits first, *u* discards both the messages. As illustrated in Fig. 10, a double-direction face traversal is started at a node *s* and the two traversal messages *MSG*(*s*, *d*, *trav*(*v, w, Left, s*)) and *MSG*(*s*, *d*, *trav*(*y, w, Right, s*)) reach the node *w*. According to the termination condition, *w* determines completion of face traversal and discards both the messages.

The preceding termination condition only considers the situation in which no SKIP is used and will fail when SKIP is applied. As shown in Fig. 11, *u* sends a traversal message to *y* (skip *v*, *w*, and *x*), and *z* sends a message to *v* (skip *y*, *x*, and *w*). Since these two messages do not meet at an intermediate node, by using the termination condition, it is possible that the face will be traversed forever. We remedy this problem by adding a special field in each traversal message containing the list of nodes which are skipped by the message sender. Then a typical message of node *u* in Fig. 11 is *MSG*(*s*, *d*, *trav*(*u, y, Left,* v_i, [*v*, *w*, *x*])), where [*v*, *w*, *x*] is the skipping node list between *u* and *y*. When the nodes *v*, *w* and *x* overhear this message, they store the message locally but not forward the message. Then, after *v* receives the message *MSG*(*s*, *d*, *trav*(*z, v, Right,* v_i, [*y*, *x*, *w*])) from *z*, *v* can use the termination condition to determine whether to forward the new message.

4 Performance Evaluation

We compare the performance of SFR with the existing approaches by using simulation. The metric used in the comparison is the total number of transmissions to complete face traversals. Two sets of simulation results are presented for different algorithms. In the first set of experiments, we compare performance of two face routing algorithms, SFR and FACE [3], during face traversal. In the second set of experiments, we embed these algorithms in GFG and evaluate their performance.

4.1 Comparison of Total Number of Transmissions in Face Routing

The first experiment shows the total number of transmissions related to face traversal for two face routing algorithms: SFR and FACE. To make a fair comparison, for each *UDG*(*V*), FACE are applied to three planar graphs, *GG*(*V*), *RNG*(*V*), and *UDel*(*V*), derived from *UDG*(*V*). FACE algorithms applied on these graphs are denoted by F-GG, F-RNG, and F-UDel, respectively.

Simulation is done by using a routing-level simulator, operating on a set of randomly generated networks. In each sample network, nodes are randomly distributed in a 20 × 20 square area such that the average degree (the average number of neighbors for all nodes) is *g*. We vary the value of *g* to observe the relationship between the number of transmissions and the network density. All nodes have an identical transmission radius 1. These sample networks are called *base networks*. To simulate the performance of face traversal, we create a 5 × 10 rectangular void at the center of each base network. All the nodes in a void are removed, which guarantees that the average degree of the remaining nodes is unchanged in the rest of network area. All reported results are computed by taking the average value of 20 sample networks. For simplicity, we omit the collisions involved in data transmissions.

Fig. 12 shows the total number of transmissions related to traversing the faces created by the rectangular voids, and Fig. 13 shows the average distance achieved by all transmissions. The *x*-axis denotes the average degree of sample networks. According to Fig. 12, F-UDel performs best, and followed by those of SFR, F-GG, and F-RNG in a decreasing order. In addition, we have the following comparison results.

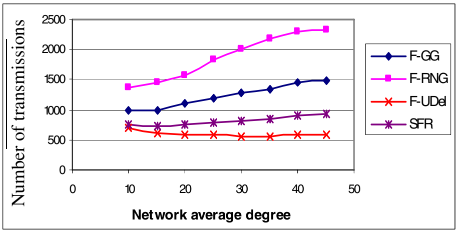


Fig. 12. Total number of transmissions involved in face traversal

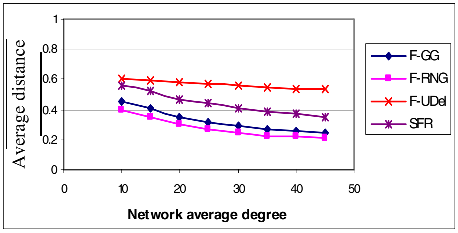


Fig. 13. Average distance of one-hop transmission involved in face traversal

Comparison between SFR and F-GG

In sparse networks (*g* = 10), SFR reduces approximately 25% of the total number of transmissions required by F-GG. In dense networks (such as *g* = 45), these reduction percent of SFR comparing with F-GG can be up to 40%. Since each node, performing SFR and F-GG, requires its one-hop neighbor information only, SFR has the exact performance gain stated above comparing with F-GG. In addition, F-RNG performs worst among all these approaches. This is because RNG is a sub-graph of GG and RNG is sparser than GG in terms of the edge density.

Comparison between SFR and F-UDel

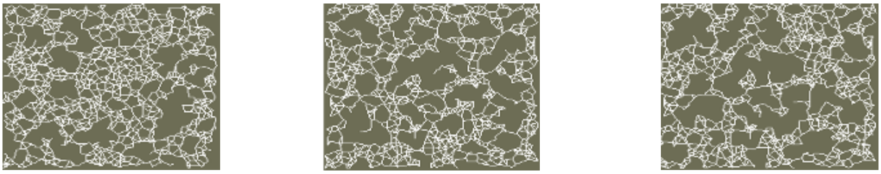
F-UDel reduces 15% to 30% of the number of transmissions of SFR in networks with various densities. However, *UDel*(*V*) can not be constructed locally. By other words, the construction of *UDel*(*V*) requires global knowledge of the network topology, and the construction cost is therefore very high (not realistic). A localized Delaunay triangulation LDel was proposed in [16] with a construction cost no more than (37*q* + 13*p* + 100)*n* bits data transmission, where *p* is the number of bits to represent a node location, q is the number of bits to present a node ID, and n is the total number of nodes in the network. Meanwhile, LDel does not have non-trivial upper bounds on maintaining *LDel*(V). Since the constructed graph of LDel is very similar to *UDel*(V), we use its construction cost as an achievable cost to construct *UDel*(V). On the other hand, SFR only requires $(p + q + 1)n$ bits data transmission to construct *GG*(V), which is much smaller than the construction cost of UDel and LDel.

From Fig. 12, we observed that the higher the network density is, the larger the percentage of saved transmissions can be achieved by SFR. The transmission distances of F-GG and SFR decrease when the network density increases. This observation results in the positive slopes of the F-GG and SFR curves in Fig. 12.

4.2 Comparison of Total Number of Transmissions in GFG Routing

In the second set of experiments, we embed the face routing algorithms discussed in Section 4.1 in GFG routing [4, 5] and evaluate their performance. GFG routing switches between two modes: greedy mode and face mode. The currently adopted algorithm in face mode is FACE [3]. Therefore, we compare SFR embedded in GFG routing with the original GFG. Similar to the notations of the algorithms in Section 4.1, in this section, we use SFR, F-GG, F-RNG, and F-UDel to denote the GFG routing embedded with these five algorithms, respectively.

Simulation is performed on a set of sample void networks generated from the base networks stated in Section 4.1. To simulate the performance of GFG routing, for each base network with a fixed average degree, we randomly place m number of 1.5 × 1.5 square voids within the network area, and all the nodes in the voids are removed. The value of m is varied from a list {10, 15, 20, 25, 30, 35, 40}. Fig. 14 shows Gabriel graphs of three sample void networks with 15, 25, and 35 voids, generated from three base networks with g = 10. Using these void networks is because in practical applications, due to node mobility and existence of obstacles, the networks shown in Fig. 14 are more realistic than networks with uniformly distributed nodes.



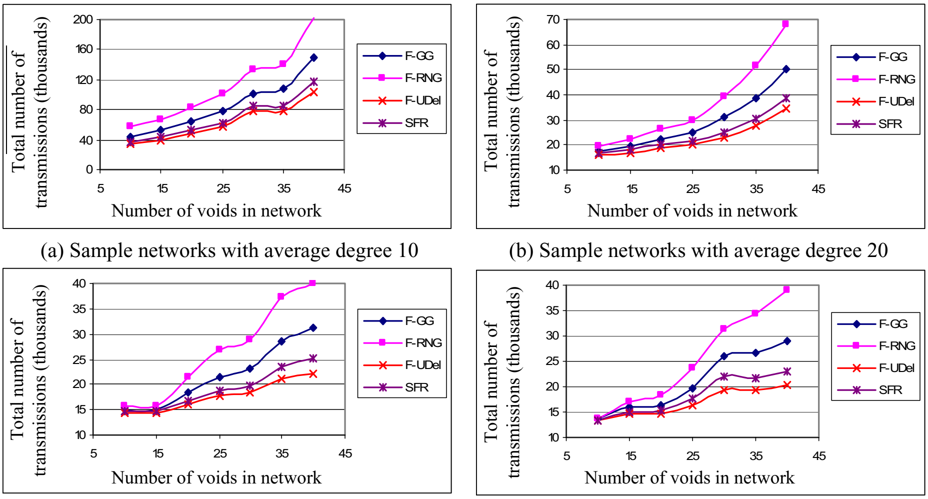
(a) Network with 15 voids (b) Network with 25 voids (c) Network with 35 voids

Fig. 14. Void networks generated from base networks with average degree 10

For each void network, we randomly select 100 source-destination pairs, perform the five GFG algorithms for these node pairs, and record the total transmission number. The experimental results are shown in Fig. 15 for networks with different void number and different density. All reported results are computed by summing up the total transmission numbers generated by 10 sample void networks.

Since all the compared algorithms perform the same algorithm in greedy mode, the differences of total transmission numbers among these algorithms are generated from the transmissions associated to face mode. According to Fig. 15, performance of

F-UDel is superior to all other algorithms, and followed by SFR, F-GG, and F-RNG. In addition, we have following observations.



(a) Sample networks with average degree 10

(b) Sample networks with average degree 20

(c) Sample networks with average degree 30

(d) Sample networks with average degree 40

Fig. 15. Total number of transmissions in GFG embedded with four algorithms

First, in sample networks with a small number of voids, the performance gain of SFR is not significant comparing with F-GG. This is because the total number of transmissions is dominated by the transmissions in greedy mode, which are same for all compared algorithms. Second, for void networks generated from the base networks with a fixed average degree, the larger the number of voids in the network, the higher the performance gain can be obtained by using SFR. The reduced transmissions by using SFR can be up to 30% of the total transmission number of SFR-GG. Third, network density has no significant impact on the relative ratio between the total transmission numbers of two compared algorithms, but has a significant impact on the total transmission numbers for each algorithm. The higher the density of a base networks, the smaller the total number of transmissions required by each algorithm.

5 Concluding Remarks

Face routing has been designed to guarantee message delivery and played an important role in wireless ad hoc and sensor networks. The existing face routing algorithms have a major drawback: longer traversal paths. This drawback is caused by the short average transmission distance among Gabriel neighbors or RNG neighbors. Even though face routing applied on unit Delaunay triangulation traverses a face with a relative long average transmission distance, the construction and maintenance cost of unit Delaunay triangulation is very high.

In this paper, we propose a face routing techniques, Skipping Face Routing (SFR), to overcome the problem in the existing approaches. In SFR, each node knows the locations of its one-hop neighbors only. According to the simulation studies, SFR outperforms the existing face traversal approaches in terms of total number of transmissions. For dense networks, the reduced total number of transmission by using SFR can be up to 40% of the total transmissions in the existing approaches on Gabriel graph. Furthermore, SFR can be embedded in GFG routing and have significant performance gain (up to 30% transmission saving) comparing with the existing GFG in networks with a large number of voids.

References

[1] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or why Six is a Magic Number," Conference Record, National Telecommunications Conference, December 1978, pp. 432-435.

[2] G.G. Finn, "Routing and Addressing Problems in Large Metropolitan-scale Internetworks," *ISI Research Report ISU/RR-87-180*, 1987.

[3] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," In *Proc. Canadian Conference on Computational Geometry*, Vancouver, August, 1999.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks" in *Proc. ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications* DIAL M99, 1999, pp. 48-55.

[5] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," ACM Wireless Networks, vol. 7(6), 2001, pp. 609-616.

[6] Brad Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. ACM/IEEE MOBICOM*, August 2000. pp. 243-254.

[7] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically Optimal Geometric Mobile Ad-Hoc Routing," in *Proc. Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM Press, 2002, pp. 24-33.

[8] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," in *Proc. MobiHoc*, 2003.

[9] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric Ad-hoc Routing: Of Theory and Practice," in *Proc. PODC*, 2003.

[10] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," IEEE Wireless Communication, vol. 11(6), December. 2004, pp. 6-28.

[11] J. Liu, "A Distributed Routing Algorithm in Mobile Packet Radio Networks," University of Illinois, Urbana, TR, 1980.

[12] R. Nelson and L. Kleinrock, "The Spatial Capacity of a Slotted ALOHA Miltihop Packet Radio Network with Capture," IEEE TON, vol. 32(6), 1984, pp. 684-649.

[13] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," IEEE TON, Vol. 32(3), 1984, pp. 246-257.

[14] M. Zorzi, and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multi-hop Performance," IEEE TMC, vol. 2(4), 2003, pp. 337-348.

[15] I. Stojmenovic, "Geocasting with Guaranteed Delivery in Sensor Networks," IEEE Wireless Communications, vol. 11, no. 6, 2004, pp. 29- 37.

[16] X. Li, G. Calinescu, P. Wan, and Y. Wang, "Localized Delaunay Triangulation with Applications in Ad Hoc Wireless Networks," IEEE TPDS, vol.14, 2003, pp. 1035-1047.

[17] G. Calinescu, “Computing 2-hop Neighborhoods in Ad Hoc Wireless Networks,” AD-HOC Networks and Wireless (AdHoc-NOW), Oct 2003, pp. 175-186.

[18] F. P. Preparata and M. I. Shamos, “Computational Geometry: An Introduction,” Springer-Verlag, 1985.

[19] J. Lian, K. Naik, and G. Agnew, “Data Capacity Improvement of Wireless Sensor Networks Using Non-Uniform Sensor Distribution,” *International Journal of Distributed Sensor Networks*, 2005.

[20] S. Giordano, I. Stojmenovic, “Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy,” Ad Hoc Wireless Networking, Kluwer, 2004, 103-136.

[21] Y. Liu, L. Xiao , X. Liu, L. M. Ni, and X. Zhang, "Location Awareness in Unstructured Peer-to-Peer Systems," IEEE TPDS, vol. 16(2), 2005, pp. 163-174.

[22] S. Datta, I. Stojmenovic, J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," Cluster Computing, vol. 5(2), April 2002, pp. 169-178.

[23] L. M. Ni, Y. Liu, Y. Lau, and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," ACM Wireless Networks, vol. 10(6), 2004, pp. 701-710.

[24] P. Boone, E. Chavez, et al, “Morelia Test: Improving the Efficiency of the Gabriel Test and Face Routing in Ad-hoc Networks,” in *Proc. SIROCCO’04*, 2004, pp. 23-34.

[25] W. Xue, Q. Luo, L. Chen, Y. Liu, "Contour Map Matching For Event Detection in Sensor Networks," in *Proc. ACM SIGMOD*, June, 2006.

[26] J. Wu, and H. Li, “On calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks,” Telecommunication Systems, vol. 18(1-3), 2001, pp. 13-36.

[27] Stojmenovic, M. Seddigh, J. Zunic, "Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks," IEEE TPDS, vol. 13(1), 2002, pp. 14-25.

An Anti-void Geographic Routing Algorithm for Wireless Sensor Networks[*]

Ming Xu, Yingwen Chen, Yi Wu, and Wanrong Yu

School of Computer, National University of Defense Technology,
410073 Changsha, Hunan Province, China
xuming64@public.cs.hn.cn, ywch_nudt@hotmail.com,
wswymagic@163.com, yu.wanrong@gmail.com

Abstract. Wireless sensor networks have attracted great attention in research and industrial development due to their fast-growing application potentials. Most of the existing geographic routing algorithms for wireless sensor networks are based on maintaining *one-hop neighbors* on the sensor node, leading to a well-known void problem. In this paper, we demonstrate that we can improve routing performance by storing more neighbors (called *spatial neighbors*) on the sensor nodes so as to avoid the void problem. We propose a neighborhood discovery algorithm and a neighborhood maintenance strategy to collect and maintain the *spatial neighbors* for each node. Based on the *spatial neighbors* on each node, we propose an *Anti-Void Geographic Routing* algorithm. Simulation results show that the AVGR routing algorithm outperforms the typical routing algorithm GPG/GPSR, especially in networks with more voids.

1 Introduction

A *Wireless Sensor Network* (WSN) consists of a collection of communicating nodes, which are placed across a distributed geographical area. Each node is incorporated with one or more sensors for measuring parameters or identifying control states in the environment, collecting real-time data, or extracting value-added information to the *sink nodes*. This new kind of networks exhibits a number of advantages over traditional sensing methods and is becoming increasingly popular in military applications and other risk-associated applications[1].

A popular routing algorithm for WSNs that has been widely studied is geographic routing[2,3,4]. In a geographic routing scheme, a source node knows the location of the destination node, either by acquiring it from a GPS device[3] or a location service[5], or by computing it using a hash function in a data-centric storage scheme[6]. Each node in the WSN maintains the information of its neighbors by adopting *Neighborhood Discovery Protocol.* Data packets are greedily forwarded to the neighbor node which is closer to the destination than the current node. This process is repeated until the packet reaches the destination. In a non-localized routing algorithm, each node

[*] This research is partially supported by the Doctoral Foundation of National Education Ministry of China under Grant No.20059998022.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 60 – 71, 2006.

maintains an accurate description of the overall network topology to compute the next hop, so that the routing path with global minimal hop count can be found. However, considering the scalability and pragmatic issues of a large WSN with an arbitrary number of nodes, each node decides the next hop based only on its *one-hop neighbors* (localized information) in most of the existing routing algorithms[2,3,7,8]. Thus, the aforementioned geographic greedy forwarding process might suffer from the so called *void problem*, that is, a packet gets stranded at a node whose *one-hop neighbors* are all further away from the destination.

Since the geographic routing algorithms explore the geographic greedy forwarding process in the neighbor list of each node, existence of more neighbors for each node might increase the chance to avoid the void problem. In this paper, we demonstrate that we can avoid the void problem of the traditional geographic routing algorithms by maintaining the information of the *spatial neighbors* (defined in Section 3) other than *one-hop neighbors* on each node, while only increasing the storage and memory requirements from $O(\frac{2n_e}{n_v})$ to $O(\frac{4n_e}{n_e - n_v + 2})$, where n_e and n_v are the numbers of edges and vertices of the communication graph of WSN, respectively.

The contribution of the paper is threefold. First, we introduce the definition of the *spatial neighbor* of a node and construct an efficient neighborhood discovery algorithm and an efficient neighborhood maintenance strategy. Second, we propose the AVGR algorithm with two greedy packet forwarding strategies and AVGR can improve the performance of the geographic routing by avoiding the void problem. Third, we conduct simulation studies in NS2[9] to evaluate the performance of our routing algorithm and compare it with GPG/GPSR algorithm. Our simulation results indicate that AVGR always outperforms GPG/GPSR, especially when there are more topology voids in the network. In some cases, the hop count of the routing path found by AVGR is only ***half*** of that found by GPG/GPSR.

The remainder of this paper is organized as follows. Section 2 introduces the related work of geographic routing algorithms. Section 3 presents some preliminary definitions and observations of our AVGR algorithm. In Section 4, we construct the spatial neighborhood discovery and maintenance algorithm and propose our AVGR algorithm. In Section 5, we conduct simulated experiments to evaluate the performance. Finally, we conclude the paper briefly and outline some of our future research directions.

2 Related Work

The early proposals of geographic routing algorithm, suggested about a decade ago, purely adopted the geographic greedy forwarding process[10,11,12]. Thus, when suffering the *void problem*, these routing algorithms could not guarantee the messages be delivered to the destination.

To overcome the limitation of the early algorithms, there have been later suggestions to guarantee the message delivery[2,13]. In literature [13], the authors proposed the first geographic routing algorithm called *Face Routing*, which does guarantee delivery. *Face Routing* proceeds towards the destination by exploring the boundaries of the faces of a planarized network graph, employing the local *right hand rule* (in analogy to following the right hand wall in a maze). *Face Routing* guarantees to reach the destination after $O(n)$ steps, n being the number of network nodes. However, if n becomes very large, *Face Routing* may be very in-efficient because the hop count of the routing path may be very large. To improve the performance of *Face Routing*, Bose et al. [2] described a Greedy-Face-Greedy (GFG) algorithm that guarantees delivery of messages in MANETs. The transformation of this algorithm into a protocol named Greedy Perimeter Stateless Routing (GPSR) was later presented by Karp and Kung[3]. GFG/GPSR combines both the geographic greedy forwarding process and the *right hand rule*. It can switch between two of them when necessarily. GFG/GPSR also provides delivery guarantees and is somewhat more efficient in the average case than *Face Routing*, though it does not outperform *Face Routing* in the worst case. Literature [14] studied the geographic routing issues and proposed a Distance Updating Algorithm (DUA) in a relaxed environment, where all the sensor nodes need to send packets to a common specified sink node. DUA replaces the *right hand rule* by distance upgrading and eliminates the void problem, so that it can find routing path with smaller hop count than GFG/GPSR. However, DUA is not flexible enough to support the typical sensor network application in [15]. Our Anti-Void Geographic Routing algorithm is related to the routing algorithms above.

3 Preliminaries

In this section we introduce some preliminary definitions and observations for our AVGR algorithm. The key contribution of AVGR algorithm is that we can avoid the void problem based on the *spatial neighbors* (to be discussed in Section 3.3) of each sensor node. Since the definition of *spatial neighbor* relies on some other definitions, we will introduce them one by one.

3.1 Planarized Graph

Because the concept of the *face* (to be discussed in Section 3.2) is defined on a planar graph that has no crossing links, we assume the sensor network is planar. Although a randomly deployed WSN may not be planar, we can planarize the network graph by using the *Gabriel Graph* (GG) [17], the *Relative Neighborhood Graph* (RNG) [18], or *Restricted Delaunay Graph* (RDG) [19]. These graph constructs provably yield a connected, planar graph so long as the connectivity between nodes obeys the unit-disk assumption. Without loss of generality, we adopt GG graph in our cases.

A sensor network can be modeled as a graph G=<V, E, E'>, where V represents the set of all the sensor nodes, and E represents the set of all the edges between two nodes when they are within each other's communication range, and E' represents the set of all the edges that after the graph being planarized.

Definition 1. Two nodes u, v are mutually called *one-hop neighbors* if and only if $e \underline{\underline{\Delta}} (u,v) \in E$. Denoting $\delta(e)$ as the set of the two vertices of edge e, given node u, the *one-hop neighbors* of u can be defined as $HN(u) = \bigcup_{(e \in E) \wedge (\delta(e) \cap \{u\} \neq \phi)} \delta(e) - \{u\}$.

Definition 2. Two nodes *u*, *v* are mutually called *planar neighbors* if and only if $e \triangleq (u,v) \in E'$. Denoting $\delta(e)$ as the set of the two vertices of edge *e*, given node *u*, the *planar neighbors* of *u* can be defined as $PN(u) = \bigcup_{(e\in E')\wedge(\delta(e)\cap\{u\}\neq\phi)} \delta(e) - \{u\}$.

Observation 1. Since the planarization of WSN does not change the unit-disk assumption, we have $E' \subseteq E$.

Observation 2. Planarization of WSN may increase the hop count between *one-hop neighbors.*

In a planar graph, denoting $d(u,v)$ as the minimal hop count between any two nodes, we have $d(u,v)=1$ if and only if $(u,v)\in E'$. For any $(u,v)\in E$ but $(u,v)\notin E'$, we have $d(u,v)>1$. Taking the GG graph in Fig.1 as an example, all the four nodes are in the communication range of each other. However, edge (A,C) is removed to eliminate the crossing links. As a result, even though *A*, *C* are mutually *one-hop neighbors*, their hop count $d(A,C)=2$.

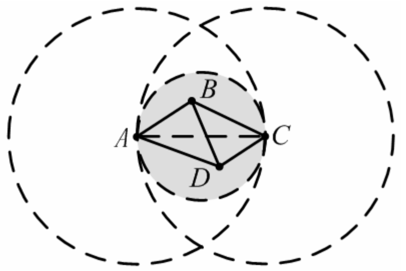


Fig. 1. The GG graph

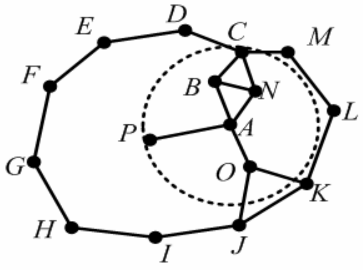


Fig. 2. Faces in a planar graph

3.2 Faces of a Graph

In order to clarify the definition of the *face* of a graph, we introduce the notation for the *right hand rule* first. We note three nodes v_1, v_2, v_3 fulfilling the *right hand rule* as *Right*(v_1, v_2)=v_3, where $v_1, v_2, v_3 \in V$, and v_3 is the first *planar neighbor* of v_1 that intersects with the counter-clockwise rotating radial centered at v_1 and started from $\overrightarrow{v_1 v_2}$.

Definition 3. A sequence of nodes $f=(v_1,v_2,...,v_{n-1},v_n)$ is called a *face* if and only if: (1) $(v_i,v_{i+1})\in E', 1\le i\le n-1$; (2) $(v_n,v_1)\in E'$; (3) $Right(v_{i+1},v_i)=v_{i+2}, 1\le i\le n-2$; (4) $Right(v_n,v_{n-1})=v_1$ and $Right(v_1,v_n)=v_2$.

For instance, in the planar graph as shown in Fig. 2, node *P* only has one face and *A* has three adjacent faces. Node that the boundary node *M* of the network has two adjacent faces. One of them is the *inner face* formed by nodes *M-L-K-O-A-N-C*. The other one is the *out face* formed by nodes *M-C-D-E-F-G-H-I-J-K-L.*

Definition 4. Two nodes are mutually called *face neighbors* if and only if they are on the same face. Denoting Γ as all the faces in the planar graph and $\sigma(f)$ as the set of all the nodes on face *f*, given node *u*, the *face neighbors* of *u* can be defined as $FN(u) = \bigcup_{(f\in\Gamma)\wedge(\sigma(f)\cap\{u\}\neq\phi)} \sigma(f) - \{u\}$.

Observation 3. The geographic greedy forwarding process can avoid the *void problem* by using the information of the *face neighbors.*

Taking Fig. 2 as an example, suppose node *A* need to send data packet to node *F*. Based on the information of *one-hop neighbors*, *A* greedily forwards the packet to *P*, which is the nearest to *F* among the *one-hop neighbors* of *A*. However, when the packet is forwarded to *P*, there comes the *void problem* that there is no *one-hop neighbor* of *P* closer to *F* than *P* itself. As a result, the routing path found by GPSR is *A-P-A-O-J-I-H-G-F*, which is 8 hops. On the other hand, by adopting the definition of *face neighbors*, we can see that *F* is the *face neighbor* of *A*, and there is no void between *A* and *F*. If *A* has maintained the information of its *face neighbors*, it can directly forward the packet to *F* along a shorter side of the face *A-B-C-D-E-F.* which is only 5 hops.

Observation 4. Combing the *one-hop neighbors* of a node together with its *face neighbors* can reduce the hop count of the routing path.

As we have explained in the above example, based on the information of the *face neighbors* we can find a 5-hop routing path from *A* to *F*. However, node *C* is *one-hop neighbor* of node *A*. We can short cut the path directly from *A* to *C* other than trace along the reversed face *A-B-C*, further shortening the routing path to 4 hops.

3.3 Spatial Neighbors

Learning from the properties in Observation 3 and Observation 4, we define the *spatial neighbors* of a node by combining its *one-hop neighbors* with its *face neighbors* as follows.

Definition 5. The *spatial neighbors* of a node *u* can be defined as $SN(u) = HN(u) \bigcup FN(u) = \bigcup_{(e\in E)\wedge(\delta(e)\cap\{u\}\neq\phi)} \delta(e) \bigcup \bigcup_{(f\in\Gamma)\wedge(\sigma(f)\cap\{u\}\neq\phi)} \sigma(f) - \{u\}$.

In literature [16], the authors also proposed a definition for *spatial neighbors*. However, their definition is different from ours. The *spatial neighbors* defined here include both the *one-hop neighbors* and *face neighbors*. Therefore, they have good properties of Observation 3 and Observation 4. We can construct routing algorithm based on our *spatial neighbors* to avoid the *void problem* and reduce the hop count of the routing path.

Observation 5. The average number of *one-hop neighbors* for a node is $O(\frac{2n_e}{n_v})$, while that of its *spatial neighbors* is $O(\frac{4n_e}{n_e - n_v + 2})$, which is acceptable for a localized routing algorithm. The parameters n_e and n_v are the numbers of edges and vertices of the communication graph of WSN, respectively.

It is obvious that the average number of *one-hop neighbors* of a node equals to its average node degree, which is $O(\frac{2n_e}{n_v})$. As is proved in 16, the average size of a face is $O(\frac{2n_e}{n_e - n_v + 2})$, and the average number of *spatial neighbors* is $O(\frac{2n_e}{n_e - n_v + 2} \cdot \frac{2n_e}{n_v})$, which can be simplified as $O(\frac{4n_e}{n_e - n_v + 2})$.

4 Anti-void Geographic Routing

In Section 4, we introduce all the components of our AVGR algorithm, including spatial neighborhood discovery, spatial neighborhood maintenance, and packet forwarding strategies.

4.1 Spatial Neighborhood Discovery

The *spatial neighbors* of a node are composed of *one-hop neighbors* and *face neighbors*. The purpose of the spatial neighborhood discovery process is to identify all these two kind of neighbors. Since the *one-hop neighbors* can be easily identified by exchanging beacons with immediate neighbors once, we focus on the process for face neighborhood discovery. In order to reduce the communication overhead of the face neighborhood discovery process, we adopt a two-stage method. First, an initiator node is elected by a distributed leader election algorithm to initiate the discovery process. After that, the initiator node will create the discovery message to discover the neighborhood for all the nodes on the same face.

Applying the Gabriel planarization method[17], each node v not only knows who their *one-hop neighbors* are, but also who among them are its *planar neighbors*. Adopting the *right hand rule* on the *planar neighbors* of each node, we can identify all the faces in the planar graph. We assume that there are no two sensor nodes located at the same coordinates. Therefore each node can be uniquely identified by its coordinates. By adopting the distributed leader election algorithm in each ring or on each face[20], we can elect the left-down most node as the initiator node to initiate the face neighborhood discovery process. That is, the node on a face with the minimal x-coordinates is the initiator node; if there are more than one node with the minimal x-coordinates, the node with minimal y-coordinates is the initiator node.

The elected initiator node creates a discovery message and the discovery message is forwarded by each node on the face adopting the *right hand rule*. As the discovery message traverses the face, the coordinates of the nodes it has traversed are added to the message. After the discovery message finishes traversing the face, all nodes' locations on the face are collected and the complete discovery result traverses the same face another time to inform every node on the face. Then the node can obtain the information of all its *spatial neighbors* by combing all its *one-hop neighbors* and *face neighbors*.

The spatial neighborhood discovery process is executed only once as soon as the sensor network is deployed. Even though the topology of the sensor network is dynamic and the *spatial neighbors* of each node may keep changing, we can construct an efficient spatial neighborhood maintenance mechanism so as to localize the updating of the *spatial neighbors*. We introduce the spatial neighborhood maintenance mechanism in the following part.

4.2 Spatial Neighborhood Maintenance

Although in most cases the sensor nodes are not mobile, we cannot assume that the *spatial neighbors* of each node will stay constant all the time. This is due to the inherently dynamic nature of sensor networks, involving node failures, new nodes joining the network, etc. In this section, we discuss how to effectively maintain the *spatial neighbors* of each node without global information exchange.

Similar to other geographic routing algorithms, every node in AVGR periodically broadcasts a beacon packet to its neighbors. This periodic beaconing is used for exchanging location information between neighbors. Authors of [7] argue that the beaconing rate can be very low when nodes inside the sensor network are stationary or slow moving. Moreover, piggybacking[2] methods can also be exploited to reduce this beacon overhead. In AVGR, each node keeps an *ExpireTime* parameter for each *one-hop neighbor* to timeout this neighbor. If the beacon of any *one-hop neighbor* can not be heard after a certain timeout, it will be removed from the *one-hop neighbor* table. If the beacon of a new *one-hop neighbor* is heard, it will be added to the *one-hop neighbor* table.

Whenever a *changed node* joins or leaves the network, the appearance or disappearance of its periodical beacons will inform its *one-hop neighbors*, called *influenced nodes*. Then the *spatial neighbors* of all the *influenced nodes* should be rediscovered. This operation can be described in a three-step manner. Firstly, the *influenced node* re-computes its *planar neighbors* based on the new *one-hop neighbors*. Secondly, if the *planar neighbors* of the *influenced node* get changed, the *influenced node* sends a probing message to repair the broken face. Note that, in order to reduce the overhead of this repairing operation, the probing message is forwarded towards the direction of the *changed node* by adopting the *right hand rule* or *contradictory right hand rule*. Finally, when the face gets repaired, the left-down most *influenced node* on the face will act as the initiator node to forward an updating message adopting the *right hand rule*. When the updating message has toured around the face, the *face neighbors* of each node have been refreshed. Fig. 3 illustrates the spatial neighborhood maintenance process when node *B* is eliminated from the graph.

As Fig. 3 shows, we assume that node *B* leaves the network. Node *D*, *C*, *M*, *N*, *A* are all *one-hop neighbors* of node *B*, and they are all *influenced nodes*. However, only the *planar neighbors* of node *C*, *N*, *A* get changed. Both node *A* and node *C* find that the original face *A-P-A-O-J-I-H-G-F-E-D-C-B* is broken. As the dashed arrows show, both of them send out a probing message towards the direction of node *B* to repair the face. As a result, node *A* sends out the message by adopting the *contradictory right hand rule* while node *C* sends out the message by adopting the *right hand rule*.

Because node *A* is on the left of node *C*, when *A* gets the probing message from node *C*, it will initiate the updating message to tour along the path *P-A-O-J-I-H-G-F-E-D-C-N-A*. Note that, even though the *planar neighbors* of node *N* also get changed and the face *N-A-B* and *N-B-C* are broken, there are no other *planar neighbors* for node *N* to repair the broken face. As a result, no probing message is initiated by node *N*, and the information of face *N-A-B* and *N-B-C* can be deleted.

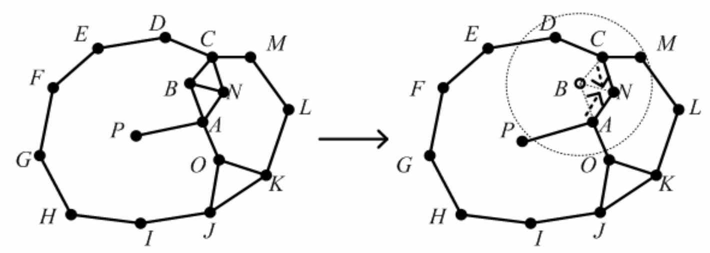


Fig. 3. Spatial neighborhood maintenance process when node *B* is eliminated from the graph

As is described above, when a node joins or leaves the network, the topology updating message only needs to be forwarded to a small part of all the sensor nodes. The number of nodes involved in the spatial neighborhood maintenance process is proportion to the number of *spatial neighbors* of the node, which is $O(\frac{4n_e}{n_e - n_v + 2})$ in average. We can conclude that our spatial neighborhood maintenance strategy is efficient and does not need global periodic broadcasts.

By adopting the spatial neighborhood discovery and spatial neighborhood maintenance process, we can temporally store the local topology information on each node. Next we will discuss how to make use of this local topology information to effectively forward the packet to the destination.

4.3 Greedy Forwarding

As we have introduced in section 3, *spatial neighbors* of each node have many good properties. Taking full advantage of the *spatial neighbors* leads to great improvement of our AVGR routing algorithm. We discuss the packet forwarding strategies in this section and introduce the improved performance of our AVGR algorithm by simulation results in section 5.

In order to clarify the greedy forwarding process, we give some notations first. Suppose there is a node *u* that has a packet to be forwarded to a destination. Learned from Observation 3, node *u* should lookup its *spatial neighbors* to find the node *v*, which has the shortest Euclidean distance to the destination node. Because the face information is maintained on node *u*, it is easy for node *u* to find the minimal hop count path from *u* to *v*, noted as P_{uv}. Learned from Observation 4, path P_{uv} should be short cut as $\widetilde{P}_{uv}$ if possible. Noting node *w* as the next hop for node *u* in the path $\widetilde{P}_{uv}$, then node *u* has the following two different greedy forwarding strategies:

Greedy 1: Just forward the packet to node w.

The basic idea of this forwarding strategy is to forward the packet to the face that contains the *spatial neighbor* closest to the destination. This process is repeated until the packet gets to the face containing the destination. Finally, the packet is forwarded along the minimal hop count path on the face to the destination.

Greedy 2: Forward the packet directly to node v along the path $\tilde{P}_{uv}$.

The basic idea of this forwarding strategy is to directly forward the packet to the *spatial neighbor* closest to the destination. This process is repeated until the packet finally gets to the destination.

Since it is hard to tell which is better between *Greedy 1* and *Greedy 2*, we will evaluate them by simulations in the next section.

5 Simulation Results and Evaluation

In this section, we present the results of our simulation study. We evaluated the performance of our AVGR algorithm and made a comparision with GPSR[21]. The performance of the routing algorithm is mearsured by the hop count of the routing path. We also investigated the effects of some parameters such as the communication distance from the source node to the destination node and the node density.

In our simulation, the sensor nodes are distributed in a region δ, according to the uniform distribution. A communication graph is generated under the assumption that all the nodes have the same transmission range ρ. A summary of the communication and sensor network parameters and their default values is presented in Table 1.

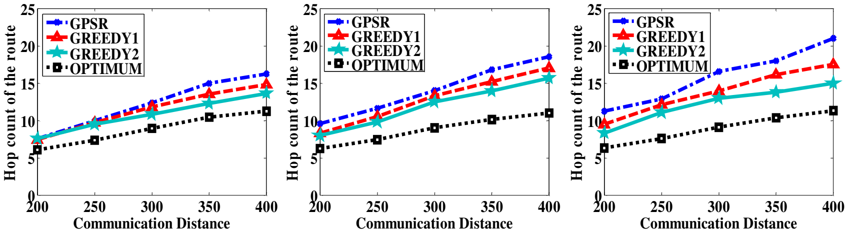
Table 1. Parameters of communication and sensor network

Parameter	Symbol	Default value
Coverage of sensor network	δ	500 by 500
Number of sensor nodes	*N*	200-500
Transmission range	ρ	50
Communication distance	*D*	200-400

We simulated different kind of routing algorithms and strategies, including GPSR, GREEDY1, GREEDY2, and OPTIMUM. Both GREEDY1 and GREEDY2 are AVGR oriented algorithm with packet forwarding strategy *Greedy 1* and *Greedy 2* separately. In OPTIMUM, the routing path is the minimal hop path found by global flooding in the overall network. Because global flooding is very expensive in distributed systems, it is not practical in a real sensor network. We just use the performance of OPTIMUM as a benchmark for comparison. We generate 30 connected network instances for each simulation and spawn 100 communications in each network instance. The average performance for each communication in each network topology is measured and the overall performance is obtained as an average over all the 30 topologies. In order to evaluate the performance of the routing algorithms under the condition with different topology voids, zero to two artificial voids are randomly placed in the area. The artificial void is a rectangular with a default area size of 50 by 100. All the node in the void are eliminated from the network.

5.1 Impact of Communication Distance

The first set of simulated experiments aims at evaluating the performance of the routing algorithms with different communication distance *D*. The communication distance is the Euclidean distance between the source and the destination, and it reflects how far it is from the source node to the destination node. In order to make the communication distance controllable, for each communication we temporarily add distance-specified source node and destination node pair into the network. In this experiment, we fix the number of sensors *N* to 300. The results are depicted in Fig. 4.



(a) with 0 artificial void (b) with 1 artificial void (c) with 2 artificial voids

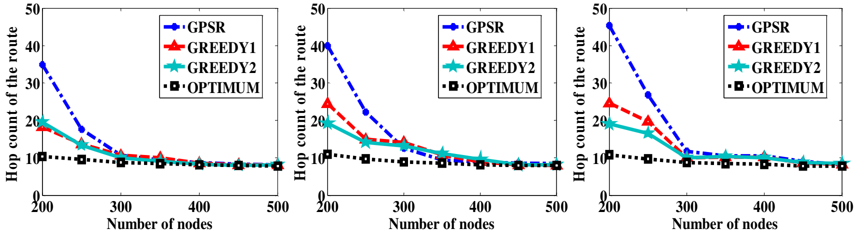
Fig. 4. Performances according to different communication distance

From Fig. 4, it is obvious that AVGR always outperforms GPSR, especially when adopting the *Greedy 2* forwarding strategy. The *Greedy 2* forwarding strategy is better than the *Greedy 1* forwarding strategy. Fig. 4(a) shows that even though there is no artificial void, AVGR also performs better than GPSR when the communication distance is increasing. This is because the topology void inevitably exists due to the randomly deployed sensor nodes. As the communication distance *D* increases, it is more likely to meet with a void when forwarding a packet from the source node to the destination node. Combining the results of Fig. 4(b) and Fig. 4(c), we can conclude that GPSR performs very poor in case of topology voids. However, AVGR-GREEDY2 performs much better than GPSR by avoiding the void problem under the condition that there are more topology voids.

5.2 Impact of Node Density

Since the topology of the sensor network is affected greatly by the node density, we investigate how the node density will affect the performance of the routing algorithms. In this experiment, the distance of each two communication nodes is fixed to 300, and varying the number of nodes *N*, and hence node density. The results are depicted in Fig. 5.

From Fig. 5, it is obvious that AVGR still always outperforms GPSR. GPSR is very density sensitive. In the case that there are two artificial voids in a 200-node sensor network, the routing path found by GPSR is ***two times longer*** than that found by AVGR-GREEDY2. That is because, the lower the node density is the more the topology voids might come into being. However, when the node density is large enough, both GPSR and AVGR perform closely to OPTIMUM. This is reasonable, because when there are enough nodes, any kind of greedy routing algorithm will get enough nodes to make a decision to choose a more suitable next hop for packet forwarding. Summarized from all the three figures in Fig. 5, AVGR-GREEDY2 performs nearly the same even though there are more topology voids. This once again proves that AVGR can avoid the void problem, and it is more suitable than GPSR for WSNs with low node density.



(a) with 0 artificial void (b) with 1 artificial void (c) with 2 artificial voids

Fig. 5. Performances according to different node density

6 Conclusions

This paper proposed a new geographic routing algorithm that can avoid the void problem of existing routing algorithms for WSNs. It produces more efficient routing paths than the traditional GFG/GPSR algorithm. Furthermore, No global periodic broadcasts are required to maintain the routing paths. The algorithm can respond to topology changes instantly with localized operations. Our future research work include more efficient data structure for the *spatial neighbors* and more efficient neighbor lookup algorithm for the AVGR algorithm.

References

1. D.C. Steere, A. Baptista, D. McNamee, C. Pu, and J.Walpole. Research challenges in environmental observation and forecasting systems. In *Proceedings of International Conference on Mobile Computing and Networking*, pages 292–299, 2000.
2. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM '99), Aug. 1999.
3. B. Karp and H. T. Kung. GSPR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 243–254, Boston, MA, August 2000.
4. R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communications*, 8(1):48–57, February 2001.

5. J. Li, J. Jannotti, D. DeCouto, D. Karger, and R. Morris, A scalable location service for geographic ad-hoc routing. In *Proc. 6th Annu. ACM/IEEE International Conference on Mobile Computeing and Networking*, 2000.
6. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, GHT: A geographic hash table for data-centric storage in sensornets. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002, pp. 78–87.
7. T. He, J. A.Stankovic, C. Lu, and T. F. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS'03)*, May 2003.
8. T. Melodia, D. Pomppili, and I. F. Akyildiz. Optimal local topology knowledge for energy efficient geographical routing in sensor networks. In *Proceedings of IEEE Infocom*, Hong Kong, March 2004.
9. http://www.isi.edu/nsnam/ns/
10. G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. *Technical Report ISI/RR-87-180*, ISI, March 1987.
11. T.Hou and V.Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(2):38-44,1986.
12. H.Takagi and L.Kleinrock. Opimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246-257, 1984.
13. E.Kranakis, H.Singh, and J.Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, pages 51-54, Vancouver, August 1999.
14. Shigang Chen, Guangbin Fan, Jun-Hong Cui. Avoid "Void" in Geographic Routing for Data Aggregation in Sensor Networks. *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Special Issue on Wireless Sensor Networks*, vol. 2, no. 1, 2006.
15. Hyung Seok Kim, Tarek F. Abdelzaher, Wook Hyun Kwon. Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks. In *Proceedings of ACM SenSys*, Nov. 2003.
16. Qingfeng Huang, Chenyang Lu, Gruia-Catalin Roman. Reliable Mobicast via Face-Aware Routing. In *Proceedings of IEEE Infocom*, Hong Kong, March 2004.
17. K.Gabriel, R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*. 18 (1969), 259–278.
18. G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*. 12, 4 (1980), 261–268.
19. J. Gao, L. Guibas, J. Hershberger, L. Zhang, A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. ACM MobiHoc*, pp. 45–55. Oct. 2001.
20. Roger Wattenhofer. Principles of Distributed Computing. Lecture Notes, www.dcg.ethz.ch/lectures/ ss05/distcomp/lecture/chapter2.pdf
21. http://www.icir.org/bkarp/gpsr/gpsr.html

A Correctness Proof of the DSR Protocol*

Huabing Yang, Xingyuan Zhang, and Yuanyuan Wang

PLA University of Science and Technology, Nanjing 210007, China
yanghuabing@gmail.com, xyzhang@public1.ptt.js.cn, wwyyyy@263.net

Abstract. The correctness of a routing protocol consists of two kinds of properties: safety and liveness. Safety properties specify that every route found by the protocol is well formed, while liveness properties specify that useful routes will eventually be found and data messages be eventually delivered to recipients. Many safety properties for routing protocols have been verified; however, the verification of liveness properties was overlooked. This paper stresses the importance of liveness properties of routing protocol and presents a formal verification of the DSR (Dynamic Source Routing) protocol dealing with both safety and liveness properties. The results are checked with Isabelle/HOL/Isar.

1 Introduction

DSR[1] is a MANET (Mobile Ad hoc Network) routing protocol. MANET is a kind of wireless network in which nodes cooperate by forwarding messages to each other so that communication beyond transmission range is achieved. Since MANET is completely self-organizing and self-configuring with no need of existing network infrastructure, its protocols tend to be complicated and formal verification of these protocols is indispensable.

There are a number of formal verifications for MANET routing protocols[2–6]. As shown in Table 1, none of these researches covers liveness properties, and all of them can only deal with network with limited number of nodes.

Although being difficult to prove, liveness properties are important for MANET routing protocols. The liveness property proved in this paper is of the following form: $?\sigma \models \Box\Diamond\, ?P \hookrightarrow \Box\Diamond\, ?Q$, in which $?P$ specifies the moment when a data message is sent and $?Q$ specifies the moment when the data message is received. The liveness property means: if a data message is sent infinitely often, then it will be received infinitely often. The establishment of this liveness property assures people that the routing protocol will never enter a trap in which no data message can be delivered.

This paper contributes by presenting a formal verification of the DSR protocol dealing with both safety and liveness properties. The liveness result is obtained

* This research was funded by National Natural Science Foundation of China, under grant 60373068 'Machine-assisted correctness proof of complex programs'.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 72–83, 2006.

Table 1. A comparison with previous works

Work	Protocol	Tool	Arbitrarily many nodes?	Properties	Conclusion
[2]	AODV	SPIN (Model checking)	No (*35* hops)	Safety	AODV is loop free
[3]	DSR	SDL	No (*5* nodes)	Safety	whether *Route Request* table correctly updated after receiving a *RREP*
[4]	WARP	SPIN (Model checking)	No (*5* nodes)	Safety	protocol reliability can be checked
[5]	GPSAL	SMC (Model checking)	No (*5* nodes)	Safety	GPSAL is loop free
[6]	AODV	Athena	No (*4* nodes)	Safety	route stability will be violated if attacks exist
This paper	DSR	Isabelle/HOL (Theorem proving)	Yes (N nodes)	Safety and liveness	DSR is loop free; data can be transmitted successfully in DSR

using a liveness proof approach devised by ourselves[7]. The results in this paper are checked with Isabelle/HOL/Isar[8, 9]. Scripts are available on demand. Because of theorem proving technology we used, all the results are about networks with arbitrarily many nodes. Such kind of results is not achieved by the researches in Table 1. And this comprises the second contribution of ours.

The rest of this paper is organized as follows. Section 2 introduces the definition of concurrent systems. Section 3 gives the formal description of DSR. Section 4 describes the safety properties of DSR. Section 5 presents the liveness proof. Section 6 concludes.

2 Concurrent Systems

In proving safety properties, only *finite* executions need to be used. The system state is identified with the finite execution, which is written as τ and represented as lists of events[10]. These events are arranged in reverse order of happening. Which event is to happen in next step is decided according to current system state.

In proving liveness properties, *infinite* executions should be considered. An infinite execution is written as σ, which is of type $nat \Rightarrow {}'a$. The event happened at step i is $\sigma\ i$ and usually abbreviated as σ_i. The first i events of an infinite execution σ can be packed into a list in reverse order to form a finite execution. Such a packing is written as $[\![\sigma]\!]_i$.

A routing protocol can be viewed as a concurrent system. A concurrent system is often written as cs and its type is $({}'a\ list \times {}'a)\ set$. The expression $(\tau,\ e) \in cs$ means that the event e is legitimate to happen under the system state τ, according to cs. The notation $(\tau,\ e) \in cs$ is abbreviated as $\tau\ [cs\!>\ e$. The set of *valid* finite executions of a concurrent system cs is written as $vt\ cs$, which is inductively defined. The expression $\tau \in vt\ cs$ means that the finite execution τ is a valid finite execution of cs. The expression $\tau \in vt\ cs$ is usually abbreviated as $cs \vdash \tau$. The operator $\vdash$ is overloaded, so that $cs \vdash \sigma$ can express that σ is a valid infinite execution of cs. Fig. 1 presents previous definitions.

constdefs i-th :: (nat $\Rightarrow$ 'a) $\Rightarrow$ nat $\Rightarrow$ 'a (-_ [64, 64] 1000)

$\sigma_i \equiv \sigma$ i

consts prefix :: (nat $\Rightarrow$ 'a) $\Rightarrow$ nat $\Rightarrow$ 'a list ($[\![$-$]\!]$_ [64, 64] 1000)

primrec $[\![\sigma]\!]_{0} = []$

$[\![\sigma]\!]_{(Suc\ i)} = \sigma_i \# [\![\sigma]\!]_i$

constdefs may-happen :: 'a list $\Rightarrow$ ('a list$\times$'a) set $\Rightarrow$ 'a $\Rightarrow$ bool (- [-> - [64,64,64] 50)

τ [cs> e $\equiv$ (τ, e) $\in$ cs

consts vt :: ('a list $\times$ 'a) set $\Rightarrow$ 'a list set

inductive vt cs **intros**

vt-nil [intro] : [] $\in$ vt cs

vt-cons [intro] : $[\![\tau \in$ vt cs; τ [cs> e$]\!] \Longrightarrow$ (e # τ) $\in$ vt cs

consts derivable :: 'a $\Rightarrow$ 'b $\Rightarrow$ bool (- $\vdash$ - [64, 64] 50)

defs (**overloaded**)

fnt-valid-def: cs $\vdash \tau \equiv \tau \in$ vt cs

inf-valid-def: cs $\vdash \sigma \equiv \forall$ i. $[\![\sigma]\!]_i$ [cs> σ_i

Fig. 1. The definitions of concurrent system

3 Overview and Formalization of DSR

3.1 DSR Overview

DSR is an on-demand reactive routing protocol. As shown in Fig. 2, DSR consists of two main processes: *Route Discovery* and *Route Maintenance*. *Route Discovery* is used by source nodes to find routes before sending data messages. Routes found by each source node are stored in *route cache*s for later use. *Route Maintenance* is used to keep the routes in *route cache*s up to date.

When a source node S^1 initiates a *Route Discovery* process, it broadcasts a *Route Request* message $\lceil RREQ\ D\ qid\ [S]\rfloor_{S,0}$, in which D is the destination node and *qid* is a number generated by S to uniquely identify this request message. The $[S]$ is the *address list* field of this *Route Request* message. On receiving a *Route Request* message, each intermediate node, such as A and B, appends itself to *address list* field before rebroadcasting. In this way, when a *Route Request* message reaches the destination node D, a route from S to D is already accumulated in its *address list* field. In Fig. 2, this accumulated route is $[S,A,B,D]$. And the destination node D will send a *Route Reply* message $\lceil RREP\ qid\ 2\ [D,B,A,S]\rfloor_{D,S}$, in which the *source route* field $[D,B,A,S]$ is obtained by reversing the *address list* field of the original *Route Request* message. The *Route Reply* is transmitted by unicast along the *source route* back to the source node S. The *2* in $\lceil RREP\ qid\ 2\ [D,B,A,S]\rfloor_{D,S}$ is called *segments left* field, which means the number of nodes still to be visited before the message reaching its destination. Each intermediate nodes will decrease the *segments left* field by *1* before forwarding the message.

Data from S to D is sent by S in a *Data* message $\lceil DATA\ dat\ 2\ [S,A,B,D]\rfloor_{S,D}$, in which *dat* is the data content. Each node along the path $[S,A,B,D]$ is

[1] In this paper, nodes are represented by capital letters such as S, D, A, B.

responsible for checking whether the *Data* message is indeed received by the next node. A link-layer acknowledgement will facilitate such a checking. The sender treats the link to the next node as 'broken' if it does not receive any acknowledgement in a certain period of time. Suppose node B detected such a break, it will send a *Route Error* message $\lceil RERR\ B\ D\ 1\ [D,B,A,S] \rfloor_{B,S}$ back to source node S. On receiving this message, node S will remove the 'broken' route $[S,A,B,D]$ from its *route cache.*

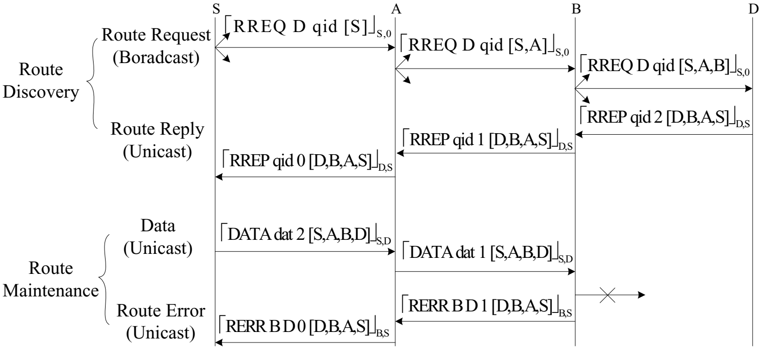

S
A
B
D
Route Discovery
Route Request (Boradcast)
Route Reply (Unicast)
⌈RREQ D qid [S]⌋S,0
⌈RREQ D qid [S,A]⌋S,0
⌈RREQ D qid [S,A,B]⌋S,0
⌈RREP qid 2 [D,B,A,S]⌋D,S
⌈RREP qid 1 [D,B,A,S]⌋D,S
⌈RREP qid 0 [D,B,A,S]⌋D,S
Route Maintenance
Data (Unicast)
Route Error (Unicast)
⌈DATA dat 2 [S,A,B,D]⌋S,D
⌈DATA dat 1 [S,A,B,D]⌋S,D
⌈RERR B D 1 [D,B,A,S]⌋B,S
⌈RERR B D 0 [D,B,A,S]⌋B,S

Fig. 2. The basic operation of DSR

3.2 DSR Formalization

Messages and Events. DSR nodes are identified with natural numbers, while the number *0* is reserved for broadcast address:

types *Node* = *nat* — *nat* expresses natural number.

As shown in Fig. 2, there are four kinds of messages: *Route Request*, *Route Reply*, *Data* and *Route Error*. Accordingly, *Msg-body* is defined as follows:

types *Qid* = *nat* — *Qid* expresses the identification of a *Route Request* message.

types *SegLeft* = *nat* — *SegLeft* expresses the *segments left* field.

typedecl *Dat* — *Dat* expresses the data content in a *Data* message, which is declared as an abstract type.

datatype *Msg-body* =
  RREQ Node Qid Node list — *Route Request*
| *RREP Qid SegLeft Node list* — *Route Reply*
| *DATA Dat SegLeft Node list* — *Data*
| *RERR Node Node SegLeft Node list* — *Route Error*

And messages are formalized as an Isabelle type *Msg*:

datatype *Msg* = *MSG Node Node Msg-body*

Therefore, the general format of a message is *MSG S D m*, in which S and D are source and destination nodes, m is the message body. With the following translation, *MSG S D m* can be rewritten more compactly as $\lceil m \rfloor_{S,D}$:

translations $\lceil m \rfloor_{S,D} \rightleftharpoons$ *MSG S D m*

The events that may happen in DSR are divided into six kinds:

```
datatype event =
    Send Node Msg                 — Send A msg: node A sends a message msg.
  | Recv Node Msg                 — Recv A msg: node A receives a message msg.
  | Move Node real×real           — Move A (x, y): node A moves to a new position (x, y).
  | Disturb real×real real        — Disturb (x, y) p: a disturbance happens at position (x, y)
                                    with the power p.
  | DatNdSnd Node Node Dat        — DatNdSnd S D dat: node S wants to transmit some data
                                    dat to node D.
  | DatRcvd Node Node Dat         — DatRcvd D S dat: node D receives some data dat from S.
```

In addition, natural number N is used to express the number of nodes:

consts N :: *nat*

Since there is no restriction on the upper limit of *N*, node number in our formal treatment of DSR can be arbitrarily large.

DSR. DSR is formalized as a concurrent system *dsr*, whose type is (*event list* × *event*) *set*. The *dsr* consists of fourteen rules. The conclusion of each rule is of form $(\tau,\ e) \in$ *dsr*, which means event e is eligible to happen under the system state τ.

The *dsr* uses *observation function*s to obtain some information of current system state so as to decide which event is eligible to happen in next step. In order to indicate how the *observation function*s are constructed, an *observation function sendbf* is explained in detail in the following paragraph. The other *observation function*s used by *dsr* are briefly explained in Table 2.

When a node wants to send some data to a destination node, if this node has no route to the destination node, it will store the data in a local buffer before initiating a *Route Discovery*. This buffer is called *send buffer*, which is formalized as function *sendbf*. The value of *sendbf* varies with system states and nodes. So the *send buffer* of node S at current system state τ is written as *sendbf* $(\tau,\ S)$ and it is defined as follows:

```
consts sendbf :: (event list × Node) ⇒ (event × nat) list
recdef sendbf measure (λ(τ, S). length τ)     — The sendbf is defined recursively.
  sendbf ([], S) = []                          — Initially, the send buffer is empty.
  sendbf (DatNdSnd S' D dat # τ, S) =
    (if S' = S ∧ (DatNdSnd S D dat, 0) ∉ set (sendbf (τ, S))
     then if |sendbf (τ, S)| < MaxCacheSize then (DatNdSnd S D dat, 0) # sendbf (τ, S)
          else (DatNdSnd S D dat, 0) # butlast (sendbf (τ, S))
     else sendbf (τ, S))
```

— When event *DatNdSnd S′ D dat* happens and '$S' = S \wedge$ (*DatNdSnd S D dat*, *0*) $\notin$ *set* (*sendbf* $(\tau,\ S)$)', which means the happened event is *DatNdSnd S D dat* and it happens for the first time, if '|*sendbf* $(\tau,\ S)$| < *MaxCacheSize*', which means the size of the *send buffer* of node S does not reach the maximum, S puts (*DatNdSnd S D dat*, *0*) in the head of its *send buffer* ((*DatNdSnd S D dat*, *0*) # *sendbf* $(\tau,\ S)$); Otherwise, S removes the last element of its *send buffer*, then puts (*DatNdSnd S D dat*, *0*) in the head of its *send buffer* ((*DatNdSnd S D dat*, *0*) # *butlast* (*sendbf* $(\tau,\ S)$)).

Table 2. The explanation of some *observation functions*

Function	Explanation
$\mathit{recvq}\ (\tau,\ A) :: \mathit{Msg\ list}$	$\mathit{recvq}\ (\tau,\ A)$ is the reception buffer of node A at current system state τ.
$\mathit{cache}\ (\tau,\ S,\ D) ::$ $\mathit{Node\ list\ list}$	$\mathit{cache}\ (\tau,\ S,\ D)$ returns the routes to node D in the route cache of node S at current system state τ. Its content is like $[[S,A,B,D],\ [S,E,F,D]]$, which expresses that S has two routes to D, '*SABD*' and '*SEFD*'.
$\mathit{ranqid}\ (\tau,\ S,\ D) :: \mathit{Qid}$	$\mathit{ranqid}\ (\tau,\ S,\ D)$ returns a new *Qid*, which has never been used by the *Route Request* messages sent from node S to node D.
$\mathit{pdREQ}\ (\tau,\ A) :: \mathit{Msg\ set}$	$\mathit{pdREQ}\ (\tau,\ A)$ is the set of *Route Request* messages which have been received by node A, but have not been disposed (retransmitted or replied).
$\mathit{pdREP}\ (\tau,\ A) :: \mathit{Msg\ set}$	$\mathit{pdREP}\ (\tau,\ A)$ is the set of *Route Reply* messages which have been received by node A, but have not been retransmitted.
$\mathit{pdERR}\ (\tau,\ A) :: \mathit{Msg\ set}$	$\mathit{pdERR}\ (\tau,\ A)$ is the set of *Route Error* messages which have been received by node A, but have not been retransmitted.
$\mathit{pdDAT}\ (\tau,\ A) :: \mathit{Msg\ set}$	$\mathit{pdDAT}\ (\tau,\ A)$ is the set of *DATA* messages which have been received by node A, but have not been disposed.
$\mathit{pdDD}\ (\tau,\ A) :: \mathit{Msg\ set}$	*pdDD* is used specifically to initiate a *Route Error* message. $\mathit{pdDD}\ (\tau,\ A)$ is the set of *DATA* messages which have been received by node A, but have not been used to initiate a *Route Error* message.
$\mathit{nodelist} \succ \mathit{segleft} :: \mathit{Node}$	When a *Route Reply*, *Route Error* or *Data* message is transmitted, the *SegLeft* field in the message decreases step by step. The node that should receive this message in next step, called *next receiving node*, can be figured out through *SegLeft* and *Node list*. Function $\succ$ is used to calculate the *next receiving node.*
$\mathit{nodelist} \succeq \mathit{segleft} :: \mathit{Node}$	Function $\succeq$ is used to calculate the next hop of *next receiving node.*

$\mathit{sendbf}\ (\mathit{Send}\ A\ \lceil \mathit{RREQ}\ D\ \mathit{qid}\ \mathit{ndl}\rfloor_{S',0}\ \#\ \tau,\ S) = (\mathit{if}\ A = S \wedge S' = S$

$\quad \mathit{then}\ \mathit{map}\ (\lambda x.\ \mathit{if}\ (\exists\, \mathit{dat}'.\ x{=}(\mathit{DatNdSnd}\ S\ D\ \mathit{dat}',0))\ \mathit{then}\ (\mathit{fst}\ x,1)\ \mathit{else}\ x)\ (\mathit{sendbf}(\tau,S))$
$\quad \mathit{else}\ \mathit{sendbf}\ (\tau,\ S))$

— When event $\mathit{Send}\ S\ \lceil \mathit{RREQ}\ D\ \mathit{qid}\ \mathit{ndl}\rfloor_{S,0}$ happens, which means node S broadcasts a *Route Request* message to find routes to node D, S changes $(\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 0)$ into $(\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 1)$ in order to avoid sending repeated *Route Request* messages.

$\mathit{sendbf}\ (\mathit{Send}\ A\ \lceil \mathit{DATA}\ \mathit{dat}\ \mathit{segl}\ \mathit{ndl}\rfloor_{S',D}\ \#\ \tau,\ S) = (\mathit{if}\ A = S \wedge S' = S$

$\quad \mathit{then}\ [x \in \mathit{sendbf}\ (\tau,S).\ x \neq (\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 0) \vee x \neq (\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 1)]$
$\quad \mathit{else}\ \mathit{sendbf}\ (\tau,\ S))$

— When event $\mathit{Send}\ S\ \lceil \mathit{DATA}\ \mathit{dat}\ \mathit{segl}\ \mathit{ndl}\rfloor_{S,D}$ happens, S removes $(\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 0)$ and $(\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 1)$ to avoid sending repeated *Data* messages.

$\mathit{sendbf}\ (e\ \#\ \tau,\ S) = \mathit{sendbf}\ (\tau,\ S)$

— The happening of any other event does not affect the *send buffer*.

The concurrent system *dsr* is inductively defined as follows:

consts $\mathit{dsr} :: (\mathit{event\ list} \times \mathit{event})\ \mathit{set}$ **inductive** *dsr* **intros**

$\mathit{Node\text{-}can\text{-}move\text{-}randomly}$: $(\tau,\ \mathit{Move}\ A\ (x,\ y)) \in \mathit{dsr}$

— Event $\mathit{Move}\ A\ (x,\ y)$ may happen under any system state τ. It means that any node can move randomly in the network.

$\mathit{disturb}$: $(\tau,\ \mathit{Disturb}\ (x,\ y)\ p) \in \mathit{dsr}$ — Any disturbance may happen in any position.

$\mathit{data\text{-}need\text{-}send\text{-}anytime}$: $(\tau,\ \mathit{DatNdSnd}\ S\ D\ \mathit{dat}) \in \mathit{dsr}$

— Any data *dat* may need to be transmitted from one node to another node.

$\mathit{recv\text{-}msg}$: $\mathit{msg} \in \mathit{set}\ (\mathit{recvq}\ (\tau,\ A)) \Longrightarrow (\tau,\ \mathit{Recv}\ A\ \mathit{msg}) \in \mathit{dsr}$

— Any node may receive the message that is in its reception buffer.

$\mathit{originate\text{-}route\text{-}request}$: $[\![(\mathit{DatNdSnd}\ S\ D\ \mathit{dat},0) \in \mathit{set}\ (\mathit{sendbf}\ (\tau,S));\ \mathit{cache}\ (\tau,S,D) = []]\!]$
$\quad \Longrightarrow (\tau,\ \mathit{Send}\ S\ \lceil \mathit{RREQ}\ D\ (\mathit{ranqid}\ (\tau,\ S,\ D))\ [S]\rfloor_{S,0}) \in \mathit{dsr}$

— If $(\mathit{DatNdSnd}\ S\ D\ \mathit{dat},\ 0)$ is in the *send buffer* of node S, which means that S needs to transmit some data *dat* to node D and S has no route to D ($\mathit{cache}\ (\tau,\ S,\ D) = []$), S may broadcast a *Route Request* message to find routes to D.

forward-request: $[\![\lceil RREQ\ D\ qid\ ndl \rfloor_{S,0} \in pdREQ\ (\tau,\ B);\ B \neq D;$
$(S,\ qid,\ D) \notin set\ (recvdREQ\ (\tau,\ B));\ B \notin set\ ndl;\ |ndl| < N - 1]\!]$
$\Longrightarrow (\tau,\ Send\ B\ \lceil RREQ\ D\ qid\ (ndl@[B]) \rfloor_{S,0}) \in dsr$

— When node B receives a *Route Request* message ($\lceil RREQ\ D\ qid\ ndl \rfloor_{S,0} \in pdREQ\ (\tau,\ B)$), if it is not the destination node ($B \neq D$), it extracts the *address list* (ndl) from the message. If B has not received this message before ($(S,\ qid,\ D) \notin set\ (recvdREQ\ (\tau,\ B))$), and it does not exist in the *address list* ($B \notin set\ ndl$), and the length of the *address list* is less than $N - 1$ ($|ndl| < N - 1$), B may rebroadcast this *Route Request* message with the *address list* appended with its own address.

reply-route-request: $\lceil RREQ\ D\ qid\ ndl \rfloor_{S,0} \in pdREQ\ (\tau,\ D)$
$\Longrightarrow (\tau,\ Send\ D\ \lceil RREP\ qid\ (\ |ndl@[D]| - 2)\ (rev\ (ndl@[D])) \rfloor_{D,S}) \in dsr$

— When destination node D receives a *Route Request* message, D may send a *Route Reply* message to the source node S, using the reversal of the *address list* as its *source route*.

forward-rrep: $\lceil RREP\ qid\ segl\ ndl \rfloor_{D,S} \in pdREP\ (\tau,\ ndl \succ segl)$
$\Longrightarrow (\tau,\ Send\ (ndl \succ segl)\ \lceil RREP\ qid\ (segl - 1)\ ndl \rfloor_{D,S}) \in dsr$

— When an intermediate node receives a *Route Reply* message and it is the *next receiving node*, it may retransmit the message with *segments left* ($segl$) decreased by *1*.

send-data: $[\![(DatNdSnd\ S\ D\ dat,\ 0) \in set\ (sendbf\ (\tau,\ S)) \vee$
$(DatNdSnd\ S\ D\ dat,\ 1) \in set\ (sendbf\ (\tau,\ S));\ cache\ (\tau,\ S,\ D) \neq [\,]]\!]$
$\Longrightarrow (\tau,\ Send\ S\ \lceil DATA\ dat\ (\ |cache\ (\tau,\ S,\ D)!0| - 2)\ (cache\ (\tau,\ S,\ D)!0) \rfloor_{S,D}) \in dsr$

— If node S needs to transmit some data *dat* to node D and it has route(s) to D, it may send out a *Data* message using the first route as *source route*.

forward-data: $[\![\lceil DATA\ dat\ segl\ ndl \rfloor_{S,D} \in pdDAT\ (\tau,\ ndl \succ segl);\ segl \neq 0]\!] \Longrightarrow$
$(\tau,\ Send\ (ndl \succ segl)\ \lceil DATA\ dat\ (segl - 1)\ ndl \rfloor_{S,D}) \in dsr$ — Similar to *forward-rrep.*

receive-data: $\lceil DATA\ dat\ 0\ ndl \rfloor_{S,D} \in pdDAT\ (\tau,\ D) \Longrightarrow (\tau,\ DatRcvd\ D\ S\ dat) \in dsr$

— When a node D receives a *Data* message, which comes from node S and its data content is *dat*, event *DatRcvd D S dat* may happen.

forward-data-error: $[\![\lceil DATA\ dat\ segl\ ndl \rfloor_{S,D} \in pdDD\ (\tau,\ B);\ segl \neq 0;$
$B = (ndl \succ segl);\ nexthop = (ndl \succeq segl);\ \neg\ adj\ \tau\ B\ nexthop]\!]$
$\Longrightarrow (\tau,\ Send\ B\ \lceil RERR\ B\ nexthop\ (\ |ndl| - segl - 2)\ (rev\ ndl) \rfloor_{B,S}) \in dsr$

— When transmitting a *Data* message, if a node B finds the link to its next hop has broken ($\neg\ adj\ \tau\ B\ nexthop$), B may send a *Route Error* message to the source node S.

forward-rerr: $\lceil RERR\ B\ C\ segl\ ndl \rfloor_{B,S} \in pdERR\ (\tau,\ ndl \succ segl) \Longrightarrow$
$(\tau, Send\ (ndl \succ segl)\ \lceil RERR\ B\ C\ (segl - 1)\ ndl \rfloor_{B,S}) \in dsr$ — Similar to *forward-rrep.*

forward-data-error-s: $[\![\lceil DATA\ dat\ segl\ ndl \rfloor_{S,D} \in pdDD\ (\tau,\ S);\ \neg\ adj\ \tau\ S\ (ndl \succ segl)]\!]$
$\Longrightarrow (\tau,\ Send\ S\ \lceil RERR\ S\ (ndl \succ segl)\ 0\ (rev\ ndl) \rfloor_{S,S}) \in dsr$

— When transmitting a *Data* message, if the source node S finds the link to its next hop has broken, S may send a *Route Error* message to itself to remove the broken route.

4 Safety Properties of DSR

The discovered routes should have at least the following safety properties:

- Any discovered route is loop free. Namely, all the nodes in the route are distinct.
- Any discovered route contains at least two nodes.
- Any discovered route contains at most N nodes.

– The first node of any discovered route is the source node.
– The last node of any discovered route is the destination node.

The theorem *sfp* can formally express these safety properties:

$$[\![dsr \vdash \tau;\ r \in set\ (cache\ (\tau,\ S,\ D))]\!] \Longrightarrow distinct\ r \wedge 2{\le}|r| \wedge |r|{\le}N \wedge hd\ r{=}S \wedge last\ r{=}D$$

The first premise $dsr \vdash \tau$ means that τ is a valid finite execution of *dsr*. The second premise $r \in set\ (cache\ (\tau,\ S,\ D))$ means that r is a route to node D in the route cache of node S.

The conclusion of the theorem *sfp* is the conjunction of five parts, which correspond to the above five safety properties respectively. The *distinct r* means that r is loop free. The $2 \le |r|$ means the length of r is not less than 2. The $|r| \le N$ means the length of r is not more than N. The $hd\ r = S$ means that the first node of r is the source node S. The $last\ r = D$ means that the last node of r is the destination node D.

Since the proof of *sfp* is not difficult, we do not dwell on it.

5 Liveness Property of DSR

5.1 Embedding LTL

LTL (Linear Temporal Logic) is widely used for the specification and verification of concurrent systems[11]. A shallow embedding of LTL is given in Fig. 3 to make sure the property proved in this paper is indeed a *liveness* property.

The type of LTL formulae is defined as *′a tlf*. The expression $(\sigma,\ i) \models \varphi$ means that LTL formula φ is valid at moment i of the infinite execution σ. The operator $\models$ is overloaded, so that $\sigma \models \varphi$ can be defined as the abbreviation of $(\sigma,\ 0) \models \varphi$. The *always* operator $\Box$ and *eventual* operator $\Diamond$ are defined literally.

An operator $\langle\text{-}\rangle$ is defined to lift a predicate on finite executions up to a LTL formula. The temporal operator $\hookrightarrow$ is the lift of logical implication $\longrightarrow$ up to LTL

```
types ′a tlf = (nat ⇒ ′a) ⇒ nat ⇒ bool

consts valid-under :: ′a ⇒ ′b ⇒ bool (- ⊨ -  [64, 64] 50)
defs (overloaded)  pr ⊨ φ ≡ let (σ, i) = pr in φ σ i
defs (overloaded)  σ ⊨ φ ≡ ((σ::nat ⇒ ′a), (0::nat)) ⊨ φ

  □φ ≡ λ σ i. ∀ j. i ≤ j ⟶ (σ, j) ⊨ φ
  ◇φ ≡ λ σ i. ∃ j. i ≤ j ∧ (σ, j) ⊨ φ

constdefs lift-pred :: (′a list ⇒ bool) ⇒ ′a tlf   (⟨-⟩ [65] 65)
  ⟨P⟩ ≡ λ σ i. P ⟦σ⟧_i

constdefs lift-imply :: ′a tlf ⇒ ′a tlf ⇒ ′a tlf (-↪- [65, 65] 65)
  φ ↪ ψ ≡ λ σ i. φ σ i ⟶ ψ σ i

constdefs last-is :: ′a ⇒ ′a list ⇒ bool
  last-is e τ ≡ (case τ of Nil ⇒ False | (e_1 # t) ⇒ e_1 = e)
translations (|e|) ⇌ last-is e
```

Fig. 3. A shallow embedding of LTL

level. For an event e, the term $(\!| e |\!)$ is a predicate on finite executions stating that the last happened event is e. Therefore, the expression $\langle (\!| e |\!) \rangle$ is an LTL formula saying that event e happens at the current moment.

5.2 Liveness Description

Theorem *send-will-recv* formally expresses the liveness property of DSR:

$$[\![dsr \vdash \sigma;\ PF\ dsr\ \{\!| F\ S\ D\ dat,\ E\ S\ D\ dat,\ M |\!\}\ \sigma]\!]$$
$$\Longrightarrow \sigma \models \Box\Diamond\langle (\!| DatNdSnd\ S\ D\ dat |\!)\ \&\&\ expath\ S\ D \rangle \hookrightarrow \Box\Diamond\langle (\!| DatRcvd\ D\ S\ dat |\!) \rangle$$

The first premise of *send-will-recv* means that σ is a valid infinite execution of *dsr*. The second premise of *send-will-recv* is a *parametric fairness* assumption, which can ensure that the concurrent system *dsr* runs fairly. In unfair executions, some events may be enabled infinitely many times, but never happen. This *parametric fairness* assumption is necessary to prevent such occasions from happening. The detailed explanation of *parametric fairness* is described in [7].

The conclusion of *send-will-recv* is a reactivity property, which is a kind of liveness property[11]. The conclusion says that if some data *dat*, which needs to be transmitted from S to D, is delivered to the network layer of node S infinitely many times and there is at least one path between S and D each time, the data *dat* will be received by D infinitely many times.

5.3 Liveness Proof

In [7], a rule *react-rule* is presented to prove liveness properties. This paper uses another liveness proof rule *react1-rule*[2]:

$$[\![REACT1\ ?cs\ ?TR\ ?N\ ?P\ ?Q;\ ?cs \vdash \sigma;$$
$$PF\ ?cs\ \{\!| (\lambda\tau.\ |cur\text{-}tr\ ?TR\ ?N\ ?P\ z\ \tau|\ ),\ (\lambda\tau.\ last\ (cur\text{-}tr\ ?TR\ ?N\ ?P\ z\ \tau)),\ ?N |\!\}\ \sigma]\!]$$
$$\Longrightarrow \sigma \models \Box\Diamond\langle ?P \rangle \hookrightarrow \Box\Diamond\langle ?Q \rangle$$

If let $?cs = dsr,\ ?P = (\!| DatNdSnd\ S\ D\ dat |\!)\ \&\&\ expath\ S\ D,\ ?Q = (\!| DatRcvd\ D\ S\ dat |\!)$, the rule turns into the following form:

$$[\![REACT1\ dsr\ ?TR\ ?N\ ((\!| DatNdSnd\ S\ D\ dat |\!)\ \&\&\ expath\ S\ D)\ (\!| DatRcvd\ D\ S\ dat |\!);\ dsr \vdash \sigma;$$
$$PF\ dsr\ \{\!| (\lambda\tau.\ |cur\text{-}tr\ ?TR\ ?N\ ((\!| DatNdSnd\ S\ D\ dat |\!)\ \&\&\ expath\ S\ D)\ z\ \tau|\ ),$$
$$(\lambda\tau.\ last\ (cur\text{-}tr\ ?TR\ ?N\ ((\!| DatNdSnd\ S\ D\ dat |\!)\ \&\&\ expath\ S\ D)\ z\ \tau)),\ ?N |\!\}\ \sigma]\!]$$
$$\Longrightarrow \sigma \models \Box\Diamond\langle (\!| DatNdSnd\ S\ D\ dat |\!)\ \&\&\ expath\ S\ D \rangle \hookrightarrow \Box\Diamond\langle (\!| DatRcvd\ D\ S\ dat |\!) \rangle$$

In this way, it is only needed to prove the first premise, *REACT1*, in order to prove theorem *send-will-recv*, which is the formal expression of the liveness property.

Fig. 4 shows the definition of the locale *REACT1*. It can be seen that it is only needed to prove the *path* assumption of *REACT1* in order to prove *REACT1*.

$$path\text{: } [\![?cs \vdash \tau;\ ?P\ \tau]\!] \Longrightarrow |?TR\ \tau| < ?N \wedge ?Q\ ((?TR\ \tau)\ @\ \tau) \wedge ?cs \vdash (?TR\ \tau)\ @\ \tau$$

According to the above discussion, we know: $?cs = dsr,\ ?P\ \tau = (\!| DatNdSnd\ S\ D\ dat |\!)\ \tau \wedge expath\ S\ D\ \tau,\ ?Q\ (?TR\ \tau\ @\ \tau) = (\!| DatRcvd\ D\ S\ dat |\!)\ (?TR\ \tau\ @\ \tau)$. In order to prove the *path* assumption, it is only needed to find a finite event list $?TR\ \tau$, which can

[2] In fact, the two rules are similar, and they can be converted each other.

```
locale REACT1 =
  fixes cs :: ('a list × 'a) set
  and TR :: 'a list ⇒ 'a list
  and N :: nat
  and P :: 'a list ⇒ bool
  and Q :: 'a list ⇒ bool
  assumes path: ⟦cs ⊢ τ; P τ⟧ ⟹ |TR τ| < N ∧ Q ((TR τ) @ τ) ∧ cs ⊢ (TR τ) @ τ
```

Fig. 4. The definition of *REACT1*

lead to the desired Q-state (the event *DatRcvd D S dat* happens) from current P-state (the event *DatNdSnd S D dat* happens and there is at least one path between S and D currently).

We have successfully found such a finite event list *?TR* τ for an arbitrarily large network. The following paragraphs take a four-node MANET as an example to show how to find such a finite event list. Fig. 5 illustrates the four-node MANET, in which node S neighbors with node A, node A neighbors with node B, and node B neighbors with node D.

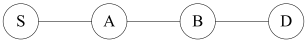


Fig. 5. An illustration for a four-node MANET

There are three cases when the event *DatNdSnd S D dat* happens (node S wants to transmit some data *dat* to node D). Fig. 6 shows the state transition diagram.

Firstly, if node S has at least one route to node D in its route cache, and the first one is unbroken,[3] S will send out the *Data* message using this route. Let *?TR* τ be the process of transmitting this *Data* message:

?TR τ_1 = [*DatRcvd D S dat*, *Recv D m(0)*, *Send B m(0)*, *Recv B m(1)*, *Send A m(1)*, *Recv A m(2)*, *Send S m(2)*]. [4]

Secondly, if node S has no route to node D in its route cache, S will begin to discover routes to D, and will be able to find at least one unbroken route because there is at least one path between S and D (*expath S D* τ). Then S begins to transmit the *Data* message using the newly discovered route. The process of transmitting *Data* message is the same as the first case. Let *?TR* τ be the process of discovering route and transmitting the *Data* message:

?TR τ_2 = *?TR* τ_1 @ [*DatNdSnd S D dat*, *Recv S rep(0)*, *Send A rep(0)*, *Recv A rep(1)*, *Send B rep(1)*, *Recv B rep(2)*, *Send D rep(2)*] @ [*Recv D req(S,A,B)*, *Send B req(S,A,B)*, *Recv B req(S,A)*, *Send A req(S,A)*, *Recv A req(S)*, *Send S req(S)*]. [5]

[3] It is assumed that nodes choose the first route to send *Data* messages.

[4] *m(?)* is the abbreviation for $\lceil DATA\ dat\ ?\ [S,A,B,D]\rfloor_{S,D}$.

[5] *rep(?)* is the abbreviation for $\lceil RREP\ qid\ ?\ [D,B,A,S]\rfloor_{D,S}$;
req(?) is the abbreviation for $\lceil RREQ\ D\ qid\ [?]\rfloor_{S,0}$.

Thirdly, if node S has at least one route to node D in its route cache, but the first one is broken, S will transmit the *Data* message using this broken route. The transmission will fail and a *Route Error* message will be sent back to S. And then S will remove the broken route from its route cache.

After removing this broken route, the situation will be one of the above three cases again. If all routes in the route cache of node S are broken, S will remove all of them because the size of the route cache is finite. In the end, S will discover a good route to transmit the *Data* message successfully.

According to the above discussion, such a finite event list *?TR* τ can be found and its length is finite because both the size of the route cache and the number of nodes are finite. So the *path* assumption of *REACT1* can be proved.

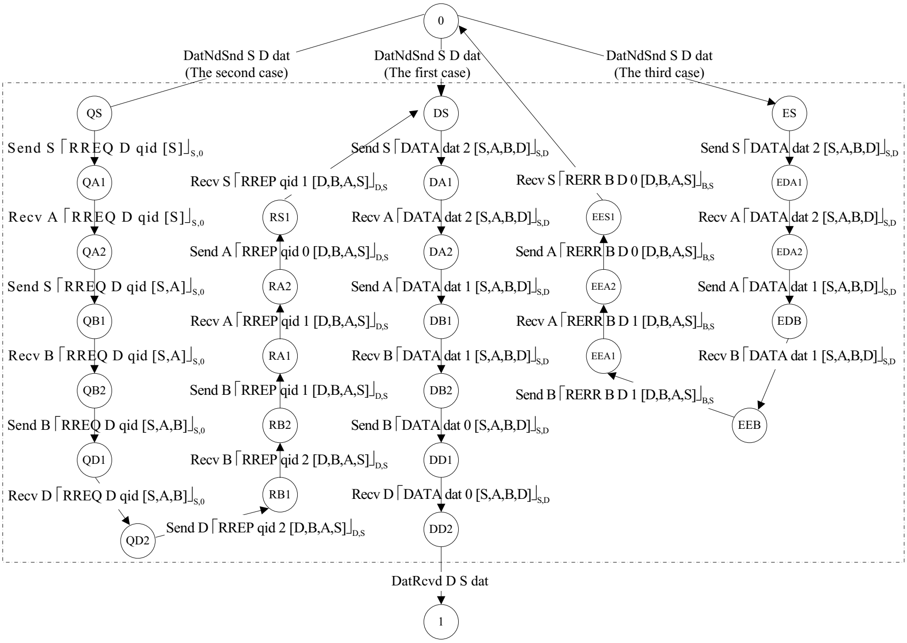

0
DatNdSnd S D dat (The second case)
DatNdSnd S D dat (The first case)
DatNdSnd S D dat (The third case)
QS
DS
ES
Send S ⌈RREQ D qid [S]⌋$_{S,0}$
QA1
Recv A ⌈RREQ D qid [S]⌋$_{S,0}$
QA2
Send S ⌈RREQ D qid [S,A]⌋$_{S,0}$
QB1
Recv B ⌈RREQ D qid [S,A]⌋$_{S,0}$
QB2
Send B ⌈RREQ D qid [S,A,B]⌋$_{S,0}$
QD1
Recv D ⌈RREQ D qid [S,A,B]⌋$_{S,0}$
QD2
Send D ⌈RREP qid 2 [D,B,A,S]⌋$_{D,S}$
RB1
Recv B ⌈RREP qid 2 [D,B,A,S]⌋$_{D,S}$
RB2
Send B ⌈RREP qid 1 [D,B,A,S]⌋$_{D,S}$
RA1
Recv A ⌈RREP qid 1 [D,B,A,S]⌋$_{D,S}$
RA2
Send A ⌈RREP qid 0 [D,B,A,S]⌋$_{D,S}$
RS1
Recv S ⌈RREP qid 1 [D,B,A,S]⌋$_{D,S}$
Send S ⌈DATA dat 2 [S,A,B,D]⌋$_{S,D}$
DA1
Recv A ⌈DATA dat 2 [S,A,B,D]⌋$_{S,D}$
DA2
Send A ⌈DATA dat 1 [S,A,B,D]⌋$_{S,D}$
DB1
Recv B ⌈DATA dat 1 [S,A,B,D]⌋$_{S,D}$
DB2
Send B ⌈DATA dat 0 [S,A,B,D]⌋$_{S,D}$
DD1
Recv D ⌈DATA dat 0 [S,A,B,D]⌋$_{S,D}$
DD2
DatRcvd D S dat
1
Send S ⌈DATA dat 2 [S,A,B,D]⌋$_{S,D}$
EDA1
Recv A ⌈DATA dat 2 [S,A,B,D]⌋$_{S,D}$
EDA2
Send A ⌈DATA dat 1 [S,A,B,D]⌋$_{S,D}$
EDB
Recv B ⌈DATA dat 1 [S,A,B,D]⌋$_{S,D}$
EEB
Send B ⌈RERR B D 1 [D,B,A,S]⌋$_{B,S}$
EEA1
Recv A ⌈RERR B D 1 [D,B,A,S]⌋$_{B,S}$
EEA2
Send A ⌈RERR B D 0 [D,B,A,S]⌋$_{B,S}$
EES1
Recv S ⌈RERR B D 0 [D,B,A,S]⌋$_{B,S}$

Fig. 6. The state transition diagram

6 Conclusion

It is shown in this paper that both safety and liveness properties of the DSR protocol can be formally verified and the results can be applied to arbitrarily large networks. The proofs have been mechanically checked using Isabelle/HOL/Isar to increase confidence. In the future, the same technology will be used to treat secure MANET protocols.

References

1. D.B.Johnson, D.A.Maltz, Y.Hu: The dynamic source routing protocol for mobile ad hoc networks(dsr). Internet Draft: draft-ietf-manet-dsr-10.txt (2004)
2. K.Bhargavan, D.Obradovic, C.A.Gunter: Formal verification of standards for distance vector routing protocols. Journal of the ACM **49**(4) (2002) 538–576
3. A.R.Cavalli, C.Grepet, S.Maag, V.Tortajada: A validation model for the dsr protocol. In: ICDCS Workshops. (2004) 768–773
4. R.Renesse, A.H.Aghvami: Formal verification of ad-hoc routing protocols using spin model checker. In: IEEE MELECON, Dubrovnik, Croatia (2004)
5. T.Lauschner, A.Macedo, S.Campos: Formal verification and analysis of a routing protocol for ad-hoc networks. (2000)
6. S.Yang, J.S.Baras: Modeling vulnerabilities of ad hoc routing protocols. In: SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA, ACM Press (2003) 12–20
7. X.Zhang, H.Yang, Y.Wang: Liveness reasoning for inductive protocol verification. In: The 'Emerging Trend' of TPHOLs 2005. Oxford University Computing Lab. PRG-RR-05-02 (2005) 221–235
8. T.Nipkow, L.C.Paulson, M.Wenzel: Isabelle/HOL — A Proof Assistant for Higher-Order Logic. Volume 2283 of LNCS. Springer (2002)
9. M.Wenzel: Isar - a generic interpretative approach to readable formal proof documents. In: TPHOLs 1999. Volume 2283 of LNCS., Springer (1999)
10. L.C.Paulson: The inductive approach to verifying cryptographic protocols. J. Computer Security **6** (1998) 85–128
11. Z.Manna, A.Pnueli: Completing the temporal picture. Theor. Comput. Sci. (1991) 91–130

Scalable Proxy Routing in Multi-hop Cellular Networks

Nan Yang[1] and Guangbin Fan[2]

[1] Department of Computer and Information Science
The University of Mississippi, USA
[2] Intel Corporation, China
nan@olemiss.edu, guangbin.fan@intel.com

Abstract. Ad-hoc and cellular networks have received great attention in recent years. To accommodate the large number of users and traffic over a large geographic area, cellular networks could take advantage of the infrastructure-less ad-hoc networks to provide extended service. One of the key issues in the integration of cellular and ad-hoc networks is to find some mobile nodes as proxies to relay the messages from base stations to destination nodes. In this paper, a scalable proxy relay routing protocol (SRP) is proposed to increase the total throughput of the system. In the strategy, the base station always sends data to the destination node through the selected proxy nodes which has minimum transmission delay. We demonstrate the advantage of our scheme over other current schemes such as UCAN and DST through simulation. The results have shown that the scheme outperforms other related schemes in terms of throughput and end-to-end delay.

Keywords: Ad-hoc network, cellular network, proxy, relay.

1 Introduction

In recent years, the integration of cellular network and ad hoc networks has attracted much attention as the downlink bottleneck in a cellular network becomes severe with the explosive increase of the number of mobile users [6][10]. Terminals with dual access interfaces could act as relay nodes to route the data. Those nodes could have a wider bandwidth when operating in ad hoc mode. For example, IEEE 802.11b offers the bandwidth up to 11Mbps while 802.11a offers bandwidth up to 54Mbps. It would be beneficial for such integration. An integrated cellular and ad hoc architecture, multi-hop cellular network, is illustrated in Figure 1.

One important issue in the integrated networks is to improve the throughput of the whole system by relaying the messages through proxy nodes, which have a much higher downlink channel rate[1][12]. Some related protocols have been proposed, such as the Unified Cellular and Ad Hoc Network Architecture (UCAN) [7] and Distributed Spanning Tree Protocol (DST) [3]. In UCAN, relay architecture is used to handle the routing. It also proposed two proxy nodes discovery schemes: greedy proxy discovery and on-demand proxy discovery. The DST utilizes a transmission tree in the order of downlink channel rates to assist the data forwarding. In UCAN and DST, the downlink channel rate of each mobile node assigned as the weight of a path is the criteria of choosing the appropriate transmission route.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 84 – 94, 2006.

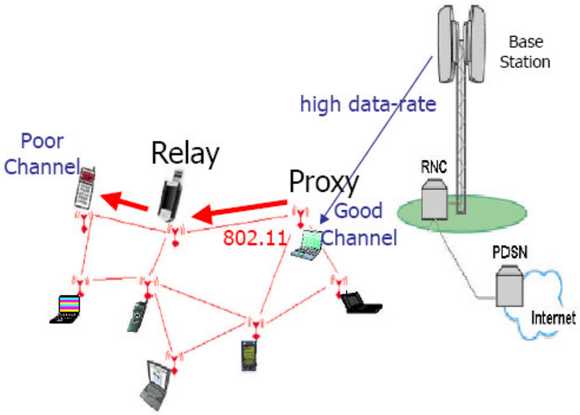


Fig. 1. Integrated Cellular and Ad-Hoc Networks

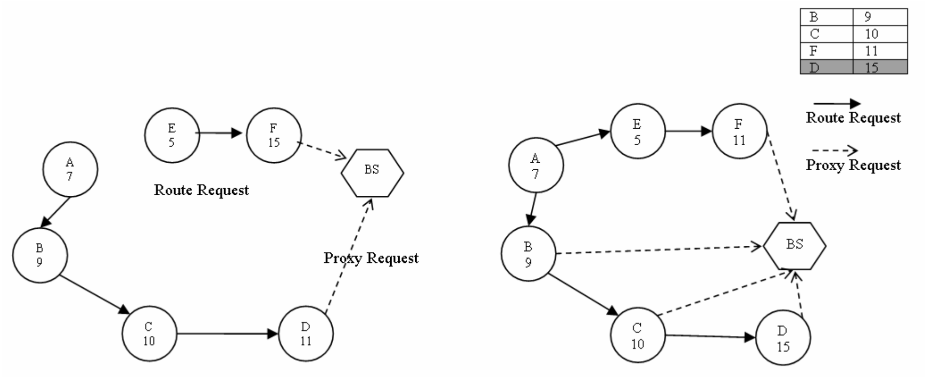
Neither of the above protocols considered the transmission delay from a mobile node to the base station. As we know, the more hops there are, the larger the delay and the more packet loss. So the system Quality of Service (QoS) cannot be guaranteed. Although both UCAN and DST increase the whole system throughput, the relay path might not be efficient. There is also flooding problems in UCAN's on-demand proxy discovery. In this paper, an efficient scalable proxy routing protocol (SRP) is proposed to improve the performance. The protocol not only increases the whole system throughput, but also reduces the delay by selecting the shortest path from a base station to the selected cluster leader which further relays the message to the destination nodes. The scheme outperforms other schemes, UCAN and DST, in both throughput and end-to-end delays.

The rest of the paper is organized as follows. Section 2 introduces the background and related works. Section 3 elaborates the proposed scalable proxy routing protocol. Section 4 gives the performance evaluation though simulations. We finally summarize the paper in Section 5.

2 Related Work

Cellular network is a communication system that provides a wireless connection to the public switched telephone network and Internet. It can accommodate a large number of users over a large geographic area within a limited frequency spectrum[9]. Cellular technologies vary from analog networks to digital networks. Currently, most a cellular network employs digital technology to improve the system quality and services. Typical present cellular networks include GSM, GPRS, CDMA, WCDMA, etc. Wireless Ad-hoc network is a temporary wireless mobile network which is organized by a collection of wireless mobile devices without the aid of any established infrastructure [5][8]. Wireless Ad hoc networks are normally used for specific strategic purposes, such as military deployment, emergency task force operation, and business conferencing activities where it is hard to have a fixed infrastructure. In those cases, a collection of mobile hosts with wireless network interfaces may form a temporary network.

The downlink bottleneck in cellular network becomes severe as the number of mobile users is exploding. All mobile users request the transmission from the base station no matter how their channel conditions are. Some users are far from the base station and have poor downlink channel condition. Traffic to those mobile nodes having weak downlink channel conditions would further increase the burden of the cellular system due to frequent retransmissions. In [7], Luo and Ramjee et al. proposed a Unified Cellular and Ad-Hoc Network Architecture (UCAN). It defines two kinds of proxy discovery. One is called greedy proxy discovery. The other is called on-demand proxy discovery.



B	9
C	10
F	11
D	15

(a) (b)

Fig. 2. Greedy and on-demand proxy discovery

In greedy proxy discovery of UCAN, a mobile node always sends the route request message to the neighbor node having the best downlink channel rate. Suppose the numbers represent the downlink channel rates, the route request for Node A and E are shown in Figure 2(a). The problem of greedy discovery is that greedy proxy discovery cannot always find the mobile node with the best channel condition as the proxy. For example, in Figure 2(a), Node A sends the Route Request to B since B has better channel condition. The procedure continues till D declares itself as the proxy. But in this scheme, A is unaware of F which has much better channel condition than D. In on-demand proxy discovery, a mobile node floods the route request to look for a node with higher downlink channel rate to be its downlink relay proxy. The process is shown in Figure 2(b). Node A initiates a route request message. This message is forwarded to all neighbors of A, i.e. B and E. When B receives this message, it compares its own downlink channel rate with the one included in the received message. B finds that it has a better downlink channel rate than A. B sends the message to the base station to request to be a proxy. C, F and D do the same thing when they find their downlink channel rates are higher than the node that forwards the message to them. The base station will pick up a node which has the best downlink channel rate from all nodes that sends the proxy request messages.

There are two problems in this protocol. First, a large number of flooding messages in this scheme will cause severe congestions. Second, because the base station always chooses the node with the highest downlink rate as the proxy, the number of hops from the proxy to the destination node is ignored. In Figure 2(b), the base station selects D as the proxy of A. The number of hops from the D to A is 3. Although the downlink rate of B is slightly weaker than D, it is only one hop away from A. We would prefer B rather than D. This algorithm could not handle such situation.

Another related protocol is the Distributed Spanning Tree protocol (DST) which was proposed in [3]. This protocol generates a transmission tree in the order of downlink channel rates. Each node has a parent node having a higher downlink channel rate. The base station is the root of this tree. Messages from the base station are always relayed by the parent node till it reaches the destination node. Figure 3 shows the topology tree.

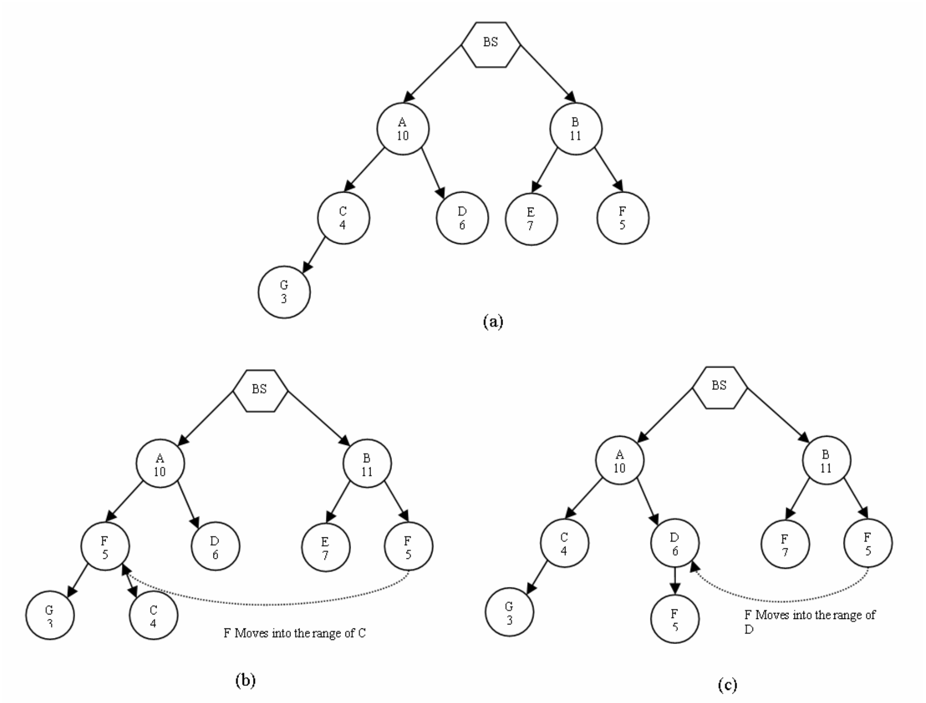


Fig. 3. DST protocol

The problem of DST is whenever there is a node moving out of the current parent and moving into a new region, the topology of the tree has to be updated as illustrated in Figure 3(b) and (c). Figure 3(b) shows the changes in the topology tree when F moves away from the range of B and moves into the range of C. Figure 3(c) shows the changes of topology tree when F moves into the range of D. The DST is not efficient when the mobile speed is high because it has to spend much time on updating the topology tree. Another problem is if the difference of downlink channel rates among all nodes is very big, e.g. from 1kbps to 500kbps, the depth of the topology tree would be very big. In other words, the message would be relayed many times. The fading of signal would be substantial. The third problem is this scheme can ensure that the message is always relayed through the node with high downlink channel rate, but it cannot guarantee the transmission path from the base station to the destination is the one with minimum transmission delay among all other possible paths.

3 Scalable Proxy Routing

3.1 Motivation

The problem in UCAN is flooding messages and inefficient relay proxy. The problem in DST is time-consuming and unnecessary relay. In order to solve those problems, we present a new scalable proxy routing protocol. The proposed routing protocol has two steps. The first step is to cluster the nodes and select cluster leaders which have the highest downlink rates among all of its m-hop-away neighbors. And the second step is to discover a shortest path from the base station to each cluster leader.

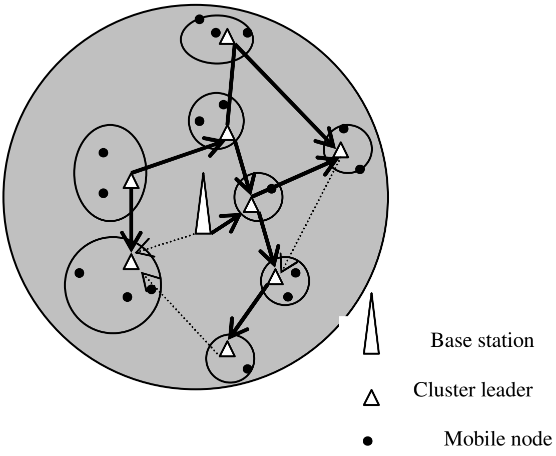
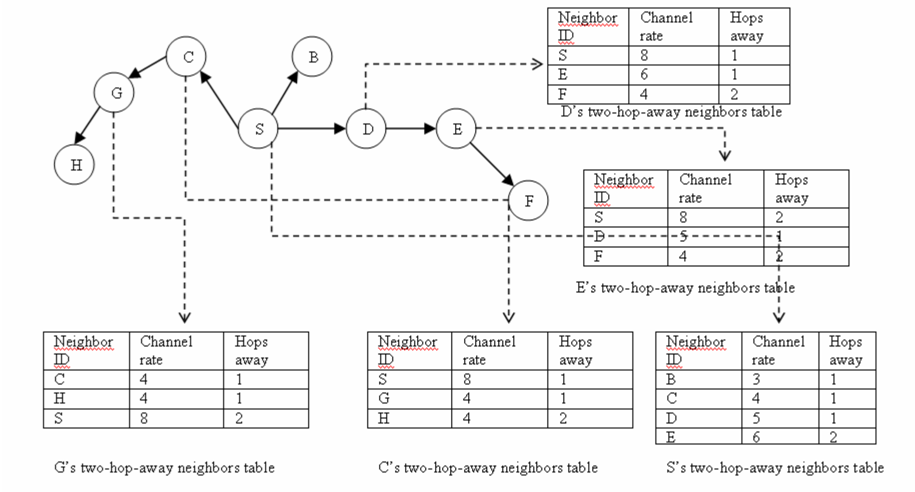


Fig. 4. Shortest path from base station to cluster leaders

The cellular network integrated with an ad-hoc network can be viewed as a directed graph G = (V, E). Here V is the set of cluster leaders and the base stations. E is the set of edges between the cluster leaders. Each edge is weighted by the transmission delay from the base station to the cluster leader or from one cluster leader to the other cluster leader.

It is obvious that w(u, v)>=0 for each edge(u, v) ∈E. By executing the variation of Dijsktra's algorithm [11], a shortest path from the base station to each cluster leader can be found. An example is shown in Figure 4. When any one mobile node wants to download data from the base station, it sends a request to its cluster leader. And then cluster leader relays this message to the base station through the shortest path.

Before the waiting timeout, the base station will send back an acknowledgement message to the cluster leader. Meanwhile the cluster leader sends the acknowledgement message to the destination node which initialized the route request



Neighbor ID	Channel rate	Hops away
S	8	1
E	6	1
F	4	2

D's two-hop-away neighbors table

Neighbor ID	Channel rate	Hops away
S	8	2
D	5	1
F	4	2

E's two-hop-away neighbors table

Neighbor ID	Channel rate	Hops away
C	4	1
H	4	1
S	8	2

G's two-hop-away neighbors table

Neighbor ID	Channel rate	Hops away
S	8	1
G	4	1
H	4	2

C's two-hop-away neighbors table

Neighbor ID	Channel rate	Hops away
B	3	1
C	4	1
D	5	1
E	6	2

S's two-hop-away neighbors table

Fig. 5. Two-hop away cluster

message. It is obvious that the system throughput will be improved by the proposed routing algorithm because of the assistance of ad-hoc networking. Also, the base station always takes the shortest path to send data to the cluster leaders.

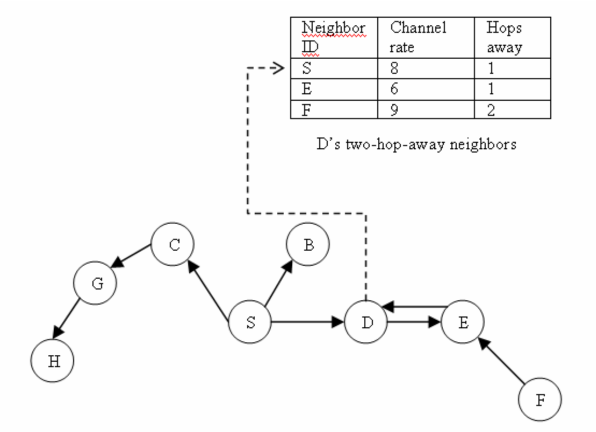
3.2 Protocol Description

The first step in the protocol is to select local leaders in small clusters. The clusters will also be formed when the leaders are elected. Those local leaders having the best channel merit among all other nodes in a cluster. This channel merit could be anything meaningful for the network. For example, it could be the downlink channel rate or the channel signal quality. This implementation uses channel downlink rate as the channel merit to choose local leaders from all of nodes in a local range. The range of a cluster is determined by the maximum hops through which the local-leader declaring message could be relayed. Once the local leaders have been set, the second step is to find a shortest path from the base station to all local leaders operating by applying Dijsktra's algorithm. Whenever a mobile station wants a route from the base station, it will send a request to its local leader. The local leader replies this request by an acknowledgement message if there is a shortest path to the base station. Meanwhile, the local leader sends a communication request to the base station through that shortest path. Before the waiting time out, the base station sends an acknowledgement back to the cluster leader. The cluster leader will notify the destination mobile station to be ready for the coming transmission.

Local leader selection in m-hop-away range

Each mobile node exchanges the message of its own downlink channel rate with all of its neighbors. If a mobile node finds that it has the best downlink channel rate either among its neighbors or among the neighbor nodes of its neighbors, it will declare itself as a cluster leader by broadcasting a cluster-leader declaration message. Any node that receives this message checks its own m-hop-away neighbor table. If the node sending the cluster-leader declaration message is the node having the best channel rate, it will reply to this message to be under the leadership of the node broadcasting the cluster-leader declare message.

Figure 5 shows the process when node S is selected to be the cluster leader of node B, C, E, G and D. S finds itself having the best downlink channel rate, 8, among its two-hop-away neighbors B, C, D and E when S checks its neighbor table. S broadcasts the cluster-leader-declare message to all of its neighbors. Nodes B, C, E, G and D reply to this message to become members of S's cluster because S is the node with the best downlink channel rate in their two-hop-away neighbor tables.



Neighbor ID	Channel rate	Hops away
S	8	1
E	6	1
F	9	2

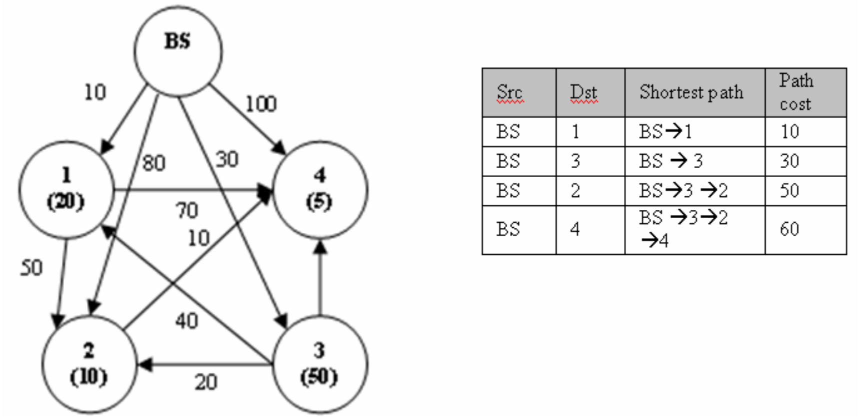
D's two-hop-away neighbors

Fig. 6. Two-hop away cluster

There is a special situation if we change the downlink channel rate of F from 4 to 9 as illustrated in Figure 6. When D checks its two-hop-away neighbor table, it finds F has a higher downlink channel rate than S. In this situation, D checks if the downlink-channel rate difference between F and S is over a threshold t. If the difference is smaller than t, D will select S as its cluster leader because S has much better downlink channel rate than F. Suppose we set the threshold to be 5. Since the downlink channel rate difference of F and S is 1 that is smaller than 5, so D replies the cluster-leader declaration message from S to accept S as the cluster leader.

Shortest path discovery

The hybrid a cellular network and ad-hoc network can be viewed as a directed graph G = (V, E) as mentioned before. Again, V is the set of local leaders and base stations. E is the set of edges between the cluster heads. The direction is always from the base to cluster leaders and from one cluster leader with higher downlink channel rate to the other one with lower downlink channel rate. Based on Dijstra's algorithm, we can always find a shortest path, i.e. the smallest delay from the base station to any one of cluster leaders. Figure 7 shows the process of shortest path discovery. The source node is the base station. The destination nodes are cluster leader 1, 2, 3 and 4. Each time, the discovery routine starts from BS station to find a route to each destination with the smallest delay.



Src	Dst	Shortest path	Path cost
BS	1	BS→1	10
BS	3	BS → 3	30
BS	2	BS→3 →2	50
BS	4	BS →3→2 →4	60

Fig. 7. Shortest route discovery

Cluster leader maintenance

Changes in the shortest path can be caused by one of the following situations: a) a cluster leader node moves out of current cluster range; b) the downlink channel rate of some nodes change substantially due to movements; c) a node moves out of current cluster range and become the node having the greatest downlink channel rate among its m-hop-away neighbors.

If a cluster leader node moves out of current cluster range, its one-hop neighbor nodes will notice its missing. Those nodes broadcast this message to all of their m-hop-away neighbors to resume cluster leader selection. The new cluster leader nodes send messages to the base station to report the changes. The substantial change of downlink channel rate will be noticed because each node broadcast the downlink channel rate message periodically. When a node moves into a cluster, it decides to join the cluster or declares itself as a new cluster leader through exchanging messages with its one-hop-away neighbors. If the new node has a larger downlink channel rate than the current cluster leader, it declares itself as the new cluster leader and broadcasts this messages.

Shortest path maintenance

The shortest path is updated only when there is a change of any one of cluster leaders. Current cluster leaders send a message to the base station periodically. When the base station finds that new nodes have appeared or that an old one has disappeared, the base station executes the shortest path discovery to generate new paths.

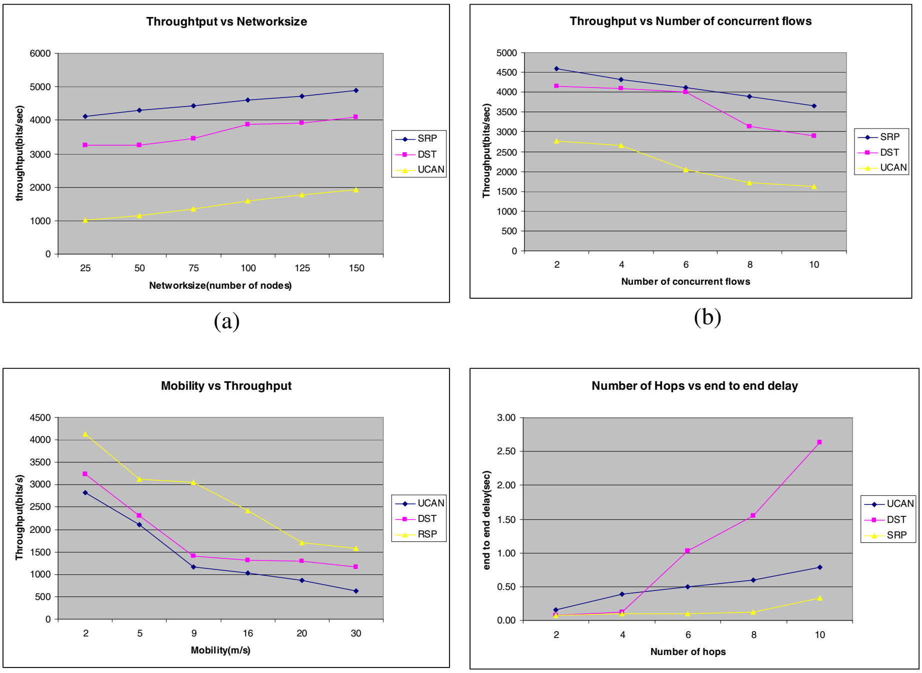
4 Performance Evaluation

In this section we present an experimental analysis to compare the proposed algorithm with UCAN's on-demand proxy discovery algorithm and DST algorithm. We use the QualNet, a network simulation software developed by Scalable Network Technologies, as the simulation tool. For all experiments in this project, we use the Constant Bit Rate (CBR) as application traffic. The propagation model is two-ray propagation path loss. IEEE 802.11 is the MAC layer protocol. Initially, all nodes are placed randomly in a square of 1500 square meters. Each node moves in the random waypoint model [4]. In our model, each mobile node inside this area is coved by the cellular transmission range of the base station with various downlink channel rates. The parameters used in the experiment are as shown in Table 1. In the first three experiments, we compare the total throughput of three protocols (SRP, DST and UCAN). In the fourth experiment, we compare the end to end delay of the three protocols based on various numbers of hops from the cluster leader to the destination node.

Table 1. Parameters in the simulation

Experiment Number	Network Size	Concurrent Flows	Mobility Speed	Number # hops
1	various	1	1m/s to 10m/s	2
2	50 nodes	various	1m/s to 10m/s	2
3	50 nodes	5	various	2
4	50 nodes	5	1m/s to 10m/s	various

We first control the value of network size from 25 nodes to 150 nodes placed in a square of 1500 by 1500 meters and study the network throughput. Only one CBR flow is used in this experiment. Figure 8(a) shows the result. In the figure, we can see that our protocol achieves the highest throughput with different network sizes. The average throughput is 70% higher than that of UCAN and 30% more than that of DST.



(a)

(b)

(c)

(d)

Fig. 8. Network throughputs and delay performance

In the second experiment, we increase the total number of concurrent CBR traffic from 2 to 10. The network size is fixed at 50 nodes. Figure 8(b) shows that our protocol always gets the highest throughput when the number of concurrent flows increases from 2 to 10. Our protocol improves the system throughput by 72% of UCAN and 37% of DST.

In the third experiment, we increase the maximum mobile speed of each node from 2m/s, 5m/s, 9m/s, 16m/s, 20m/s, to 30m/s. A total of 50 nodes are placed randomly in each experiment. Figure 8(c) shows the result. It is shown the SRP achieves the highest throughput when the mobile stations moves at different speeds. In other words, the SRP has high tolerance of mobility in the networks.

In the last experiment, we control the maximum number of hops a cluster leader can reach. We increase the maximum hops from the destination node to its cluster leader from 2 to 10 in increments of 2. As in Figure 8(d), the result shows that the SRP protocol has the minimum end-to-end delay as the number of hops increases. DST has small delay when the number of hops is small. However, when the number of hops increases, it has to spend much more time on maintaining its topology tree. The delay of DST becomes very noticeable.

5 Conclusion

This paper presented a scalable proxy routing protocol for hybrid cellular and ad hoc networks. The protocol utilizes mobile stations as proxy to relay messages from the base station to destination nodes to avoid the bottleneck of downlink channels. The protocol shows advantages over some previous protocols such as UCAN and DST. It improves the flooding problem in UCAN's on-demand proxy discovery and the non-shortest path problem in DST. The simulation results also show that the proposed scheme can achieve higher system throughput and lower end to end delay than previous schemes.

References

1. G.N. Aggelou and R. Tafazolli. On the relaying capacity of next-generation gsm cellulat networks. February 2001.
2. H. Y. Hsieh and R. Sivakumar. On using the ad-hoc network model in wireless packet data networks. In Proceedings of ACM MOBIHOC, 2002.
3. I. Ioannidis and B. Carbunar, Scalable routing in hybrid cellular and ad-hoc networks. Proceedings of IEEE INFOCOM, 2004.
4. D.B. Johnson and D.A. Malta. Dynamic source routing in ad hoc wireless networks. In Mobile Computing, volume 353, pages 153-181. Kluwer Academic Publishers, 1996
5. S. Lee, S. Banerjee, and B. Bhattacharjee. The case for a multihop wireless local area network. In Proceedings of IEEE INFO-COM, 2004.
6. Y. Lin and Y.-C. Hsu. Multihop A cellular: A New Architecture for Wireless Communications. Proceedings of IEEE INFOCOM, volume 3, pages 1273-2000.
7. H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu, UCAN: a unified a cellular and ad-hoc network architecture. Proceeding of the 9th annual international conference on Mobile computing and networking, pages 353-367, 2003.

8. S. Radhakrishnan, G. Racherla, C. Sekharan, N. Rao, S. Batsell, Protocol for dynamic ad-hoc networks using distributed spanning trees, Wireless Networks, 9:673-686, 2003.
9. T. S. Rappaport. Wireless Communications: Principles and Practice. Prentice Hall, 1996.
10. N. B. Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson. A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop cellular Networks. In Proceedings of ACM MOBIHOC, pages 1324, 2003.
11. T. Wang, W. Li, A fast low-cost shortest path tree algorithm. Journal of Software. 2004 Vol.15, No.5.
12. X. Wu, S.-H. Chan, and B. Mukherjee. MADF: A Novel Approach to Add an Ad-hoc Overlay on a Fixed A cellular Infrastructure. In Proceedings of IEEE WCNC, volume 2, pages 549, 2000.

An On-Demand Routing Protocol in Ad Hoc Network Using Label Switching

Shaohe Lv, Xingming Zhou, Xiaodong Wang, and Chi Liu

National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, Hunan, China 410073
chi.shaohe@gmail.com, xmzhou@nudt.edu.cn, xdwang@nudt.edu.cn, liuchi7669@gmail.com

Abstract. This paper presents ORAL, a new on demand routing protocol in mobile ad hoc network, which incorporates the on demand source routing, local repair and MPLS like label switching to achieve higher throughput at the expense of reasonable overhead. ORAL separates the routing and forwarding explicitly: in the former process a label switch path and corresponding traversed node sequence, e.g., the source route, are constructed and maintained similar to DSR except the label signaling embedded in control packets. In the latter process, packet forwarding is based on label matching and switching rather than destination address or prefix searching and matching which is well-known computationally intensive. Integrating label and refined source route, ORAL can control the route table size equal to the amount of actively communicating nodes if no other optimization is deployed. Numerous simulation experiments show that ORAL achieves significant higher throughput in comparison with other protocols like DSR and AODV but with much less protocol overhead than AODV though a bit more than DSR.

1 Introduction

Mobile ad hoc network (MANET) is composed of variable wireless devices that can move freely and interconnect with each other via multi-hop connections without any pre-configuration and infrastructure support. This style network is suitable for various scenarios such as search-and-rescue and disaster-and-relief and so on.

Routing in such network is a fundamental issue that has received considerable concern in recent years. However, the characteristics of MANET make the design of efficient routing protocol a challenge task. First, due to node movement, the whole network including both network topology and traffic condition is quite dynamic. Second, in MANET, every node may act as a router. But this relaying operation should take as less resource as possible because of the essential limit of power and computing capacity at each node. In addition, there may be no optimization for forwarding at nodes as the routers in Internet do. Third, bandwidth is still a scarce resource, so to reduce overhead is important for all routing protocol in MANET.

In this paper we present a novel routing protocol called ORAL which integrating on demand source route, local repair and MPLS-alike label switch to meet the above requirements together.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 95 – 106, 2006.

On demand source route, originated by DSR [1], has been shown to be suitable for MANET. Using this type routing mechanism, node should only discover and maintain the routes to the communicated peers and only respond to the change related with these routes. No local repair is deployed in DSR make it simple, but also leads to some performance loss. Combining local repair and source route, ORAL gains obvious greater packet delivery ratio and further promotes throughput.

Label switch, borrowed from MPLS, multi-protocol label switch [2], is a promising technology used in Internet. One original design aim of MPLS is to pro-vide faster forwarding based on label matching and switching than address or prefix matching which is well known computationally intensive. This advantage is trivial in Internet due to the optimal hardware implementation in router to improve the address matching operations. For MANET, however, this merit is more significant since the router here is just a common node without any special optimization.

Integrating label and refined source route, moreover, ORAL can control the total amount of route table equal to the number of actively communicating nodes, not exceed the network size, if no other optimization like multi-path is used. Like MPLS, the use or distributing of label is completely local and independent among nodes. The use of label instead of node address in packet forwarding, make the whole network design quite flexible and scalable.

The rest of the paper is organized as follows: section 2 provides a brief survey of the routing protocols and label switch mechanism used in MANET. Section 3 presents ORAL protocol including the operations in routing and forwarding process. Section 4 shows that ORAL is loop-free. The experimental setup and results are shown in section 5 and 6 respectively. Finally, the conclusions and future work are given in section 7.

2 Related Work

There have mainly three categories of routing protocols in MANET so far: on demand, proactive and the hybrid ones. Using reactive protocol such as AODV [3] or DSR, a node discovers (or re-constructs if necessary) the route to another node only when it wants to communicate with that node.

In opposite, proactive protocol such as DSDV [4], OLSR [5] and TBRPF [6], requires node to maintain routes to all reachable nodes at any time, and the periodic exchange of route information is used to keep route table up-to-date. Using this type protocol, obviously, node can easily determine whether a node is reachable and the corresponding route if it is reachable. However, this type protocol also incurs a constant and significant overhead though many routes may not be used at all.

The last category is the hybrid ones such as ZRP [7], SHARP [8], which divide the network into different zone and maintain or search a route to another node according to the location of the target node, e.g., whether in the same zone or not. In addition, hierarchical protocols such as ZHLS [9] are also proposed, though to maintain an efficient hierarchical structure in dynamic network is still challenge.

Considerable efforts aim to improve the above basic protocols, such as route cache used in DSR [1]. In OLSRv2 [10], the designers also propose to exchange only partial route information and other protocol, i.e. FSR [11], requires the node to maintain the proactive route table in a reactive update fashion, to reduce the control overhead. In addition, DART [12] explores the use of dynamic address allocation and address aggregation based routing to reduce the overhead of proactive protocols.

Though there have numerous routing protocols proposed for MANET, no system work has been done to embrace the label switch mechanism. A label switched packet forwarding architecture for ad hoc network is proposed in [13], with major focus on enhancing MAC protocol assuming that a standard label signaling protocol is used. In [14], a wireless ad hoc label switching protocol WALS is proposed, which focuses mainly on multi-path transmission. The work most closed to us is in [15], where Lilith is proposed to address the inter-operation issue over heterogeneous networks via wireless ad hoc mode with MPLS. However, these works do not detail the incorporation between MPLS with routing protocol in MANET and the intuitive integration is insufficient. ORAL, to our best knowledge, is the first to describe in detail the incorporation among on demand source route, local repair and label switch to provide loop free route and better performance with a bit more protocol overhead.

3 Description of ORAL

Operations in ORAL comprise routing and forwarding process where the latter is to transmit data by means of label switching and the former is to discover and maintain route via three basic control packet, e.g. route request, reply and update packets, termed as RREQ, RREP and RRUP respectively. Except the broadcast of route request, both reply and update packets are send via unicast fashion.

ORAL shares the most common assumptions with other protocols as follows: each node has a unique identifier, referred to as node address in general, within the entire network; all channels are symmetric and lastly, all route update can be transmitted reliably except that the used link is unavailable (i.e. broken).

The channel assumption is required because of the bidirectional communication in the construction of LSP. Though strict, this assumption is still reasonable and acceptable due to the widespread use of 802.11 based wireless devices, which always need two-way handoff in data transfer. Moreover, all updates are only sent to neig-hbor node, when all links are symmetric, the last assumption implies that no data can be transmitted successfully over the route where update fails to be transferred.

3.1 Fundaments

In route table, each entry represents a route to a destination node and composed of a label component and a route component. Each label component contains outbound label (oLabel), next hop and destination address and a list of upstream neighbor of the nodes taking use of the label. Route component stores the post route, the node sequence from the current node to destination. Each route entry is index-ed by a label. We will back to why choose post route than entire path in section 3.3.3.

After created, a route or label is active before it becomes unavailable due to timeout or invalidated by a link broken event or route update packet. And a label is activated after it has been used to send data successfully. Furthermore, an entry is called valid when it is active and activated with non-empty post route, only valid route can respond route request at intermediated nodes. Moreover, a node is to say that has an available route to a destination if such route to that node exists and active.

Consider a path S…XAY…D, we refer to node X and Y as the upstream and downstream neighbor of node A and path segment S…X and Y…D as the preceding and post path respectively. In this paper we use the term route and path interchangeably. In addition, every node in S…X or Y…D is called the upstream or downstream node of A respectively. Finally, besides the source and destination node, the sending node of a packet is defined as the node sending it immediately, for example, when a packet arrives A, its sending node is X. Obviously, different from source or destination, sending node changes dynamically along the transmission of the packet. The extension required by ORAL at packet header includes a label and the address of sending node.

3.2 Forwarding Phase

Forwarding process handles the data packets from other nodes or the upper layer such as transport layer. When a node receives a packet from transport layer, it first tries to find an active route by searching route table with the destination address, similar to other protocols. If such entry is found, the packet is sent to the next hop specified in the found route after its ORAL extension is created and filled with the oLabel of the entry and the address of the node itself. Otherwise, the node buffers the packet and enters into routing phase to find an available route.

The great difference between ORAL and other protocols is in relaying data packet from other nodes. When such packet arrives, node first attempts to get the route according to the label specified in the extended header. For example, in current ORAL, node uses the label as index to get corresponding element of the route table implementing by an array as the route. After this, the following operations such as to transmit or buffer the packet is the same as above.

The most distinguishing feature here is that node is to match a route entry with the incoming label rather than to search the route table based on the destination address, which is computationally intensive without special optimization when the route table size is very large such as in a network with large size or a great many of flows. In the opposite, the label matching and switch in ORAL is a trivial operation no matter how large the route table size is, reducing the required resource, speeding up packet forwarding and finally resulting in the decrease of transmission delay and the promotion of throughput.

3.3 Routing Phase

Routing phase is to discover and maintain available routes to desire targets when needed. As many on demand protocols, this phase in ORAL also comprises route discovery and maintenance.

Route discovery is called when a node needs to communicate with another node but without any appropriate route available. And route maintenance is called when the status of a route should be changed, i.e., the route will be deleted or updated upon the arrival of new route to the same target with more preferable property or the old route is already expired by link broken. The use of update packet, hence, is not only to invalidate the unavailable route as done by route error packet in DSR or AODV, but also to propagate the new alternative or better route if exists.

In follow we first discuss the handle of route request and reply packets in route discovery, then describe the local repair and handle of route update packet in route maintenance and finally, some comments are given on why we prefer the post route than entire path.

3.3.1 Route Request

A route request packet mainly includes the initiator addresses, destination address, and a partial route composed of the nodes the RREQ already traversed, which, at the beginning, contains only the initiator node and is revised further by intermediate nodes.

When a node has data transferred to another node but has no available route, it will enter into route discovery scheme and may initiate new route request. After a route request is sent, node should wait for a specified interval, during which no extra RREQ is allowable to be sent to the same destination even when additional data to that node comes (these data will be buffered). If no reply is received during the waiting period, however, the node should rebroadcast the request and the new waiting time length is twice as the old one until it reaches the maximum permitted length. This exponential back-off algorithm is to limit the protocol overhead. Finally the node should delete all data in buffer to the target node if no route is returned after the maximum permission waiting period.

When a node receives a RREQ, it will reply the request if it is the destination or it has a valid route to the target. Or else, it should append its own address to partial path and re-broadcast the request. Of course, a node should handle the same request only once and the request that has too long partial path, i.e., exceeds the maximum hop count, should be saliently discarded.

3.3.2 Route Reply

A reply packet is launched when node decides to respond a route request as it is the intended destination or it has a valid route to the intended target node.

The basic reply packet mainly contains a label and a complete path. If this reply is initiated at the destination, the label is a reserved self-indication label, *SIL*, and the complete path is the partial path recorded in route request adding the destination address. Or else, when an intermediate node responds a request, the label is the index of the route element to the target node, and the complete path is the partial path in route request and the post route, concatenating by the intermediate node itself. A limitation here is that the constructed path must be loop-free and the total length no more than maximum hop count.

When a reply arrives, the intermediate node needs to update its route table in case that no route to the destination or the existing route is unavailable or not as good as the one carried in RREP with respect to some metrics such as path length or QoS and so on. The change to a new route must be propagated via route update packets to all upstream neighbors. Now all data in buffer to the target can be sent by means of the new coming route and the request to the destination can be deleted.

Nodes except the first node of the complete path should relay the reply to the upstream neighbor specified in the complete path after change the label to the index label of the relative route at sending node.

3.3.3 Local Repair and Rout Update

Currently, no neighbor detection is deployed except when data transfer is failure, the link between current node and the next hop is regarded as broken. Local repair is called to remedy the broken route. The node first inactivates the route. And then a new route discovery process is executed to find an alternative path. During this repair period, all arriving data to the repaired target are buffered. If repair is failed, all buffered data and the route entry are deleted, or else, the route is now set to active and the update should be propagated to all upstream neighbors.

We now discuss the handle of route update packet. A route update packet includes the latest post route of the updated route and the label indexing the entry at the sending node. In addition, a flag is carried in RUPP to indicate whether this packet is to invalidate the route or just to advocate a new path. Note here that the IP address of current sending node is in the source address field in IP header update packet.

A node needs to invalidate a route upon the following three conditions: the route is timeout, or a local repair is completed but still no alternative route is found or an arriving RUPP indicates that the route is already invalid. Similarly, an update to alter a route is invoked after a node replaces one route with new outbound label or path when a local repair succeeds to find a route, or a better route is found or an arriving RUPP indicates the change of the route.

When a route update packet is received, the node must update the matched route entry. A route is a match to an update packet when the next hop of the route entry is the source of RUPP; the outbound label is the same as label carried in RUPP and the destination is also the same as the target address specified in RUPP.

If the update is to invalidate route, the match route is deleted. If the update is just to modify current route, otherwise, the post route must be changed according to the update packet. In both cases, if some upstream neighbors exist, this change should be further propagated via new update packets.

In order to avoid possible loop, before a node updates the post route, it must first ensure that the new route has not included itself. If the node is already in the new path, the only allowable operation is to invalidate current route even the received update is to change route. We will back to this issue in section 4.

Finally we emphasize that the use of post route is indispensable and crucial for the control of route table size. Suppose, at some time, node C has a route to E with post route DE, which is created with complete path ABCDE. Now a request for E from node F with partial path FGK arrives, if C decides to respond this request with path FGKCDE, it is no need to create a new entry to store the path FGKC DE but only to add K into the upstream neighbor list of the already existing entry. This, as result, make the total route table size in ORAL proportional with the amount of actively connected nodes, rather than the number of active flows, superior to these protocols like DSR choosing to store the complete path.

3.4 Example

Now we give an example to illustrate the above scheme with topology shown in Fig.1, where the node pair connected by solid line can communicate directly. Initially, every node has an empty route table. The operations similar to other protocols as DSR are omitted or discussed curtly.

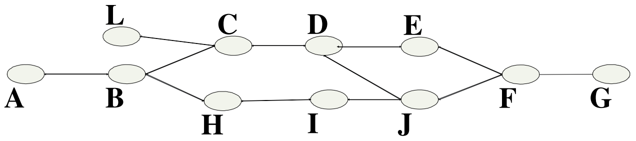


Fig. 1. Network Topology

At time t_0, node A needs to connect with node G but no route is available, then A broadcasts its request to G, now the partial path is with A only. This request is further propagated to node B, C, H and so on.

Suppose at time t_3, G receives the request at the first time and decides to reply rather than propagate the request. Without loss of generality, suppose the request is with partial path as ABCDEF. Then G sends a RREP with *SIL* and the complete path ABCDEFG back to node F via unicast fashion.

Table 1. The route table status at time t_6

Node #	Upstream	Destination	Next hop	Out Label	Post route
A	#	G	B	L_G(B)	BCDEFG
B	A	G	C	L_G (C)	CDEFG
C	B	G	D	L_G (D)	DEFG
D	C	G	E	L_G (E)	EFG
E	D	G	F	L_G (F)	FG
F	E	G	G	SIL	G

Table 2. The packet extension content

Node #	When packet arrives		When packet leaves	
	Label	Node Address	Label	Node Address
A	#	#	L_G(B)	B
B	L_G(B)	B	L_G(C)	C
C	L_G(C)	C	L_G(D)	D
D	L_G(D)	D	L_G(E)	E
E	L_G(E)	E	L_G(F)	F
F	L_G(F)	F	SIL	G
G	SIL	G	#	#

At time t_4, F receives the RREP. F allocates a new entry to store the route and sets the destination and next hop as node G, the oLabel as *SIL* and post route as "G". Suppose the entry is indexed by a new label, L_G(F). Here, L_X(Y) indicates the label allocated by node Y to index the route to node X. Finally, F changes the label in

RREP to $L_G(F)$ and relays it to node E. And now the route in F is in active. If there has some data to G in buffer, F can send them and then activate the entry. Similar operation is executed at node E, D, C, B and A along the propagation of RREP. A will not to further propagate the reply if that no request to G is stored at A. Table 1 summarizes the final route entries from node A to F.

At time t_7, node L also needs communication with G. assuming the route at node C is still available and already activated. Then once node C receives the request, it can respond a reply with path LCDEFG and label $L_C(G)$ to node L. And the request is no longer propagated. Now the upstream list at node C changes to "B, L".

Table 2 gives the ORAL extension at packet header upon a data packet arrives at and leaves a node from A to G along the path ABCDEFG. When packet first come to network layer from upper layer like transport layer at node A, no ORAL extension exists and finally at node G, no further forwarding is needed so that the ORAL extension is removed. We do not present the more detail operations which are quite similar to existing routing protocol except the embedding label switch process.

4 Route Loop

This section sketch the property that ORAL is loop-free, in the sense that every loop can be detected before the route is formed stably, that is , no RREP or RUPP is propagated along the route, provided all assumptions in section 3 are satisfied.

In fact, only two types of control packet RREP and RUPP can construct or alter a route. Firstly, if a route is constructed by RREP only, this case is the same as the route construction in DSR. And now every node receiving RREP has a complete view about the route, it can easily determine whether the route contains a loop or not.

One could be noted that update packet is used to maintain the latest status of existing route as consistent as possible across the entire network. Thus, at the very beginning, routes are only constructed by means of RREP, which is already shown to be loop-free. So in follow we only consider the form of loop upon a loop-free route.

Secondly, if at least a RUPP takes part in the construction or alteration of a route, loop may incur since RUPP only gives the post route rather than entire path.

A loop can be formed as follow: consider two different node A and B, for a intended target, A at some time becomes the downstream node of node B but before this fact is known by B, a request for the same target from A reaches B and invokes its response, then a loop between A and B incurs. The key point here is that B responds a request from one of its downstream nodes, which must be prohibited to avoid loop. Finally, B is placed in the revised route after A accepts the reply from B.

Furthermore, the handle of route update prevents the loop from keeping permanently. After a route is modified, node must inform the update to its upstream neighbors unless the node has no upstream neighbors or all of them are unreachable. In the latter case, obviously, the loop has no effect since the route is already unavailable though may not be discovered. For the former, however, we can ensure that, B, the upstream node of A, must receive an update with the mistaken route initiated at node A in future. Then B finds that it is already in the new route, so node B chooses not to modify the route, but to invalidate and delete it, resolving the loop.

Comments: first, the major drawback here is that ORAL can not guarantee instantaneous loop-freedom and a packet may still arrives at a node at most twice, wasting bandwidth resource. Actually, this problem also exists in salvaging packets in DSR. Second, the control packets interchanged between the form and resolve of route loop have no use for packet forwarding and incur only the waste of network resources. Finally, we note that using complete path to solve the route loop is a difficult task and requires nontrivial overhead under several strict constraints. Due to space limitation, however, the detail discussion is omitted.

5 Experiments Setup

We have evaluated the performance of ORAL using GlomoSim ver 2.03 [16]. In simulation, all nodes have a uniform radio range of 250m and place over a rectangular area of 1000m*1000m except the case of 25 nodes is in 500m*500m. The movement is according to the random waypoint model. The values of pause time used are from 100s to 500s with step 100s. Besides, the static topology is also explored.

We compare ORAL with two on-demand protocols DSR and AODV, all at network layer. In addition, the MAC layer uses 802.11 at 2 Mbps. All flows are CBR with packet size kept at 512 bytes. The run time of each simulation is 600s. We run each simulation using five different seeds and the final results are the average. We also perform the experiments with different traffic type such as FTP or the mixture of FTP and CBR, and similar results are obtained but not presented here.

The following metrics are used:

Average Packet delivery ratio: the ratio between the number of packets sent out by the sender application and the number of packet correctly received by the target.

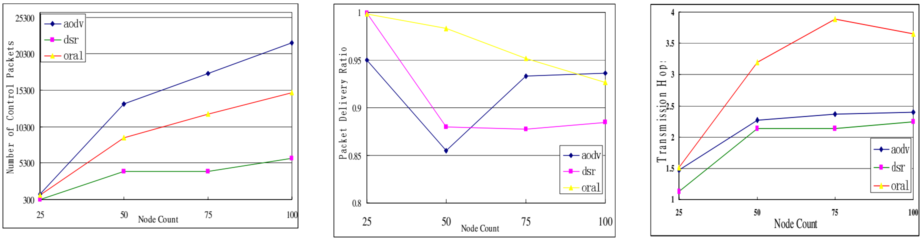
Control packet overhead: the number of control packets sent out during the simulation. The packet is counted once it has been transmitted. Thus, for a control packet, it may be counted multiple times in the case of traversed over more than one hop.

Average transmission cost: the ratio of total number of transmitted data packets at all nodes over the number of packets received successfully at all destinations. This metric, actually, is the average transmission cost per packet.

6 Results and Analysis

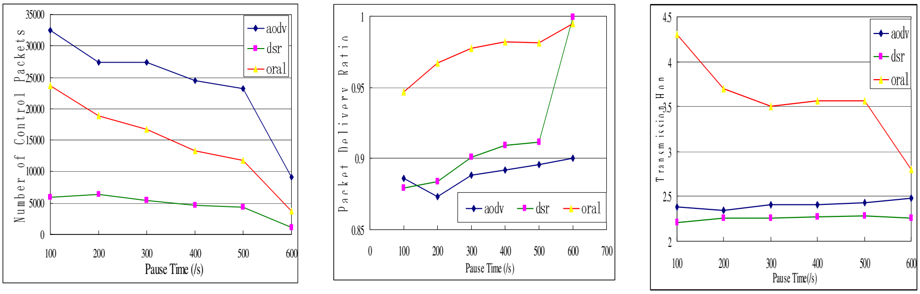
In this section we discuss the results under different node count, flow count and mobility. Fig.2 presents the experiment results under different node count from 25 to 100 with step 25 when all traffic is 30 CBR flows and the pause time is 200s. Fig.3. gives the experiment results under different node mobility where the pause time from 100s to 500s with step 100s at the scene of 50 nodes and 30 CBR flows. The results of pause time 600s refers to the case of static topology. Fig.4. shows the experiments results under different flow count from 10 to 50 CBR flows when the node count is 50 and pause time 200s.

From these figures, several conclusions can be drawn: first, ORAL has much greater packet delivery ratio at most cases over the other two protocols, which is very good for improving the system throughput.



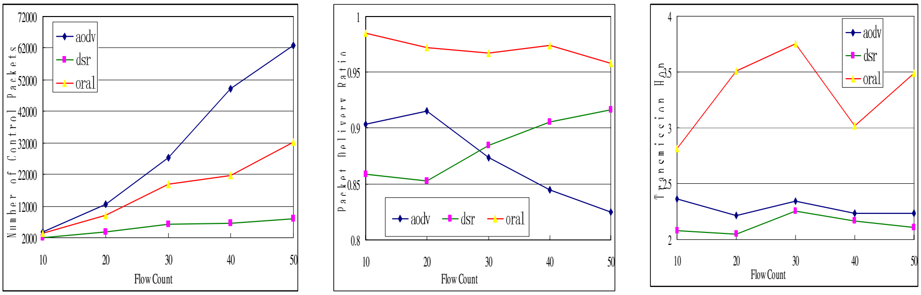
(a) Control Packet Overhead (b) Packet Delivery Ratio (c) Average Transmission Cost

Fig. 2. Results under different node count



(a) Control Packet Overhead (b) Packet Delivery Ratio (c) Average Transmission Cost

Fig. 3. Results under different node mobility



(a) Control Packet Overhead (b) Packet Delivery Ratio (c) Average Transmission Cost

Fig. 4. Results under different flow count

Along with the increase of node count, though some decrease, ORAL still gains the best delivery ratio except when count is up to 100, where AODV gets 0.9% more ratio but at the expense of about 30% more control overhead than ORAL.

No matter how large the pause time is, ORAL always gets the much highest delivery ratio than DSR and AODV, only when the topology is static shown as pause time 600s, DSR get the same delivery ratio as ORAL.

Along with the increase of flow count, the delivery ratio of AODV decreases and of DSR increases. In all cases, however, ORAL always outperforms than DSR at most about 10% and at least about 5% though both ORAL and DSR deploy similar source route mechanism.

Second, the higher throughput of ORAL is at the expense of larger overhead. DSR always gains the shortest average transmission cost per data packet and needs the least amount of control packet. AODV, in the opposite, almost needs the largest number of control packets and the increase with respect to the growth of flow count, mobility or node count is also the fastest. Especially in the case of different flow count, this increase is close to linear. ORAL stays at the balance point between AODV and DSR. The control overhead of ORAL is larger than DSR but less than AODV. Along with the increase of network dynamic, ORAL also consumes more resource but at much less increase rate compared to AODV.

The major drawback of ORAL is that it always has the greatest transmission hop count per data packet. In the worst case, ORAL has about two more hops than DSR and 1.5 more hop than AODV and on average, one more hops than DSR and 0.7 hops more than AODV. Local repair, we believe, plays an important role in the higher throughput and longer transmission hop since the remedy at intermediate node resists a large number of packet loss but these packets are transmitted over an alternative path which is longer than the broken route because at current ORAL, the shortest path selection policy is use.

7 Conclusions and Future Work

Routing is a key issue in mobile ad hoc network. We present a new on demand ad hoc routing protocol taking use of MPLS like label switch. ORAL separate the protocol behaviors into routing and forwarding phases, while the former is to discover and maintain the label switch path, as well as the traversed node sequence, e.g., source route, similar to other on-demand protocol especially DSR. In the forwarding process, however, ORAL use label switch rather than destination address or prefix matching to find the intended route, fastening the packet forwarding and finally improving system performance.

ORAL is scalable with respect to network dynamics. Integrating label and post route mechanism, a refined source route, ORAL can limit the route table size not exceed the amount of active nodes, rather than the greater network size or the amount of active flows, given no optimization such as multi-path route is deployed.

We implement ORAL in simulation and compare it with two other on-demand protocols, DSR and AODV. The results show that ORAL outperform much greater than DSR and AODV with protocol overhead much less than AODV but a bit more than DSR. The main drawback is the consistent relative larger hop count for the data packet transmission by ORAL compared to the other two protocols, which will be further examined in future work.

Acknowledgement. This work is supported by National Natural Science Foundation of China under Grant No. 60273068 and National Grand Fundamental Research 973 Program of China under Grant No.2006CB303000. We also give thanks to the reviewers for their valuable comments.

References

[1] D. Johnson, D. Maltz and Y-C Hu. The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR). IETF Draft, drat-ietf-manet-dsr-10.txt, July, 2004.

[2] E. Rosen, A. Viswanathan and R.Callon. Multi-protocol Label Switching Architecture. RFC 3031, Jan. 2001.

[3] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, Jul 2003.

[4] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In Proc. SIGCOMM, August 1994, pages 234-244.

[5] T. Clausen, Ed., P. Jacquet, Optimized Link State Routing Protocol (OLSR), Network Working Group, Request for Comments: 3626.

[6] R. Ogier, F. Templin and M. Lewis. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). IETF RFC 3684, Feb.2004.

[7] Z. J. Haas and M. R. Pearlman, ”The performance of a new routing protocol for the reconfigurable wireless networks,” Proc. ICC 1998.

[8] V. Ramasubramanian, Z J. Haas and E. G. Sirer. SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks. ACM MobiHoc,2003.

[9] M. Joa-Ng, 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 17 (8) (1999).

[10] T. Clausen. The Optimized Link State Routing Protocol Version 2. IETF MANET Draft, Aug. 2005.

[11] M. Gerla, Fisheye state routing protocol (FSR) for ad hoc networks, Internet Draft, draft-ietf-manet-aodv-03.txt, work in progress, 2002.

[12] J. Eriksson, M. Faloutsos and S. Krishnamurthy. DART: Dynamic Address RouTing for Scalable Ad Hoc and Mesh Networks. IEEE/ACM Transactions on Networking. Feb. 2006.

[13] A. Acharya, A. Misra and S. Bansal. A Label-switching Packet Forwarding Architecture for Multi-hop Wireless LANs. ACM WoWMoM 2002.

[14] H. Liu and D. Raychaudhuri. Label Switched Multi- path Forwarding in Wireless Ad-hoc Networks. IEEE PERCOM Workshops, 2005.

[15] V. Untz, M. Heusse, F. Rousseau, and A. Duda. On Demand Label Switching for Spontaneous Edge Networks. In Proc. SIGCOMM FDNA, Portland, August 2004.

[16] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks. In 12th workshop on Parallel and distributed simulation, pages 154–161, Banada, Canada, May 1998.

[17] M. Abolhasan, T. Wysocki and E. Dutkiewics. A Review of Routing Protocols for Mobile Ad Hoc Networks. In Ad Hoc Networks. 2004.

[18] S. Pleisch, M. Balakrishnan, K. Birman and R. Renesse. MISTRAL: Efficient Flooding in Mobile Ad hoc Networks. In ACM MobiHoc. 2006.

[19] J.J. Garcia-Luna-Aceves and S. Roy, ``On-demand Loop-Free Routing with Link Vectors," Proc. IEEE ICNP 2004.

[20] H. Lundgren, E. Nordstrom, C. Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc network, 5th ACM WoWMoM, Sep., 2002

Source-Based Multiple Gateway Selection Routing Protocol in Ad-Hoc Networks*

Sang-Jo Yoo and Byung-Jin Lee

Multimedia Network Laboratory, The Graduate School of IT&T,
Inha University 253, Yonghyun-Dong, Nam Gu,
Incheon 402-751, Korea
sjyoo@inha.ac.kr, arin0213@empal.com

Abstract. A mobile ad-hoc network (MANET) is one consisting of a set of mobile hosts capable of communicating with each other without the assistance of base stations. It is necessary to use bandwidth effectively because MANET has limited bandwidth. In this paper, we propose SMGS in which each node estimates its expected life time and if its ELT is larger than that of current gateway it becomes a candidate node. When a source node establishes a path, in each grid the candidate node will take the route request and be a gateway node for the each source node. The node that is expected to stay the longest time in the grid is selected so that we can reduce frequent gateway handoff, packet loss, and handoff delay.

Keywords: Geocasting, routing, ad-hoc network, gateway.

1 Introduction

According to the advancement in portable computing devices and wireless communication, it is possible to communicate for mobile devices. One research issue that has attracted a lot of attention recently is the design of mobile ad-hoc network (MANET). MANET is one consisting of a set of mobile hosts capable of communicating with each other without the assistance of base stations. Applications of MANETs will occur in many situations such as battle fields or major disaster areas, where networks need to be deployed immediately but base stations or fixed networks infrastructures are not available. Network topology of MANET isn't fixed because nodes in MANET move continuously. So we cannot utilize wired routing protocols for MANET and need a new routing protocol.

Many routing protocols have been proposed for MANET. Among them, protocols using location information [1-6] have been researched to reduce network overhead. To use these protocols, they assumed that each node knows its current location information provided by positioning devices such as global positioning systems (GPS).

LAR (location aided routing)[1] proposes an expected zone using location information of destination provided by past communication records. If it has no

* "This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)" (IITA-2005-C1090-0502-0030).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 107 – 118, 2006.

communicationrecords, it uses flooding routing protocol as like AODV. The expected zone is the circular region. And LAR protocol also proposes request zone which is rectangle including source and the expected zone. The request zone is used to confine the zone to be searched for a route from source to destination. When a source node wants to send data, it sends a RREQ (route request) packet first. If the node, which is located in the request zone, receives a RREQ packet, it forwards a RREQ packet. If not, it discards a RREQ packet. LAR protocol can reduce network overhead caused by route discovery process by the method to restrict the packet broadcasting area. But LAR protocol still has network overhead problem because all nodes in the request zone also use broadcasting scheme though the range of route discovery is restricted.

GRID[2] protocol partitions LAR's request zone into small grids, each as a square. One mobile node is elected as the leader of the grid. It is termed gateway and responsible for route discovery, packet relay and route maintenance. When the gateway node leaves out the grid, it transfers routing table to a next gateway node. GRID protocol is more efficient than LAR protocol. Only one of several nodes in the grid is responsible for communication, so it can reduce network overhead. But the gateway node can have traffic concentration problem which can lead to network partition or disconnection. And GRID protocol generates delay to elect gateway when a gateway node leaves its grid. If a gateway node leaves its grid while it transmits data, it would lead to data packet loss.

In this paper, we present a source-based multiple gateway selection routing protocol (SMGS). Each node estimates its expected life time (ELT) in the current grid and if its ELT is larger than that of the current gateway node it can be a candidate node of the grid. When a source node builds a path to the destination, a candidate node at that moment of each grid will take the route request from the source node. Because the gateway of each source is the node that is expected to stay longest time in that grid, we can reduce the frequent grid changes. The proposed SMGS can avoid traffic concentration on a gateway node so that it is able to distribute power consumption to each node of the ad-hoc network.

The rest of this paper is organized as follows. Section 2 explains the proposed SMGS protocol and Section 3 shows our experimental results. In Section 4, we conclude this paper.

2 Source-Based Multiple Gateway Selection Routing Protocol

2.1 Structure of Grid-Based Routing Protocol

Grid based routing protocols partition whole area into many small grids and elect one gateway node in each grid which is responsible for communication. Routing is performed in a gird-by-gird manner through gateways. Grid based routing protocols can eliminate the broadcast storm problem that is associated with existing protocols when searching for new routes. The GRID[2] protocol is a representative protocol among grid based routing protocol. The node, which is the closest node at the geographical grid center, is elected as a gateway. The gateway node broadcasts GATE message periodically.

GRID route discovery procedure begins when a source node S sends RREQ (route request) as in Figure 1. If nodes, which receive RREQ packets, are satisfied with the following conditions, the nodes discard RREQ packets.

1) Nodes are out of zone like node I.
2) Nodes in the request zone receive duplicated RREQ packets.
3) Nodes are not gateway or destination node.

GRID can utilize network resource more efficiently than other protocols by reducing the number of control packets. Disconnection or network partition can occur because one gateway node is in charge of whole communication coming to its grid. In Figure 1, gateway node B broadcasts a RREQ packet and node E receives the RREQ packet. Node E first checks whether it is located in the request zone or not. Then node E rebroadcasts the RREQ packet and saves a reverse path point to the previous grid. The RREQ packet is forwarded and sometimes later a node D receives the RREQ packet.

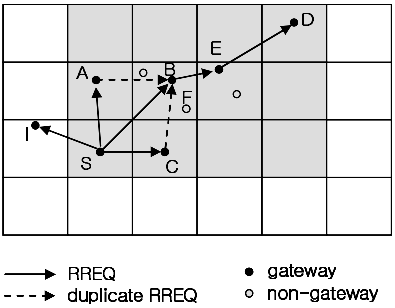


Fig. 1. RREQ forwarding

In Figure 2, as the destination D receives the RREQ, it responds a RREP (route reply) packet by unicasting to source S. This packet follows the reverse path that was established by the RREQ packets. Each gateway establishes an entry in its routing table indicating the next grid leading to destination D. The route discovery process has been finished when node S receives a RREP packet. And the path is established.

When a gateway leaves its current grid, it should broadcast a RETIRE (**g**, **T**) packet, where **g** is the grid coordinate where it served as a gateway and **T** is the routing table. Other nodes in this grid, on hearing this packet, will inherit the routing table T and broadcasts a BID (Broadcast ID) packet including the distance from the node to geographical center in order to compete as a gateway node. Those nodes which received the BID packet compare with its own distance and if the node's distance is larger than distance in received the BID packet, the nodes stop broadcasting the BID packet. If not, the node broadcasts continually own BID packet. After the time defined in advance expires, if only one node broadcasts the BID packet, the node is elected as a gateway node and broadcasts GATE message. This hand-off procedure has problem when gateway node is transmitting data packet. This protocol needs time to elect new gateway, that is why packet loss can occur during that time.

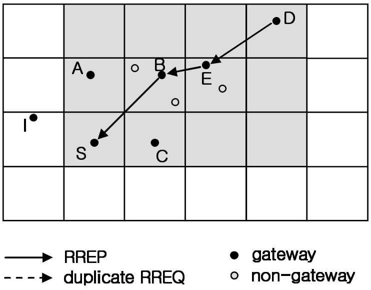


Fig. 2. RREP forwarding

2.2 The Proposed SMGS Routing Protocol

In this section, we present SMGS (source-based multiple gateway selection routing) protocol based on grid mechanism. Our proposed protocol can reduce network overhead as like GRID protocol and can solve problems which GRID protocol has. SMGS protocol can reduce the number of gateway hand-off, so it can reduce data packet loss rate and time delay for electing gateway. SMGS also can distribute traffic to several gateway nodes by multiple gateway election.

We assume SMGS protocol has the following conditions. First, each node knows its current location by using positioning device (e.g., GPS). Each node can calculate moving velocity by comparing current location with past location and can acquire the expected life time (ELT) value which is remained staying time before its leaving out the current grid. ELT value indicates the node's predicted staying time in current grid and is derived as (1).

$$ELT = \frac{S}{V} \tag{1}$$

(S = remained distance from the grid boundary, V=node velocity)

Second, node transmission range must include neighbor grid like Figure 3. Each gateway node must be able to listen periodic GATE and CANDIDATE messages which are broadcasted from the neighbor grids.

In SMGS protocol, we elect the node which has largest ELT in the grid to be a candidate node. If a RREP packet passes this node, it changes to be a gateway node.

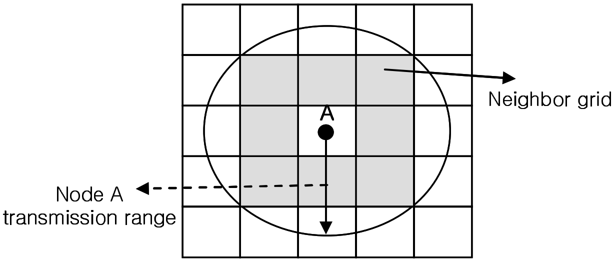


Fig. 3. Gateway signal transmission range

Data is forwarded by gateway nodes. And with the concept of the candidate node we can solve delay and packet loss problem which is originated from hand-off. The candidate node is elected as Figure 4. In Figure 4-a, node S is sending data through established path (S-A-D). Some time later, several nodes that have larger ELT value than that of gateway appear in the grid (Figure 4-b). They become CANDIDATE nodes and broadcast CANDIDATE packets periodically. Candidate nodes compare own ELT with other candidate node's ELT values. If ELT in the candidate packet is larger than own ELT, the candidate node stops broadcasting. Else the node continues to broadcast candidate packets. As shown in Figure 4-c, only one candidate node remains finally.

Table 1. Terminologies of SMGS

ELT	The node's remained time until it leaves out current grid.
Gateway node	In the grid, the node which can forward route discovery packet and data packets and maintain route for a specific source. A gateway node broadcasts GATE message periodically. In a grid, there can be multiple gateways.
Gateway candidate node	A node that its ELT is the largest in the grid at that moment. It broadcasts CANDIDATE message periodically.
GATE message	The packet broadcasted by a gateway node (node ID, location, ELT).
CANDIDATE message	The packet broadcasted by a candidate node (node ID, location, ELT)
GATE_CHANGE message	When a gateway is changed (hand-off), a changed gateway node sends this packet.

In SMGS, a source node broadcasts a RREQ packet for establishing route as shown in Figure 5. If a gateway node, which is out of request zone as node I receives the RREQ packet, the node discards the RREQ packet. If a candidate node (or a gateway node when there is no candidate node in the grid) that receives the RREQ packet and it is not duplicated packet, the node records source address and sequence number in its routing table in order to check duplicated RREQ packet. It records its grid number (not node id) in the RREQ packet and forwards the RREQ packet. When the node, that is neither a candidate node (or gateway node) nor a destination node receives the RREQ packet, the RREQ packet is discarded. SMGS protocol can reduce the number of hand-off by electing the nodes which have largest ELT value to be gateway nodes when the RREP packet comes back. The RREQ packet is forwarded until RREQ packets arrive at the destination node.

A destination node receiving a RREQ packet replies with a RREP packet. As shown in Figure 6, destination node D sends the RREP packet to node E which is a gateway node in grid 2 because it received only GATE packet from grid 2. If the destination node D receives CANDIDATE and GATE packets from grid 2, it sends the RREP packet to a candidate node in grid 2. Node E received both GATE and CANDIDATE packets from grid 3 which is its backward grid of grid 2. So node E elects candidate node B to be a gateway in grid 3 by sending the RREP packet to node B which is a candidate node in grid 3. If there are several candidate nodes in backward grid, RREP is sent to the candidate node which has the largest ELT value among them. The Route discovery process will be finished after the RREP packet arrives at source node S.

In the conventional GRID protocol, node F is elected to be a gateway node in grid 3 even if node B is expected to stay longer than node F in grid 3 when a new source node initiates a connection. In GRID protocol, once a gateway node is decided, it continuously serves before it leaves out the grid. In SMGS protocol, at the moment that any source node starts its route discovery procedure, the node which is expected to stay the longest at each grid, is elected to be a gateway so that the selected gateway can serve the connection with little handoff possibility.

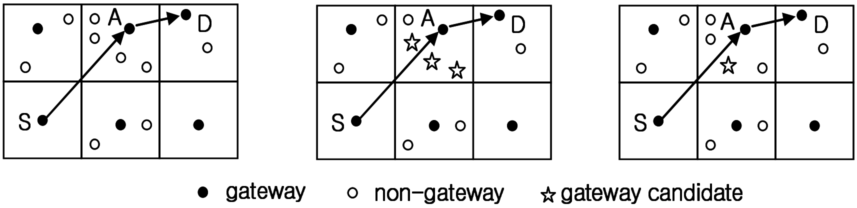


Fig. 4-a **Fig. 4-b** **Fig. 4-c**

Fig. 4. Candidate node election

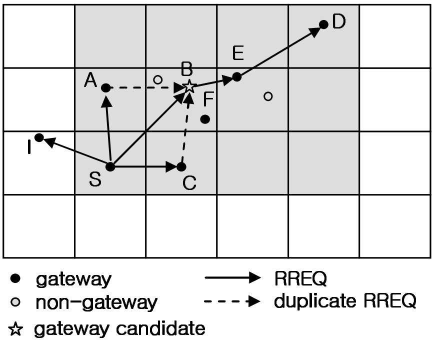


Fig. 5. RREQ forwarding

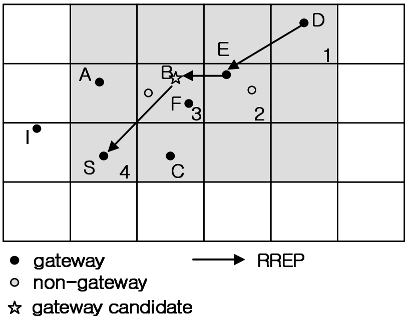


Fig. 6. RREP forwarding

SMGS protocol can also reduce time delay and data packet losses caused by frequent hand-off. In SMGS, when a gateway node of a certain data flow leaves out the grid, hand-off will occur. Hand-off procedure is shown in Figure 7. Node A and node C node transmit data to node D and F through path 1 (A-B-F) and path 2 (C-B-D) respectively. Node E is a candidate node which has the largest ELT value in the grid. If ELT value of node B comes to be smaller than ELT_limit value. It means that B will leave out the grid before long. Therefore node B sends its routing table to a candidate node E (Figure 7-b). When node E received routing table from node B, it broadcasts the GATE_CHANGE packet to report that it becomes a new gateway node in this grid (Figure 7-c). If node A and node C receive the GATE_CHANGE packet, they establish new route using node E as a intermediate node (Figure 7-d). In GRID protocol[2], hand-off mechanism leads to delay for electing gateway. Therefore, if the gateway node that is sending data packets leaves out grid, disconnection can occur temporarily during the time of electing gateway. SMGS protocol solves problems such as packet loss and delay for electing gateway, which are caused by gateway hand-off, by carrying out handoff procedure in advance through the method to predict the time when a gateway leave out its grid.

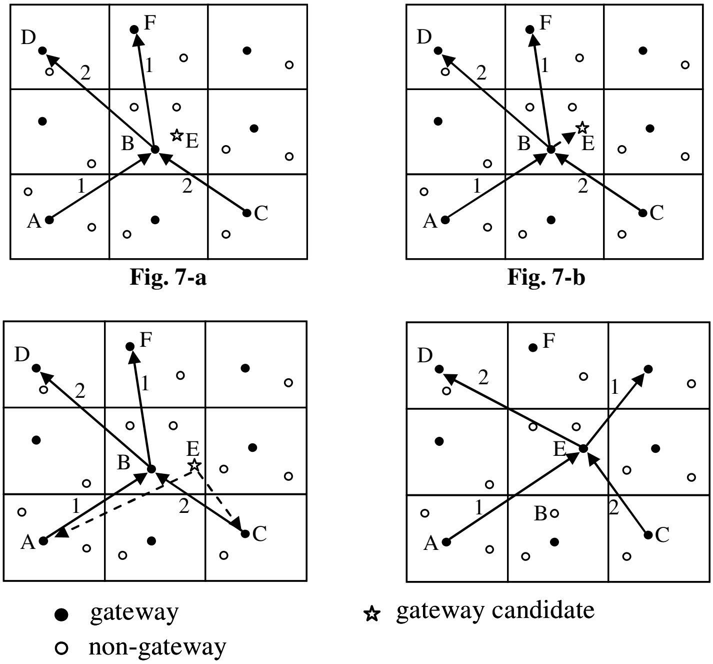


Fig. 7-a

Fig. 7-b

Fig. 7-c

Fig. 7-d

Fig. 7. Procedure of gateway handoff

SMGS protocol presents multiple gateway election algorithm. There can be multiple gateway nodes in each grid in SMGS protocol unlike GRID protocol. Multiple gateway election procedure is as follows.

In Figure 8-a, we assume node A in grid 7 is transmitting data packet to node F through the following path (A-E-F). Node B in grid 4 broadcasts a RREQ packet when it wants to transmit data to node C. At this time, candidate node G, which has the largest ELT in grid 5 and is not a gateway node, will be responsible for forwarding a RREQ packet (Figure 8-b). If it has no candidate node at the moment, it means that the gateway node E has the largest ELT and node B sends a RREQ packet to a gateway node E. So the RREQ packet is forwarded to the destination node C. And destination node C received the RREQ packet sends a RREP packet. When the RREP packet arrives at the source node B, Route discovery process has been finished and the source node B sends data packet to the destination node C. At this time there are two gateway nodes (G, E) in grid 8 (Figure 8-d).

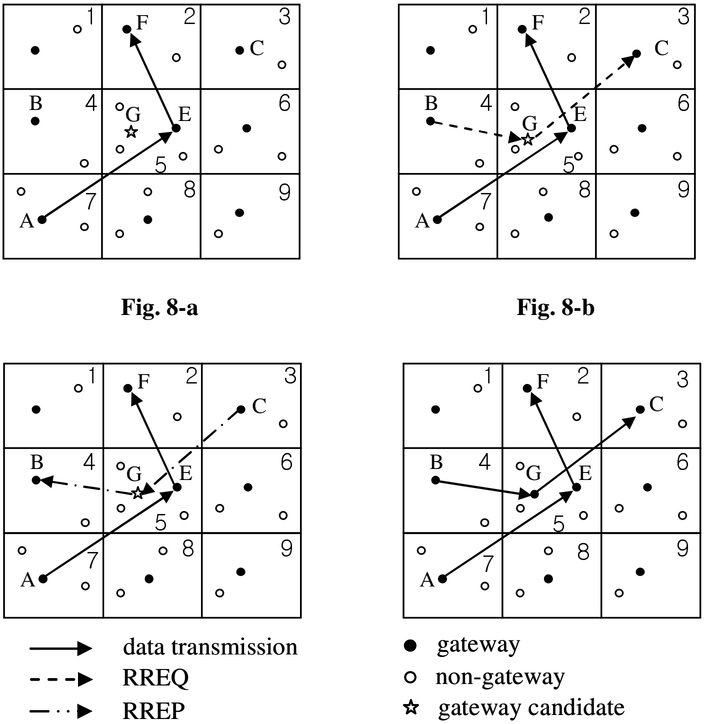


Fig. 8-a

Fig. 8-b

Fig. 8-c

Fig. 8-d

Fig. 8. Multiple Gateway Election

In GRID protocol, regardless each node's ELT time, once a node is selected as a gateway node. It serves all data flows from different sources until it leaves out the grid. If traffic is heavy in grid 5, traffic concentration problem can occur at node E and node E consumes its battery rapidly. But SMGS protocol can prevent rapid battery power consumption at one certain node because multiple gateway nodes can exist in a grid. Multiple gateway election procedure is simple as shown in Figure 8. SMGS protocol elects the node that has the longest predicted staying time in its grid at the moment so that for each flow of the source-based gateway will serve relatively long time compared with GRID.

3 Simulation Results

In this paper, we used the ns-2 simulator for evaluating performance. We compared the performance of SMGS with the GRID. The simulation model is explained in Table 2.

Table 2. The Simulation Model

Simulation area	1000 * 1000 m
Grid region	100 * 100 m (one hundred of grids)
Mobile node	50 ~ 600 (Each node knows its location)
Moving velocity	10~30m/s (36 ~ 108 km/h)
The number of connection	10 ~ 50
Traffic	1Mbps CBR traffic
ELT_limit	0.3

The first simulation experiment compared average gateway lifetime of two approaches (SMGS, GRID), by varying node velocity. For this experiment it is assumed that the number of mobile nodes is 100; the number of connections is 50. Figure 9 shows, SMGS protocol shows 20% increased gateway lifetime compared to GRID. Because in SMGS node sojourn time in its grid is considered to elect gateway node by using ELT value so that the number of handoff can be reduced. Network overhead can also be reduced because the amount of control packets that are originated when handoff occurs is reduced.

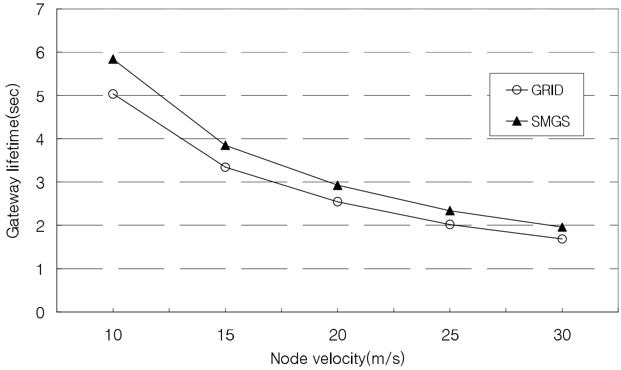


Fig. 9. Gateway lifetime varied by velocity

The second simulation experiment compared average gateway lifetime of two approaches (SMGS, GRID), by varying node the number of nodes. For this experiment it is assumed that the node velocity is 10m/s; the number of connections is 50. In

GRID protocol, gateway lifetime does not change significantly the number of nodes because GRID protocol elects a node which is the closest from geographical center in grid to be a gateway. On the other hand, in SMGS protocol, gateway lifetime is changing significantly as the number of nodes is increasing because the possibility to elect a node having comparatively long sojourn time in grid as a gateway is larger than grid.

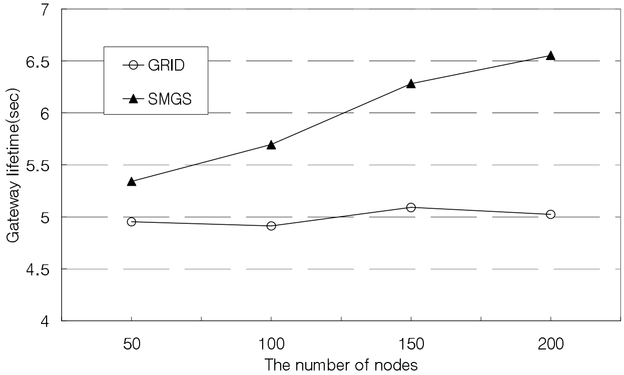


Fig. 10. Gateway lifetime varied by the number of nodes

The third simulation experiment compared the average number of gateway nodes by varying node the number of connections. For this experiment it is assumed that the node velocity is 10m/s; the number of nodes is 600. In Figure 11, the average number of gateway nodes in each grid increases as the number of nodes attempting to communicate in SMGS. When the number of connections is 50, the average number of gateway nodes in SMGS is about 2.4 times as many as GRID. SMGS relieves disconnection problem caused by node energy shortages because it makes power consumption to be distributed to multiple gateways in each grid by using multiple gateway election method. Therefore, the more number of connections there are in grid, the more gateway nodes there will be.

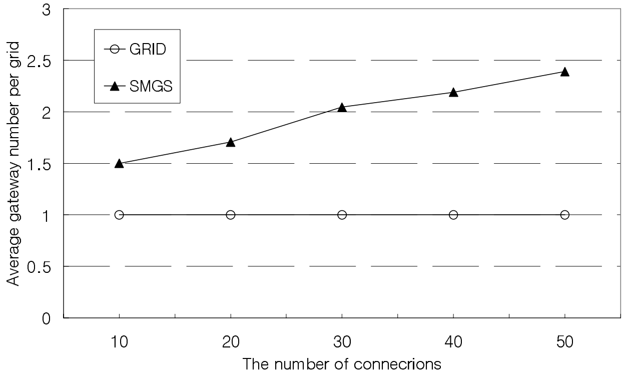


Fig. 11. The number of average gateway nodes varied by the number of connections

The fourth simulation experiment compared amount of data transmission per gateway node by varying the number of connections. For this experiment it is assumed that the node velocity is 10m/s; the number of nodes is 600. As shown in Figure 12, in SMGS amount of data transmission per gateway slowly increases as the number of connections increases because SMGS allows multiple gateways to be elected in a grid. But in GRID, although the number of connections increases, the number of gateway node in one grid is just one. So the gateway node suffers from heavy traffic. In Figure 12, we can know that the amount of data transmission per a gateway node was reduced below 50% of GRID when the number of nodes was larger than 25.

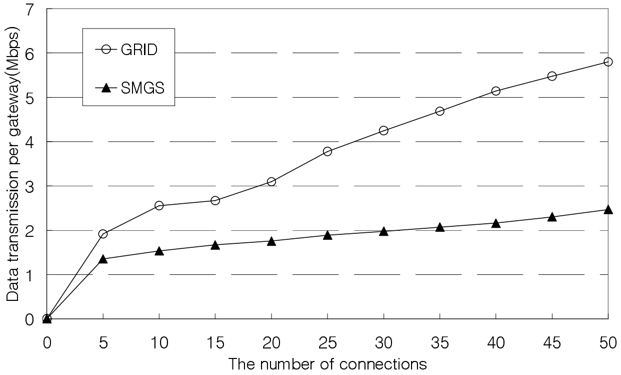


Fig. 12. The amount of data transmission per gateway varied by the number of connections

Finally, we fixed the number of connections (50) and velocity (10m/s). The number of nodes increases from 50 to 300. We compared the amount of data transmission per a gateway. In the GRID protocol, although the number of nodes was increased, only one gateway node forwards data packet. However SMGS protocol distributed data packet processing to several gateway nodes. And the more number of nodes there are, the number of gateway nodes there are also.

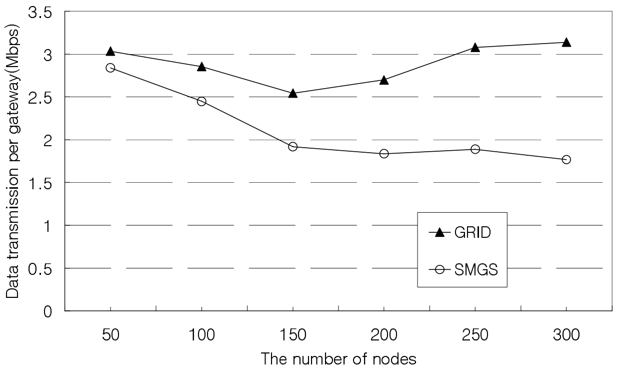


Fig. 13. The amount of data transmission per gateway varied by the number of nodes

4 Conclusion

In this paper, we present SMGS (source based multiple gateway selection routing protocol) which elects multiple gateways by using ELT. SMGS protocol solves network overhead problem caused by control packet broadcasting. It also solves disconnection and network partition problems through multiple gateway election method. It reduces gateway election time delay by using candidate nodes and also reduces data packet loss and transmission delay, which occur during gateway handoff, by electing new gateway before old gateway leaves out its grid. As we can see the experimental results, In SMGS protocol, the amount of average traffic that a gateway node should handle is below 50% compared to GRID protocol so that we can avoid rapid power consumption of gateway nodes. And gateway lifetime also increases by 20% compared to GRID because we take node's staying time into consideration to elect a gateway. So the number of handoff is reduced.

References

1. Young-Bae Ko and Nitin H. Vaidya: Location Aided Routing (LAR) in mobile ad hoc networks. ACM/IEEE International Conference on Mobile Computing and Networking (1998) 66-75
2. Wen-Hwa Liao, Yu-Chee Tseng, and Jang-Ping Sheu: GRID: A Fully Location Aware Routing Protocol for Mobile Ad Hoc Network. Telecommunication Systems, Vol. 18, No. 1-3 (2001) pp. 37-60
3. Chun Huang, Fei Dai, Jie Wu: On-Demand Location-Aided QoS Routing in Ad Hoc Networks. International Conference on Parallel Processing (2004) pp 502-509
4. Jian Li, Prasant Mohapatra: LAKER: Location Aided Knowledge Extraction Routing for mobile ad hoc networks. IEEE Wireless Communications and Networking Conference, Vol. 4, No 1 (2003) pp. 1180-1184
5. W.H Liao, Yu-Chee Tseng, Kuo-Lun Lo, Jang-Ping Sheu: GeoGRID: A Geocasting Protocol for Mobile Ad hoc Networks Based on GRID. J.Internet Tech., Vol.1 No. 2 (2003) pp 23-32
6. Ljubica Blazevic, Jean-Yves Le Boudec, Silvia Giordano: A Location-Based Routing Method for Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing, Vol. 4, No. 2 (2005) pp.97-110

Smart Path-Finding with Local Information in a Sensory Field

Wenzhe Zhang[1], Minglu Li[1], Wei Shu[2], and Min-You Wu[1]

[1] Dept. of CSE, Shanghai Jiaotong University, Shanghai, China
{wzzhang, mlli, mwu}@sjtu.edu.cn
[2] Dept. of ECE, the University of New Mexico, NM, USA

Abstract. Field surveillance is one of the most important applications for wireless sensor networks. Many sensors are deployed in a region of concern to detect any potential targets. On the contrary, intelligent target looks for the best path to traverse the sensing field for fear of being detected and leads to defunct surveillance. In this paper, we focus on how an intelligent target traverses the sensing field. We model this traversing problem, design, implement and evaluate a number of path-finding algorithms. Different from previous works which assume complete information of the sensing field, we assume that the target only can detect part of the sensor network in its detection radius. This makes the proposed methods more practical. Extensive experiments with a target and a sensor network confirm the validity of the approach.

1 Introduction

Sensor networks are usually deployed to monitor a region. Their foreseeable applications include protecting and monitoring military, environmental, safety-critical, or domestic infrastructures and resources [1]. Networked sensors measure signals at a given sampling rate to produce a time series data that are processed by the sensor nodes. The sensor nodes obtain energy measurement accounts for the behavior of the target. Giving the energy, processing the series, and possibly fusing the information between different sensor nodes, the sensor network may detect the target traversing the sensing field. If the target is intelligent, such as an enemy at battlefield, the target may search for a "better" traversing path to escape from being detected. So there is a couple of opposite notions between sensors and the target: monitoring and anti-monitoring, detection and anti-detection. Many works point out and analyze the vulnerability of sensor networks about Denial of Service [2, 3]. Some works look for the best path for the target [4, 5, 6]. However, almost all previous works are based on complete information of sensor deployment. Different from previous works, in this paper we focus on how the target traverses the sensing field with incomplete information.

To evaluate the probability of the target being detected, the authors in [3, 4, 7, 8] introduced the notion of exposure. In [3], exposure is defined as the net likelihood of detecting a target. Exposure in [4] is defined as signal energy gathered by the sensor node. Both of the papers compute the exposure and find the least exposure path with Dijkastra's single-source single-destination algorithm. The author of [7] relates the

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 119 – 130, 2006.

exposure of a path to the distance from the path to the sensors and finds two kinds of paths: the maximal breach path and the maximal support path. Exposure in [8] is defined as the total energy that the sensors gathered when the target moves following the path. All of these works are based on complete information while ignoring the fact that the target cannot know the entire deployment of sensor network in reality. In this paper we propose to search for traversing path based on incomplete information.

As sensors can detect the traversing target, the target can find the deployed sensors, too. Depending on the application and type of target, there are several ways to detect sensor nodes. Practically the target may detect any sensors with different probability. We refer to detection problem as detection probability. Obviously, the larger the detection probability is, the more information of deployment the target knows. The notion of detection probability will be explicated in detail in section 3. In this way complete information in previous works only is a special case of our model.

Obtaining reliable location information is essential to many location-aware tasks in sensor networks such as detection, tracking and mobility management. Fortunately, it is easy to gather such information by the Global Positioning System (GPS) [9]. Since our traversing algorithms rely on location information, we have implemented the location procedure as the initial step. In this localization approach, sensor nodes know their coordinates either from satellite information or pre-deployment. The target also knows it location and destination in advance, where it needs to go.

The reminder of this paper is organized as follows: Section 2 introduces concept of exposure of the path. In section 3, we propose five path-finding algorithms. Section 4 presents the results of our experiments. After analyzing the experimental results, we conclude the paper with section 5.

2 Metrics of the Path

We propose two metrics to evaluate the path: exposure and length. Similarly to [4] and [7], we define exposure as the energy collected by sensors. Exposure of the given path measures the probability of being detected when a target follows the path. Exposure varies with the path the target follows. However, it is not obvious as to which path is the least exposure. The length is considered as the distance the target travels to its destination. In fact, some paths are short but traverse the part of region with high exposure, other paths are long while traverse the part of low exposure. An intelligent target prefers the path with low exposure and short length as well.

2.1 Exposure

Sensor generally has widely different theoretical and physical characteristics. Thus, numerous models of varying complexity can be constructed based on application needs and device features. As the sensing ability diminishes as distance increases, we express sensing model S at an arbitrary point p for a sensor s:

$$S(s,p)=\frac{\lambda}{[d(s,p)]^k} \tag{1}$$

Where *d(s, p)* is the Euclidean distance between the sensor *s* and the point *p*, and positive constants λ and *k* are technology- and environment-dependent parameters. Therefore, the closer the target is near to the sensor, the more dangerous its status.

In order to introduce the notion of exposure in sensor fields, we first define the *Sensor Field Intensity* for a given point *p* in the sensor field *F*. Depending on the application and the type of sensor models at hand, the sensor field intensity can be defined in several ways. Here, we present a model for the sensor field intensity *Nearest-Sensor Field Intensity (*I_N*)*. Nearest-Sensor Field Intensity $I_N(F\ ,p)$ for a point *p* in the field *F* is defined as the sensing measure at point *p* from the nearest sensor in *F,* i.e. the sensor that has the smallest Euclidean distance from point *p*. I_N can be expressed as:

$$\begin{aligned} s_{\min} &= s_m \in S \mid d(s_m, p) \le d(s, p)\ \forall s \in S \\ I_N(F, p) &= S(s_{\min}, p) \end{aligned} \tag{2}$$

where s_{min} is the nearest sensor to *p*.

Suppose a target *T* is moving in the sensor field *F* from point $p(t_1)$ to point $p(t_2)$ along the path *p(t)*. The *Exposure* for a target *T* in the sensor field during the interval $[t_1,t_2]$ along the path *p(t)* is defined as:

$$E(p(t), t_1, t_2) = \int_{t_1}^{t_2} I(F, p(t)) \left| \frac{dp(t)}{dt} \right| dt \tag{3}$$

Since $p(t) = (x(t), y(t))$,

$$\left| \frac{dp(t)}{dt} \right| = \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} \tag{4}$$

2.2 Length

The length of the path is the total Euclid distance of adjacent point on the path when a target *T* is moving in the sensor field *F* from point $p(t_1)$ to point $p(t_2)$ along the path *p(t)*. The length of the path is defined as:

$$L(p(t), t_1, t_2) = \int_{t_1}^{t_2} \left| \frac{dp(t)}{dt} \right| dt \tag{5}$$

3 Path-Finding Algorithm

Assume that a sensor network is pre-deployed to establish field surveillance by sensing existence and activities of any target. Although sensors carry on tasks cooperatively with each other, a target behaves *adversely* with the sensor network. An intelligent target, equipped with an omni-directional detector, attempts to detect the immediate environment it resides, such as finding out where the sensors are and what are their sensing capacities. We use detection probability to express the ability of the target. Practically the target can detect the sensor nearby with high probability than remote one. Define the distance from the target t to a point p as *d(t,p)*, and detection probability of t at p as *P(t,p)*. *P(t,p)* generally diminishes as *d(t,p)* increases. The specific function of *P(t,p)* depends on the type of target. A Gaussian function can model most generic targets.

$$P(t,p) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d(t,p)^2}{2\sigma^2}} \tag{6}$$

where $\sigma > 0$, and is a manufacture-dependent parameter, and it denotes the detection ability of the target. *P(t,p)* represents the probability the target detect the sensor at point of p. Without loss of generality, we use it as the detection function of target in our simulation.

3.1 Voronoi Diagram

In order to reduce the complexity when finding traversing path with less exposure and shorter length, we resort to some technique of partitioning 2-D plane. In the previous works [7, 8], the authors make use of grid to approximate exposure integrals. Here, we utilize Voronoi diagram, which has been re-invented, used, and studied in many domains. According to [10], the Voronoi diagram is a fundamental construction defined by a discrete set of sites. The Voronoi diagram partitions the plane into a set of convex polygons such that all points inside a polygon are nearest to only one site. The boundary segment of a Voronoi region is called the *Voronoi edge*. The intersection point of two or more Voronoi edges is called the *Voronoi vertex*. The path of the least exposure with respect to Nearest-Sensor Field Intensity exactly lies on the polygon edges. This prompts us to find the least exposure path traversing along the Voronoi polygon edges. Fig 1(a) shows an example of the Voronoi diagram for a randomly placed sensor network.

3.2 LVD and Its Update

An intelligent target tries to find out the sensors nearby with probability as Formula (6), and it can detect the sensor if the probability is larger than a threshold. Furthermore the detection probability is independent of direction because of an omnidirectional detector. In this work, we denote the range of successful detection above the threshold as *detection radius* (r_D). r_D is usually longer than sensors can detect, but it is relatively smaller than the diameter of the sensing field. Then, it reacts by hiding itself or finding the best path to its destination based on some criteria, such as minimal exposure and/or shorter path. Due to the role of adversary, a target cannot expect information relay and gathering from surrounding sensors. Instead, such a target can only rely on the partially available information in its detection scope. Thus, the target can only figure out a partial imprecise Voronoi diagram based on those exposed sensors. We call this partial Voronoi diagram *Local Voronoi Diagram* (*LVD*). As shown in Fig 1 (b), the target, represented as the pentagon in the centre of the circle, computes LVD of the eight sensors exposed. Note that the LVD is different from the corresponding part of the *Complete Voronoi Diagram* (*CVD*).

When the target moves along the polygon edge, it may detect more newly exposed sensors. Thus it needs to update LVD at time and reevaluate the condition of monitoring. According to the properties of Voronoi diagram [11], the update of LVD includes adding the peripheral polygon edges when new sensors are detected in the circumjacent region. As shown in Fig 2, if the target (the pentagon) moves to a new location along the arrow, it detects S_7 and modifies LVD by adding two dotted lines as polygon edges. At the same time, the target loses sight of S_0, S_4, S_5 and S_6. However, the information can still be retained if needed.

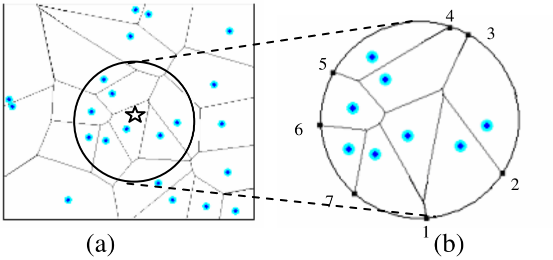


Fig. 1. Voronoi diagram of a set of sensors in a plane

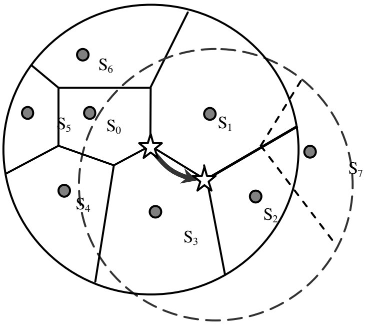


Fig. 2. An example of LVD update

3.3 Base Path-Finding Framework

In this section, we investigate a path-finding approach based on LVD. According to the density of sensors, there are three situations to be considered. First, if a target does not detect any sensor, there is no LVD to construct. So the exposure of any path is zero. Apparently, the path can be a straight line from the current site to its destination. Next, if the target detects only one sensor, the moving direction should change to the perpendicular of the line from the current site to the sensor. Otherwise, if the target detects more than one sensor, it needs to construct a local Voronoi diagram, to evaluate the exposure and to select a favorite path to traverse.

The path-finding algorithm is described as follows. First, the intersecting points of polygon edges and the LVD contour are denoted as Vertex Candidates (*VCs*). For example, there are seven VCs as shown in Fig 1 (b). Second, we compute the exposure and length along the path from the center to these VCs, based on which the best path can be determined. Finally, the target is moving towards the Selected Vertex (*SV*). During the move, upon detection of any newly exposed sensor, the procedure will be repeated. The base path-finding algorithm is described in Fig 3.

```
Base path-finding algorithm
a) Let $s$ be the source and $d$ the destination; let $V_{cur} = s$
b) Loop while $V_{cur} \neq d$
c)   Compute a LVD centered at $V_{cur}$ ;
d)   Search all intersecting points of polygon edges
       and LVD contour as $VC_i$, (i=1,…,n)
e)   For each Voronoi edge within LVD, let
       weight(e) be the least exposure of that particu-
       lar line segment with $I_N$ used as its sensing in-
       tensity function.
f)   For each $VC_i$ , compute $\Psi_i$:
       i) calculate $Proximity_i$ at $VC_i$;
       ii)find the least exposure path, $P_{minE}$, from $V_{cur}$ to
           $VC_i$ with Dijkastra's single-source shortest
           path algorithm, based on weight(e)
       iii) compute $\Psi_i = Proximity_i / (\mu*Length_i+Exposure_i)$
              where, $Exposure_i = \sum$ weight(e), $e \in P_{minE}$
                $Length_i = \sum len(e), e \in P_{minE}$
g)   Decision-making: Select $SV=VC_k$, where $\Psi_k \geq \Psi_i$
       for all $1 \leq i \leq n$.
h)   Move the target towards SV along the selected path
       until detection of newly exposed sensor; record
       the location as a new $V_{cur}$, go to step (b);
i) End of loop
```

Fig. 3. Pseudo-code of the base path-finding algorithm

In this algorithm, *proximity* is used to measure the relative closeness to the destination. The function to compute proximity will be defined later. Variable Ψ_i is introduced as a priority indicator for VC_i to be selected as the next point to move. Here we define Ψ_i as:

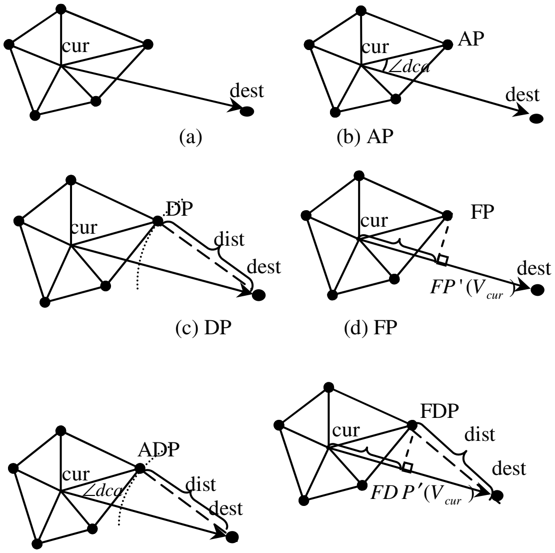
$$\Psi_i = \frac{Proximity_i}{\mu * Length_i + Exposure_i} \tag{7}$$

where the length is normalized by the field dimension; and μ is a regulative parameter and its influence on performance of the path selection will be discussed later.

3.4 Proximity Definition

In the base path-finding framework above, we need to compute the proximity of VCs. In this section we propose five different functions defined to compute proximity to be used in the path-finding algorithms.

Suppose the target starts from the current vertex (V_{cur}), and goes to the destination vertex (V_{dest}), see Fig 4 (a). We can express the forwarding direction contribution with three basic schemes: angle, forwarding length, and distance to the destination. Thus we introduce three measures of proximity as Angle Preference (*AP*), Distance Preference (*DP*), and Forward Preference (*FP*), as shown in Fig 4 (b), (c), and (d), respectively. We also introduce two hybrids of these preferences: *ADP* as the combination



(a) (b) AP

(c) DP (d) FP

(e) ADP (f) FDP

Fig. 4. Definitions of various algorithms

Table 1. Definition of Proximity function

Proximity	Definition of the proximity function
AP	$\Pr oximity = \cos(\angle dca)$, where $\angle dca$ is measured in both clockwise and counter-clockwise directions.
DP	$\Pr oximity = 1/dist(DP(V_{cur}), V_{dest})$, where *dist (p,q)* denotes the Euclidean distance between *p* and *q*. And $DP(V_{cur})$ is the distance preferential vertex of V_{cur} .
FP	$\Pr oximity = dist(V_{cur}, FP'(V_{cur}))$, where $FP'(V_{cur})$ is the projection of $FP(V_{cur})$ on segment V_{cur} to V_{dest}. And $FP(V_{cur})$ is the forward preferential vertex of V_{cur} .
ADP	$\Pr oximity = \cos(\angle dca)/dist(ADP(V_{cur}), V_{dest})$, where $ADP(V_{cur})$ is the angle and distance preferential vertex of V_{cur} .
FDP	$\Pr oximity = dist(V_{cur}, FDP'(V_{cur}))/dist(FDP(V_{cur}), V_{dest})$ where $FDP'(V_{cur})$ is the projection of $FDP(V_{cur})$ on segment V_{cur} to V_{dest}. And $FDP(V_{cur})$ is the forward and distance preferential vertex of V_{cur} .

of AP and DP, and *FDP* as the combination of FP and DP, as shown in Fig 4 (e) and (f). The definition of all these proximities is summarized in Table 1.

The algorithms described above borrow ideas from the corresponding geographical routing algorithms [12, 13, 14]. AP is similar to compass routing, DP greedy routing and FP most-forwarding routing. Thus they have the similar properties. Morin [13] proved that greedy routing guarantees the target moves to the destination if the underlying structure is Delaunay triangulation. The compass routing guarantees reach of the destination if the topology is Delaunay triangulation [14]. The most-forwarding routing works for all triangulations, i.e. it guarantees the target move to the destination as long as a triangulation is used as the underlying structure [12]. Based on analysis above, the notion of proximity based on LVD provides guarantees on the feasibility of our framework.

4 Experimental Results

In order to gain a better understanding of LVD-assisted traversing, we have performed a wide range of simulation studies. In this section, we present several interesting results and discuss their implications and possible applications.

The main simulation platform is written in C++. The visualization and user interface elements are currently implemented with Visual C++ and OpenGL libraries. Network Simulator (NS2) and CrossBow® $MICA_Z$ sensor nodes are also used to verify the sensing models and the qualitative performance of the exposure model in a realistic environment. The sensor field in our experiments is defined as a 500 * 500 square. N sensor nodes are randomly deployed in the region.

For simplicity, parameters k and λ of sensing model in Formula (1) are set to 2 and 1, respectively. Suppose an intelligent target starts at midpoint of the left edge and monitored by the networked sensors. It searches for traversing path with our algorithms to its destination, the midpoint of the right edge. As shown in Fig 5, we calculate the traversing path with AP, FP, DP, ADP, and FDP, respectively. All methods are based on locally available information.

The target selects the path based on the information within the range of r_D, which has a great impact on the performance of algorithms. We conducted 50 independent trials and the outcome is averaged. Shown in Fig 6 (a), when r_D is zero, a target is not aware of any sensor deployed, therefore, has no choice but moving straightforward to the destination. With r_D increases, the target is able to obtain more information, and can choose a better traversing path. When r_D increases over 550 ($\approx 250\sqrt{5}$) or so, it can find the best traversing path due to available global information.

Fig 6 (a) also illustrated that the exposure of all algorithms decreases as r_D grows, and the curve becomes smooth when r_D adds up to 300 or so. This phenomenon can be due to the fact that the target can detect most part of the sensors' deployment when r_D reaches about half of the field edge. Besides, FDP exhibits a significantly lower exposure.

Different from the exposure, the length increases proportionally to r_D, as shown in Fig 6 (b). This is because of the fact that the target would like to take a longer path to steer clear of the close-by sensors. The length of all algorithms is unit when r_D is zero. Then the length grows as r_D augments from zero. Among the five algorithms with incomplete information, FDP performs better than the other four.

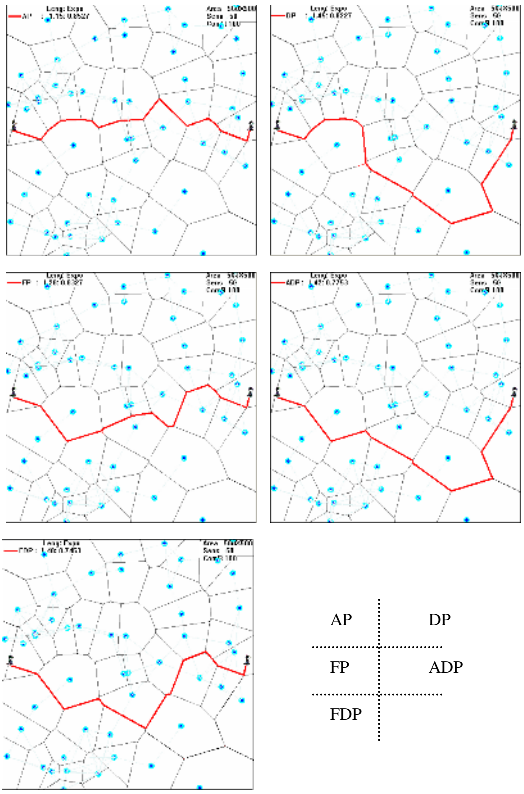
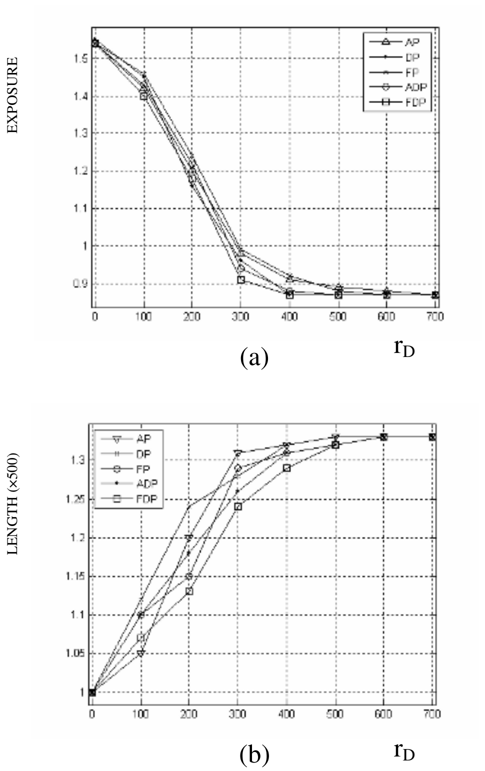


Fig. 5. A snapshot of the experimental result (N=50, r_D=320, μ=0.5)

As to path-selecting criterion of Formula (7), we examine on performance impact with variable parameter μ. Given the same deployment of the sensors, we perform the traversing procedure with different values of μ. Empirically, μ has little influence on a sparse-deployed sensor field, therefore, 110 sensors are placed in the region. We conduct fifty trials with different settings of μ. And the averaged exposure and length are presented in Fig 7. It suggests that if μ is set larger, the selected path emphasizes a shorter length but has a larger exposure. If μ is set near zero, a path with less exposure becomes a favorite. This behavior may be explained that the target makes a detour to keep off sensors. Fig 7 shows there is a trade-off between the exposure and the length. And parameter μ can be used as an adjustor to adapt to needs of real applications.

Suppose to deploy more sensors randomly in the region of 500*500, the density of sensor nodes really has an influence on path-finding algorithms too. Generally for sparse fields, there are a wide range of least exposure paths that can be expected from uniform random deployments. As sensor density increases in the field, the least exposure value and path length tend to stabilize as Table 2 shows, including the average, standard deviation (*SD*) and relative standard deviation (*RSD*) of exposure and length for 50 such cases. This effect can be observed from the diminishing RSD as the number of sensor increases. The results suggest that there is a saturation point after which randomly placing more sensors does not significantly impact the least exposure in the field. In our experiments we have observed that as the number of sensors increase, the least exposure path generally gets closer to the bounding edges of the field, and the path length approaches the half field perimeter value. This behavior is caused by the fact that sensors are only allowed to exist in the field and thus the boundary edges of the field are generally farther from the mass of sensors.



(a)

(b)

Fig. 6. Performance vs. Detection Radius (N=50, μ=0.5)

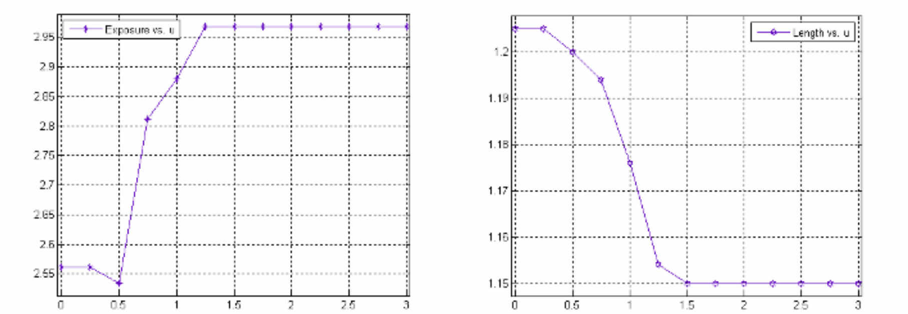


Fig. 7. FDP Performance vs. μ (N=110, r_D=320)

Table 2. Path-selecting Performance vs. N (FDP, r_D=320, μ=0.5)

Sensors	Exposure			Length		
	Avg.	SD	RSD	Avg.	SD	RSD
10	0.18	0.1041	57.83%	1.17	0.1389	11.87%
30	0.42	0.0778	18.52%	1.20	0.1261	10.50%
50	0.80	0.1637	20.46%	1.26	0.1659	13.16%
70	1.34	0.1674	12.49%	1.23	0.0963	7.83%
90	1.62	0.2646	16.33%	1.20	0.0734	6.12%
110	2.66	0.4026	15.13%	1.28	0.0927	7.24%

5 Conclusion and Future Work

In this paper, the typical behavior of an intelligent target has been thoroughly studied. We focus on how an intelligent target traverses the monitoring region. With modeling of this anti-detection problem, we propose and evaluate a number of path-finding algorithms. Different from the previous works, we investigate the case with a realistic assumption. That is, the target only can detect part deployment of the sensor network. Experimental results showed the effectiveness of the path-finding algorithms. These path-finding algorithms are the enormous menace to the quality of surveillance in sensor networks. Hiding techniques may be a good idea for sensor networks against being detected. The key idea is to hide the sensor network from the scope of intelligent target, like the natural immune system covered on the sensor networks. In the near future, we would like to make effort on hiding technology in each phase – initialization, surveillance and reporting phase respectively.

References

1. A. Mainwaring, "Wireless sensor networks for habitat monitoring," ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA), pp. 88-97,Atlanta, Georgia, USA, 2002
2. A. D. Wood, John A. Stankovic, "Denial of Service in Sensor Networks," IEEE Computer, 35(10), pp. 54-62, October 2002
3. V. Phipatanasuphorn, P. Ramanathan, "Vulnerability of Sensor Networks to unauthorized Traversal and Monitoring," IEEE Transactions on Computers, 53(3), pp. 364-369, March 2004
4. Veltri G, Huang Q, Qu G, Potkonjak M. "Minimal and maximal exposure path algorithms for wireless embedded sensor networks," In: Akyildiz IF, Estion D, eds. Proc. of the ACM Int'l Conf. on Embedded Networked Sensor Systems (SenSys). New York: ACM Press, 2003. 40-50
5. X. Y. Li, "Coverage in Wireless Ad Hoc Sensor Networks," IEEE Transactions on Computers, Vol.52, No.6, pp. 753-763, June, 2003
6. S.Megerian, F. Koushanfar, M. Potkonjak, M. Srivastava, "Worst and Best-Case Coverage in Sensor Networks," IEEE Transactions on Mobile Computing Vol.4, No.1, pp. 84-92, 2005

7. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” IEEE INFOCOM, vol. 3, pp. 1380–1387, April 2001
8. S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, “Exposure in wireless ad-hoc sensor networks,” ACM MobiCom, pp.139-150, July 2001
9. A. Savvides, “Location Discovery in Ad-hoc Wireless Sensor Networks,” unpublished, Dept.of EE and CS, UCLA
10. F. Aurenhammer, “Voronoi Diagrams – A Survey of A Fundamental Geometric Data Structure,” ACM Computing Surveys 23(3), pp. 345- 405, September 1991
11. K. Mulmuley, “Computational Geometry: An Introduction through Randomized Algorithms,” Prentice-Hall, 1994
12. P. Bose, P. Morin, “Online routing in triangulations,” the 10th International Symposium on Algorithms and Computation (ISAAC’99), vol. 1741 of LNCS, pp. 113-122, Springer-Verlag, 1999
13. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” ACM/Kluwer Wireless Networks, 7(6), pp. 609-616, November 2001
14. E. Kranakis, H. Singh, and J. Urrutia, “Compass routing on geometric networks,” the 11th Canadian Conference on Computational Geometry, pp.52-54, August 1999.

An Efficient Fibonacci Series Based Hierarchical Application-Layer Multicast Protocol

Jing Li[1,2,3], Naijie Gu[1,2], and Weijia Jia[2,3]

[1] Department of Computer Science and Technology,
University of Science & Technology of China, Hefei, China
[2] Joint Research Lab of Excellence,
CityU-USTC Advanced Research Institute, Suzhou, China
[3] Department of Computer Science,
City University of Hong Kong, Hong Kong, China
jennylee@mail.ustc.edu.cn, gunj@ustc.edu.cn,
itjia@cityu.edu.hk

Abstract. In this paper, an efficient Fibonacci series based hierarchical protocol- HFTM (Hierarchical Fibonacci Tree Multicast) is proposed for application-layer multicast. It adopts the idea of layer and cluster to construct multicast group members into a hierarchical architecture. During the cluster formation, it considers the underlying network properties to reduce packet delivering on costly links. In each cluster, a Fibonacci multicast tree is constructed by recursively partitioning the member sequence into two halves with different length. Moreover, the size of cluster is taken into account in order to obtain a balanced architecture. The considering of underlying network properties and the construction of Fibonacci multicast tree improve the delay performance of the novel protocol. The simulation shows that HFTM is an efficient and scalable application-layer multicast protocol.

1 Introduction

Multicast is an efficient delivery mechanism in one-to-many data transfer applications. Network-layer multicast [1, 2] is a native multicast. It is efficient because of its best-effort multi-point content delivery over the internet. However network-layer multicast is limited to 'islands' of network domains because of its inherent drawbacks [3]. It lacks ubiquitous multicast support among all internet service providers. So network-layer multicast had not been adopted widely in the past decade.

At first, some alternative schemes [4, 5, 6] were proposed to try to address the above problem. But these schemes still depended on routers unavoidably. In this situation, application-layer multicast was proposed [7, 8, 9, 10, 11].

Application-layer multicast is a different kind of multicast. It builds a multicast architecture by having the end hosts self-organize into logical overlay networks. It offers multicast function by unicasting on underlying links. As shown in Figure 1, one of the dominant differences between application-layer multicast and network-layer multicast is that the multicast packets are replicated and forwarded by end hosts instead of intermediate routers. It demonstrates that excessive identical packets induced

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 131–142, 2006.

by end host replicating and forwarding will occupy network resources. Therefore one motivation is to build appropriate topologies to improve the scalability and the efficiency of application-layer multicast.

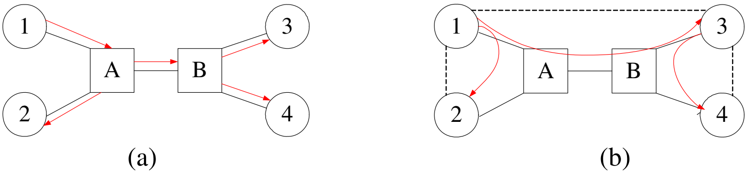


Fig. 1. (a) Network-layer multicast; (b) Application-layer multicast. Node 1, 2, 3 and 4 are end hosts. Node A and B are routers. The dotted lines represent peers on the overlay.

According to overlay topology design, current proposed application-layer multicast protocols can be classified into three flavors: the mesh-first, tree-first and implicit application-layer multicast protocols.

Take NARADA [7] and ALMI [8] as examples of mesh-first application-layer multicast protocol. NARADA firstly organizes the multicast group members into a mesh topology and then constructs a spanning tree whose root is the multicast source. To guarantee robust, every multicast group member maintains a state list about all members. But this condition compromises the scalability of NARADA. ALMI constructs a MST (Minimum Spanning Tree) among members which reduces maintenance cost.

YOID [9] is a case of tree-first application-layer multicast protocol. It firstly builds a shared data delivery tree among members. The tree structure is easy to construct and has logarithmic scaling behavior. Its drawback is a direct control over every aspect of tree structure and this will result in high costs. Host Multicast [10] is another case of tree-first application-layer multicast protocol.

NICE [11] and CAN-based multicast [12] is two representatives of implicit application-layer multicast protocol. CAN-based multicast is based on a special infrastructure - Content-Addressable Network (CAN). CAN is an overlay network whose constituent nodes form a virtual *d*-dimensional Cartesian coordinate space. Each member owns its individual distinct zone in this space. The "flooding scheme" is used to multicast packets. Unlike CAN-based multicast, NICE is hierarchical infrastructure. It involves several layers and each layer has a set of clusters. Each cluster has a cluster leader. The cluster members only communicate with each other in their own cluster. So NICE is a scalable protocol. However, it does not consider underlying topology when running hierarchy and clusters. This compromises the delay performance.

Therefore designing an efficient application-layer multicast protocol is still an open problem. We propose a novel application-layer multicast protocol based on Fibonacci series-HFTM (Hierarchical Fibonacci Tree Multicast). It is a hierarchical multicast. Each layer includes some clusters. Each cluster has a cluster leader. The members in each cluster are constructed into a Fibonacci multicast tree by a Fibonacci series based multicast algorithm [13].

The rest of the paper is organized as follows: Section 2 gives the detail of Fibonacci tree. Section 3 provides the detail design of the protocol--HFTM. Simulation evaluations and performance comparisons between NICE, CAN-based multicast and HFTM are provided in Section 4. Section 5 concludes the paper.

2 Fibonacci Multicast Tree

2.1 Detail Design

In HFTM, an innovation is the use of Fibonacci multicast tree. In each cluster, a Fibonacci multicast tree is created by using a Fibonacci series based multicast algorithm. The input of the algorithm is a cluster member sequence constructed with some special means (described in the next section). This algorithm adopts the idea of Fibonacci series to partition the members sequence into parts with different sizes. The Fibonacci series $\{f_i\}$ satisfies the following condition: f_0=0, f_1=1; $f_n = f_{n-1} + f_{n-2}$, *if* $n > 1$. It guarantees the root of Fibonacci multicast tree serves for the members as many as possible.

Algorithm 1 (Fibonacci series based multicast):

```
Input: member sequence $\Phi = \langle d_1, d_2, ..., d_K \rangle$, $d_s$ is the cluster leader which serves as the source node, $d_s \in \Phi$. The number of members in $\Phi$ is $K$. $f_n \le K < f_{n+1}$
Output: a multicast tree constructed for all members in $\Phi$
1   If ( $K = 2$ ) $d_s$ send packets to the only destination;
2   If ( $K > 2$ ) $\Phi$ is partitioned into two subsequences $\Phi_1$ and $\Phi_2$ where $d_s$ is in the larger subsequence and the smaller one includes $f_{n-2}$ members;
    2.1  If ( $s > f_{n-2}$ ) { $\Phi_1 = \langle d_1, d_2, d_3 ..., d_{f_{n-2}} \rangle$ ; $\Phi_2 = \langle d_{f_{n-2}+1}, d_{f_{n-2+2}}, ..., d_K \rangle$ ;}
         Else { $\Phi_1 = \langle d_1, d_2, ..., d_{K-f_{n-2}} \rangle$ ; $\Phi_2 = \langle d_{K-f_{n-2}+1}, d_{K-f_{n-2+2}}, ..., d_K \rangle$ ;}
    2.2  If ( $s > f_{n-2}$ ) { $d_s$ firstly sends packets to $d_1$, then $d_1$ is in charge of multicasting in $\Phi_1$ and $d_s$ is in charge of multicasting in $\Phi_2$ ; }
         Else { $d_s$ firstly sends packets to $d_{K-f_{n-2}+1}$, then $d_s$ is in charge of multicasting in $\Phi_1$ and $d_{K-f_{n-2}+1}$ is in charge of multicasting in $\Phi_2$ ;}
3   Multicast packets from $d_1$ to all members in $\Phi_1$ and from $d_s$ to all members in $\Phi_2$ (or multicast packets from $d_s$ to all members in $\Phi_1$ and from $d_{K-f_{n-2}+1}$ to all members in $\Phi_2$) by recursive calls Algorithm 1.
```

Figure 2 shows an example of Fibonacci multicast tree achieved by this algorithm.

2.2 Analysis

In the process of Fibonacci multicast tree building, each member in sequence is processed once. Obviously, the time complexity of tree building is $O(K)$, where K is the number of members in the initial input sequence.

The following is the analysis of time complexity in the process of multicast packets to K members by using the Fibonacci series based multicast algorithm. Several notations used in this section are given first. L denotes the propagation time on an overlay link and t is the packet processing time on each end host. For simplicity, L is supposed to be same for each overlay link. $T(K)$ denotes the time used to multicast a packet from one source to K destinations. Obviously, $T(K)$ is a monotonous increasing function of K. When packet multicast begins, the source firstly sends the packet to the partition node. It takes $L+t$ seconds to deal with the packet. As a result, the initial member sequence is divided into two sub-sequences.

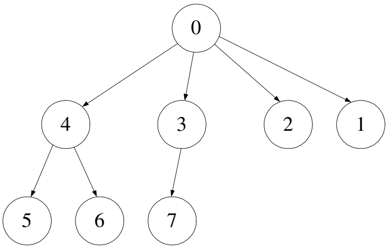


Fig. 2. The multicast tree achieved by the Fibonacci series based multicast algorithm on member sequence $\Phi = \langle 4,6,5,3,7,2,1,0 \rangle \cdot$ $K=8, n=6, f_{n-2}=3$, the source node is node 0.

Because of the recursive character of this algorithm, $T(K)$ is expressed as $T(K)=\max\{T(f_{n-2})+L+t, T(K-f_{n-2})\}$, where $T(1)=0$ and $T(2)=L+t$.

If $f_n \le K < f_{n+1}$, $T(K)=max\{T(f_{n-2})+L+t, T(K-f_{n-2})\}$

$$\le max\{T(f_{n-2})+L+t, T(f_{n+1}-f_{n-2})\}$$

$$\le max\{T(f_{n-2})+L+2t, T(f_{n+1}-2f_{n-2})\}$$

In addition, we have $f_{n+1}-2f_{n-2}=f_{n-1}+f_{n-3}$

From the above equation, it is easily to obtain

$$f_{n-1} \le f_{n+1}-2f_{n-2} < f_n$$

So $T(K) \le max\{T(f_{n-2})+L+2t, max\{T(f_{n-3})+L+2t, T(f_{n-1})\}\}$

$$= max\{T(f_{n-2})+L+2t, T(f_{n-3})+L+2t, T(f_{n-1})\}$$

$$= max\{T(f_{n-2})+L+2t, T(f_{n-1})\}$$

In addition, $T(f_{n+1}) \le max\{T(f_{n-2})+L+2t, T(f_{n-1})\}$

So the following two expressions can be achieved:

$$T(f_{n-1}) \le max\{T(f_{n-4})+L+2t, T(f_{n-3})\}$$

$$T(f_{n-2}) \le max\{T(f_{n-5})+L+2t, T(f_{n-4})\}$$

Then, if $f_n \le K < f_{n+1}$, $T(K) \le max\{T(f_{n-2})+L+2t, T(f_{n-1})\}$

$$\le max\{T(f_{n-5})+2(L+2t), T(f_{n-3})\}$$

$$\le \ldots\ldots$$

$$\le \left\lfloor \frac{n}{2} \right\rfloor \cdot (L+2t)$$

Moreover, $f_n = \left\lfloor \frac{\varphi^n}{\sqrt{5}} + \frac{1}{2} \right\rfloor$, where $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.62$ (proved in [14])

$$\Longrightarrow \quad \log_2 f_n \approx n \cdot \log_2 \varphi + \frac{1}{2}\log_2 5$$

$$n \approx \frac{\log_2 f_n - \frac{1}{2}\log_2 5}{\log_2 \varphi} \approx 1.44(\log_2 f_n - 1.16)$$

In addition, we have $f_n \le K < f_{n+1}$

So we can obtain $n < 1.44(\log_2 K - 1.16)$

$$\Longrightarrow \quad T(K) \le \left\lfloor \frac{n}{2} \right\rfloor \cdot (L+2t) \le \left\lfloor 0.72 \log_2 K \right\rfloor \cdot (L+2t)$$

From the above analysis process, a conclusion is obtained that it takes $O(\log_2 K)$ time to multicast a packet to a member group by using the Fibonacci series based multicast algorithm, where K is the number of members.

3 HFTM Design

The hierarchy and cluster in NICE are proved effective in improving the scalability and efficiency of application-layer multicast. But NICE does not consider the underlying topology characteristic when clustering. This will induce improper partition which wastes resource. Focusing on this problem, HFTM makes some improvements. It introduces a new conception: local area. Local area means that end hosts in it all attach to the same router directly or through several local network components (e.g. the hubs or switches). The following is the detail design of HFTM.

3.1 Hierarchy Design

The multicast members belong to several local areas. In each local area, group members are partitioned into different clusters with the size between k and $3k-1$ where k is a constant. When the number of unassigned end hosts is smaller than k, these remaining members form one cluster. Unlike NICE, we use k=6 in our experiment. Every cluster has a cluster leader. It is achieved by using the probe scheme. The leader has the minimum maximum distance to all other members in the cluster.

The members in each local area are divided into different layers. They all firstly lie in the lowest layer called L_0, then the cluster leaders in layer L_0 compose higher layer L_1. The members in layer L_1 continue to be partitioned into clusters and their leaders go on acting as members in layer L_2 and so on. The hierarchy will continue until the number of members is not larger than $3k-1$. The cluster leader in the highest layer is considered as the core of the local area. Then these cores of local areas are partitioned into clusters and form layers with the same method above. Ultimately, we achieve a final core serve as the root of the whole hierarchical architecture.

3.2 Cluster Member Construction

In NICE, the leader is responsible for delivering the packet to all its members in turn. This means is not very efficient because the delivery can only be performed serially. Unlike NICE, in each cluster, HFTM adopts the Fibonacci series based multicast algorithm to organize the members into a Fibonacci multicast tree which is a shared-tree.

Before the Fibonacci series based multicast algorithm is performed, the members of the cluster should be organized into a member sequence. When members of a cluster all belong to the same local area, the delay distance of logic link from each member to the cluster leader is defined as the weight of these members (these values have been got when selecting leader). Then these members are sorted into a sequence (i.e. the input of the algorithm) with a descending order of the delay distance value. This guarantees that the most time-cost end host is treated first. It makes the total delay performance better. Moreover, if the members of a cluster do not belong to the same local area, the organizing method is different. In this case, in each cluster, member m_i should maintain two parameters: 1) n_i: the number of the members of the local area it belongs to; and 2) d_i: the delay distance of logic link from itself to the cluster leader. The value of n_i has been achieved when selecting leader in local area and is maintained by the core of local area. The value of d_i is achieved by using the same method as above. $n_{\max}$ denotes the maximum of n_i and $d_{\max}$ represents the maximum of d_i in the cluster. Then there is $w_i = \beta \frac{n_i}{n_{\max}} + (1-\beta)\frac{d_i}{d_{\max}}$, where β is a balance factor. We use β=0.4 in our experiment in order to give greater weight to delay distance. The parameter w_i is adopted as the weight of the member m_i. The cluster members are sorted into a sequence with a descending order of w_i value. In decision of member's weight, it considers not only the delay distance but also the size of local area. This avoids the situation that the resulting tree is seriously unbalanced. It is the benefit of HFTM compared with NICE.

3.3 Cluster Maintenance

HFTM employs the refresh messages to follow the membership alterations in the group. The maintenance paths are defined as the overlay paths that carry the refresh messages. Now the comparison of worst-case cluster maintenance overheads created by all the cluster members in HFTM and NICE are stated. The sizes of the largest clusters in HFTM and NICE are $3k_1 - 1$ and $3k_2 - 1$ respectively, and $k_1 = 2k_2$. The worst-case cluster maintenance overheads appear in the largest clusters for both HFTM and NICE. Suppose each refresh message has *r* bits. In NICE, the alteration of an end host should be noticed to all other cluster members. Hence the worst-case cluster maintenance overhead in NICE is $r(3k_2-1)(3k_2-2)$. However, in HFTM, each end host constructs the maintenance paths with its direct upstream member and its direct child members. Therefore the worst-case cluster maintenance overhead in HFTM is $2r(3k_1-1)$. It shows that the maintenance overheads created in HFTM are much less than the ones in NICE. As a result, the less control traffic contributes to a shorter delay to multicast the data packets in the HFTM.

***New Host Joins*:** A set of Rendezvous Points *RPS* are supposed to exist in the HFTM protocol. When a new end host *v* wants to join in the group, it uses the DNS of *RPS* to find the closest Rendezvous Point RP_r and sends it JOIN REQ. RP_r then searches the closest local core (say l_a) to *v* in its local core list and informs it the *v*'s join. l_a then checks all its member lists of clusters in all layers which it joins in order to find the closest cluster member to *v*. In NICE, the new end host needs to enquire each closest member in each layer of the hierarchy architecture. The existence of local core lists achieves the less maintenance traffic created in HFTM than NICE. After new end host joins the right cluster, the number of cluster members is increased by one denoted by K_{new}. The leader will make some modifications depending on the value of K_{new}. The old number of members is represented by K and there exists $f_n \le K < f_{n+1}$. We will simply build a direct link between the new end host and the leader until $K_{new} > f_{n+1}$. Once there is $K_{new} > f_{n+1}$, new leader should be chosen among all current cluster members including new joining end hosts and a new Fibonacci multicast tree is constructed.

***Member Departure*:** When a host H leaves the multicast group, it sends a *Remove* message to the leader. The leader then computes the current number of member denoted by K_{new}. When H is leader or $K_{new} < f_n$, new leader is selected first and then the Fibonacci tree is reconstructed for current cluster members. Otherwise, H is just deleted from the member sequence.

3.4 Architecture Analysis

The analysis of HFTM architecture is given by formulating it. Let the multicast group with n members be $G=\{g_0,g_1,\ldots,g_i,\ldots,g_{n-1}\}(i\in[0,n-1])$ and the source set with $|S|$ sources be $S=\{s_0,s_1,\ldots,s_j,\ldots,s_{|S|-1}\}(j\in[0,|S|-1])$. Denote the overlay path between the source s_j and the group member g_i as $\langle s_j-g_i\rangle$. The set of packet forwarders with $|F|$ forwarders on $\langle s_j-g_i\rangle$ is denoted as $F=\left\{f_0,f_1,\ldots f_m,\ldots,f_{|F|-1}\right\}(m\in[0,|F|-1)$. Suppose the overlay path $\langle s_j-g_i\rangle$ covers L underlying physical links. The delay experienced by a packet p at the time t through the path $\langle s_j-g_i\rangle$ is $d_{\langle s_j-g_i\rangle}(p,t)=\sum_{l=0}^{L-1}d_l^{pr}+\sum_{l=0}^{L-1}d_l^{tr}(p,t)+\sum_{m=0}^{|F|-1}d_{f_m}^{e}(p,t)$, where d_l^{pr} is the propagation delay of the *l*-th physical link, $d_l^{tr}(p,t)$ is the packet delay in the *l*-th physical link at the time *t* that includes the transmission delay in the link and the packet queuing delay at the router connecting to the link, and $d^e_{f_m}(p,t)$ is the packet processing delay (i.e., the delay to replicate and forward the packet) at the end host f_m. The average multicast delay of packet p from the source s_j at the time t is $\overline{d_{s_j}(p,t)} = \frac{\sum_{i=0,g_i \neq s_j}^{n-1} d_{\langle s_j - g_i \rangle}(p,t)}{n-1}$. In order to achieve the short average multicast delay, we can decrease $\sum_{l=0}^{L-1} d_l^{pr}$, $\sum_{l=0}^{L-1} d_l^{tr}(p,t)$ and $\sum_{m=0}^{|F|-1} d^e_{f_m}(p,t)$.

The HFTM protocol makes as many local area links as possible to transmit the multicast packets which contribute to a shorter $\sum_{l=0}^{L-1} d_l^{pr}$. In addition, the data transferring on the costly links between different local areas is reduced so that more network resource is reserved for data traffic. Moreover, the hierarchy and cluster divides the whole group into several “mini-groups”. It greatly decreases the periodical group maintenance traffic among group members that belong to different “mini-groups” and therefore $\sum_{l=0}^{L-1} d_l^{tr}(p,t)$ is decreased. Because the cluster members are constructed into a Fibonacci multicast tree, the height of the whole HFTM hierarchical architecture is decreased than that of NICE. As a result, the number of forwarders on the path $\langle s_j - g_i \rangle$ can be reduced so that $\sum_{m=0}^{|F|-1} d^e_{f_m}(p,t)$ is decreased. Hence the average end-to-end delay in HFTM is proved to be decreased than that in NICE.

Moreover, the case of the asymmetrical distribution of group members among local areas is also considered. It is improved by taking the size of local area into account.

4 Simulation Evaluation

4.1 Model Design

We use the well-known tool NS-2 [15] to accomplish our simulation experiments. The backbone network in the simulation is shown in Figure 3. It is the well-known MCI ISP backbone that is a representative backbone model of Internet used in many researches. The backbone includes 19 routers. The end hosts in multicast group are connected to routers directly or indirectly. In the simulation, the bandwidth of links in the backbone network is 1000Mbps and the bandwidth of links in the local area is 100Mbps. The cost of each link in backbone network is a random integer between 20 and 40. The cost of each link in the local domains is a random integer between 1 and 4. The simulation traffic is the 1.5Mbps MPEG-1 video streams.

In the simulation, performance comparisons are given among three application-layer multicast protocols: NICE, CAN-based multicast and HFTM. To facilitate the comparison, the following classical metrics are adopted: 1) Average End-to-end Delay (AED): it is the ratio of the sum of end-to-end delay from a multicast source to each group member to the number of group members; 2) Average Link Stress (ALS): it is the ratio of the sum of times that identical packet copies traverse over the underlying links to the number of links in the group; 3) Average Cost Stretch (ACS): it is the ratio of the sum of the consuming costs of packets multicasting to all the members to the sum of group members.

4.2 Comparison Results

Two simulations are done to compare the performances of different protocols along the AED, ALS and ACS metrics. The first simulation is to observe the AED, ALS, ACS performances of different protocols when the number of group members varies from 50 to 600 and the number of sending sources is 1.

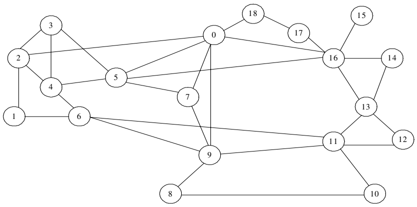


Fig. 3. The experimental MCI ISP backbone network used in the simulation

Figure 4 illustrates the comparison of AED performances of the three application-layer multicast protocols. The curves show that the flooding routing in CAN-based multicast incurs much longer AED than the other two protocols. HFTM achieves better AED performances than NICE because of the architecture characters of HFTM. The introduction of local area reduces the packet delivering along the costly links across different local areas. In addition, the size of cluster in HFTM is larger than in NICE. So it decreases the height of the whole tree architecture. Hence, the existence of the local area and the Fibonacci multicast tree in HFTM results in its better AED performance.

Figure 5 illustrates the ALS performances of the three protocols. HFTM achieves better ALS performances than NICE. In NICE, the packets are distributed to each cluster member by the cluster leaders. The outgoing links of cluster leaders in some higher layers usually suffer from large link stress. However, in the HFTM, the burden of cluster leaders is shared by other cluster members through the use of the Fibonacci multicast tree. Hence, the ALS of HFTM is less than NICE. In CAN-based multicast, the flooding routing is used to multicast packets to group. This flooding routing scheme evenly distributes the data traffic and link stress among all the links. It enables CAN-based multicast to achieve the smallest ALS.

Figure 6 gives the ACS performance comparison of the three protocols. The curves have similar trends with the AED curves in Figure 4. The curves reflect the consuming cost of each member of the three protocols. For the same reasons as above, the ACS of HFTM is the lowest one in the three protocols and the ACS of NICE is less than CAN-based multicast. Moreover, the designs of NICE and CAN-based multicast neglect the underlying network properties. It results in the packet delivering transferring more costly links and some redundant links in the two protocols. HFTM reduces the consuming cost of each member by considering the underlying network properties. The ACS performance of HFTM is the best because of its architect characters.

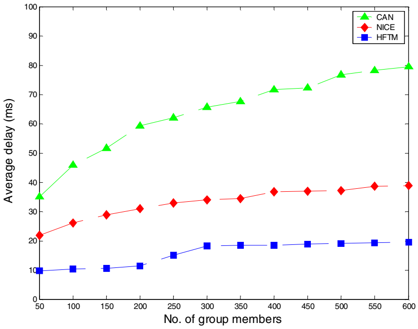


Fig. 4. The AED performances of the three protocols with the increasing number of members in the group

In the second simulation, the AED, ALS and ACS performances of the three protocols are observed under the situation that the number of the sending source varies and the number of group members is always 100. The additional sources are determined randomly.

Figure 7(a) illustrates the AED performances of the three protocols when the number of sending source varies from 1 to 10. The relation of the curves in Figure 7 is similar with that in the Figure 4. The flooding scheme leads the CAN-based multicast to create the lowest AED performance. The cluster construction and the consideration of the underlying network properties in HFTM make that the data packets take less delay to reach group members than in NICE.

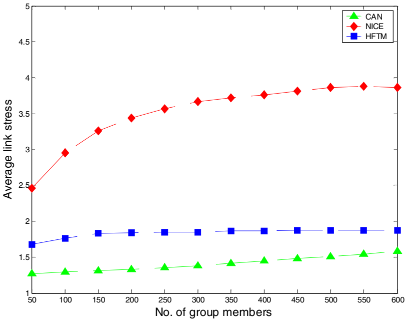


Fig. 5. The ALS performances of the three protocols with the increasing number of members in the group

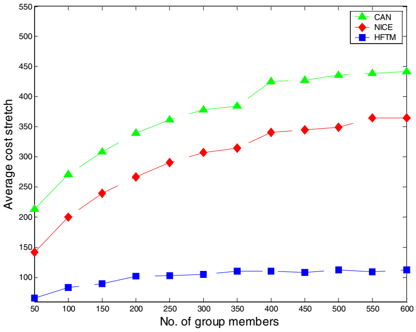
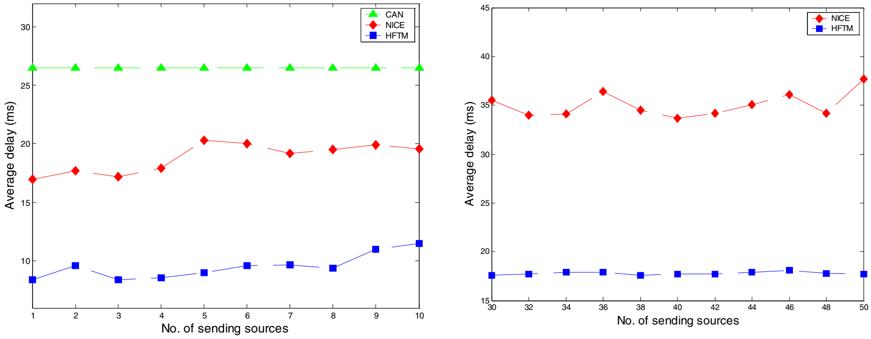


Fig. 6. The ACS performances of the three protocols with the increasing number of members

Figure 7(b) illustrates the AED performances of the two protocols when the number of sending sources varies from 30 to 50. Because the value of CAN-based multicast in this case is very large, its curve is not plotted in Figure 7(b). The simultaneous traffic sent by all sources cause the heavy traffic in the network. In NICE, more cluster leaders suffer from the heavy traffic because it neglects the underlying network properties. However, in HFTM, the existence of Fibonacci tree distributes some traffic load to more members.



(a) (b)

Fig. 7. The AED performances with the increasing number of sending sources. There are 100 members. (a)the number of sources varies from 1 to 10;(b) the number of sources varies from 30 to 50.

In the second simulation, the comparison of link properties of the three protocols is shown in Table 1. The figures illustrate that HFTM is more efficient and scalable in term of ALS than NICE. Moreover, in HFTM, few links are used than in CAN-based multicast that makes more resources are reserved for the data transmission.

Table 1. The comparison of link properties of the three protocols

Protocol	ALS	No. of actual links used
CAN-based multicast	1.464	197
NICE	3.122	130
HFTM	1.854	123

The simulations illustrate that HFTM is the most efficient one in term of AED metric among the three protocols. Its delay performance can satisfy the demands of real-time streams. Moreover, its ALS performance is better than that of NICE. Hence, the integrated performance of HFTM is better than the other two protocols.

5 Conclusion

Current application-layer multicast protocols have some inherent drawbacks. Focusing on the performance limitations of NICE, a novel application-layer multicast protocol based on the Fibonacci series–HFTM is proposed. In HFTM, a hierarchical architecture is constructed by partitioning the group members into several clusters and layers. It adopts the Fibonacci series based multicast algorithm to organize cluster members. In addition, the underlying network properties are considered when constructing clusters. It alleviates the problem of redundant packet transferring on costly links. For the above characters of HFTM, its delay performance is improved. The simulation results show that HFTM has its benefits compared with traditional solutions and it is an efficient and scalable application-layer multicast protocol.

Acknowledgements

The research described here is supported by the National Basic Research Fund of China ("973" Program) under Grant No.2003CB317003, and has been benefited from various discussions among the group members of the Joint Research Lab between CityU and USTC in their advanced research institute in Suzhou (China).

References

1. S. Deering and D. Cheriton.: Multicast Routing in Datagram Internetworks and Extended LANs. ACM Transactions on Computer Systems (May 1990)
2. S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, L. Wei.: The PIM Architecture for Wide-Area Multicast Routing. IEEE/ACM Transactions Networking (December 1997)
3. C. Diot, B.N. Levine, B. Lyles, H. Kassan, D. Balensiefen.: Deployment issues for the IP multicast service and architecture. IEEE Networks Spec (2000)
4. R. Perlman, C. Lee, A. Ballardie, J. Crowcroft, Z. Wang, T. Maufer, C. Diot, J. Thoo, M.: Green, Simple multicast: a design for simple, low-overhead multicast. IETF draft, draft-perlman-simple-multicast-03.txt (October 1999)
5. H. Hoolbrook, D. Cheriton.: IP multicast channels: EXPRESS support for large-scale single source applications. Proc. ACM SIGCOMM (September 1999)
6. H. Hoolbrook, B. Cain.: Source specific multicast. IEFT draft, Holbrook-ssm-00.txt (March 2000)
7. Y.H. Chu, S.G. Rao and H. Zhang.: A case for end system multicast. Proc. of ACM SIGMETRICS (June 2000)
8. Pendarakis, S. Shi, D. Verma, and M.Waldvogel.: ALMI: An Application Level Multicast Infrastructure. The 3rd Usenix Symposium on Internet Technologies and Systems (USITS 2001), San Francisco (March 2001)
9. P. Francis.: Yoid: Extending the multicast internet architecture. White paper http://www.aciri.org/yoid/ (April 2000)
10. B. Zhang, S. Jamin and L. Zhang.: Host multicast: A framework for delivering multicast to end users. Proc. of IEEE INFOCOM (June 2002)
11. S. Banerjee, B. Bhattacharjee and C. Kommareddy.: Scalable application layer multicast. Proc. of ACM SIGCOMM (August 2002)
12. S. Ratnasamy, M. Handley, R. Karp and S. Shenker.: Application-level multicast using content-addressable networks. Proc. of NGC (2001)
13. N.J. Gu, W. Li and J. Liu.: Fibonacci Series-based Multicast Algorithm. Chinese computers, Vol. 25 No.4 (Apr 2002)
14. R.M. Karp, A.Sahay *et al.*.: Optimal broadcast and summation in the Log*P* model. Proc the 5th Annual ACM Symposium on Parallel Algorithms and Architectures, Velen, Germany (1993)
15. UC Berkeley, LBL, USC/ISI, and Xerox PARC.: ns Notes and Documentation.(October 1999)

Maximizing the Probability of Delivery of Multipoint Relay Broadcast Protocol in Wireless Ad Hoc Networks with a Realistic Physical Layer[*]

François Ingelrest and David Simplot-Ryl

IRCICA/LIFL, University of Lille 1.
CNRS UMR 8022, INRIA Futurs, France
{Francois.Ingelrest, David.Simplot}@lifl.fr

Abstract. It is now commonly accepted that the unit disk graph used to model the physical layer in wireless networks does not reflect real radio transmissions, and that the lognormal shadowing model better suits to experimental simulations. Previous work on realistic scenarios focused on unicast, while broadcast requirements are fundamentally different and cannot be derived from unicast case. Therefore, broadcast protocols must be adapted in order to still be efficient under realistic assumptions. In this paper, we study the well-known multipoint relay protocol (MPR). In the latter, each node has to choose a set of neighbors to act as relays in order to cover the whole 2-hop neighborhood. We give experimental results showing that the original method provided to select the set of relays does not give good results with the realistic model. We also provide three new heuristics in replacement and their performances which demonstrate that they better suit to the considered model. The first one maximizes the probability of correct reception between the node and the considered relays multiplied by their coverage in the 2-hop neighborhood. The second one replaces the coverage by the average of the probabilities of correct reception between the considered neighbor and the 2-hop neighbors it covers. Finally, the third heuristic keeps the same concept as the second one, but tries to maximize the coverage level of the 2-hop neighborhood: 2-hop neighbors are still being considered as uncovered while their coverage level is not higher than a given coverage threshold, many neighbors may thus be selected to cover the same 2-hop neighbors.

1 Introduction

Nowadays, wireless networking has become an indispensable technology. However, the most deployed technology, known as WiFi, is too restrictive, as users must stay near to fixed access points. Therefore, the latter must be sufficiently deployed and correctly configured to offer a good quality of service. Moreover, there exists some more unusual situations where an infrastructure may be unavailable (*e.g.*, rescue areas). The future of this technology probably lies in wireless ad hoc networks, which are designed to be

[*] This work was partially supported by a grant from CPER Nord-Pas-de-Calais/FEDER TAC COM'DOM, INRIA research action IRAMUS and CNRS national platform RECAP.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 143–154, 2006.

functional without any infrastructure. They are defined to be composed of a set of mobile or static hosts operating in a self-organized and decentralized manner, which communicate together thanks to radio interfaces. Hosts may be either terminals or routers, depending on the needs of the system, leading to a cooperative multi-hop routing.

Broadcasting is one of the most important communication task in those networks, as it is used for many purposes such as route discovery (*e.g.*, OLSR [1]) or synchronization. In a straightforward solution to broadcasting, hosts blindly relay packets upon first reception to their neighborhood in order to fully cover the network. However, due to known physical phenomena, this leads to the broadcast storm problem [2]. Moreover, this is a totally inefficient algorithm, because most of the retransmissions are not needed to ensure the global delivery of the packet, and a huge amount of energy is thus unnecessarily wasted. Many other algorithms have been proposed in replacement. Some of them are centralized (a global knowledge of the network is needed), while the others are localized (hosts only need to know their local neighborhood to take decisions). Obviously, the latter better fit to ad hoc networks and their decentralized architecture.

All the proposed broadcast schemes have always been studied under ideal scenario, where the unit disk graph is used to model communications between hosts. In this model, two hosts can communicate together if the distance between them is no more than a given communication radius, and packets are always received without any error. Recently, this model has been highly criticized as it does not correctly reflect the behavior of transmissions in a real environment [3]. Indeed, signal strength fluctuations have a significant impact on performance, and thus cannot be ignored when designing communication protocols for ad hoc and sensor networks. Unfortunately, this has been the case until now for broadcast protocols.

In this paper, we consider the well-known multipoint relay protocol (MPR) [4], used for broadcasting in ad hoc networks, under a more realistic scenario where the probability of correct reception of a packet smoothly decreases with the distance between the emitter and the receiver(s). We thus replace the unit disk graph model by the lognormal shadowing model [5] to simulate a more realistic physical layer, and provide experimental results. As they demonstrate the need for a more suitable algorithm, we also propose several modifications to MPR in order to maximize the delivery ratio of the broadcast packet, while minimizing the number of needed retransmissions. By experimentation, we show that these new versions are much more efficient than the original one under the considered realistic scenario.

The remainder of this paper is organized as follows: we first provide the definitions needed by our models, while in Sec. 3 a detailed description of MPR is proposed. In Sec. 4, we provide an analysis of the behavior of the original algorithm used in MPR with the realistic physical layer. We then describe in Sec. 5 new algorithms that better fit the latter. We finally conclude in Sec. 6 and give some directions for future work.

2 Preliminaries

The common representation of a wireless network is a graph $G = (V,E)$, where V is the set of vertices (the hosts, or nodes) and $E \subseteq V^2$ the set of edges which represents the available communications: there exists an ordered pair $(u,v) \in E$ if the node v is able to physically receive packets sent by u (in a *single-hop* fashion). The neighborhood set $\mathrm{N}(u)$ of the node u is defined as $\{v : (u,v) \in E \vee (v,u) \in E\}$. The density of the network is equal to the average number of nodes in a given communication area. Each node u is assigned a unique identifier (this can simply be, for instance, an IP or a MAC address).

We assume that nodes are aware of the existence of each neighboring node within a distance of 2 hops (we call this a 2-hop knowledge). In ad hoc networks, the neighborhood discovery is generally done thanks to small control (HELLO) messages which are regularly sent by each host. A 2-hop knowledge can easily be acquired thanks to two rounds of HELLO exchanges: nodes can indeed insert the identifiers of their neighbors in their own beacon messages.

In our mathematical model, the existence of a pair $(u,v) \in E$ is determined by the considered physical layer model and depends on several conditions, the most obvious one being the distance between u and v. In the most commonly used model, known as the unit disk graph model, a bidirectional edge exists between two nodes if the distance between them is not greater than a given communication radius R (it is assumed that all nodes have the same communication radius). In this model, the set E is then simply defined by:

$$E = \{(u,v) \in V^2 \mid u \neq v \wedge \mathrm{dist}(u,v) \leq R\}, \tag{1}$$

$\mathrm{dist}(u,v)$ being the Euclidean distance between nodes u and v.

This model, while being well spread, cannot be considered as realistic. Indeed, it is assumed that packets are always received without any error, as long as the distance between the emitter and the receiver is smaller than the communication radius. This totally ignores random variations in the received signal strength, while it was demonstrated that their impact is really significant.

These fluctuations generate erroneous bits in the transmitted packets. If the error rate is sufficiently low, these bits can be repaired thanks to correction codes. However, if it is too high, then the packet must be dropped and a new emission must be done. This supposes the existence of an acknowledgement mechanism (ACK packets) that cannot be used in broadcasting tasks due to the really high number of emitters. Our work thus only relies upon the probability of correct reception, which is influenced by a lot of factors (*e.g.*, power of emission, distance with the receiver(s), presence of obstacles). We suppose that all nodes have the same transmitting radius, so the power of emission does not have to be taken into account here.

To consider the signal fluctuations, we change G into a weighted graph where each edge $(u,v) \in E$ holds the probability $\mathrm{p}(u,v)$ of correct reception between the two nodes u and v. To determine these probabilities, we chose to consider the lognormal shadowing model [3] in our simulations. We used an approximated function $\mathrm{P}(x)$ described in [6]:

$$\mathrm{P}(x) = \begin{cases} 1 - \frac{(\frac{x}{R})^{2\alpha}}{2} & \text{if } 0 < x \leq R, \\ \frac{(\frac{2R-x}{R})^{2\alpha}}{2} & \text{if } R < x \leq 2R, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

α being the power attenuation factor, and x the considered distance. Fig. 1 illustrates this model with $R = 100$ and $\alpha = 4$.

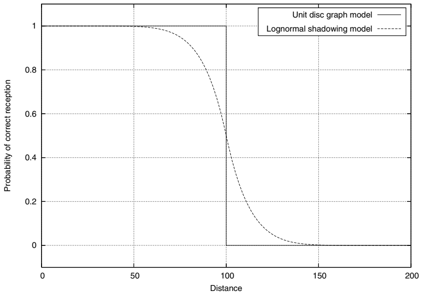


Fig. 1. Unit disk graph and lognormal shadowing models ($R = 100$, $\alpha = 4$)

We assume that each node u is able to determine the probability $\mathrm{p}(u,v)$ of correct reception of a packet that would be sent to a neighbor v. The gain of this knowledge may be simply achieved thanks to beacon messages: based on the quantity of correctly received HELLO packets, v is able to determine an approximated value of $\mathrm{p}(u,v)$. Node v may then include this value in its own beacon messages.

One of the major criticisms of the unit disk graph model is that it does not model the presence of obstacles between nodes. The lognormal shadowing model neither considers them, but we argue that it is sufficient enough for simulations. The most important factor is the weighting of edges by reception probabilities, the method used to distribute the latter is not important to compare protocols in general cases. A realistic model would be mandatory to simulate existing situations and to extract exact values. But in real cases, an obstacle would decrease the probability held by the corresponding edge and would thus be detected by nodes when counting HELLO messages (if such a method is used). This means that in those cases, the broadcast algorithm would use 'real' probabilities and its behavior would be adapted to the situation.

The two previous physical models introduce two different behaviors:

- In the unit disk graph model, one has to maximize the length of each hop so that a single emission is able to reach as many mobiles as possible. The quantity of needed emitters is thus greatly reduced.
- In the lognormal shadowing model, maximizing the length of each hop leads to smaller probabilities of correct reception, but minimizing them leads to a lot of spent energy.

Some papers have already been published about routing in a realistic environment. Amongst them, DeCouto et al. [7] and Draves et al. [8] investigate the question of routing metrics for unicast protocols in wireless networks with a realistic physical layer: the key insight in most of this work is that hop-count based shortest-path routing protocols result in transmissions over long links. While this reduces the hop-count of routes, it also decreases the received signal strength at the receiver of these links, leading to very high loss rates and low end-to-end throughput. These papers also propose other routing metrics which incorporate link-quality (*e.g.*, in terms of error, congestion).

To the best of our knowledge, this paper is the first one to consider broadcasting over a realistic physical layer. Broadcast fundamentally differs from unicast, and leads to a different tradeoff between the length of each hop and the number of relays. Indeed, in a broadcast process, a node can rely on the redundancy introduced by other emitters. Further relays may thus be selected without decreasing the final delivery ratio. This is not possible in routing, as a given emitter is the only one able to transmit the packet to the next hop. The redundancy of broadcasting must be fully considered in order to improve the performance of the underlying protocol.

3 Related Work

As stated in Sec. 1, the easiest method for broadcasting a packet is to have all nodes forward it at least once to their neighborhood: this method is known as *blind flooding*. However, such a simple behavior has huge drawbacks: too many packets are lost due to collisions between neighboring nodes (this can lead to a partial coverage of the network) and far too much energy is consumed. Many other solutions have been proposed to replace it, and an extensive review of them can be found in [9].

Among all these solutions, we have chosen to focus on the multipoint relay protocol (MPR) described in [4] for several reasons:

- It is efficient using the unit disk graph model.
- It is used in the well-known standardized routing protocol OLSR [1].
- It can be used for other miscellaneous purposes (*e.g.*, computing connected dominating sets [10]).

In this algorithm, it is assumed that nodes have a 2-hop knowledge: they are aware of their neighbors (1-hop distance), and the neighbors of these neighbors (2-hop distance). Its principle is as follows. Each node u that has to relay the message must first elect some of its 1-hop neighbors to act themselves as relays in order to reach the 2-hop neighbors of u. The selection is then forwarded within the packet and receivers can thus determine if they have been selected or not: each node that receives the message for the first time checks if it is designated as a relay node by the sender, and if it is the case, the message is forwarded after the selection of a new relaying set of neighbors. A variant exists where nodes proactively select their relays before having to broadcast a packet, and selection is sent within HELLO messages.

Obviously, the tricky part of this protocol lies in the selection of the set of relays $\mathrm{MPR}(u)$ within the 1-hop neighbors of a node u: the smaller this set is, the smaller the number of retransmissions is and the more efficient the broadcast is. Unfortunately, finding such a set so that it is the smallest possible one is a NP-complete problem, so a greedy heuristic is proposed by Qayyum et al., which can be found in [11]. Considering a node u, it can be described as follows:

1. Place all 2-hop neighbors (considering only outgoing links) in a set $\mathrm{MPR}'(u)$ of uncovered 2-hop neighbors.
2. While there exists a 1-hop neighbor v which is the only common neighbor of u and some nodes in $\mathrm{MPR}'(u)$: add v to $\mathrm{MPR}(u)$, remove its neighbors from $\mathrm{MPR}'(u)$.
3. While the set $\mathrm{MPR}'(u)$ is not empty, repeatedly choose the 1-hop neighbor v not present in $\mathrm{MPR}(u)$ that covers the greatest number of nodes in $\mathrm{MPR}'(u)$. Each time

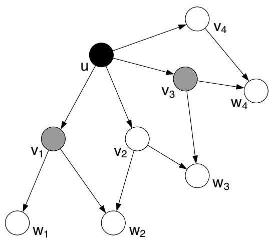


Fig. 2. Applying MPR at node u: $\mathrm{MPR}(u) = \{v_1, v_3\}$

a new node is added to $\mathrm{MPR}(u)$, remove its neighbors from $\mathrm{MPR}'(u)$. In case of tie, choose the node with the highest degree.

An example of this heuristic is given in Fig. 2, starting with $\mathrm{MPR}(u) = \emptyset$. The node v_1 is the only one able to reach w_1, so it is added to $\mathrm{MPR}(u)$ and nodes w_1 and w_2 are removed from $\mathrm{MPR}'(u)$. No other mandatory 1-hop neighbor of u exists, so other relays are selected according to the number of nodes in $\mathrm{MPR}'(u)$ they cover. Nodes v_2 and v_4 cover only one node in $\mathrm{MPR}'(u)$, while node v_3 covers at the same time w_3 and w_4, so v_3 is chosen and added to $\mathrm{MPR}(u)$. The set $\mathrm{MPR}'(u)$ being empty, no other nodes are selected. We finally have $\mathrm{MPR}(u) = \{v_1, v_3\}$.

Being the broadcast protocol used in OLSR, MPR has been the subject of miscellaneous studies since its publication. For example in [12], authors analyze how relays are selected and conclude that almost 75% of them are selected in the first step of the greedy heuristic, so that improving the second step is not really useful. This conclusion seems correct, as long as the unit disk graph model is used.

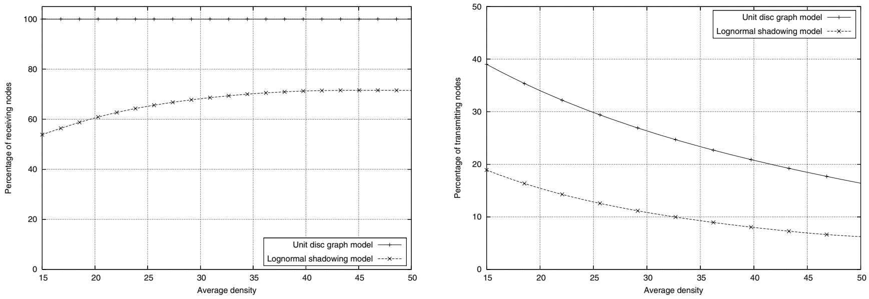
4 Original Greedy Heuristic

4.1 Graphs Generation

In this section, we provide results about the performance of MPR over our considered realistic physical layer, the lognormal shadowing model. We chose not to use a general purpose simulator in order to focus on the area of our study: we thus implemented algorithms and models in our own simulator, so that we had to decide how to generate 'realistic' graphs considering the realistic model.

We chose to consider the method cited in Sec. 2, which is based on HELLO messages. Neighborhood information is stored in a table which is regularly cleaned in order to remove too old entries. An entry is too old when the corresponding host has not signaled itself since a given amount of time, that we denote by x. Beacon messages are regularly sent by each host to signal itself. Let us denote by y the time between two HELLO messages (we have $x > y$). A node u sees a neighbor v if it has received at least one HELLO message during the last y seconds. The probability $\mathrm{p}_n(u,v)$ for this event to occur is equal to:

$$\mathrm{p}_n(u,v) = 1 - \overline{\mathrm{p}(u,v)}^{\frac{x}{y}}. \qquad (3)$$



(a) Receiving nodes.

(b) Transmitting nodes.

Fig. 3. Performance of MPR with the two considered physical models

For each directional edge, a random number is thus drawn to determine if it exists. This way, when a node *u* is aware of the existence of a neighbor *v*, it can decide to send messages to the latter. Of course, *u* cannot be ensured that its messages will reach *v*. We can easily conclude that long edges have a high probability to be unidirectional while short edges have a high probability to be bidirectional.

All the results were obtained with the following parameters. The network is static and always composed of 500 nodes randomly distributed in a uniform manner over a square area whose size is computed in order to obtain a given average density. Edges are created using the method previously described, and for each measure, we took the average value obtained after 500 iterations. We fixed the communication radius to be equal to 75 in both physical models. An ideal MAC layer is considered to isolate the intrinsic properties of the selected relays: collisions of packets could skew both results and analyses.

4.2 Experimental Results

We provide in Fig. 3(a) the delivery ratio of MPR using the two considered physical layers. When using the unit disk graph model, a total coverage of the network is achieved as MPR is a deterministic algorithm. However, this is no more the case with the lognormal shadowing model due to the multiple errors of transmission: the delivery ratio is under 70% for each considered density, and is as low as 55% for a density $d = 15$.

This poor performance can be explained by the fact that, as highlighted by Busson et al. in [12], the chosen relays are located at the limit of the communication range, where the probability of correct reception is low. This is confirmed in our experiments, as illustrated by Fig. 4: the average distance between a node and its multipoint relays is almost equal to 68, while the maximal communication range is 75. Moreover, [12] also states that 75% of the relays are chosen during the first step: this means that, when a relay does not correctly receive the message, there is a risk of 75% that this relay was the only one able to reach an isolated node, which will thus not receive the message, potentially leading to a partition of the network.

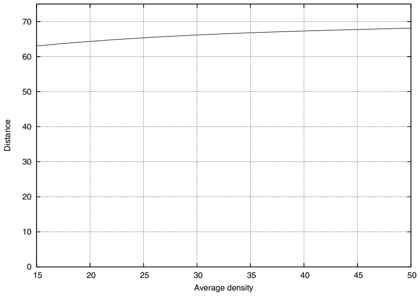


Fig. 4. Average distance between a node and its relays

We also provide in Fig. 3(b) the percentage of nodes which correctly received and then relayed the message. It is interesting to note that this percentage is different with the two models. Indeed, as only nodes which received the message are taken into account, one would have expected to observe the same values in both cases. This means that the needed number of relaying nodes does not linearly vary with the number of covered nodes: obviously, only a few relays are needed to cover a high number of different nodes, but a larger number is needed to cover the last few remaining ones.

5 New Heuristics for MPR

As illustrated in the previous section, the original greedy heuristic used by Quayyum et al. in [4] is not suitable for a realistic physical layer. An average delivery of 70% is indeed not sufficient for most of applications, and an alternative solution must thus be used.

In this section, we propose miscellaneous replacement heuristics in order to improve the performance of MPR. They aim at maximizing the average coverage, while minimizing the number of needed relays (and thus the energy consumption). In all our proposals, the first step of the original heuristic which allows isolated 2-hop neighbors to be covered is kept (it is mandatory), only the second step is replaced.

We keep notations introduced in Sec. 3. Thus, considering a node u, the set $\mathrm{MPR}(u)$ contains the multipoint relays chosen by u, while the set $\mathrm{MPR}'(u)$ contains 2-hop neighbors of u not yet covered.

5.1 First Proposal: Straightforward Approach

As previously explained, the low delivery ratio of MPR is caused by the too high distance between a node and its relays. The latter having little chance to correctly receive the broadcast packet, they also have little chance to be able to relay this packet and thus to cover the 2-hop neighbors of the emitter.

A first and straightforward idea could be, when choosing a relay, to balance the coverage it offers and its probability to correctly receive the packet. Thus, at each step considering a node u, a score can be computed for each potential relay v. The node with

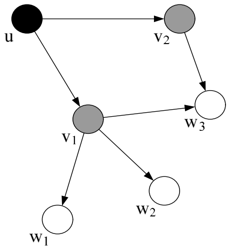


Fig. 5. A case where the node u has to select its multipoint relays between its neighbors v_1 and v_2 ($\mathrm{MPR}(u) = \emptyset, \mathrm{MPR}'(u) = \{w_1, w_2, w_3\}$

the highest score is selected and placed in $\mathrm{MPR}(u)$. We denote by $\mathrm{c}_u(v)$ the *additional* coverage offered by v to u:

$$\mathrm{c}_u(v) = |\mathrm{MPR}'(u) \cap \mathrm{N}(v)|. \tag{4}$$

The score obtained by v at a given iteration for a node u, denoted by $\mathrm{s}_u(v)$, is thus defined by:

$$\mathrm{s}_u(v) = \mathrm{c}_u(v) \times \mathrm{p}(u, v). \tag{5}$$

In simple terms, the additional coverage offered by v is weighted by its probability to correctly receive the broadcast packet. In Fig. 5, the score $\mathrm{s}_u(v_1)$ of v_1 is equal to $3 \times \mathrm{p}(u, v_1)$.

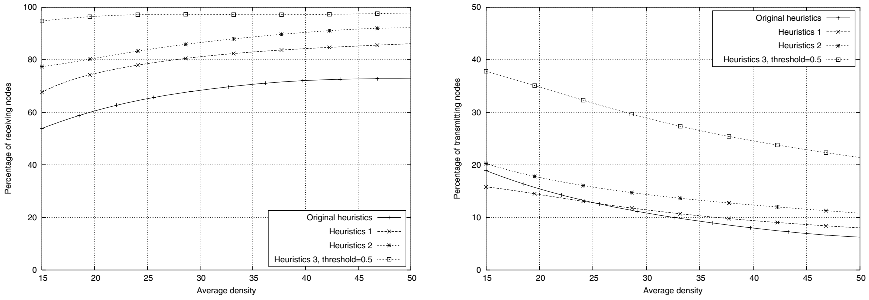
5.2 Second Proposal: Clever Approach

The previous heuristic, while being more suitable for a realistic environment than the original one, still has an obvious flaw: it still takes into account additional coverage in a too simple way. One can thus easily imagine a situation where a very distant 1-hop neighbor would offer an additional coverage such that the latter would compensate a low probability of correct reception. In this case, this neighbor would be selected as relay while its probability to correctly receive the packet, and thus to be able to relay it, would be very low. One can also imagine a situation where the distance between the relay and the 2-hop neighbors it covers would be very high, such that the re-emission of this relay would have little chance to reach these 2-hop neighbors.

We propose to extend the concept used in the first proposal, by taking into account the probabilities of correct reception between the potential relay and the 2-hop neighbors it covers. We thus replace the additional coverage offered by a relay by the average probability of correct reception by 2-hop neighbors. We thus obtain:

$$\mathrm{s}_u(v) = \mathrm{p}(u, v) \times \sum_{i=1}^{i=|\mathrm{c}_u(v)|} (\, \mathrm{p}(v, w_i) \,/\, |\mathrm{c}_u(v)| \,). \tag{6}$$

This way, multipoint relays offering a low coverage in terms of probabilities have little chance to be selected. In Fig. 5, the score $\mathrm{s}_u(v_1)$ of v_1 is now equal to $\mathrm{p}(u, v1) \times ((\mathrm{p}(v_1, w_1) + \mathrm{p}(v_1, w_2) + \mathrm{p}(v_1, w_3))/3)$.



(a) Receiving nodes.

(b) Transmitting nodes.

Fig. 6. Performance of the different heuristics using the lognormal shadowing model

5.3 Third Proposal: Robustness Approach

In the previous proposals, as soon as a 2-hop neighbor has a non-null probability to be covered, it is removed from $\mathrm{MPR}'(u)$. This removal is done even with a very low probability, which in this case may be meaningless. It can be more interesting to consider a 2-hop neighbor as covered when its probability to correctly receive the broadcast packet is over a given threshold, in order to increase the delivery ratio.

We thus propose to keep the score computation used in the previous heuristic, while modifying how 2-hop neighbors are removed from $\mathrm{MPR}'(u)$. For such a 2-hop neighbor w of u, its removal from $\mathrm{MPR}'(u)$ is done only if its coverage level $\mathrm{t}_u(w)$ is over a given threshold. The value of $\mathrm{t}_u(w)$ is given by:

$$\mathrm{t}_u(w) = 1 - \prod_{i=1}^{i=|\mathrm{MPR}(u)|} \overline{\mathrm{p}(v_i,w)}, \tag{7}$$

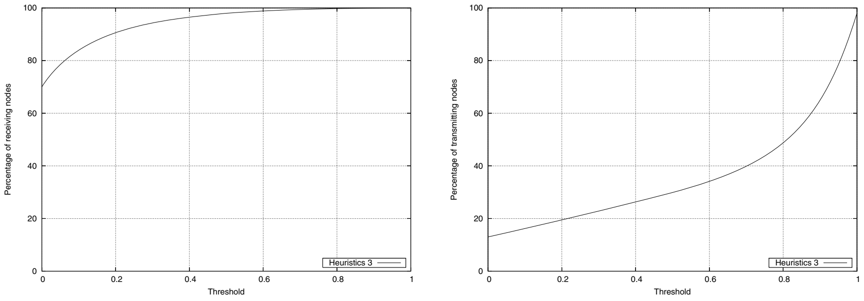
$\overline{\mathrm{p}(v_i,w)}$ being equal to $1-\mathrm{p}(v_i,w)$. In simpler terms, the coverage level of a 2-hop neighbor is equal to its probability to correctly receive the packet from at least one of the chosen relays.

Still considering Fig. 5, if the nodes v_1 and v_2 are selected as relays, then the coverage level $\mathrm{t}_u(w_3)$ of w_3 is equal to $1-(\overline{\mathrm{p}(v_1,w_3)} \times \overline{\mathrm{p}(v_2,w_3)})$. Several relays can thus now be selected to cover the same set of 2-hop neighbors, in order to increase the delivery ratio.

5.4 Performance

We provide in Fig. 6 the performance of the new heuristics presented in this section, considering the lognormal shadowing model. We use the same parameters as in Sec. 4.

Not surprisingly, we observe in Fig. 6(a) that the new heuristics lead to a far better delivery ratio than the original algorithm. This improvement is of course due to the use of the probabilities of correct reception given by the physical model. As expected, the second heuristic offers a higher percentage of covered nodes simply because it prevents



(a) Receiving nodes. (b) Transmitting nodes.

Fig. 7. Performance of the third heuristic for varying thresholds and a density $d = 30$

too far neighbors to be selected as relays. Considering the density $d = 30$, the original heuristic only covers 67% of nodes, against 81%, 85% and 98% for our three proposals. The delivery ratio has thus been greatly improved, as aimed by our heuristics.

As illustrated by Fig. 6(b), the third heuristic, used with a threshold equal to 0.5, requires the participation of 28% of the receiving nodes for the density $d = 30$ to provide a delivery ratio of 98%. This may seem a high value compared to other curves, but considering the results given in Fig. 3 with the unit disk graph model, one can observe that values are almost the same for the original heuristic. This means that the number of chosen multipoint relays for a given node is approximately the same, but their choice is of better quality.

We finally provide in Fig. 7 the performance of the third heuristic for different values of the threshold parameter, considering a density $d = 30$. As expected, the delivery ratio is proportional to the value of the threshold while the number of relays is inversely proportional to it. Choosing a threshold equal to 1 is almost useless as a total coverage can nearly be achieved with a value between 0.4 and 0.5 with far less relaying nodes. Using a threshold of 0 does not lead to a null delivery ratio, because the first step is still applied to cover isolated nodes.

6 Conclusion

From the variety of results presented, we can observe that a realistic physical layer leads to miscellaneous problems while broadcasting. The MPR protocol is a good example: while being very efficient with the unit disk graph, its delivery ratio is not sufficient for most applications with a realistic model. While this study focused on MPR, we believe that other main broadcasting methods, such as dominating sets, will exhibit the same flaws. However, some small modifications, which takes into account probabilities of correct reception, may correct these flaws. Thus, the new heuristics we presented for MPR keep the principle of the protocol, only the selection process of multipoint relays is modified. The latter, while being approximately as many as in the original heuristic, are generally better chosen and provide a higher delivery ratio.

More generally, a huge amount of work is left to be done about this subject. As previously stated, other well-known algorithms will probably need to be modified in order to provide correct performance. Other mechanisms, such as the neighbor elimination scheme [13], may be of prime importance in the quest for the optimal tradeoff between robustness and efficiency. Other aspects of communications, such as neighborhood discovery protocols must also be studied and probably adapted to realistic environments.

References

1. Jacquet, P., Mühlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L.: Optimized link state routing protocol for ad hoc networks. In: Proc. IEEE Int. Multi-topic Conf. (INMIC'01)
2. Ni, S., Tseng, Y., Chen, Y., Sheu, J.: The broadcast storm problem in a mobile ad hoc network. In: Proc. Int. Conf. on Mobile Computing and Networking (MobiCom'99)
3. Stojmenović, I., Nayak, A., Kuruvila, J.: Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer. IEEE Communications Magazine **43**(3) (2005) 101 – 106
4. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint relaying for flooding broadcast messages in mobile wireless networks. In: Proc. Hawaii Int. Conf. on System Sciences (HICSS'02)
5. Quin, L., Kunz, T.: On-demand routing in MANETs: The impact of a realistic physical layer model. In: Proc. Int. Conf. on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW'03)
6. Kuruvila, J., Nayak, A., Stojmenović, I.: Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer. In: Proc. IEEE Int. Conf. on Mobile Ad Hoc and Sensor Systems (MASS'04)
7. DeCouto, D., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. In: Proc. Int. Conf. on Mobile Computing and Networking (MobiCom'03)
8. Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks. In: Proc. ACM SIGCOMM 2004
9. Ingelrest, F., Simplot-Ryl, D., Stojmenović, I.: 17 – 'Energy-Efficient Broadcasting in Wireless Mobile Ad Hoc Networks'. In: Resource Management in Wireless Networking, edited by M. Cardei, I. Cardei and D.Z. Du. Kluwer (2004)
10. Adjih, C., Jacquet, P., Viennot, L.: Computing connected dominated sets with multipoint relays. Ad Hoc & Sensor Wireless Networks **1**(1 – 2) (2005) 27 – 39
11. Lovasz, L.: On the ratio of optimal integral and fractional covers. Discrete Mathematics **13** (1975) 383 – 390
12. Busson, A., Mitton, N., Fleury, E.: An analysis of the multi-point relays selection in OLSR. Technical Report 5468, INRIA (2005)
13. Stojmenović, I., Seddigh, M.: Broadcasting algorithms in wireless networks. In: Proc. Int. Conf. on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (2000)

On Improving Wireless Broadcast Reliability of Sensor Networks Using Erasure Codes

Rajnish Kumar[1], Arnab Paul[2], Umakishore Ramachandran[1], and David Kotz[3]

[1] College of Computing, Georgia Tech, Atlanta, GA 30332
[2] Intel Corp., Hillsboro, OR 97124
[3] Computer Science Depratment, Dartmouth College, Hanover, NH 03755

Abstract. Efficient and reliable dissemination of information over a large area is a critical ability of a sensor network for various reasons such as software updates and transferring large data objects (e.g., surveillance images). Thus efficiency of wireless broadcast is an important aspect of sensor network deployment. In this paper, we study FBcast, a new broadcast protocol based on the principles of modern erasure codes. We show that our approach provides high reliability, often considered critical for disseminating codes. In addition FBcast offers limited data confidentiality. For a large network, where every node may not be reachable by the source, we extend FBcast with the idea of repeaters to improve reliable coverage. Simulation results on TOSSIM show that FBcast offers higher reliability with lower number of retransmissions than traditional broadcasts.

1 Introduction

We consider the problem of information dissemination in wireless sensor networks (WSN). This is an important domain of research because of the multitude of potential applications such as surveillance, tracking and monitoring. WSN nodes are resource constrained, and thus they are initially programmed with minimal software code, and are updated whenever needed. For such on-demand programming, broadcast is typically used to disseminate the new software, making broadcast-efficiency a very important aspect of WSN deployment.

An efficient wireless broadcast scheme must solve two key interrelated challenges:

(*i*) *Messaging Overhead:* Traditionally, each node of a WSN rebroadcasts any new data packet, resulting in too many unnecessary transmissions [14]. For example, if a software update of k packets is to be sent over a WSN of n nodes, potentially k times n broadcasts could be sent out. The larger the number of broadcasts, the more cumulative power is consumed because of the communication. Furthermore, the increased messaging overhead introduces more collisions and thus affects the channel reliability.

(*ii*) *Reliability:* The reliability of message dissemination is a key requirement for a sensor network to function properly. For example, in case of a global software update, if the software at all nodes are not updated reliably, the collected data may become erroneous, or the network may run into inconsistent state. To avoid such problems, reliable code dissemination becomes important. But, empirical results establish that

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 155–170, 2006.

wireless channels are often lossy [2], and in presence of channel loss and collisions, achieving high reliability becomes difficult.

So far, two baseline approaches have been proposed in the literature, *viz.,* deterministic and probabilistic flooding [13]. It turns out that simple deterministic flooding protocols are quite inefficient to address the issues mentioned above. In a probabilistic approach, each node randomly decides whether or not to broadcast a newly seen data packet. These baseline approaches do not assume any extra information about the networks. Several variants and optimizations over these two baseline schemes have also been introduced [14,16,5]. Typically, these derivatives either assume some structural information about the networks, e.g., the knowledge of the network neighborhood, inter-node distances, views on the possible local clusters and so on, or, the protocols rely upon additional rounds of messages, such as periodic ' 'Hello" packets, and ACK/NACK packets following every broadcast.

However, it may not often be possible to depend on any additional information for reasons that are specific to sensor networks. For example, the nodes may not be equipped with GPS, or deployed in an area with very weak GPS signals. The information on neighborhood, distance, location, etc., may continue to change due to mobility and failures. The periodic ' 'gossip" becomes expensive to support because of the transmission overhead incurred and dynamic nature of WSN.

Instead of a protocol that relies completely on controlling the communication, our intuition is to aid the messaging with computational pre/post-processing. The emerging hardware trend suggests that future sensors would have significant computing power. For example, devices such as an iMote have up to 64 KB of main memory and can operate at a speed of 12 MHz. Extrapolating into the future, the motes will soon possess as much computing resources as today's iPAQs. However, while processor efficiency (speed, power, miniaturization) continues to improve, networking performance over the wireless is not expected to grow equally, merely because of the physical ambient noise that must be dealt with. Thus trading processor cycles for communication can offer many-in-one benefits in terms of smaller messaging overhead, less collision, enhanced reliability and reduced power consumption. Following this intuition, we propose a new baseline protocol that is based on a fundamentally different principle, that of the forward error correcting codes (FEC). [1]

The contributions and the findings of this paper can be summarized as follows:

(*i*) We present a new design principle for wireless broadcast in sensor networks. The idea is to combine erasure coding with probabilistic broadcast technique. Founded on this FEC principle, the new WSN broadcast protocol, FBcast, offers high reliability at low messaging overhead. The new scheme also provides additional confidentiality. FEC has earlier been used for asynchronous data delivery and IP multicast in wired networks, but to the best of our knowledge, ours is the first work to explore the viability of applying FEC in wireless sensor networks that have unique requirements and packet loss characteristics substantially different from wired networks. Ours is a vanilla data

[1] Erasure codes are a class of encoding; a data packet (seen as a collection of small blocks) is blown up with additional redundant blocks (such as parity checksums) so that if some blocks are lost due to any kind of noise (external signal, faults, compromises), the original packet may still be reconstructed from the rest.

dissemination protocol that assumes no extra information about the underlying network. As we observe through our experiments, the transmission characteristics (such as signal strength and packet loss) vary fairly randomly as one goes radially outward from a data source; thus common assumptions, such as regular signal strength distribution over concentric circles or sphere, that are made by many of the other protocols do not hold true in reality. Using FEC based vanilla protocols in such a scenario becomes quite useful.

(*ii*) We compare FBcast with probabilistic broadcast through simulation studies using the TOSSIM simulator [4]. Protocol efficiency can be evaluated over more than one axis, each of which can be potentially traded for another, e.g., reliability can be enhanced at the cost of extra messaging overhead, or spatial density of the sensors and so on. Thus a point-by-point fair comparison between these approaches may not always be possible. However, our experiments do suggest that, FBcast performs better over a larger parameter space formed by the metric-axes.

(*iii*) We propose FBcast with *repeaters* to disseminate code over large area network. Over a multi-hop network, especially in sparse network deployment, traditional broadcast reliability decreases as one goes away from the data source. We extend the basic FBcast protocol with the idea of repeater nodes. Where to place the repeaters without much network information is a challenge. We present a novel heuristic to solve the repeater placement problem. We compare the performance of the extended FBcast protocol against a similar variant of probabilistic protocol, and find the new protocol more effective in increasing the broadcast coverage.

The paper is organized in the following way. Section 2 looks at the motivation and related broadcast protocols to place our work in context. Section 3 provides details of FBcast protocol. It also explains the encoding scheme used. Section 4 presents the implementation details of FBcast and simulation results.

2 Background and Related Work

Baseline Approaches: So far, the data dissemination techniques for wireless networks can be divided into two major approaches, deterministic or probabilistic broadcasts. Simple flooding [10] is the most naive implementation of the first class. However, naively re-broadcasting can easily lead to broadcast storm problem [14], and hence the need for controlled density-aware flooding and multicast algorithms for wireless networks [1]. Simple flooding (depending on the placement of neighboring nodes) also suffers from the severe inefficiency that the effective additional coverage of a new broadcast can be as low as 19% of the area of the original circle that a broadcast can effectively reach.

Variants and Optimizations: Several other optimizations can be applied to these baseline schemes. For example, *pruning* is the strategy of selectively suppressing the broadcast activity. The objective here is to find out exactly a set of nodes (that will broadcast data) so that no other node need to rebroadcast. These nodes constitute a *Flooding Tree*. Finding a minimal flooding tree is NP-complete [6]. The *Dominant Pruning* (DP) algorithm is an approximation to finding the minimal flooding tree [6]. Lou and Wu proposed further improvements over DP that utilize two-hop neighbor information more

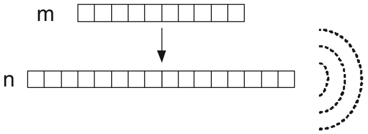


Fig. 1. FBcast at source: m original packets are encoded into n packets and injected in the network

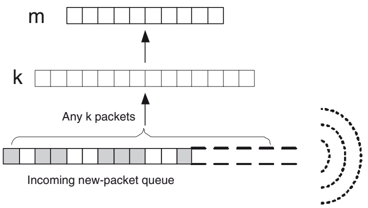


Fig. 2. FBcast at recipients: k packets are chosen from received queue and decoded back to m

effectively than DP [7]. Again, the neighborhood information is maintained by periodic gossiping that add additional transport overhead. Garuda [12] provides reliable downstream data broadcast using a minimum dominating set of core nodes, which provides a loss recovery infrastructure for remaining nodes. The overhead incurred by core selection and maintenance in Garuda may make it an expensive solution for dynamic networks.

Similarly, since collision is a critical obstacle, one intuition is to have many of the nodes stay off from transmissions, and thus create a virtual sparser topology that will have less collisions. Irrigator protocol and its variants are based on this idea [11]. For a comparative and comprehensive survey of these protocols, the reader can refer to a related work[16]. PSFQ [15] uses hop-to-hop error recovery by injecting data packets slowly, while neighbor nodes use NACK-based fast repair. On a similar note, Trickle [3] combines epidemic algorithms and density-aware broadcast towards code dissemination in WSN.

FBcast, as a base approach, provides another alternative to simple deterministic and probabilistic broadcasts. Other smart adaptations such as location-aware retransmission, maintaining neighborhood and routing information are expected to boost its performance. Rate-less erasure codes, such as Fountain codes, form a critical ingredient of our approach. They were first proposed as a tool for efficient multicasting. Later on, versions of such codes have been shown as useful tool for Peer to Peer file sharing and download purposes [9]. FBcast applies the idea of fountain encoding with the previously known scheme of probabilistic gossip, to achieve high reliability without the extra messaging overhead, in the domain of wireless networking where bandwidth, power and reliability are very critical issues to be addressed.

3 FBcast Protocol

Figure 1 and 2 pictorially represent FBcast broadcast protocol. The data to be disseminated consists of m packets. The source blows up these m packets into n packets, using the encoding scheme described below, and the encoded data is injected in the network. Each recipient rebroadcasts a packet, if it is new, with a probability p. When a recipient node has received enough data packets ($k \geq m$). The exact value of k depends on the specific encoding scheme used. for decoding, it reconstructs the data and passes it to the application. In order to encode and decode (using FEC) the nodes would need to know a random seed from which the rest of the code parameters can be generated using a pseudo random number generator. We assume that this seed and the generator (typically a very light-weight algorithm) is shared by all nodes.

Erasure codes provide the type of encoding that is needed to accomplish the protocol. In particular, it is desirable that the codes have following properties. (*i*) The ratio n/m (also known as the *stretch factor*) can be made arbitrarily large or small flexibly. In other words, one can generate as many redundant packets as needed, by a decision that can be taken online. (*ii*) There is no restriction on the packet length. (*iii*) Both encoding and decoding should be inexpensive.

Standard erasure codes, such as the Reed-Solomon codes, are *inflexible* in the first two aspects, and are quite inefficient in performance. Although these codes allow any desired stretch factor, this can only be done statically. It is not easy to change n and k *on the fly* during the application runtime for the following reasons. First, these codes require that every time the stretch factor is to be readjusted, a new code needs to be defined and disseminated to all the participating nodes. Second, the code length parameter n is always upper bounded by the order q of the underlying field; every time a higher stretch factor needs to be applied, a great deal of meta-information needs to be disseminated and computational overhead incurred. Third, the size of a symbol, i.e., the number of bytes treated as one unit of information, is also upper bounded by the field size; for a field size q, the largest unit of information treated at one time can be at most $\log q$ bits. In our setting, the size of one packet is essentially the symbol length of the code being used, and thus essentially the chunk of data that can be handled at one time is limited by this *a priori* fixed parameter. For a comprehensive discussion regarding the kind of problems posed by standard codes, the reader can refer to the works of Luby, Mitzenmatcher *et al.* [8].

Fortunately, a modern class of erasure codes solves these problems effectively. These are called *Fountain codes*. The category-name *fountain* is suggestive - when one needs to fill up a cup of water from a fountain, there is no need to bother about which particular droplets are being collected, rather just enough number of drops to fill in the glass would be sufficient. Not all Fountain codes are equally flexible. The candidate ones that we are particularly interested in are the Luby Transform codes (LT codes [8]), Raptor and Online codes. We are interested in the codes that are *rateless*, *i.e.*, can produce on-the-fly a *fountain* of encoded blocks from k original message blocks. For a pre-decided small number ϵ, only $k = (1 + \epsilon)m$ number of data blocks out of this fountain suffice to reconstruct the original document. Moreover, there is no limitation on the symbol-size, i.e., the packet length - any number of bytes can be considered as a single unit of information. An example of a rate-less code is the Luby-Transform codes [8]. Our idea is to generate the blocks and sprinkle them over to the neighbors who would re-sprinkle a small fraction of the fountain.

The main benefit of data encoding is three fold. (*i*) Enhanced reliability, which is achieved by adding extra information encoded in the data packets. Thus, if a node has noisy reception, it may not be able to receive all the data packets, yet, it can generate the original data. (*ii*) Data encoding decreases transmission overhead. Because of the redundancy, the recipients do not need to retransmit all the packets; each transmits only a few of what it receives, thus alleviating contention for the shared channel. (*iii*) The scheme provides data confidentiality as an extra benefit. Because of the shared nature of the wireless channel, confidentiality is often a requirement in wireless network deployment. To encode and subsequently decode the same data, the sender and receiver need to have a shared random seed. Hence, no eavesdropper can decode the information from the wireless channel.

4 FBcast Evaluation

We have implemented the communication aspect of FBcast protocol in TinyOS, i.e., we account for only packet transmission and reception. Though we do not implement the data encoding and decoding in TinyOS, we utilize our experience of fountain code implementation, discussed below, on Linux to tune the FBcast parameter (stretch factor). While we explore the effect of encoding/decoding control parameters upon FBcast reliability, we do not evaluate their effect on the energy consumption or computational latency they add to the broadcast because of the focus of this paper. To understand the protocol behavior, we simulated FBcast using TOSSIM [4] for different network sizes. For comparative study, we also implemented traditional probabilistic broadcast, *pbcast* in TinyOS.

We have looked at three aspects of FBcast and *pbcast*: reliability, transmission overhead, and latency. Reliability is measured as the percentage of motes that receive the original message being disseminated. If a mote receives only some of the injected packets, it may not be able to reconstruct the original data; we assume this to be true for both FBcast and *pbcast*. Transmission overhead is the sum total of transmitted packets on all the nodes during the simulation time. The simulation time is kept long enough such that retransmissions by every mote is complete. Latency is the average time when motes are able to reconstruct original data after receiving enough packets, and it does not include the data encoding or decoding time. For FBcast, latency is the expected time when motes have received k packets, and for *pbcast* it is the expected time when motes have received all the injected packets.

The FBcast parameters are set as follows: $m = 10, n \in \{20, 40, 60\}, k = 14$, and p is adjusted in proportion to n. More precisely, p varies from $1/n$ to $8/n$, thus, for n=20, p is varied from 0.1 to 0.4. Putting this in words, the number of packets in the original data is 10. With a stretch factor of 2, FBcast encodes the data to 20 packets and injects them at the source. Our experiments reveal that a factor of 1.4 is sufficient, i.e., a mote that receives at least 14 distinct packets, can reconstruct the original data. In case of simple broadcast, only 10 packets are injected. For FBcast, value of p is kept proportionally low as n is varied. For $n = 20$ and $p = 0.4$, a mote is expected to retransmit 8 out of 20 new packets it receives. The retransmission overhead here thus becomes equivalent to that of *pbcast* with $p = 0.8$. For *pbcast* experiments, due to absence of any encoding or decoding, $n = m$.

A few words about the implementation of Fountain code. Our experience of implementing fountain codes suggests that by choosing $m' \approx 1000$ (number of message symbols) and $n' \approx 6000$ (number of encoded symbols), data can be reliably reconstructed from $k' \approx 1400$ symbols. However, a bunch of symbols can be coalesced together to form a packet, e.g., by coalescing 100 symbols one generates message blocks of size

$m = 10$ packets and encoded blocks of size $n = 60$ packets. The memory requirement is also within the limits of present motes. For example, to implement LT codes (a class of fountain codes), one needs to have in memory a bipartite graph of size $n' \times \log(m/\delta)$ (see Theorem 13 in [8]). δ is a small constant (e.g., $\delta = 10^{-3}$ gives us very high accuracy in the decoding). Thus, for the parameter set we have presented in this paper, and the most space-efficient representation of the data structures, the memory requirements would be a little over 60 KB, which is clearly not far from the current limits. Moreover, it is expected that the main memory will soon touch the limits of megabytes, thus paving for more time-efficient representations of the structures for these algorithms. In our TOSSIM experiments, we simulated a network of mica2 motes. These motes presently have only 8 Kilobytes of memory, not enough for a full-scale implementation. However, devices such as iMotes already have 64 KB memory, and it is only fair to assume that very soon, enough space will be available on these motes for running both OS and the applications of this order.

Results Summary. FBcast and *pbcast* both can achieve reliability close to 100%, but FBcast can do so for larger window of variation in the mote density, and at lower transmission overhead than *pbcast*. Also, while *pbcast* exposes only the forwarding probability to control its behavior, FBcast exposes the forwarding probability and the stretch factor as control. Thus FBcast can be adapted more flexibly to suit different deployment densities and reliability requirements. The repeater variants of *pbcast* and FBcast, designed for large network deployments, both have higher reliability compared to their original counterparts. However, FBcast variant is easier to configure and it attains more than 99% reliability for various deployment parameters at lower transmission overhead compared to the $pbcast$ variant.

The rest of this section is organized as follows. First, after explaining the network model used, we start with simple experiments, where there is no rebroadcast, and observe the possible benefits of using FEC. Then we add probabilistic retransmissions to increase the reliability and increase the broadcast coverage. We also explore different ways in which FBcast can be configured. Finally, we add the idea of repeaters in FBcast to overcome the limitation observed for FBcast without repeaters, namely, broadcasting over a very large area.

4.1 Network Model and Assumptions

Unless specified otherwise, the following experiments are based on the empirical radio model supported in TOSSIM, where every link is used as a lossy channel with loss probability based on empirical data. Instead of a perfect radio model, we use the empirical radio model because it allows us to see the effect of packet loss upon broadcast. TOSSIM provides a Java tool, $LossyBuilder$, for generating loss rates from physical topologies. The tool models loss rates observed empirically in an experiment performed by Woo et al. on a TinyOS network [2]. LossyBuilder assumes each mote has a transmission radius of 50 feet. Thus, each mote transmits its signal to all nodes within 50 feet radius range, and the quality of received signal decreases with the increase in distance from the transmitter. Given a physical topology, the tool generates packet loss rates for each pair based on the inter-mote distance.

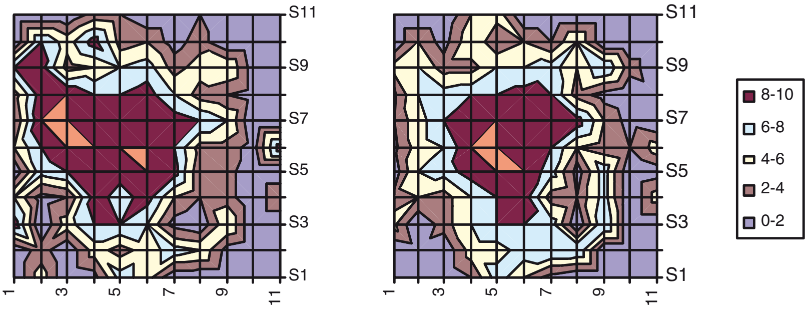


Fig. 3. Topographical picture of reliability for two typical runs showing how the reliability decreases as we move away from the grid center. 121 motes placed on 11x11 grid with inter-mote spacing of 5 feet.

For experiments that use the empirical radio model, we use a grid-based topology to get the loss pattern. By varying the grid size, inter-mote distance is varied, thus affecting the loss probability. The data source is assumed to be at the grid center because of the nature of the experiments. For experiments that use the simple radio model, the transmission loss probability between any two motes is the same for all mote pairs. Nodes are assumed to be located such that each node can listen to all the other nodes. In TOSSIM, network signals are modelled such that distance does not affect their strength, thus making interference in TOSSIM generally worse than the expected real world behavior. However, due to the TinyOS CSMA protocol, the probability of two motes, within a single cell, transmitting at the same time is very very low.

4.2 FBcast Without Any Rebroadcast

We distinguish between rebroadcast and retransmission that we will maintain throughout the rest of the discussion. Whenever a node transmits the same message that it has transmitted in the past, we refer to the event as a *retransmission*. However, when a node is broadcasting a message that is received from another node, the event is called a *rebroadcast*.

We first consider the case when a single source broadcasts a message and there is no other rebroadcast following this event. However the source may retransmit the message multiple times. Reliability is defined as the fraction of nodes that are able to receive (reconstruct) the original message. Thus, reliability depends on packet loss, which in turn depends on multiple factors, including the bit-error rate and interference at the receiving node. Since there is no rebroadcast, there will be no interference. Results (omitted because of lack of space) show that using FEC improves reliability compared to simply re-injecting the original packets multiple times, but without any extra mechanism, or rebroadcasts, it does not provide enough reliability.

If we look at the number of packets being received at different motes, because of the probabilistic nature of channel error, the resulting topological distribution pattern for successful reception is quite dynamic across different simulation runs. Topologies obtained for two typical runs are shown in Figure 3. There exists a set of concentric

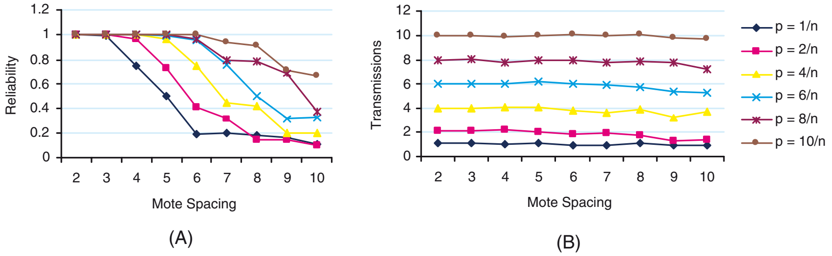


Fig. 4. Pbcast performance for 121 motes deployed on a 11x11 grid with varying inter-mote spacing (x-axis)

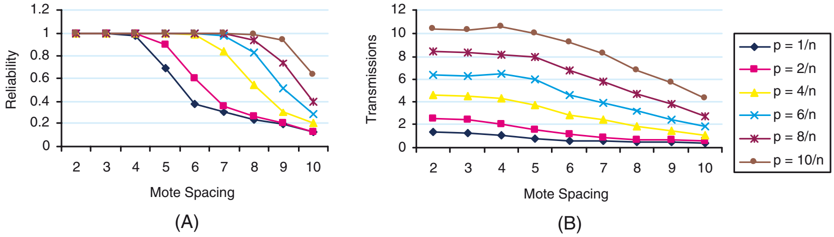


Fig. 5. FBcast (with probabilistic rebroadcast) performance for 121 motes deployed on a 11x11 grid with varying inter-mote spacing (x-axis)

bands of motes that receive similar number of packets, but the bands are not circular, nor are they identical across different runs. The bands are neither circular nor repeatable because of the nature of wireless medium and mutual interferences. This means there is no simple way in which we can divide a large area into smaller cells and put a broadcast server into each cell to provide reliability in a large area. Hence, we resort to another intuitive alternative to increase reliability, i.e., by doing a probabilistic rebroadcast at intermediate motes.

4.3 FBcast with Probabilistic Rebroadcast

When a node is allowed to do probabilistic rebroadcast of new packets, the results show that for the same deployment of 121 motes as before, we can do reliable broadcast to all the motes even at higher inter-mote spacing (than mere 2 feet) is achievable by increasing the forwarding probability at intermediate motes. Both Pbcast (probabilistic broadcast variance without FEC) and FBcast (FEC variant) achieve complete reliability, but as shown in Figures 4 and 5, FBcast achieves higher reliability (Figure 4-A and 5-A) than Pbcast at lower transmission overhead (Figure 4-B and 5-B).

Let us represent the forwarding probability p as α/n, where α is the number of forwarded packets, and n is the original number of packets. At first glance, it may appear that for a given α, say $p = 10/n$, FBcast always gives higher reliability than Pbcast. But if we look closely, we find that there is no direct correlation in the reliability of Pbcast and FBcast for the same α in p. For example, with $p = 10/n$, and spacing

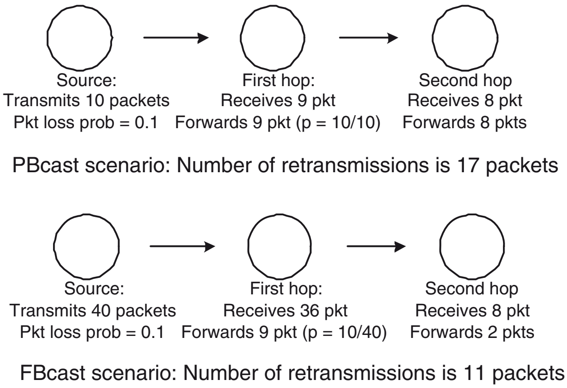


Fig. 6. Transmission overhead comparison of Pbcast and FBcast for a simple topology

of 10 feet, Pbcast has better reliability than FBcast, but at spacing of 8 feet, FBcast gives better reliability. This can be explained if we look at the transmission overheads of Pbcast and FBcast that are shown in Figures 4-B and 5-B.

Contrary to intuition, we see that at $p = 10/n$, at spacing of 10 feet, Pbcast transmits about 10 packets per mote, while FBcast transmits only about 4 packets per mote. We expected both Pbcast and FBcast to incur same amount of transmissions, i.e., about 10 packets per mote (p being $10/n$). The amount of transmissions explain why FBcast has lower reliability than Pbcast. But to understand why FBcast has lower transmission than Pbcast for the same α, we can look at a simple model shown in Figure 6. The three motes, placed in a straight line with one-hop spacing, incur only 11 retransmissions in the case of FBcast, compared to 17 extra transmissions (due to rebroadcast) in the case of Pbcast. This is because though we limit the number of extra transmissions by changing α, the amount of new packets received at distant hops is not proportional to n for Pbcast and FBcast, thus FBcast does fewer transmissions due to rebroadcasting. Also, we have observed that at higher inter-mote spacing, the number of transmissions for Pbcast decreases and the curve becomes similar to what we show in the case of FBcast.

From the above results, we learn that FBcast can provide higher reliability than Pbcast for similar amount of retransmissions; but it may not necessarily mean that FBcast will provide higher reliability than Pbcast for same α. Still, the reliability is limited at higher inter-mote spacing. How much is it possible to stretch the reliability by increasing FBcast's stretch factor? To answer this, we look into the results shown in Figure 7.

For a deployment of 121 motes with 10 feet inter-mote spacing, we can achieve close to 100% reliability at higher stretch factors (e.g. 6) and at high forwarding probability, shown in Figure 7. As expected, increasing stretch factor improves the reliability and also increases the number of retransmissions. Also, for same the stretch factor, increasing the forwarding probability improves the reliability. However, for stretch factor of 2 ($n = 20$), we observe that reliability first improves, peaks at $p = 16/20$, and then it goes down rapidly. To understand this anomaly, we look at the effect of the rate of broadcast at the data source.

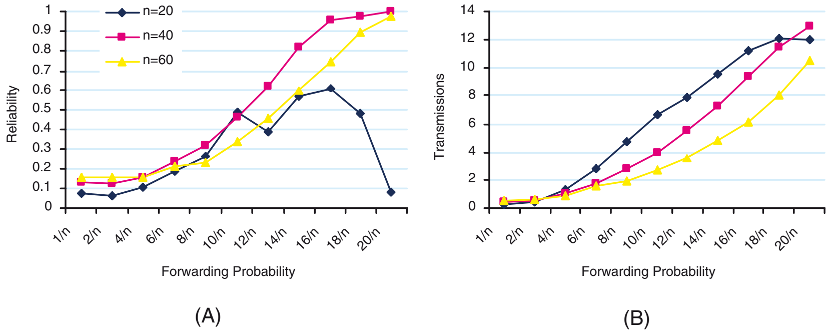
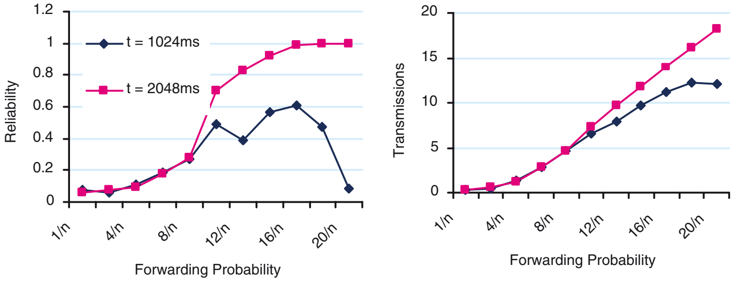


Fig. 7. Effect of stretch factor on reliability. Mote spacing = 10 feet. 121 motes deployed on a 11x11 grid. Forwarding probability is varied along x-axis.

Consider two extreme cases: first, when the source packets are inserted into the network at a very slow rate, and second, where data packets are being inserted without much delay between the transmissions. At a slow rate of source broadcast, there is less interference, and thus higher reliability, compared to the case where data packets are being inserted rapidly. The interference becomes more severe in the presence of probabilistic rebroadcast at the intermediate motes. This is because, when there is a large number of new packets being inserted at the source, there is an increase in the number of retransmissions at the other nodes, and this leads to collision due to hidden terminal problems or other interference issues.

The effect of interference is also observed in our experimental results shown in Figure 8, where we compare the reliability of FBcast with $n = 20$ for 121 motes deployed at 10 feet inter-mote spacing. When the data source injects roughly one packet per second, we observe that reliability suffers heavily at high forwarding probability: though the number of retransmissions shown in Figure 8-B is very high, the number of successful receipt is low (see Figure 8-A). However, when the data source slows down the rate of packet broadcast (a packet roughly every 2 seconds), the reliability increases continuously until it reaches 100% at higher forwarding probabilities. The amount of transmission overhead increases, but so does reliability, indicating that the interference effect is subdued because of the slower rate of source data broadcast. The effect of interference is less apparent for higher stretch factors because of the basic property of FEC-based data recovery, i.e., even if some of the packets are lost due to interference, other motes will be able to reconstruct the original data.

Need for FBcast Extension. From the above results, it is clear that FBcast can be adapted more flexibly to suit different deployment densities and reliability requirements. However because of the number of parameters involved, the complexity of packet loss characteristics, and the probabilistic nature of FBcast, there is no simple expression that captures FBcast reliability for different parametric settings and network conditions. In the following discussion we explore how we can achieve high reliability for various network sizes without dynamically adapting the stretch factor or forwarding



(A) (B)

Fig. 8. The effect of injection rate: how interference causes the reliability to drop at higher forwarding probability, though the amount of retransmissions increases as expected

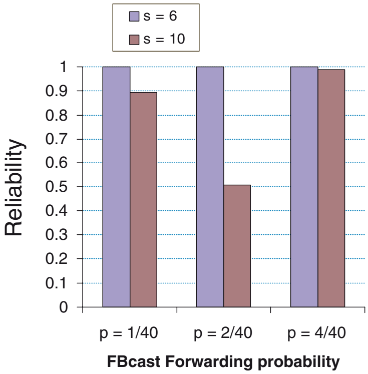


Fig. 9. Performance of FBcast ($n = 40, p = 2/40$) with repeaters for 441 motes deployed with inter-mote density s = 6' and 10'

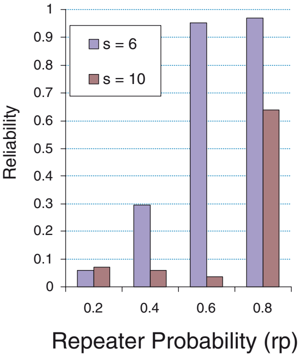


Fig. 10. Performance of pbcast ($p = 0.8$) with repeaters for 441 motes deployed with inter-mote density s = 6' and 10'

probability. In doing so we look at the limitation of FBcast in covering large deployments, which leads to our solution using repeater extensions.

4.4 Protocol Extension with Repeaters

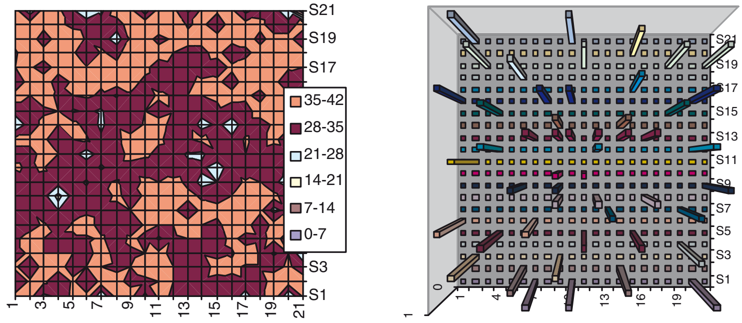
In the experiments in the Section 4.3, all the motes are within the broadcast range of the source (referred to as single-hop experiments). The network topology used here is once again a grid of motes, but unlike the earlier single hop experiments, here the mote density is kept the same while increasing the number of motes, thus expanding the deployment area. For example, a grid of 441 motes deployed with inter-mote spacing of s = 10', will cover 200' X 200' area. With increase in the deployment area, the number of hops between the data source and the peripheral motes increases, realizing the effect of multi-hop communication. In the presence of such multi-hop communication, we want to measure the reliability of *pbcast* and FBcast protocols. For the following experiments, $pbcast$ is set with $p = 0.8$ because $pbcast$ with lower forwarding probability value has very low reliability. Experiments reveal that even though, FBcast provides higher reliability than $pbcast$, the reliability decreases with increasing deployment area. The fraction of motes being able to reconstruct the original data decreases with increase in the deployment area. There are two possible reasons for this result. First, hidden terminal problem is more severe here than in the single-hop experiments. For example, for a small deployment area, the source mote was found to be able to inject all 10 packets, but for a larger deployment area, the source mote had to retry injecting the original packets several times. Second, the peripheral motes are able to receive only a few or no packets.

Because of channel loss and probabilistic retransmission, as we go away from the data source in the center, the number of received packets decreases. This is observed in single hop scenario also (see Figure 3), but it is more evident for multi-hop scenario. For these experiments, the inter-mote spacing is 10’. With 441 motes placed uniformly in a 200’x200’ area, the figure shows the number of packets received in different zones. For *pbcast*, with $p = 8/10$, the broadcast coverage is less than 5% of the area. As we increase n, we observe an increase in the coverage. But increasing n also has inherent cost (encoding/decoding cost), a very high n may not be the desirable engineering choice. Also, even with $n = 60$, the coverage is less than even 20%. Next, we explore how extending FBcast with repeaters extends the broadcast coverage.

A repeater is a mote that reconstructs the original data and acts as the data source, thus injecting the encoded data packets. For $pbcast$, being a repeater just means retransmitting the original data packets, and for FBcast, being a repeater means decoding the received packets to reconstruct the original data, encoding the data again to generate n packets, and re-injecting all the packets. Hence, only a mote that has received at least k packets can be a repeater. We design and evaluate an FBcast protocol with repeater motes. For a fair comparison of $pbcast$ and FBcast, we also develop a repeater variant of $pbcast$ and compare it with FBcast.

FBcast Extension with Repeaters. Because of unknown data source mote, unknown network topology and radio behavior, and probabilistic nature of the broadcast, a priori placement of repeaters is not desirable. A repeater should be selected dynamically, and such a selection poses a question: how can a mote decide to be a repeater based on the received packets? If the condition for being a repeater is very relaxed, there may be too many motes serving as repeaters (over-coverage), and if the condition is very tight, there will be too few repeaters to cover the whole area (under-coverage). Our repeater algorithm strikes a balance by utilizing the rate of packet receptions and the number of received packets.

The repeater algorithm works as follows. Every mote calculates how long it should listen before it decides about becoming a repeater, call this time is the *listen window*. At the end of a listen window, if a mote has enough packets to construct original data, but less than a threshold number of packets (k_{th}), the mote becomes a repeater. By having threshold check as a condition, we ensure that not every mote becomes a repeater, but only those that are able to receive a small fraction of the injected packets. This threshold



A. Topography for multi-hop scenario with repeaters (200'x200' area with grids of size 10'10')

B. Position of the repeater motes

Fig. 11. FBcast with repeaters for motes deployed with 10’ spacing in a 200’x200’ area: (A) shows the coverage, and (B) shows the overhead in terms of number of repeaters and their positions

condition will be satisfied by a band of motes, as it is clear from the one-hop results; around the data source, there exist concentric bands of motes with similar number of received packets. To ensure that not all motes in a particular band become repeates, we carefully randomized the length of initial listen window. Listen window consists of two parts: first part is the expected time when a mote will receive the threshold number (k_{th}) of packets, and the second is a random duration from 0 to $t \in [0, t_k]$, where t_k is the duration between reception of first and kth packets. The length of listen window affects the latency of packet dissemination in large area. For faster propagation, the listen window can be reduced; though the flip side of this choice is an increase in the number of repeaters per unit area.

Figure 9 shows the results of FBcast deployed over a coverage area of 200’x200’ with motes spaced every 10’ along the grid points. Reliability is plotted on the Y-axis against the forwarding probability on the X-axis. We present the results for 2 different spacings, $s = 6'$ and $s = 10'$. We see that for most of the cases the attained reliability is very high, except for one instance (because of the probabilistic nature of the protocol). The algorithm parameters are set as follows: $n = 40$, $p = 8/40$, and $k_{th} = 21$. The value of threshold packet count, k_{th} is important. Since a mote becomes repeater only if it receives packets between k and k_{th}, setting k_{th} too close to k decreases the probability of a mote becoming a repeater.

In Figure 11A we show the topographical coverage attained by FBcast with repeater; for the chosen setting, complete coverage of the 200’x200’ area is attained. Most of the motes receive more than 21 packets. We found that setting $k_{th} = 1.5k$ for mote deployments with 10’ spacings or less enables this complete coverage. Figure 11B shows the position of repeater motes.

pbcast Extension with Repeaters. For *pbcast*, typically there is no way for a mote to know how far it is located from the data source (unless some extra information such as location about the neighborhood is provided). Thus, in this variant we assign a predefined probability (rp) of being a repeater to any mote that has received 10 packets.

Figure 10 shows that *pbcast* with repeaters can provide complete coverage of the deployment area, albeit at the cost of high repeater probability. For $rp = 0.6$, the reliability is 85% for inter-mote spacing of $s = 6'$, but the reliability goes below 10% for spacing of $s = 10'$ (sparser mote density). Increasing the repeater probability helps achieving high reliability for sparser deployments, but that also increases interference; high values of rp essentially amounts to data flooding, and the consequent interference leads to the well known broadcast storm situation [14]. At an inter-mote spacing of 10', we noticed more than 99% coverage only for very dense repeater deployments $(rp > 0.8)$.

Protocol Comparisons. The repeater variants of *pbcast* and FBcast both have higher reliability compared to their original counterparts. For sparser deployment, *pbcast* yields high reliability only with very high repeater probabilities, thus causing high transmission overhead. FBcast (with the aid of listen window and threshold number of received packets) is able to control the number of repeaters while ensuring more than 99% reliability for various deployment parameters.

5 Conclusion

We have presented a new broadcast protocol that exploits data encoding technique to achieve higher reliability and data confidentiality at low overhead. The simulation experiments show that with increased network density, traditional broadcast become quite unreliable, but FBcast maintains its reliability. The forwarding probability parameter of FBcast can be tuned to decrease the number of transmissions with higher density. FBcast with repeaters allow nodes to reconstruct the original data and then re-inject the new packets into the network (when the number of received packet falls below a threshold). FBcast trades off computation for communication. The data encoding (source) and decoding (recipients) consume computation cycles, but since computation is order of magnitude less power-expensive than communication, we expect to save power. Also, considering the computation needs of the encoding scheme, FBcast is suitable for computationally rich nodes. Based upon the continuing trend we believe that today's handhelds are tomorrow's motes, and FBcast will be quite suitable for future WSN.

References

1. P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Trans. Comput. Syst.*, 21(4):341–374, 2003.
2. D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks, 2002. Technical Report, Intel Research.
3. Jae-Hwan Chang and Leandros Tassiulas. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the First ACM/Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.
4. P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *Proceedings of the first international conference on Embedded networked sensor systems*, pages 126–137. ACM Press, 2003.

5. L. Li, J. Halpern, and Z. Haas. Gossip-based ad hoc routing. In *Proceedings of the 21st Conference of the IEEE Communications Society (INFOCOM'02),.*, 2002.
6. H. Lim and C. Kim. Flooding in wireless networks. *Computer Communicatins*, 24(3-4):353–363, 2001.
7. W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing', volume =.*
8. M. Luby. Lt codes. In *Proceedings of 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.
9. P. Maymounkov and D. Mazieres. Rateless codes and big downloads. In *Proc. of the 2nd International Workshop on Peer-to-Peer Systems*, 2003.
10. K. Obraczka, K. Viswanath, and G. Tsudik. Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks*, 7(6):627–634, 2001.
11. L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti. Localized techniques for broadcasting in wireless sensor networks. In *Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pages 41–51. ACM Press, 2004.
12. S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 78–89, New York, NY, USA, 2004. ACM Press.
13. W. Peng and X.-C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 129–130. IEEE Press, 2000.
14. Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wirel. Netw.*, 8(2/3):153–167, 2002.
15. C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 1–11, New York, NY, USA, 2002. ACM Press.
16. B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194–205. ACM Press, 2002.

Cost-Aware Route Selection in Wireless Mesh Networks

Junmo Yang[1], Kazuya Sakai[2], Bonam Kim[1], Hiromi Okada[2], and Min-Te Sun[1]

[1] Department of Computer Science and Software Engineering,
Auburn University, Auburn, Alabama 36849–5347
{yangjun, kimbona, sunmint}@eng.auburn.edu
[2] Department of Electronics Engineering
Kansai University, 3-3-35 Yamate-cho, Suita, Osaka, Japan 564-8680
{sakai, okada}@jnet.densi.kansai-u.ac.jp

Abstract. Wireless mesh networks have emerged to be one of the promising applications of ad hoc networks. The idea of installing multiple radio interfaces at each mesh router allows a mesh network to better utilize the available wireless bandwidth, but at the same time complicates the issue of route selection. In this paper, we propose a novel metric that measures the bandwidth and cost ratio of each route. Based on this metric, a Cost-Aware Route Selection (CARS) scheme is proposed to improve the overall throughput of a mesh network. The simulation results confirm that our scheme is able to better utilize the limited wireless resource and improves the overall network throughput by more than 95% with different types of traffic and communication patterns when it is compared against the past route selection schemes.

1 Introduction

Wireless Mesh Networks (WMNs) have emerged as one of the most promising applications of ad hoc networks. By connecting inexpensive mesh routers with multiple radios wirelessly, WMNs can quickly provide broadband networking infrastructure for large business enterprizes and bring Internet access to residence in rural areas.

To take full advantage of WMNs, many research issues, such as backbone construction, cross-layer design, multi-channel MAC, and fault tolerance [1], are yet to be addressed. Among them, routing is perhaps one of the most important topics. At first glance, since WMNs are considered as a special type of ad hoc network, it seems appropriate to use one of the routing protocols originally developed for ad hoc networks [2] for WMNs. However, such an approach overlooks the following three key differences between research in WMNs and traditional ad hoc networks, and is thus likely to result in poor performance.

- Node classification - Traditional ad hoc networks are formed by nodes that are commonly assumed to be homogeneous in terms of the hardware/software configuration and degree of mobility. In contrast, wireless mesh networks are composed of two distinct types of nodes - mesh routers and mesh clients. Mesh routers, similar to conventional wireless access points, are generally assumed to be built using inexpensive parts, to be stationary, and to be connected to an external power supply. Notice that most mesh routers are not connected directly to the wired backbone. If a mesh router is connected to the wired backbone, we referred it as the gateway or gateway mesh router in particular. The mesh clients, such as laptops and handheld

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 171–184, 2006.

PDAs with wireless LAN [3] capability, run on their own batteries and move at moderate speed.

- Multiple antennas - To increase the capability of WMNs, mesh routers can be equipped with multiple radio interfaces. Each interface can adopt one of the three wireless standards: IEEE 802.11a [3], 802.11b [3], and 802.11g [3]. The different standards present distinct by different physical characteristics, particularly with regard to their radio spectrum, transmission rate, and transmission radius. This immediately presents two challenges for the protocol design. First, the topology of WMNs is no longer a simple graph. Depending on which radio interfaces are available, a mesh router can have several different sets of neighbors. Second, the channels used by different radio interfaces can interfere with each other if the portion of the radio spectrum used by these interfaces overlap with each other.
- Adaptive transmission rate - In most research on ad hoc networks, the unit disk model is used [4,5]. In this model, the transmission rate between two nodes within a predefined transmission range is assumed to be a constant. However, it is known that the transmission rate between two wireless LAN entities can automatically step down if the quality of the link between them degrades. For instance, depending on the distance between two nodes, the transmission rate of a IEEE 802.11b link can be either 11Mbps(0m - 50m), 5.5Mbps(51m - 62m), 2Mbps(62m - 68m), or 1Mbps(68m - 85m) [6].

These differences further complicate the issue of routing in WMNs. In [7], it was shown that finding the optimal route in a multi-radio WMN is NP-hard. As the first step toward solving this problem, most previous proposals [7,8,9,10,11] suggested different metrics that can be used to help identify the best one out of a set of candidate routes. It is expected that by using these metrics for route selection, the overall throughput of the mesh network can be improved.

In this paper, we propose a novel route selection scheme is proposed, namely Cost-Aware Route Selection (CARS), for WMNs. Unlike the past route selection schemes, which are primarily based on the quality of links in a route, the new scheme takes the interference cost and traffic aggregation into consideration. By selecting the route with the best bandwidth and cost ratio from a set of candidates, the limited wireless resources (i.e., the available channels for mesh routers) can be better utilized. This will automatically lead to better overall network throughput. The simulation results show that the proposed CARS scheme significantly improves the overall network throughput by more than 150% in the case of burst traffic and the number of connections by more than 95% in the case of constant bit rate traffic.

The remainder of this paper is organized as follows. In Section 2 we review the existing route metrics and route selection schemes for WMNs are required. The proposed route metric and scheme are described in Section 3 and the simulation results and analysis are provided in Section 4. Finally, the chapter concludes by summarizing the research and pointing out the future research directions in Section 5.

2 Survey of Existing Route Selection Schemes in WMNs

Routing is one of the most fundamental issues in WMNs. In the past, several metrics were proposed for multi-hop wireless networks in order to measure the quality of a route. In [10], the Expected Transmission Count (ETX), which is based on link layer frame loss rates, was used to locate a path with higher throughput in a multi-hop wireless network. However, ETX does not take into account the bandwidth of links in a path. In addition, ETX does not give preference to channel diversity.

In [12], a link quality source routing (LQSR) protocol was proposed which selects a route according to a specified link quality metric. LQSR is an extension of the dynamic source routing protocol [13]. In [12], three different link quality metrics: ETX, per-hop round-trip time, and per-hop packet pair, were evaluated and compared, along with LQSR. However, LQSR was designed primarily for nodes with a single radio interface.

In [11], the authors promoted the uses of multiple radio interfaces at each mesh router for the improvement of network capacity. Since then, most research on WMNs has adopted this idea. However, while such configurations enable a mesh router to simultaneously transmit and receive packets, it also complicates the selection of routes. It has been shown that finding the optimal route for a given source-destination pair with the best radio and channel in a multi-radio WMN is an NP-hard problem [7].

In [11], a multi-radio LQSR (MR-LQSR) was proposed for mesh routers with multiple radio interfaces. MR-LQSR incorporates several performance metrics. The Expected Transmission Time (ETT), which is essentially the expected time to transmit a packet of a certain size over a link, is introduced to measure the quality of a link. ETT accounts for both packet loss rate and link bandwidth. The Weighted Cumulative Expected Transmission Time (WCETT) is used to measure the quality of a path. WCETT is a combination of the Summation of ETT (SETT) and Bottleneck Group ETT (BG-ETT), which is the sum of expected transmission time of a bottleneck channel. WCETT takes into account both link quality metric and the minimum hop-count. Depending on the parameter set for SETT and BG-ETT in WCETT, MR-LQSR generally achieves a good tradeoff between delay and throughput. However, MR-LQSR does not consider interference, as the authors assumed that all the radio interfaces on each mesh router are tuned to non-interfering channels. In reality, the number of available channels is limited, so when multiple traffic flows are running on the network the impact of interference should not be overlooked.

In [8], a centralized channel assignment and routing algorithm were proposed. The proposed heuristic improves the aggregate throughput of WMNs and balance loads among gateways. For the channel assignment algorithm, load balancing is the first criterion assessed. The routing algorithm used both shortest path routing and randomized routing.

In [7], in order to solve a joint channel assignment and routing problem, a traffic flow based channel assignment was proposed to maximize the bandwidth allocated to each traffic aggregation point, subject to the fairness constraint. Unlike the heuristic approach in [7,8] took into account the interference constraints at each mesh router in the formulation of the joint channel assignment and routing. As a result, the proposed algorithm was able to increase overall throughput.

The authors in [7,8,9] do not consider the use of scheduling in the event of multiple links being assigned to the same channel. In their algorithms, the mesh routers may need to buffer data packets, introducing extra hardware requirements for mesh routers. Moreover, these algorithms do not consider some of the physical characteristics inherent in the IEEE 802.11 standards, such as an adaptive transmission rate and the existence of multiple neighboring sets for a multi-radio mesh router due to the different transmission ranges of the radios.

3 Cost-Aware Route Selection

3.1 Motivation

Prior studies [8,11] have pointed out the shortcomings of the shortest-path routing approach in WMNs. As a result, most of the proposed route selection schemes for WMNs such as [10,11] are instead based on the quality of links in a route. While these schemes favor routes with higher throughput, they do not take into account the cost of a route. As a result, in cases where multiple active connections are present, these schemes do not scale up well and tend to produce lower overall throughput in multi-radio WMNs.

Figure 1 shows 5 candidate routes between source S and destination D. The values of various metrics for these candidate routes, including the shortest-path, SETT, and BG-ETT, are presented in Table 1. As can be seen, the shortest-path will select path 2 or 4, since these two routes consist of only 3 hops; SETT will favor path 4 because it has the lowest value of SETT among all routes (In general, SETT tends to favor shorter paths.); and BG-ETT will favor path 3 because of its radio diversity. However, none of these metrics considers the channel diversity, as the impact of interference has not been treated as one of the primary factors for route selection. Another drawback of these metrics is that their route selections are based solely on the individual traffic flow instead of multiple simultaneous flows. As a result, none of these metrics will be in favor of traffic aggregation, which will lead to better utilization of wireless resources (i.e., channels).

Given a set of candidate routes, the problem of route selection in WMNs can be considered as a resource allocation problem, where the limited resource is the wireless medium. When a route is an active, it prevents the mesh routers close to it from accessing the channels used by the active route due to the impact of co-channel interference. This limits the available routes for nearby mesh routers for other connections. In this paper, a new route selection scheme, namely Cost-Aware Route Selection (CARS) will be proposed. In this scheme, the interference cost of a route is measured quantitatively. By choosing the route with the highest bandwidth and cost ratio, the overall network throughput can be improved. In the following subsections, this approach will be explained in detail.

3.2 Problem Formulation

The assumptions below were made for the WMN in which the route selection scheme is expected to operate. Note that these assumptions do not conflict with any of the IEEE 802.11 specifications, and the frame format in the specifications is never changed.

- All mesh routers in WMNs are stationary.
- Assume that each mesh router has a set of 802.11 radio interfaces. The type of a radio interface can be either 802.11a [3], 802.11b [3], or 802.11g [3].

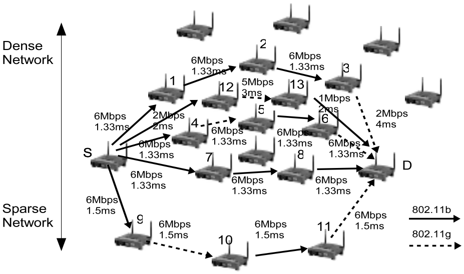


Fig. 1. 5 candidate routes from source S to destination D

Table 1. Performance metrics in Figure 1

path ID	route	hop	throughput	SETT	BG-ETT
1	s-1-2-3-d	4	2Mbps	8ms	4ms
2	s-12-13-d	3	1Mbps	7.0ms	4.0ms
3	s-4-5-6-d	4	3Mbps	5.32ms	2.66ms
4	s-7-8-d	3	2Mbps	4ms	4ms
5	s-9-10-11-d	4	3Mbps	6.0ms	3.0ms

Table 2. Physical characteristics

	802.11b	802.11g	802.11a
Maximum rate	11Mbps	54Mbps	54Mbps
Outdoor trans. range	300 feet	250 feet	175 feet
Indoor trans. range	100-150 feet	100-150 feet	100-150 feet
Non-overlapping ch	1, 6, 11	1, 6, 11	1 - 12
Spectrum	2.4GHz	2.4GHz	5GHz

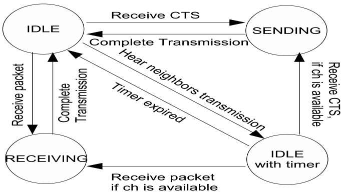


Fig. 2. The state transition diagram of radio interface

- A radio interface is always in one of four MAC states: SENDING, RECEIVING, IDLE and IDLE with TIMER. The transitions between these states are illustrated in Figure 2. The state IDLE with TIMER means that the radio is unused, but some neighboring mesh routers are using the same type of radio.
- Each type of radio has a number of available channels. If a nearby mesh router is using a channel for communications, the state of the channel is set to be IN-USE. Otherwise, the state of the channel is set to be UNUSED.
- Assume that the primary cause of packet loss is co-channel interference. The other factors that may affect the packet transmissions, such as multi-path fading [2], are assumed to be fixed by incorporating simple error recovery techniques (e.g., CRC [15]).
- To simplify the hardware design and lower the cost, assume a mesh router does not have a large data buffer. Packets received by an intermediate mesh router are always quickly forwarded to the next hop.
- The communications between mesh clients and their associated mesh router are assumed to be handled separately by a different set of wireless radio interfaces. In other words, in this research a route consists of only mesh routers.

In WMNs, since each mesh router can be equipped with multiple radio interfaces, the traditional graph denotation is not sufficient to describe the network topology of a WMN. Before formulating the problem, consider the nomenclature used in this paper.

A mesh-graph, $G_M = (V_R, E_R)$, is composed of a set of node vectors V_R and a set of link vectors E_R. A node vector is defined as $v =< n, r >$, where n is a mesh router and r is one of n's radio interfaces. A type function $type(v)$ is defined to take a node vector $v =< n, r >$ as its argument and return the type of radio r (i.e., 802.11a, b, or g) of mesh router n. A link vector $< v_1, v_2 >$, where $v_1 =< n_i, r_p >$ and $v_2 =< n_j, r_q >$, represents a mesh link between sender mesh router n_i using radio interface r_p to communicate with receiver mesh router n_j using radio interface r_q. Note that for a given mesh link $< v_1, v_2 >$, the constraint $type(v_1) = type(v_2)$ must be satisfied.

The open neighbor set of a node vector $N(v)$ $(v =< n, r >)$ is defined as a set of mesh routers within transmission range of the radio transmission r of mesh router n, excluding n itself.

As mentioned earlier, the transmission rate of an 802.11 wireless link may step down automatically when the signal strength is weakened. Since the signal strength is closely related to the distance between sender and receiver, the available bandwidth of a mesh link is formulated as follows. Given a link $< v_1, v_2 >$ where $v_1 =< n_i, r_p >$ and $v_2 =< n_j, r_q >$, the available bandwidth of link $< v_1, v_2 >$, denoted as $b(< v_1, v_2 >)$, is defined by Equation 1. In Equation 1, the available bandwidth of a link is inversely proportional to the physical distance between two ends of the link.

$$b(< v_i, v_j >) = Maximum\, rate \cdot (1 - \frac{dist(n_i, n_j)}{R}) \\ \text{(refer to Table 2 for maximum rate \& transmission range, R)} \tag{1}$$

In WMNs, a path χ connects the source node vector v_s and the destination node vector v_d and is composed of a set of ordered link vectors as follows.

$$\chi = \{< v_s, v_1 >, < v_1', v_2 >, \cdots, < v_h', v_d >\}$$

In a path, any two adjacent links $< v_{k-1}', v_k >$ and $< v_k', v_{k+1} >$, where $v_k =< n_i, r_p >$ and $v_k' =< n_j, r_q >$, should satisfy the constraint $(n_i = n_j) \wedge (r_p \neq r_q)$.

The path bandwidth of a path χ, denoted as $B(\chi)$, is defined as a function of the available bandwidths of the links in the path. Depending on the type of traffic along a path, the function may be defined differently. (Note that the terms *path bandwidth* and *path throughput* are identical and are used interchangeably in this paper.) Two types of traffic are considered in this paper: Burst Traffic (BT) and Constant Bit Rate (CBR) traffic. For the burst traffic, path bandwidth function is defined as the minimum available bandwidth of links in the path as, shown in Equation 2. If a path is assigned more bandwidth than the available bandwidth of any link in the path, an intermediate mesh router will have to buffer the data packets and this violates the no data buffer assumption. For instance, in Figure 1, path 2 has 3 links with 2Mbps, 5Mbps, and 1Mbps available bandwidth. Consequently, the path bandwidth of path 2 is 1Mbps.

$$B(\chi) = \min\{b(< v_s, v_1 >), b(< v_1', v_2 >), \cdots, b(< v_{h-1}', v_h >), b(< v_h', v_d >)\} \tag{2}$$

For CBR traffic, $B(\chi)$ is a constant value b_c. Note that the available bandwidth of any of the links in the path has to be larger than b_c.

Let a set of all active connections be S, so the overall throughput B_{all} is defined as the sum of the path bandwidths of all the paths in S, as shown in Equation 3. The goal of this research is to design a route selection scheme to maximize overall throughput

B_{all} of a WMN, which is the number of bits the WMN can transport between all source and destination pairs simultaneously. The higher the overall throughput B_{all} allows a WMN to support more end-user flows.

$$B_{all} = \sum_{\forall\, \chi \in S} B(\chi) \tag{3}$$

3.3 Physical Layer Constraints

To help formulate the physical layer constraints, we first define the radio state function and the channel state function must be defined. The radio state function $rs(v)$ takes a node vector $v = (n, r)$ as input parameter and returns the state (i.e., SENDING, RECEIVING, IDLE, and IDLE w/ TIMER) of the radio interface r of node n. The channel state function $cs(v, c)$ takes a node vector $v = (n, r)$ and a channel c as its input parameters and returns the state of the channel c (i.e., IN-USE or UNUSED) for radio interface r of node n. Note that even if a radio is in IDLE or IDLE w/ TIMER state, a channel may still be the in IN-USE state if it is used by one of n's neighbors.

To establish a link $< v_1, v_2 >$, where $v_1 =< n_i, r_p >$ and $v_2 =< n_j, r_q >$, the physical layer constraints can be formulated as follows :

1. ***Before establishing link*** $< v_1, v_2 >$
   Resource Allocation:
   $(rs(v_1) = \text{IDLE} \vee rs(v_1) = \text{IDLE w/ TIMER}) \wedge$
   $(rs(v_2) = \text{IDLE} \vee rs(v_2) = \text{IDLE w/ TIMER}) \wedge$
   $\exists\, c_k\ cs(v_1, c_k) = cs(v_2, c_k) = \text{UNUSED}$
2. ***After link*** $< v_1, v_2 >$ ***is established using channel*** c_k
   Resource Allocation:
   $rs(v_1) = \text{SENDING} \wedge rs(v_2) = \text{RECEIVING}$
   Interference Avoidance:
   $\forall\, n \in N(v_1) \cup N(v_2)\ cs(< n, r >, c_k) = \text{IN-USE} \wedge$
   $\forall\, n \in N(v_1) \cup N(v_2) \setminus \{n_1, n_2\}$
   $\forall\, r$ if $type(n, r) = type(n_i, r_p) \Rightarrow rs(< n, r >) = \text{IDLE w/ TIMER}$

To successfully establish a link $< v_i, v_j >$, the components of the link vector and the neighboring routers should satisfy the above constraints. Some of the constraints need to be enforced by the DCF function (e.g., exchange RTS and CTS so the radio interfaces of neighbors will be in the IDLE w/ TIMER state) defined in the 802.11 specification [3].

3.4 Cost and Throughput Metrics

The purpose of WMN research is to facilitate rapid Internet access for a large number of mesh clients. Hence, network throughput should be the primary performance measurement. Since the number of radios and channels in WMNs is limited, if the impact of interference can be reduced when routing a traffic flow, the overall throughput can naturally be increased. In this subsection, two metrics used in our CARS scheme to evaluate a path are introduced, the cost metric that measures the degree of interference of a path, and the bandwidth metric that measures the throughput of a path.

To measure the degree of interference of a path, compute the number of mesh routers that will experience interference along the path if the path is chosen for a connection and becomes active. If all candidate routes provide the same amount of bandwidth between source and destination, by selecting the path which creates the least interference, more network resources (e.g. radios and channels) can be utilized by other traffic flows.

Given an active link $< v_1, v_2 >$ where $v_1 =< n_i, r_p >$ and $v_2 =< n_j, r_q >$, the mesh routers in $N(n_i, r_p)$ cannot use the channel currently occupied by radio r_p of node n_i for communications (see interference avoidance physical layer constraint in Subsection 3.3). Hence, the cost of using the link $< v_1, v_2 >$ can be defined as $|N(v_1)|$. For a given path $\chi = \{e_0, e_1, \cdots, e_k\}$ where $e_i =< v_i', v_{i+1} >$, the cost of a path $C(\chi)$ is defined as follows:

$$C(\chi) = \sum_{i=0}^{k} |N(v_i')| \tag{4}$$

The throughput of a path, on the other hand, is measured by the path bandwidth, $B(\chi)$, which has been defined in Subsection 3.2. In general, we prefer a path with lower cost and higher path bandwidth is preferable.

When mesh routers are distributed uniformly, a shorter path contains a smaller number of hops and thus is likely to suffer less interference from neighbors. In addition, if a specific region has too many active communications, a path traversing that region is likely to result in a lower available path bandwidth. By choosing a path with a higher path bandwidth, a path that goes through a lighter traffic area can implicitly gain priority and load balancing can be achieved.

3.5 Traffic Aggregation

In addition to path metrics, route selection also takes into account traffic aggregation. If a link is simultaneously used by multiple active connections, we say that traffic is aggregated on that link. Suppose that a link $< v_1, v_2 >$ is already a portion of an active connection. If the same link is reused by another connection, This will not create additional interference. In other words, the cost function of a path should take traffic aggregation into consideration. For a given path $\chi = \{e_0, e_1, \cdots, e_k\}$ where $e_i = < v_i', v_{i+1} >$, if a subset of links in the path S have already been used by other active connections, the cost function should be modified as in follows:

$$C(\chi) = \sum_{e_i \in \chi \setminus S} |N(v_i')| \tag{5}$$

Additionally, the definition of the available bandwidth of a link needs to be modified so that the remaining bandwidth of a link can be utilized by aggregated traffic. Given a link $< v_1, v_2 >$, let S be a set of active connections that includes the link, so the available bandwidth of link $< v_1, v_2 >$, where $v_1 =< n_i, r_p >$ and $v_2 =< n_j, r_q >$, is defined as follows:

$$b(< v_1, v_2 >) = Maximum\ Rate \cdot (1 - \frac{dist(n_i, n_j)}{R}) - \sum_{\forall\ \chi \in S} B(\chi) \tag{6}$$

($Maximum\ rate$ and transmission radius R are shown in Table 2)

For instance, suppose that in Figure 1 the links in path 2 have already been used by an active connection and the bandwidth of that path is 1Mbps. The available bandwidth of the first, second and third links of path 2 will then be 1Mbps, 4Mbps, and 0Mbps, respectively.

3.6 Proposed Cost-Aware Route Selection Scheme

In this subsection, the proposed Cost-Aware Route Selection (CARS) scheme for WMNs is presented. The new scheme consists of two steps: radio selection and path selection. For a given source-destination pair and the sequence of intermediate mesh routers between them, the first step is to select the radio and channel to be used for the adjacent mesh routers in the sequence. (Note that according to our path definition, even with the same sequence of intermediate mesh routers, if the radio interface used by any intermediate mesh router is changed the path is considered to be different.) After the radio and channel used for have been intermediate mesh router are identified, a new path metric called CARS is then used to identify the path with the best bandwidth-cost ratio for communications.

Given two adjacent mesh routers n_i and n_j in a sequence within close proximity, up to three sets of radio and channel will be returned as candidates for path consideration. First, the radio n_i with the smallest transmission range (i.e., the smallest number of that neighbors interfere) and one of its unused channels is returned. If no channel of that radio channel is available or n_j does not have an available radio channel with the matched type, the radio n_i with the next smallest transmission range and one of its available channels is returned. This process continues until a set of radio and channel is found. Second, the radio n_i with the highest available bandwidth and one of its unused channels is returned. Similarly, if no channel of that radio is available or n_j does not have an available radio channel with the matched type, the radio of n_i with the next highest available bandwidth and one of its available channel is returned. This process continues until a set of radio and channel is found. Last, these choices are examined to determine if there is an active link from n_i to n_j. If there is, the radio and channel used by the active link with the most remaining available bandwidth will be returned.

After the radio selection step, each sequence of mesh routers between source and destination will produce a number of candidate routes. Given a pair of source and destination, the candidate routes (i.e., the sequence of intermediate mesh routers) are found by doing breadth-first search starting from the shortest path until the number of candidate routes reaches 10000. In Subsection 3.4, two metrics that measure the cost and bandwidth of a path have been introduced. In Equation 7, these two metrics are combined into one single Cost-Aware Route Selection (CARS) metric for path evaluation:

$$CARS(\chi) = \frac{(B(\chi))^{\beta}}{(C(\chi))^{\alpha}} \tag{7}$$

In Equation 7, β is assumed to be $1 - \alpha$ and $0 \leq \alpha$, $\beta \leq 1$. The greater the value of α, the more weight is put on cost for path selection. On the other hand, the greater the value of β, the more weight is put on path bandwidth for path selection. When $\alpha = \beta$, the CARS metric represents the amount of earned bandwidth for a unit of interference cost. By comparing the CARS metrics for the candidate routes, it is possible to identify the most efficient path that produces the most bandwidth per unit of interference. Hence, Equation 7 captures our design goals.

For instance, in Figure 1, if α is assigned a larger value (i.e., cost is heavily weighted), CARS will tend to favor path 5 as the sparse network area path 5 traverses has fewer neighbors to cause interference. On the other hand, if β is assigned a larger value (i.e., more weight is given to path bandwidth), CARS will tend to favor path 3. This is because, according to Equation 1, the available link bandwidth is inversely proportional to the distance between sender and receiver, so the dense network area that path 3 traverses, will tend to have a higher link bandwidth.

Note that for CBR traffic, path bandwidth is a fixed value b_c. Thus, Equation 7 can be simplified as $CARS(\chi) = \frac{1}{C(\chi)}$. Consequently, the CARS metric will give priority to the path with the lower interference cost.

4 Simulation Result and Analysis

This section presents the simulation results in order to evaluate the performance of the proposed CARS scheme. For the purpose of comparison, the other route selection schemes, including the shortest path and WCETT with different values of α, are implemented along with CARS by C++ on different hardware and environment configurations.

4.1 Simulation Environment

In the simulations, mesh routers are randomly placed within a $400m$ by $400m$ two dimensional square region. Each mesh router in our simulation has a small number of radio interfaces. Each interface has a number of available channels. The channels from different types of radio can be either shared or exclusive. Two channels from different types of radio with the same ID are said to be shared if both radios utilize the same spectrum i.e., only one channel can be used at a time. Two channels from different types of radio with the same ID are said to be exclusive if radios are using different spectra i.e., both channels can be used simultaneously. The transmission rate of a link is determined by Equation 6 based on the type of radio and the physical distance between the two ends of the link. The transmission range of a radio is set according to the type of the radio and the location mesh routers (i.e., indoors or outdoors). The values used for the computation of the transmission rate and the transmission range can be found in Table 2.

Two types of traffic flow, BT and CBR, are generated in the simulation. For a BT traffic flow, the rate is computed by Equation 2. For a CBR traffic flow, the rate is set to be 1024Kbps. Additionally, two different network flow patterns, Peer-to-Peer (P2P) and gateway-oriented, are simulated. In a P2P connection, source and destination are mesh routers randomly selected in WMNs. In a gateway-oriented connection, one of the few sinks are used as one end of the traffic flow. Since the primary cause of packet loss is co-channel interference, the ETT of a link in these candidate routes can simply be calculated as the inverse of the available bandwidth of the link.

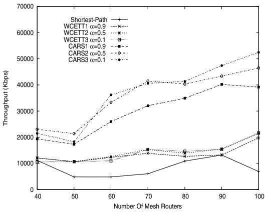


Fig. 3. Total throughput w/ BT, 3 NICs & 3 exclusive channels, P2P, indoors

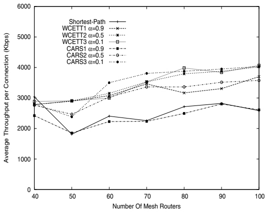


Fig. 4. Average throughput per connection w/ BT, 3 NICs & 3 exclusive channels, P2P, indoors

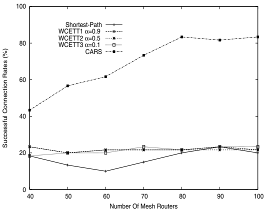


Fig. 5. Successful connection rates w/ CBR, 3 NICs & 3 exclusive channels, P2P, indoors

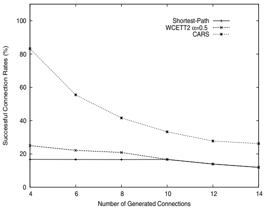


Fig. 6. Successful connection rates w/ 80 mesh routers (3 work as gateways), CBR, 2 NICs & 3 shared channels, indoors

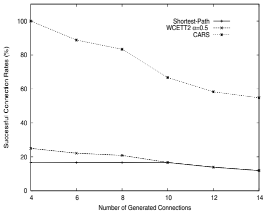


Fig. 7. Successful connection rates w/ 80 mesh routers (3 work as gateways), CBR, 2 NICs & 3 exclusive channels, outdoors

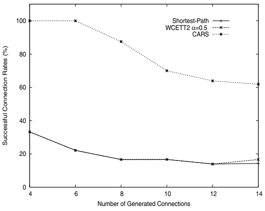


Fig. 8. Successful connection rates w/ 80 mesh routers (3 work as gateways), CBR, 2 NICs & 3 exclusive channels, indoors

4.2 Throughput of Traffic Patterns

In this subsection, the simulation results of different route selection schemes on BT and CBR traffic are presented.

Figure 3 shows the overall network throughput B_{all} for the different route selection schemes for the burst traffic scenario based on the number of mesh routers in the simulated region. In the simulations, a mesh router is set to have 3 Network Interface Card (NIC) radios, and each radio has 3 exclusive channels. P2P connections are generated until the network is saturated. The location of the simulated WMN is assumed to be indoors.

As illustrated in Figure 3, no matter what values of α is used in CARS and WCETT, CARS can always produce more than twice as much of the overall network throughput as WCETT's and the shortest path's. This is a big improvement over the past route selection schemes. While all three different CARS versions have similar performance, the that with $\alpha = 0.1$ is slightly better than the other two. This suggests that the path bandwidth metric is slightly more important than the path cost metric. Additionally, the overall network throughput of the three different CARS versions is a lot more responsive to an increase of the number of mesh routers in any of the simulated region than the other route selection schemes. This suggests that the new CARS scheme is more scalable in terms of overall throughput. This feature is especially important for

WMNs. In addition, Figure 4 shows the average path throughput of different route selection schemes with respect to the number of mesh routers in the simulated region under the same simulation settings. As illustrated in Figure 4, the schemes that produce the highest average path throughput are CARS with $\alpha = 0.1$, WCETT with $\alpha = 0.1$, and WCETT with $\alpha = 0.5$. The average path throughput of CARS with $\alpha = 0.5$ and WCETT with $\alpha = 0.9$ is just slightly lower than the highest group. It is interesting to observe from Figure 4 that CARS with $\alpha = 0.1$ achieves a significant improvement in the overall network throughput without sacrificing individual path throughput.

Figure 5 shows the successful connection rates for different route selection schemes in the case of the CBR traffic scenario with respect to the number of mesh routers. In these simulations, each mesh router is set to have 3 NIC radios, and each radio has 3 exclusive channels. 20 P2P-type connections are attempted. The location of the simulated MWN is also assumed to be indoors. Note that for the CBR scenario, the CARS metric is essentially reduced to the path cost function.

As illustrated in Figure 5, no matter what values of α is used in WCETT, CARS can successfully establish more than twice as many connections as either WCETT as the shortest path. This suggests that the cost metric still plays a crucial role in route selection. Additionally, the results of this simulation suggest that CARS allows more mesh clients to be supported than either WCETT or the shortest path. This feature is also very important for WMNs.

4.3 Successful Connection Rates of Shared and Exclusive Channels

In this subsection, the simulation results of different route selection schemes for shared and exclusive channels are presented. Here, each mesh router is set to have 2 NIC radios, and each radio has 3 channels. The network size is fixed at 80 routers with the assumption that 3 of them work as gateways to connect to the Internet. Gateway-oriented CBR connections are used and the WMN is assumed to be indoors.

Figure 6 shows the successful connection rates of different route selection schemes with respect to the number of generated connections in the simulated region in the case of the shared channels. As illustrated in Figure 6, CARS has more than twice of the successful connection rate of either WCETT's or the shortest path. This suggests that CARS also performs well for the gateway-oriented connections. However, when the number of generated connections increases, the successful connection rates for CARS decreases. This is because the wireless resource (i.e., radios and channels) close to the gateways is quickly exhausted.

Figure 8 shows the successful connection rates of different route selection schemes with respect to the number of generated connections in the simulated region in case of the exclusive channels. In Figure 8, the successful connection rates for CARS are approximately three times the rates for WCETT and the shortest path. This is because the assumption of exclusive channels actually means less possibility of interference. In other words, more resources are available in the case of the exclusive channels. When Figure 6 and Figure 8 are compared together, it can be seen that the successful connection rates of the other route selection schemes are not sensitive to the extra resources than become available when the channel type switches from shared to exclusive. This suggests that CARS can better utilize the extra resources in the network.

For the purpose of comparison, Figure 7 shows the successful connection rates of different route selection schemes with respect to the number of generated connections in the simulated region for the case of exclusive channels under the same configuration, with the only difference being that the network is located outdoors. As can be seen, the rates in Figure 7 are slightly lower than the rates in Figure 8. Because in the outdoor case, the transmission range is increased, as indicated in Table 2. At the same time, more interference will be created when a path is established.

5 Conclusion and Future Works

In this paper, a novel route selection scheme, namely Cost-Aware Route Selection, is proposed for WMNs to improve the overall throughput. The scheme incorporates a path metric which captures the bandwidth and cost ratio and the introduces idea of traffic aggregation. Simulation results show that the new CARS scheme improves the overall throughput by up to 165% in the case of the burst traffic and boosts the number of connections by up to 300% in the case of constant bit rate traffic and is also more scalable in terms of the size of the network compared to both WCETT and the shortest path route selection schemes.

Although for a given set of candidate routes this scheme is able to identify the best choice to improve overall network throughput, it has yet to completely solve the routing issue as no protocol is provided to locate those candidate routes. In addition, the new route selection scheme is centralized in the sense that the source node must collect and process all the necessary information. While the nature of WMNs (i.e., mesh routers are fixed and connected to external power supplies) allows this assumption to hold, future research on a distributed routing protocol that runs only on the basis of localized information would definitely be of interest.

References

1. Ian F. Akyildiz, and Xudong Wang, “A Survey on Wireless Mesh Networks,” *IEEE Radio Communications*, Sep. 2005.
2. C.Siva Ram Murthy and B.S. Manoj, “Ad Hoc Wireless Networks :Architechtures and Protocols,” *Peason Education*, ISBN 013147023X, 2004
3. IEEE 802.11 The working group setting the standards for Wirless LANs, http://grouper.ieee.org/groups/802/11/
4. K. M. Alzoubi, P.-J. Wan, and O. Frieder, “Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks,” *Proc. Third ACM Int’l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.
5. F. Kuhn, R. Wattenhofer and A. Zollinger, “Ad-hoc networks beyond unit disk graphs,” *Proc. Joint Workshop on Foundations of Mobile Computing*, 2003
6. K. Siwiak, “Advances in Ultra-Wide Band Technology,” *Radio Solutions*, 2001.
7. M. Alicherry, R. Bhatia, and L. Li, “Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks,” *Proc. Mobile Computing and Networking (MobiCom)*, pp. 58-72, Aug. 2005.
8. A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks,” *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*, Vol 8, No 2, Apr. 2004.

9. A. Raniwala and T. Chiueh, “Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network,” *Proc. IEEE Conference on Computer Communcation (INFOCOM)*, Vol 3, pp. 2223- 2234, Mar. 2005.
10. D.S.J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing,” *Proc. ACM International Conference on Mobile Computing and Networking Review*, Sep. 2003.
11. R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 114-128, Sep. 2004.
12. R. Draves, J. Padhye, and B. Zill, “Comparisons of Routing Metrics for Static Multi-Hop Wireless Networks,” *Proc. ACM Annual Conf. Special Interest Group on Data Communication (SIGCOMM)*, pp. 133-144, Aug. 2004.
13. D. B. Johnson, D. A. Maltz, and Y-C Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR),” *IETF Mobile Ad Hoc Networks Working Group*, Internet Draft, Feb. 2003.
14. Infopeople Webcasts, http://www.infopeople.org/training/webcasts/03-01-05/Wireless_Webcast.ppt
15. D.V. Sarwate, “Computation of Cyclic Redundancy Check via Table Look-Up,” *Communications of the ACM*, Vol.31, No.8, Aug. 1988.

Novel Route Metric for High-Throughput in Multi-rate Wireless Ad Hoc Networks

Joo-Sang Youn and Chul-Hee Kang

Department of Electronics and Computer Engineering, Korea University,
5-1ga, Anam-dong, Sungbuk-gu, Seoul, Korea
{ssrman, chkang}@widecomm.korea.ac.kr

Abstract. This paper presents the route metric, the Expected Cumulative Service Latency (ECSL), which allow any routing protocol to select a route with the high-throughput in the multi-rate wireless ad hoc networks. The ECSL route metric finds a route with minimal end-to-end forwarding time expected to successfully deliver a packet to the destination. In such networks, existing route metrics for route discovery only use the link quality metric based on the expected amount of medium time it would take by successfully transmitting a packet. Thus, in these schemes, the current state of traffic in network is not taken into account. Therefore, this may easily result in the congested network. In addition, the application has no way to improve its performance under a given network traffic condition. In this paper, we propose the route metric, the expected end-to-end latency that is estimated with the number of packets waiting for transmission in queue at relay nodes and the link transmission time as well as a new routing protocol using the proposed metric. The performance of the proposed route metric is evaluated by OPNET simulator. We show that the performance of the AODV routing protocol using the ECSL route metric outperforms that of the routing protocols using other existing route metrics in multi-rate ad hoc network.

1 Introduction

Wireless ad hoc networks are formed by a set of nodes connected to others through wireless links. Recently, these networks have been adopted in commercial environments. In addition, as various wireless networks technology evolves into the next generation internet to provide better performance for diverse applications, key technologies, such as multi-rate technology and the multi-channel assignment schemes for enabling multi-interface, have emerged recently [1]. We also believe that the development of techniques for QoS requirement, such as high end-to-end throughput and low end-to-end latency, is important for many of the communal applications likely to be enabled by such networks. In such networks, the relatively low spatial reuse of a single radio channel in multi-hop wireless environments due to wireless interference remains an impediment to the wide-spread adoption of wireless ad hoc networks. Also, as the number of nodes are increased in single-channel wireless networks, it has been shown that

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 185–196, 2006.

network capacity decreases [2]. To this problem, recent advancements in wireless technology rendering the multi-rate scheme and a multi-channel scheme for the usage of multi-interface are used to improve the capacity of wireless ad hoc networks. Specially, with multi-rate and multi-channel schemes, to get the high capacity of networks, it is necessary the combined route discovery through the routing protocol, the medium access control protocol, and physical properties of a wireless network. Researchers have proposed many metrics for the combined route discovery to find a route between a source and a destination. The traditional technique used by existing ad hoc routing protocols selects routes with minimum hop-count. The selected routes tend to contain long range links that have low effective throughput and intermediate nodes with high queuing delay. In addition, each node in multi-rate ad hoc networks can utilize the flexibility of multi-rate transmissions to make appropriate range and throughput / latency trade-off choices across a wide range of channel conditions. While, this flexibility has traditionally been used only link condition, it has recently been proposed for use in route metrics with the expected lowest cumulative link transmission time in terms of the throughput as well [8, 11]. However, these metrics do not find high throughput path in the networks where there is much amount of traffic, which because the value about queue delay in intermediate nodes is not taken into account in exiting route matric. therefore, in this paper, new route metric, with considering both the amount of traffic per link in each node and link reliability, is proposed.

This paper makes the following main contribution. First, it explores the problems of existing route metrics to be used to route discovery. Second, it presents the design and implementation of the proposed route matric, and the evaluation of the proposed route metric that affect route discovery in multi-rate wireless ad hoc networks.

The remainder of the paper is organized as follows. Section 2 presents the related works of multi-rate technology in ad hoc networks. Section 3 describes the problems of exiting route metrics. Section 4 and 5 describes the proposed route metric and the proposed routing protocol. Section 6 summarizes results of the simulation studies. This paper concludes in section 7.

2 Related Works of Multi-rate Technology

Multi-rate transmission technologies based on the 802.11 standard [5] is to leverage information which is already being collected by the MAC and Physical layers. An alternate technique used in [4] is to perform active probing at the network layer in order to measure loss rates and estimate link speeds. This approach is unable to take advantage of the more advanced channel quality estimators which are available at the lower layers. In addition, active probing techniques introduce additional network overhead proportional to the accuracy and rate at which they gather information. In this work, we strongly advocate inter-layer communication; particularly between the MAC and Network layers. Several auto rate protocols have been proposed. The most commonly used protocol is Auto Rate Fallback (ARF) [6]. ARF operates using the link level ACK frames specified by the 802.11 standard. Each node increases the rate it is using to communicate with its neighbor after a number of consecutively received ACKs, and decreases the rate after a number of consecutively missed ACKs. As an alternative, the Receiver Based Auto Rate (RBAR) protocol was presented in [7]. RBAR allows the receiving node to select the rate. This is accomplished by using the SNR of the RTS packet to choose the most appropriate rate and communicating that rate to the sender using the CTS packet. This allows much faster adaptation to the changing channel conditions than ARF. In addition, the Opportunistic Auto Rate (OAR) protocol, which is presented in [15], operates using the same receiver based approach, but allows high-rate multi-packet bursts to take advantage of the coherence times of good channel conditions. These bursts also dramatically reduce the overhead at high rates by amortizing the cost of the contention period and RTS/CTS frames over several packets. By picking appropriate sized bursts, OAR also changes the fairness characteristic from each node sending an equal number of packets to each node getting an equal allocation of medium time. Therefore, in this paper, OAR is used as multi-rate scheme.

3 Problems on Existing Route Metric

Early exiting routing protocols are originally designed for single-rate networks, and have used a min hop-count to select routes. Thus, it does not accurately capture the trade-off present in multi-rate wireless ad hoc networks. In [9], the Expected Transmission Count Metric (ETX) is proposed to select paths which minimize the number of transmissions required to transfer a packet from a source to a destination. To deal with multi-rate links, [10] defines the medium-time metric (MTM) for each transmission rate. The MTM essentially measures the time it takes to transmit a packet over a multi-rate links. It takes transmission delay into account and the overheads, which in the case of IEEE802.11 includes RTS/CTS/ACK frames and channel contention. In [11], the Weighted Cumulative Expected Transmission Time (WCETT) is proposed to use a route metric for the routing in multi-radio multi-hop static wireless networks. The WCETT refer to the combination of the MTM with ETX. However, these metrics only take link quality into account by having the metric inversely proportional to the transmission rate.

To understand the problem of the route metric only taking link quality into account, we consider simple topology shown in fig. 1. there are 3 links to a node from its neighbor nodes. We assume all links are asymmetric. The transmission rate and quality of each link is shown in fig. 1. The link quality based route metric makes a determination of link cost as to the following two parameters: current bit rate in use, that is, the modulation mode and packet drop rate at the current bit rate for a data frame with a 1000 byte payload. We assume that the source node wants to send a packet to destination. There are possible two paths: the direct path through link 1 and the alternate path through link 2 and link 3. Using the MTM metric, the link cost of link 1, link 2 and link 3 is 4ms, 1.6ms and 1.5ms, respectively. Thus, route scheme using the link quality metric selects the path using link 2 and link 3 that has the lowest cumulative value. However, if

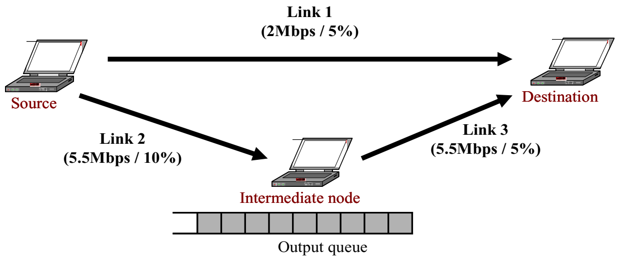


Fig. 1. The simple topology to illustrate the problem of link route metric

the intermediate node has 9 packets waiting for transmission to neighbor node in own output queue, the new packet that arrives at intermediate node have to wait in queue for a considerably long time at this time, which results in a significantly increased end-to-end delay on the path using link 2 and link 3. We believe that if the link bandwidth of the estimated path is admitted by any flow, the direct path between the source and the destination can be more effective than the path through the intermediate node in terms of end-to-end performance. This example illustrates our simple key idea. Therefore, we propose a new route metric that takes into account both link quality and queuing delay estimated with the number of packets waiting for transmission in output queue at relay nodes, and contention delay during route set up.

4 Novel Route Metric

Existing route metrics used to establish a source-destination route are not efficient in multi-rate wireless ad hoc networks because they use the expected transmission time based on only link quality and the reliability of a link, as shown in the previous section. In this paper, assuming that all nodes in the networks are stationary and have the function of multi-rate operation. In such networks, each link operates at a different transmission rate according to SNR between nodes. This section defines a novel route metric. The proposed route metric firstly estimates the per-hop forwarding delay of all nodes on a route between a source-destination pair. Per-hop forwarding delay of each node is estimated with the number of packets waiting for transmission per link at each nod, mean contention delay and link transmission time and then with the predicted per-hop service delay of each node, end-to-end service latency is made. Thus, we make more pertinent information about end-to-end service latency than existing strategies. Therefore, in this metric, the cost function for establishing a route combines two route costs consisting of the traffic load-aware and link-aware route cost. In the following subsections more details are given, which illustrates the estimation of the proposed route metric.

4.1 Expected Cumulative Service Latency

In this paper, the novel route metric, called the Expected Cumulative Service Latency (ECSL), is proposed to allow any routing protocol to find a route with the lowest end-to-end latency. The ECSL of a route is defined as the expected end-to-end latency spent in transmitting a packet successfully over the route between a source and a destination. Thus, to estimate ECSL value, each node has to get per-hop forwarding delay which is the time spent in forwarding a new arrival packet to a neighbor node. The ECSL value is determined by the total sum of per-hop forwarding delay in all nodes over a route between a source -destination pair. Thus, the ECSL of a route, p, consisting of h-hop between a source and a destination is estimated as follows:

$$\mathrm{ECSL}(p) = \sum_{n=1}^{h} d_{n,i} \tag{1}$$

where let $d_{n,i}$ be per-hop forwarding delay which is a time when a packet is forwarded through link i at node n and h be hop counter between a source-destination pair.

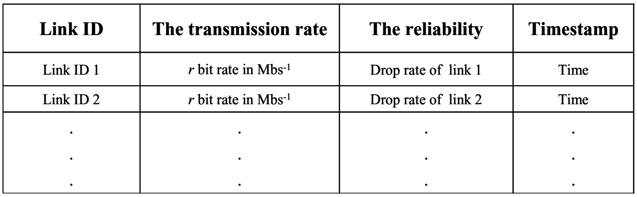
In order to estimate $d_{n,i}$ value, three values which are the Link Transmission Time (LTT) for successfully transmitting a packet on each link, the number of waiting packet and the contention delay of a node are need. In our scheme, to get these values, each node measures the number of waiting packet per neighbor node in queue, the packet drop on wireless link and average contention delay. With estimated values through the measurement, a node gets per-hop forwarding delay by multiplying the number of waiting packet per link in output queue at each node together both each link transmission time and average contention delay. Thus, $d_{n,i}$ is estimated as follows:

$$d_{n,i} = \sum_{i=1}^{x} \left(N_{n,i} \times \left(\overline{ct}_n + LTT_{n,i}\right)\right) + LLT_{n,i} \tag{2}$$

where let $N_{n,i}$ be the number of waiting packets which waits to be transmitted to a neighbor node through link i, the $\mathrm{LTT}_{n,i}$ be the link transmission time of link i at node n and $\overline{ct}_n$ be the average contention delay at node n. Assuming that there are x-neighbor nodes in transmission range of node n.

4.2 Per-hop Forwarding Delay

We have to estimate the three types of value which are the number of waiting packets, contention delay, the LLT and the reliability of each link.



Link ID	The transmission rate	The reliability	Timestamp
Link ID 1	*r* bit rate in Mbs^{-1}	Drop rate of link 1	Time
Link ID 2	*r* bit rate in Mbs^{-1}	Drop rate of link 2	Time
. . .	. . .	. . .	. . .

Fig. 2. Hello message structure

The estimation of the number of waiting packet. In order to estimate the number of waiting packets, $\overline{N}_{n,i}$, we estimate only data packet, except control packet. N value is estimated per a link. We have to obtain the link information used by an incoming packet. we get this information through the address of next hop in packet heard before a packet send to link layer. The number of waiting packets of output queue is measured every at short-term interval τ. Let $n_{n,i}$ be the current number of waiting packet which wants to use link i to send to a neighbor node at node n. The $\overline{N}_{n,i}$ value at the kth times is estimated as follows:

$$\overline{N}_{n,i} = \alpha\ \overline{N}_{n,i}(k-1) + (1-\alpha)n_{n,i} \tag{3}$$

where the parameter α is the weighting factor and α <1, whose best value has been computed to be 0.8 following a comprehensive simulation under traffic conditions.

The average contention delay estimation. The contention delay is defined as the time consumed for the head-of-line packet to be transmitted to the physical layer and is used to estimate the overhead of the transmission in the contending area. The contention delay includes the period for successful RTS/CTS exchange, if this exchange is used for that packet. Similarly, if the initial transmission of the packet is delayed due to one or more collisions generated by other nodes within the transmission range, multiple numbers of back-off periods may also be included. $\overline{ct}_{n,k}$is estimated as the running average contention delay of all packets belonging to the kth packet transmitted at node n. The weighted moving average is used to smooth the estimated value. Therefore, the mean contention delay is updated as follows:

$$\overline{ct}_{n,k} = \beta\ \overline{ct}_{n,k-1} + (1-\beta)n_{n,k} \tag{4}$$

where the parameter β is the weighting factor and β <1, whose best value has been computed to be 0.7 following a comprehensive simulation under traffic conditions and $n_{n,k}$ is the contention delay achieved by kth packet. The initial vale $ct_{n,0}$ is set to a value adding the slot-time of DIFS to the slot-time of the middle value between CW$_{min}$ and CW$_{max}$. Moreover, if a node uses multi-radio, this value is estimated per radio.

The estimation of the link transmission time. The LLT is estimated with the three information of link state which are the transmission rate of each link, and the reliability of each link which is estimated with the drop rate per link. How to estimate the three types of link state are presented. The LTT assigns a weigh to each link that is equal to the expected amount of medium time it would take by successfully sending a packet of some fixed size S on each link in the networks. The value depends on the link bandwidth and link reliability related to the drop rate on wireless link. LTT$_{n,i}$ is, first, defined as the link transmission time spent by sending a packet over link i at node n. This value is approximated and designed for ease in implementation and inter-operability. The LTT of each link is calculated as:

$$\mathrm{LTT}_{n,i} = \left[O_{control} + \frac{S_p}{r}\right] \times \frac{1}{1-R_{n,i}} \tag{5}$$

where the input parameters, r and R_i, are the bit rate in Mbs^{-1} and the frame drop rate of link i for frame size S_p, respectively. The rate, r, represents the rate at which the node would transmit a frame of standard size (S_p) based on current conditions. The R_i value is dependent on our proposed estimation below. In [3], the overhead of control ($O_{control}$) and S_p is defined as the value of 110μs and 8224, respectively.

The estimation of the reliability of a link. The reliability of each link is estimated through the number of a packet drop. The packet drop happens in the output queue of a specific interface i and on specific wireless link. However, in this paper, the packet drop on wireless link, called a collision drop, is only taken into account. In the case of a collision drop, a packet is dropped due to consistently failing retransmissions. This drop is discarding packets because the MAC's retransmission count, 3 for RTS and 7 for data packet, is expired. For convenience, the reliability, $\overline{R}_{n,i}$, is computed in a straightforward manner as follows:

$$R_{n,i} = \frac{D_{n,i}}{L_{n,i}} \tag{6}$$

where $D_{n,i}$ is the cumulative number of a packet drop on link i and $L_{n,i}$ is the cumulative number of a packet transmission of link i at node n.

5 Route Discovery

In this section, a route discovery using new route metric is proposed. It is called the AODV supporting Minimum end-to-end Latency (AODV-ML) in multi-rate multi-interface environment. It is a modified version of AODV [13] to support the ECSL metric. The AODV-ML protocol based on the basic AODV functionality, including route discovery and route maintenance, is implemented. In addition, this protocol includes the modified hello message to maintain both the reliability and transmission rate between a node and its neighbor nodes. We assume that the each link-quality is not symmetric. In considering link pairs between node a and node b, the transmission rate of the link pairs is the same, but the packet drop rate between the two links is different. this indicates that the reliability of two link between node a and node b is different. In the AODV-ML protocol, we make the table to maintain the reliability and the transmission rate of each link of all neighbor nodes. This protocol uses a proactive mechanism to update these value in the table. In order to obtain information regarding the transmission rate between a node and its neighbor nodes, and update the table of link metrics, the hello message in the AODV protocol is used. In the Hello message, the information of the transmission rate and the reliability of link i is appended onto the modified hello message. In this paper, the algorithm for multi-rate decisions between nodes is available at [7]. Once a hello message is received, a node updates the link information of neighbor node transmitting hello message in the table. In the AODV-ML protocol, the route discovery is performed in the following way. First, When a node receives a RREQ message including both a source and a destination address, it includes its per-hop forwarding delay, including the transmission time of a link which is used to transmit the RREQ message to its neighbor node, in the RREQ message. When the destination sends a route reply, the reply carries back the complete list of the per-hop forwarding delay of all nodes between a source-destination pair for the route.

6 Simulation Studies

To illustrate the effectiveness of the ECSL route metric, with comprehensive simulations, the AODV-ML protocol is evaluated and compared with other routing protocols based on the Min-HOP (MHOP) and the MTM metrics. These routing protocols represent the performance of the routing discovery schemes based on the min hop route selection and the minimum link transmission time route selection. For the simulations we consider the two topologies; one topology which is the multi-rate ad hoc networks consisting of single-interface nodes and the other topology which is the networks consisting of multi-interface nodes. All radios use auto-rate. With automatic rate control, the available rate between neighbor nodes are set from 6 Mbps to 54 Mbps. RTS/CTS are enabled. Each nodes are arranged such that several multi-hop routes to the destination are available. Both topologies are random topologies. Simulations are conducted using OPNET v11.5 simulator [12]. In these simulations, the TCP throughput, according to amount of traffic-load in the network, is studied.

6.1 Multi-rate Single-Interface Ad Hoc Networks

The performance of the our route metric in multi-rate single-interface ad hoc networks is discussed. In the simulation, there are 40 nodes in a fixed area A of 2000m x 2000 meters and the number of TCP flows varies from 5 to 25. Each TCP flow lasts until simulation time ends and sends as much data as possible. The simulation continues for 300s. The metrics used in measuring the metric's performance are the average throughput of all TCP flows and the distribution of path length. The simulation results of the average throughput are presented in table. 1. Through the results, it has been proven that the average TCP throughput of all flows using the AODV-ML protocol is better than using existing routing protocols based on other metrics, such as the MHOP and the MTM. As expected, in the simulation with low traffic load (5 flows), the average TCP throughput of all route metrics is identical. This is because these metrics almost select the same path and the amount of traffic load in the current network does not result in the saturated networks. However, as the number of TCP flow increases, the full potential of the ECSL metric is revealed. In the simulation with high traffic load, the improvement in the average TCP throughput of the AODV-ML protocol, as compared with other metrics based routing protocols, is shown. This is, the average TCP throughput on all routes through the ECSL metric is increased up to 300% compared with one on all routes through the MHOP metric and up to 150% through the MTM metric. This is because, as queue delay increases at relay nodes, the AODV-ML protocol selects routes that consist of both low queue delay and high link quality. It is verified that the route discovery using the ECSL metric almost provides end-to-end routes with high-throughput. The distribution of the path length on 25 TCP flows is shown in fig. 3(a). In the case of the MHOP metric, the routing protocol mostly selects 1-hop, 2-hop and 3-hop paths, regardless of the reliability of a link. Thus, this protocol performs well when the amount of traffic load in the networks is low, and performs poorly when the amount of traffic load is high. The MTM, however, mostly selects the 2-hop and 3-hop paths. Also, it selects 5-hop and 6-hop paths. This is, longer paths yield increased throughput than shorter paths because the path through the MTM metric utilizes the extra medium time available in long paths. However, even though the MTM selects paths with high link quality, this easily results in the network being overloaded. This is because each flow selects a similar path without considering the congested node in the network. In addition, in this situation, the application has no way of improving performance under a given network traffic condition. However, in the case of the ECSL, paths with the hop-count from 2 to 5 is usually selected. This means that load-balancing of traffic works well. Thus, when using other two metrics, the selected routes tend to contain long range links that have low effective throughput and high reliability. However, the ECSL selects a route which consists of generally clear nodes (low congested node). Thus, this results in an increase of the overall network throughput. These results show the importance of taking both link quality and queuing delay in relay nodes into account. Also, when a source wants to select routing paths with high throughput in multi-rate wireless networks, the ECSL metric provides this.

Table 1. The average throughput of all TCP flows (Mbps) in multi-rate single-interface ad hoc networks

Number of TCP flows	MHOP metric	MTM metric	ECSL metric
5	11.89	12.48	12.56
10	8.26	8.56	11.36
15	2.64	5.77	7.45
20	1.26	2.84	4.78
25	0.25	1.47	3.78

6.2 Multi-rate Multi-interface Ad Hoc Networks

To describe the performance of the ECSL metrics in multi-rate multi-interface ad hoc networks, in this simulation, we assume that all nodes have two radios which are composed of 802.11a radio and 802.11g radio. 802.11a radio operates on channel 36 and 802.11g radio operates on channel 10. Both radios use auto-rate. The available rate on both radios are also set from 6 Mbps to 54 Mbps. The simulation environments are the same as previous scenarios. As expected, the improvement in average TCP throughput of the AODV-ML protocol using

Table 2. The average throughput of all TCP flows (Mbps) in multi-rate multi-interface ad hoc networks

Number of TCP flows	MHOP metric	MTM metric	ECSL metric
5	19.85	19.43	20.58
10	10.67	14.47	17.67
20	4.56	6.56	13.47
30	1.46	3.36	5.67
40	0.78	1.89	4.13

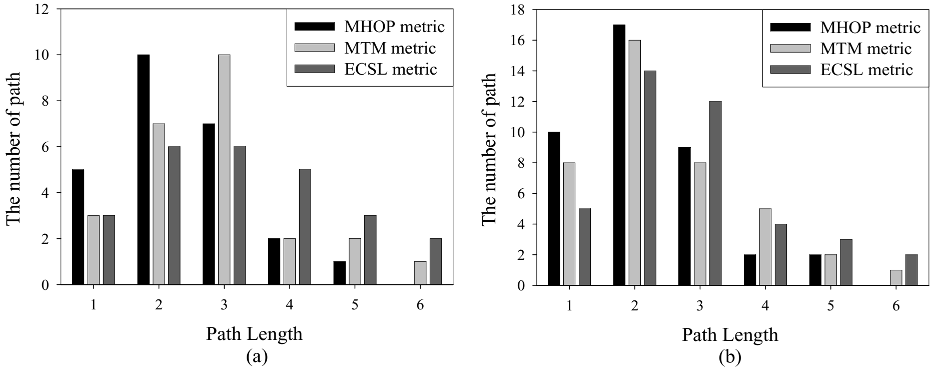


Fig. 3. The distribution of path length. (a) Results of 25 TCP flows in multi-rate single-interface ad hoc networks. (b) Results of 40 TCP flows in multi-rate multi-interface ad hoc networks.

the ECSL, as compared with other metrics based routing protocols, is shown. In particular, as the number of each TCP flows increase, the full potential of the ECSL also is revealed. The average TCP throughput using the our protocol yields more up to 300% than the MHOP based routing protocol and up to 200% than the MTM based routing protocol in higher traffic conditions. Therefore, the ECSL consistently selects the highest throughput path available in the networks. It is also verified that the ECSL metric almost provides high end-to-end throughput in multi-interface environments.

The distribution of the path length of 40 TCP flows in such environments is illustrated in fig. 3(b). The MTM usually selects paths with 3-hop and 4-hop through link quality. Longer paths yield increased throughput than shorter paths because the MTM path utilizes the extra medium time available in long paths. However, the ECSL usually selects paths with 2-hop, 3-hop and 4-hop. This also means that load-balancing of traffic works well in multi-interface environments. When using ECSL, the selected paths tend to contain long range links that have effective throughput and low packet drop rate, as compared with the MTM metric. These results show that the ECSL metric can select routes with high throughput in multi-rate multi-interface environments.

7 Conclusion and Future Work

In our work, we have shown that existing route metrics which only consider link quality are low effective throughput in multi-rate multi-interface ad hoc networks with high traffic load and tend to increase overall network congestion because these mechanisms do not take the current traffic load in a node into account. Thus, in such networks the application has no way to improve its high performance under a given network traffic condition. We have presented the novel route metric adapting per-hop forwarding delay. This metric is particularly used for a source node to find a route supporting a high throughput. This metric is proportional to the time, included both a forwarding time and the contention delay at a relay node. Especially, when congestion in networks occurs, this metric is effective. In addition, a new routing protocol using the proposed metric, called the AODV-ML protocol, is presented. Our simulation results show that the AODV-ML protocol achieves significantly higher end-to-end throughput and lower end-to-end delay than the routing protocols using alternative metrics. We observe the better average throughput of all TCP flows, as compared with the MHOP and the MTM metrics, in single-interface environments and two-interface environments. Our simulation results underscore the need for route metric embodied forwarding time and link transmission in terms of layer-2. Future studies will present the performance of the proposed route discovery using the ECSL metric in wireless ad hoc environments, with random arrivals of mobile nodes.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

References

1. I. Akyildiz, X. Wnag and W. Wang, Wireless mesh networks: a survey, Computer networks, Elsevier Science, no. 47, Jan. 2005.
2. P. Gupta and P. R. Kumar, The capacity of wireless networks, IEEE Transactions on Information Theory, IT-46(2):388404, March 2000.
3. The status of the IEEE 802.11s standard project. http://grouper.ieee.org/groups /802/11/reports/tgs/update.htm.
4. A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks. In BroadNets, 2004.
5. IEEE 802.11 Wireless Local Area Networks. http://grouper.ieee.org/groups/ 802/11/.
6. A. R. Prasad and H. Moelard, WaveLAN-II system design note 225: Enhanced data rate control, March 1999.
7. B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, Opportunistic media access for multi-rate ad hoc networks, September, 2002.

8. B. Awerbuch, D. Holmer, and H. Rubens, High throughput route selection in multi-rate ad hoc wireless networks, Lecture Notes in Computer Science, 2928:253-270, 2004.
9. D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, A high-throughput path metric for multi-hop wireless networks, in 9th annual international conference on mobile computing and networking (MobiCom 03), September 2003.
10. B. Awerbuch, D. Holmer, and H. Rubens, High throughput route selection in multi-rate ad hoc wireless networks, in Wireless On-Demand Network Systems (WONS), January 2004.
11. R. Draves, J. Padhye, and B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in ACM MobiCom, September 2004.
12. The OPNET Modeler, http://www.opnet.com/products/modeler/home.html.
13. C. E. Perkins and E. M. Royer, Ad hoc Networking, Ad hoc On-Demand Distance Vector Routing. Addison-Wesley, 2000.
14. D. Clark and W. Fang, Explicit allocation of best-effort packet delivery service, Networking, IEEE/ACM Transactions on, vol 6, issue 4, Aug. 1998 pp. 362-373.
15. G. Holland, N. H. Vaidya, and P. Bahl, A rate-adaptive MAC protocol for multi-hop wireless networks, in Mobile Computing and Networking. 2001, pp. 236-251.

Joint Power Control and Proportional Fair Scheduling with Minimum Rate Constraints in Cluster Based MANET*

Lijun Qian[1], Xiangfang Li[2], Dhadesugoor R. Vaman[1], and Zoran Gajic[2]

[1] CeBCom Research Center, Prairie View A&M University,
Prairie View, TX 77446, USA
{liqian, drvaman}@pvamu.edu
[2] WINLAB, Rutgers University, Piscataway, NJ 08854, USA
{xfli, gajic}@winlab.rutgers.edu

Abstract. In this study, the joint power control and scheduling problem in a multihop TD/CDMA MANET is investigated. A cluster based architecture is adopted to provide scalability and centralized control within clusters, and the corresponding power control and scheduling schemes are derived to maximize a network utility function and guarantee the minimum rate required by each traffic session. Because the resulted optimal power control suggests that the scheduled nodes transmit with full power while other nodes remain silent, the joint power control and scheduling problem is reduced to a scheduling problem. Proportional fair scheduling is selected to achieve the balance between throughput and fairness. The multi-link version of the proportional fair scheduling algorithms for multihop MANET are proposed. In addition, a generic token counter mechanism is employed to satisfy the minimum rate requirements. Service differentiation is also achieved by ensuring different minimum rate for different traffic sessions. Approximation algorithms are suggested to reduce the computational complexity. In networks that are lack of centralized control, distributed scheduling algorithms are also derived and fully distributed implementation is provided. Simulation results demonstrate the effectiveness of the proposed schemes.

1 Introduction

MANET has been the topic of extensive research recently. The lack of fixed infrastructure in MANET introduces great design challenges. One way to reduce the difficulty is by organizing nodes into clusters and assigning certain nodes management functions [1], such as transmission coordination. These nodes are called cluster heads. It has been shown that proper clustering in MANET reduces the complexity of link-layer and routing protocol design significantly and

* The authors would like to thank OPNET Technologies, Inc. for providing the OPNET software. This research work is supported in part by the U.S. Army Research Office under Cooperative Agreement No. W911NF-04-2-0054. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U. S. Government.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 197–208, 2006.

improves the scalability of the protocols [2]. In addition, clustering increases the network capability of supporting Quality-of-Service (QoS) [3].

Because of their poor scalability in a multihop MANET, random access protocols are not an efficient Medium Access Control (MAC) solution [5]. In [4], it is demonstrated that CDMA-based MAC protocols achieve a significant increase in network throughput at no additional cost in energy consumption compared to 802.11x MAC protocols. In this research work, we restrict our interests in clustered TD/CDMA wireless ad hoc networks. It is assumed that each user is assigned a randomly generated orthogonal code. On top of that, time is splited into equal sized slots where only the scheduled users are allowed to transmit in each slot. The cluster head functions as a manager and is responsible for scheduling the transmissions within a cluster. It is assumed that the communication links among cluster heads have sufficient bandwidth such that the bottleneck of the end-to-end traffic between nodes in different clusters resides within clusters. Hence, scheduling intra-cluster transmissions is the main concern in this paper.

Power control is employed in a MANET to control transmission range and keep the network fully connected [6]. It is a physical layer function. However, transmission power has a direct impact on multiple access of nodes by affecting the received Signal-to-Interference Ratio (SIR) at receivers. Hence, power control is strongly coupled with scheduling and has additional functions of reducing unnecessary interference among concurrent transmissions in TD/CDMA based systems [8]. Power control and scheduling is of paramount importance of ensuring the success of multiple simultaneous transmissions and is the focus of this paper. The goal is to study power control and proportional fair scheduling schemes that maximize network utility, maintain fairness among links and guarantee minimum rate of traffic sessions.

The rest of the paper is organized as follows: Section 2 states the wireless network model and formulates the joint power control and scheduling problem with QoS constraints. Both the optimal solution and the low complexity approximations are proposed, together with several algorithms that serve as lower bounds. The proposed algorithms are evaluated by extensive discrete-event simulations in Section 3. Centralized and distributed implementations are discussed in Section 4. Section 5 presents related works and Section 6 contains the concluding remarks.

2 Joint Power Control and Scheduling with Minimum Rate Constraints

In this paper, we assume that the routes for the multiple end-to-end traffic sessions are given. All the links contained in the routes form the set of "active links". Each active link is uniquely identified by its transmitter and receiver. The received SIR at the i^{th} receiver from the i^{th} transmitter is defined by

$$\gamma_i = \frac{h_{ii}p_i}{\frac{1}{L}\sum_{j\neq i} h_{ij}p_j + \sigma^2} \qquad (1)$$

where h_{ii} is the link gain from transmitter i to its designated receiver i. h_{ij} is the link gain from transmitter j to receiver i (active link i's designated receiver). p_i and p_j are the transmission power of transmitters i and j, respectively. σ^2 is the background (receiver) noise. L is the spreading gain for spread spectrum systems. The interference model adopted here is considered more realistic because the aggregated interference from a large number of nodes may not be negligible even if the interference from each of them is small.

In this paper, we assume that each link has variable rate. This rate is bounded by the feasible rate region. The link gains (channel quality) may fluctuate dramatically from one slot to another slot. A scheduling scheme should take advantage of the channel fluctuations, i.e., it should be "channel-aware". The instantaneous data rate of each active link can be evaluated by the Shannon capacity formula (for AWGN channel)

$$R_i = W_i \log_2(1 + \gamma_i) \tag{2}$$

where W_i is the bandwidth occupied by the transmission from the i^{th} transmitter to its designated receiver. Note that this formula gives the achievable rate (upper bound) of the AWGN channel. However, it is justified by the fact that with the current modulation and coding technology it can be closely approximated in most practical scenarios [16].

2.1 Problem Formulation

In this work, we will focus on end-to-end traffic sessions with minimum rate constraints. A guarantee on minimum rate is arguably the simplest possible QoS guarantee. Therefore we believe it is natural that mobile users would expect such an assurance.

Given the routes of multiple end-to-end traffic sessions with minimum rate constraints, let's define the long-term average rate vector $\bar{\mathbf{R}} = (\bar{R}_1, \ldots, \bar{R}_N)$ assuming that there are N active links resulted from routing, and each of the active link has minimum rate constraint $(\bar{R_i}^{min})$. The joint power control and scheduling problem is formulated as the following optimization problem

(P.1)

$$\max_{\mathbf{R} \in \mathcal{R}, \mathbf{p} \in \mathcal{P}} U(\bar{\mathbf{R}}) \tag{3}$$

subject to

$$\bar{R}_i \geq \bar{R_i}^{min} \ , \ \ \forall\, i \tag{4}$$

where the instantaneous rate is determined by equations (1) and (2). $\mathcal{R}$ is the rate region. $\mathcal{P}$ is the set of allowable power vector defined by

$$p_i \leq p_i^{max} \ , \ \ \forall\, i \tag{5}$$

where p_i^{max} is the maximum allowable transmission power of transmitter i. The utility function is of the form

$$U(\bar{\mathbf{R}}) = \sum_i U_i(\bar{R}_i) \tag{6}$$

where each $U_i(x)$ is an increasing concave continuously differentiable function defined for $x \geq 0$. In this work, the network utility function is chosen as $U(\bar{\mathbf{R}}) = \sum_i \log(\bar{R}_i)$ to achieve the balance between network throughput and fairness.

A node can not transmit and receive simultaneously. This primary conflict [14] is resolved by setting the link gain matrix appropriately. For example, if node i is selected to transmit in the current slot, the corresponding link gains where node i is the receiver will be set to zero. The multi-hop nature of the problem **(P.1)** reflects in the fact that the links on the same route require the same minimum rate whereas links on different routes typically have different minimum rate requirements. In other words, the order of the transmissions along a route is implicitly included in the problem formulation.

2.2 Main Results

Before introducing the Multi-link Proportional Fair algorithm with Minimum Rate constraints (MPFMR) to solve the optimization problem **(P.1)**, we observe some useful properties of the optimal solution.

Optimal Power Control

Theorem 1. *The optimal scheme has the property that each transmitting node transmits at full power, i.e.* $p_i = p_i^{max}$ *for some subset* $\mathcal{S}$ *of the nodes and* $p_i = 0$ *for the complementary set* $\bar{\mathcal{S}}$.

The proof can be found in [24]. Note that similar observations are obtained under various different contexts and assumptions [13], [7], [11], [22]. Specifically, the results reported in [7] may be viewed as a special case of the above theorem where the data rate is assumed to be a linear function of SIR instead of the more general form that adopted in this paper. Theorem 1 reveals the bang-bang characteristics of the nodes' transmission power in order to maximize the network's utility. In each time slot, selected transmitting nodes will use the maximum transmission power, while other nodes remain silent.

Scheduling Algorithms

As highlighted by Theorem 1, the joint power control and scheduling problem is reduced to a scheduling problem given the bang-bang characteristics of the optimal transmission power. The scheduling algorithm considered in this paper is the proportional fair scheduling proposed in [17], [18] and further analyzed in [19], [20]. Proportional Fair (PF) scheduling algorithm was proposed and implemented by QualComm for 3G1X EVDO (HDR) downlink. PF algorithm provides fairness among users such that in the long run each user receives the same number of time slots of services. At the same time, PF also takes advantage of channel variations. As pointed out in [21], PF scheduling maintains a balance between fairness and efficiency. However, since PF schedules users one-at-a-time, it needs to be modified for a multihop scenario.

In this paper, we are interested in proposing and studying the multi-link version of the PF algorithms for multihop MANET, called Multi-link Proportional

Fair (MPF). We are particularly interested in their modified versions that accommodate QoS constraints required by multiple traffic sessions. MPF is modified to satisfy minimum rate constraints using a token counter mechanism inspired by the scheme developed for cellular systems [12], thus it is named Multi-link Proportional Fair with Minimum Rate (MPFMR).

MPFMR: *In a time slot k, select the active links*

$$\arg\max_{\mathbf{R}\in\mathcal{R}} \sum_i e^{a_i T_i(k)} \frac{R_i(k)}{\bar{R}_i(k)} \ , \tag{7}$$

where $\bar{R}_i(k)$ is the current average service rate received by link i, $T_i(k)$ is a "token counter" for link i, and $a_i > 0$ is a parameter. The values of average rate $\bar{R}_i$ are updated as in the Proportional Fair algorithm [17]:

$$\bar{R}_i(k+1) = (1-\beta)\bar{R}_i(k) + \beta R_i(k) \ ,$$

where $\beta > 0$ is a small fixed parameter, and $R_i(k)$ is the instantaneous data rate if link i was actually served in slot k and $R_i(k) = 0$ otherwise. The token counter T_i is updated as follows:

$$T_i(k+1) = \max\{0, T_i(k) + \bar{R_i}^{min} - R_i(k)\} \ . \tag{8}$$

MPFMR may be considered as a special case of Multi-link Gradient scheduling algorithm with Minimum Rate constraints (MGMR), which solves the optimization problem **(P.1)**. The proof of optimality of MGMR (and thus MPFMR) follows our previous work in [12], and is given in the Appendix. The token counter T_i provides the key mechanism trying to ensure that the active link i received (long term) service rate stays above $\bar{R_i}^{\min}$. The dynamics of the token counter process $T_i(k)$ (see (8)) is briefly described and interpreted as follows. There is a virtual "token queue" corresponding to each flow i. The tokens "arrive in the (token) queue" (i.e. T_i is incremented) at the rate $\bar{R_i}^{\min}$ per slot. If active link i is served in slot k, then $R_i(k)$ tokens are "removed from the queue" (i.e. T_i is decremented). Thus, if in a certain time interval, the average service rate of flow i is less than $\bar{R_i}^{\min}$, the token queue size T_i has "positive drift", and therefore the chances of flow i being served in each time slot *gradually* increase. If the average service rate of flow i stays close to $\bar{R_i}^{\min}$, T_i will stay around zero and will not affect scheduling decisions.

In this study, we also considered PF scheduling without minimum rate constraint (**MPF algorithm**), and the scheduling rule is $\arg\max_{\mathbf{R}\in\mathcal{R}} \sum_i \frac{R_i(k)}{\bar{R}_i(k)}$.

2.3 Low Complexity Approximations

In this part, we provide a greedy, low-complexity, approximate solution to the optimization problem **(P.1)** that is more suitable for practical implementations.

Greedy algorithms: In each time slot, 1). Create a list by sorting active links in decreasing order of the measure v_i assuming no interference from other active links while computing R_i^0. 2). Add active link j, in order starting from the top of the list, while maintaining and updating the value of $\Phi = \sum_{i \leq j} v_i$, where R_i now takes into account interference from all added active links. 3). Stop if adding the next active link reduces Φ, and allow transmission of all added active links at their peak powers and rates as computed. The measure v_i for different algorithms are $v_i = e^{a_i T_i} \frac{R_i^0}{\bar{R}_i}$, for MPFMR; and $v_i = \frac{R_i^0}{\bar{R}_i}$, for MPF.

We also considered several algorithms that will serve *one* active link in each time slot. These algorithms serve as the lower bounds for performance comparison.

One-at-a-time algorithms: Create a list by sorting active links in decreasing order of the measure v_i assuming no interference from other active links while computing R_i^0. Serve the top on the list. $v_i = \frac{R_i^0}{\bar{R}_i}$, for PF; and $v_i = e^{a_i T_i} \frac{R_i^0}{\bar{R}_i}$, for PFMR. The various scheduling algorithms considered in this paper are summarized in Table 1.

Table 1. Scheduling algorithms for TD/CDMA wireless ad hoc networks

		Proportional Fair
Multi-Link Algorithms	without Min Rate	MPF
	with Min Rate	MPFMR
One-at-a-time Algorithms	without Min Rate	PF
	with Min Rate	PFMR
Implementation		Average rate needed
Comments		Take advantage of diversity and guarantee long-term fairness.

3 Performance Evaluation

One benchmark algorithm is the optimal (centralized) MPFMR algorithm given in the previous section. It gives the best possible performance. Other benchmark algorithms are the one-at-a-time algorithms, which will serve as lower bounds. Discrete-event simulations using OPNET have been performed to evaluate the performance of the proposed algorithms. In order to quantify the performance gain of different algorithms, all the nodes generate traffic such that the network is fully loaded. It is also assumed that the traffic sources are Poisson with different inter-arrival time for different traffic sessions. Packet length is exponentially distributed with mean 1024 bits. The QoS-support capability for specific traffic sessions is measured by the *effective rate along a route/path* ($\bar{R}_r^{eff}$) as the minimum average rate among all the links in the path r, i.e., $\bar{R}_r^{eff} = \min_{i \in r} \bar{R}_i$. Higher effective rate of a path implies higher QoS-support capability.

In this part of the simulation, there are three routes traversing through the network in Fig. 1 with crossover traffic, namely, $r_{II} : A \rightarrow D \rightarrow E \rightarrow H \rightarrow I \rightarrow L$, $r_{III} : B \rightarrow E \rightarrow G \rightarrow J$ and $r_{IV} : C \rightarrow F \rightarrow H \rightarrow K$. Suppose there are each traffic session along each route, and their respective minimum rate requirements

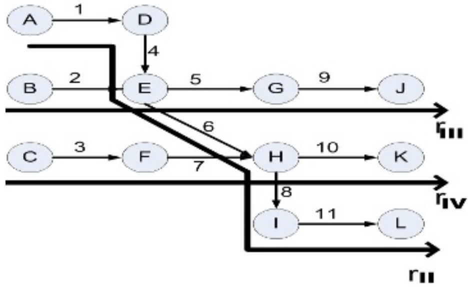


Fig. 1. A TD/CDMA wireless Ad-hoc network with Crossover Traffic

Table 2. Effective rates of route II, III and IV and total average rate (all in kbps) in the network with unbalanced traffic. (G):Greedy algorithm.

Algorithms	$\bar{R}^{eff}_{r_{II}}$	$\bar{R}^{eff}_{r_{III}}$	$\bar{R}^{eff}_{r_{IV}}$	$\bar{\mathbf{R}}$	satisfy min rates ?
PF	69.4	140.1	70.3	187.1	No
MPF (G)	101.8	191.8	101.1	271.6	Yes
PFMR	66.1	179.1	78.5	102.5	No
MPFMR (G)	108.9	226.2	122.3	188.3	Yes

are $\bar{R}^{min}_{II} = 90$kbps, $\bar{R}^{min}_{III} = 190$kbps and $\bar{R}^{min}_{IV} = 100$kbps. Instead of balanced traffic loads along the three routes (r_{II}, r_{III}, and r_{IV}), node A injected a lot of traffic into the network, to be exact, an order of magnitude higher than the other traffic sessions. The performance (especially fairness) of the proposed PF-family of algorithms will be tested against malicious node under multiple traffic sessions.

The results are listed in Table 2. It is obvious that the multi-link PF-family of algorithms still provide the required minimum rates for all the traffic sessions and surpress the disturbance caused by the malicious node. The multi-link gains are significant, 45.2% for MPF and 84.4% for MPFMR, respectively.

4 Centralized vs. Distributed Implementation

4.1 Centralized Implementation

The centralized solution needs a central controller and *global* information of all the link gains. It may be implemented, for example, in a cluster based MANET with "strong" cluster heads where centralized control is not far-fetched. In order to obtain the link gain information, each receiving node needs to measure the received SIR.

At the beginning of each time slot, the central controller will broadcast a scheduling packet (SP) that contains the schedule for all nodes within the cluster. Each node will send an acknowledgement (ACK) that includes the measured

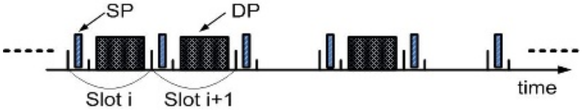


Fig. 2. Slot format in centralized implementation. SP: scheduling packet; DP: data packet.

channel gain. The central controller will decode all the replies and run a channel prediction algorithm to predict all the channel gains for the next time slot. Then it will use the predicted channel gains to calculate the schedule for the next time slot. Note that a seperate control channel may be used for the information exchange between the central controller and each node. Alternatively, it may occupy a small percentage of each slot, as illustrated in Fig 2.

4.2 Distributed Implementation

In wireless ad hoc networks where centralized control is not available, it may be very difficult to obtain the knowledge of all the link gains, and thus it is impractical to implement a centralized solution. A distributed implementation is proposed where only local information is used to perform the power control and scheduling decisions at each transmitting node individually. The procedures are as follows: 1). At the beginning of each time slot, each node i in the potential transmitter set $\mathcal{S}$ select to transmit or not by flipping a coin. (This is motivated by the work of [15] and [13].) 2). Each node that decide to transmit will send a probe packet using power equal to p^{max}. 3). Each receiver detects the probe packets from all transmitting nodes nearby, and estimate the corresponding channel gain. The receiver then sends a packet including information of all the estimated link gains using power equal to p^{max}. 4). Each node i in the potential transmitter set S detects the packets from the receivers within its transmission range. From each of these receivers, node i obtains the list of all possible interfering transmitters and their link gains toward the receiver. Then it will transmit to one of the neighboring receivers where v_i is maximized. 5). Update the token counter according to equation (8) for the algorithms using the token counter mechanism.

Discrete-event simulations have been carried out to examine the performance of the proposed distributed implementation. In this simulation study, only local information is available to each node by exchanging control messages with its neighbors as described above. The overhead of the information exchange includes a one-byte (8 bits) probe packet and the reply from the receiver (which may contain multiple bytes). The exact size of the reply depends on the number of probes that the receiver get. Each link gain in the reply is counted as one byte assuming that the link gain is quantized using a 256-level quantizer. The other parameters of the simulation are the same as in Section 3. MPF and MQR algorithms (please refer to [24] for the description of the MQR algorithm) are selected for comparison in the network with *balanced* crossover traffic.

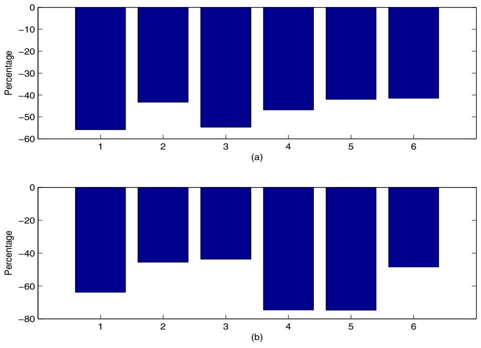


Fig. 3. Gain/Loss of distributed algorithms over their centralized counterparts: (a). Total average rate; (b). Effective rate; 1. MPF (r_{II}), 2. MQR (r_{II}), 3. MPF (r_{III}), 4. MQR (r_{III}), 5. MPF (r_{IV}), 6. MQR (r_{IV})

The percentage of rate gain/loss of distributed algorithms over their centralized counterparts is shown in Fig 3. The total average rate achieved by the distributed algorithms is about 40% less than their centralized counterparts because of lack of centralized control and global information. Because there is no global information about queue backlog or average rate, neither throughput-optimal nor fairness can be guaranteed in the distributed algorithm. The greedy nature of local decisions also results in the bigger reductions (about 50%) in all the effective rates, as expected. The overhead in all the cases is roughly the same 21%. This simple experiment demonstrates that the proposed distributed implementation achieves acceptable performance while keeps the overhead low.

5 Related Works

A power control and scheduling problem has been solved in [10] for TDMA ad hoc networks. The authors assume that each slot has *fixed* data rate rather than variable data rate. A joint time-slot and power allocation method for wireless *cellular* systems is proposed in [23]. Multi-hop scheduling (such as in MANET) is not addressed. In addition, the solution in [23] is much more complicated than that obtained in this paper and thus difficult to implement in reality. A centralized joint routing, scheduling and power control problem is formulated for TD/CDMA ad hoc networks and an approximation algorithm is derived in [14]. However, a simplified interference model is adopted, where no interference is assumed among different links. In [7], a centralized joint routing, scheduling and power control problem is solved for multihop base stations where data rate is assumed to be a linear function of SIR (in low SIR regime). The authors in [9] proposed a joint power control and scheduling scheme based on a utility function of *instantaneous* power or *instantaneous* data rate. The algorithm in [9] focused on a *snapshot* of a set of wireless links. Another work on *instantaneous* power control in wireless ad hoc networks is [8]. A randomized policy is derived to solve the multi-commodity flow problem given the long-term link capacity as weight in wireless networks [13]. Then a dynamic policy (throughput-optimal policy) is proposed for unknown arrival and channel statistics and is proven to perform better than the randomized policy. However, no fairness among users/flows is addressed in such policies. In addition, no minimum rate constraint is considered.

6 Conclusions

In this paper, the joint power control and scheduling problem for TD/CDMA wireless ad hoc networks is formulated using a utility function approach. Because the resulted optimal power control reveals bang-bang characteristics, i.e., scheduled nodes transmit with full power while other nodes remain silent, the joint power control and scheduling problem is reduced to a scheduling problem. The Multi-link Proportional Fair scheduling algorithm with Minimum Rate constraints (MPFMR) is proposed to solve the constrained optimization problem **(P.1)**. A generic token counter mechanism is employed to satisfy the minimum rate requirements. By ensuring different minimum rate for different traffic sessions, service differentiation can also be achieved. Note that the MPFMR algorithm may be modified to accomodate the maximum data rate constraints, by modifying the way that the token counter updated [24]. Maximum data rate constraints may be necessary for mobile device that has limited memory for buffering.

References

1. H. Hassanein and A. Safwat, "Virtual base stations for wireless mobile ad hoc communications: an infrastructure for the infrastructure-less," *International Journal of Commun. Syst.*, vol.14, pp.763-782, 2001.
2. C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas of Communications*, vol.15, no.9, pp.1265-1275, Sep 1997.
3. R. Ramanathan and M. Steenstrup, "Hierarchically-organized, Multihop Mobile Wireless Networks for Quality-of-Service Support," *Mobile Networks and Applications*, vol.3, no.1, pp.101-119, 1998.
4. A. Muqattash and M. Krunz, "CDMA-Based MAC Protocol for Wireless Ad Hoc Networks," *Proc. of MobiHoc'03*, pp.153-164, Annapolis, MD, 2003.
5. S. Xu and T. Sandawi. "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?," *IEEE Communications Magazine*, pp.130-137, Jun 2001.
6. V. Kawadia and P. R. Kumar, "Power Control and Clustering in Ad Hoc Networks," *Proc. of IEEE INFOCOM*, pp.459-469, 2003.
7. R.L. Cruz and A. Santhanam, "Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks," *Proc. of IEEE INFOCOM*, pp.702-711, 2003.
8. T. ElBatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad-hoc Networks," *Proc. of IEEE INFOCOM*, pp.976-984, 2002.

9. J. Fang and R. Rao, "An Integrated and Distributed Scheduling and Power Control Algorithm for Maximizing Network Utility for Wireless Multihop Networks," *Proc. of IEEE MILCOM*, 2003.
10. U.C. Kozat, I. Koutsopoulos and L. Tassiulas, "A Framework for Cross-Layer Design of Energy-Efficient Communication with QoS Provisioning in multihop Wireless Networks," *Proc. of IEEE INFOCOM*, 2004.
11. K. Kumaran and L. Qian, "Uplink Scheduling in CDMA Packet-Data Systems," *Proc. of IEEE INFOCOM*, Vol.1, pp.292-300, San Francisco, Apr 2003.
12. M. Andrews, L. Qian and A. Stolyar, "Optimal Utility Based Multi-User Throughput Allocation subject to Throughput Constraints," *Proc. of IEEE INFOCOM*, Miami, March 2005.
13. M. Neely, E. Modiano, and C. Rohrs, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," *Proc. of IEEE INFOCOM*, pp.745-755, 2003.
14. R. Bhatia and M. Kodialam, "On Power Efficient Communication over multihop Wireless Networks: Joint Routing, Scheduling and Power Control," *Proc. of IEEE INFOCOM*, 2004.
15. M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-hoc Wireless Networks," *Proc. of IEEE INFOCOM*, 2001.
16. R. Berry and E. Yeh, "Cross-Layer Wireless Resource Allocation," *IEEE Signal Processing Magazine*, pp.59-68, Sep 2004.
17. A.Jalali, R.Padovani, and R.Pankaj, "Data Throughput of CDMA-HDR, a High Efficiency - High Data Rate Personal Communication Wireless System," In *Proc. of the IEEE Semiannual Vehicular Technology Conference, VTC2000-Spring,* Tokyo, Japan, May 2000.
18. P. Viswanath, D. Tse and R. Laroia, "Opportunistic Beamforming using Dumb Antennas," *IEEE Transactions on Information Theory*, 48(6), 2002.
19. S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *Proc. of IEEE INFOCOM*, San Francisco, CA, April 2003.
20. H. Kushner, P. Whiting, "Asymptotic Properties of Proportional Fair Sharing Algorithms," *Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing.* Monticello, Illinois, USA, October 2002.
21. L. Jiang and S. Liew, "Proportional Fairness in Wireless LANs and Ad Hoc Networks," *Proc. of IEEE WCNC*, New Orleans, LA, 2005.
22. B. Radunovic, Le Boudec JY. "Optimal power control, scheduling, and routing in UWB networks," *IEEE Journal on Selected Areas in Communications*, vol.22, pp.1252-1270, Sep 2004.
23. J. W. Lee, R. Mazumdar, and N. B. Shroff, "Opportunistic Power Scheduling for Multi-server Wireless Systems with Minimum Performance Constraints," *IEEE INFOCOM'04*, Hong Kong, Mar. 2004.
24. L. Qian and D.R. Vaman, "Joint Power Control and Scheduling in Cluster based MANET," *CeBCom Technical Report*, 2006.

Appendix [Proof of Optimality of MGMR]

MGMR: *In a time slot k, select the active links*

$$\arg\max_{\mathbf{R}\in\mathcal{R}} \sum_{i} e^{a_i T_i(k)} U_i'(\bar{R}_i(k)) R_i(k) \ , \tag{9}$$

where $\bar{R}_i(k)$ is the current average service rate received by link i, $T_i(k)$ is a "token counter" for link i, and $a_i > 0$ is a parameter. The values of average rate $\bar{R}_i$ are updated as in the Proportional Fair algorithm [17]:

$$\bar{R}_i(k+1) = (1-\beta)\bar{R}_i(k) + \beta R_i(k) \ ,$$

where $\beta > 0$ is a small fixed parameter, and $R_i(k)$ is the instantaneous data rate if link i was actually served in slot k and $R_i(k) = 0$ otherwise. The token counter T_i is updated as follows:

$$T_i(k+1) = \max\{0, T_i(k) + \bar{R}_i^{\,min} - R_i(k)\} \ . \tag{10}$$

Proof. We prove the optimality of the MGMR algorithm by studying the dynamics of user throughputs and token counters under the MGMR algorithm when parameters β and a_i are small. Namely, we consider the asymptotic regime such that β converges to 0, and each $a_i = \beta\alpha_i$ with some fixed $\alpha_i > 0$. We study the dynamics of *fluid sample paths* (FSP), which are possible trajectories $(r(t), \tau(t))$ of a random process which is a limit of the process $(\bar{R}(t/\beta), \beta T(t/\beta))$ as $\beta \to 0$. (Thus, $r(t)$ approximates the behavior of the vector of throughputs $\bar{R}(t)$ when β is small and we "speed-up" time by the factor $1/\beta$; $\tau(t)$ approximates the vector $T(t)$ scaled down by factor β, and with $1/\beta$ time speed-up.) The main result is a "necessary throughput convergence" condition stated in the following theorem

Theorem 2. *Suppose FSP (r, τ) is such that*

$$r(t) \to \bar{R}^* \quad as\ t \to \infty$$

and $\tau(t)$ remains uniformly bounded for all $t \geq 0$. Then, $\bar{R}^*$ is a solution to the problem **(P.1)** *and, moreover, $\bar{R}^* \in \mathcal{R}^{cond} \cap \mathcal{R}^* \neq \emptyset$.*

Rate region $\mathcal{R}$ is a convex closed bounded polyhedron in the positive orthant. By $\mathcal{R}^*$ we denote the subset of maximal elements of $\mathcal{R}$: namely, $v \in \mathcal{R}^*$ if conditions $v \leq u$ (component wise) and $u \in \mathcal{R}$ imply $u = v$. Clearly, $\mathcal{R}^*$ is a part of the outer boundary of $\mathcal{R}$. The subset $\mathcal{R}^{cond} \subseteq \mathcal{R}$ of elements $v \in \mathcal{R}$ satisfying conditions $\bar{R}_i^{\min} \leq v_i \leq \bar{R}_i^{\max}$ for all i, is also a convex closed bounded set.

The proof of Theorem 2 follows the approach in [12]. We prove Theorem 2 for the case where there are both minimum and maximum data rate constraints, i.e., $\bar{R}_i^{\min} \leq \bar{R}_i \leq \bar{R}_i^{\max}$. The results apply to **(P.1)** by letting $\bar{R}_i^{\max} = \infty$. Theorem 2 says that if FSP is such that the vector of throughputs $r(t)$ converges to some vector $\bar{R}^*$ as $t \to \infty$, then $\bar{R}^*$ is necessarily a solution to the problem **(P.1)**. This implies that if the user throughputs converge, then the corresponding stationary throughputs do in fact maximize the desired utility function, subject to the minimum rate constraints.

Network Coding Approach: Intra-cluster Information Exchange in Wireless Sensor Networks*

Zhiqiang Xiong, Wei Liu, Jiaqing Huang, Wenqing Cheng, and Zongkai Yang

Department of Electronic and Information Engineering
Huazhong University of Science and Technology
Wuhan, Hubei Province, P.R. China, 430074
fexe@tom.com

Abstract. In this paper, we focus on the intra-cluster information exchange problem and proposed some novel solutions. Firstly, a cluster model is presented and some algorithms based on it are proposed, such as routing, flooding, cluster head relaying and network coding algorithm. The theoretical analysis and simulation comparison of these algorithms are shown subsequently. We find that network coding algorithm allows to realize significant energy and time savings, when each node of the cluster is a source that wants to transmit information to all other cluster member nodes. Energy efficiency directly affects battery life, and delay time is a very important network performance, thus both are critical design parameters in wireless ad hoc sensor networks. Further more, the network coding algorithm we proposed are efficient and implementable. We analyze theoretical cases in detail, and use the packet level simulation.

1 Introduction

The concept of network coding was introduced in a seminal paper by Ahlswede et. al. [1] and immediately attracted increasing interests. Li, Yeung, and Cai [2] showed that it is sufficient for the encoding functions at the interior nodes to be linear, i.e., a code in which each packet sent over the network is a linear combination of the original packets. In a subsequent work, Koetter and Médard [3] developed an algebraic framework for network coding and investigated linear network codes for directed graphs with cycles. This framework was used by Ho et al. [4] to show that linear network codes can be efficiently constructed by employing a randomized algorithm. Jaggi et al. [5] proposed a deterministic polynomial-time algorithm for finding a feasible network code for a given multicast network.

The basic idea in Network Coding is that intermediate nodes in the network not only forward but also process the incoming information flows, which results in significant benefits. In fact, wireless ad-hoc and sensor networks are the most

* This work is supported by the National Natural Science Foundation of China (No.60602029 and No.60572049).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 209–219, 2006.

natural setting for Network Coding because the very characteristics of wireless links that complicate routing, namely, their unreliability and broadcast nature, are the very characteristics for which coding is a natural solution. What's more, the wireless environments offer more freedom in terms of protocol design choices.

Information exchange [6] is the mutual exchange of independent information between two nodes in networks. An explicit example of information exchange in wireless networks using Network Coding is shown in figure 1. Fig. 1(a) shows that node A and B send information to each other via intermediate node C. Node C just plays a store-and-forward role. While in Fig. 1(b), after receiving the information from A and B, node C broadcasts $x_1 \oplus x_2$ (modulo 2 addition) instead of x_1 and x_2 in sequence. Thus, both A and B can recover the information of interest, while the number of transmissions is reduced. Consequently, the transmission energy cost and time consumption are reduced.

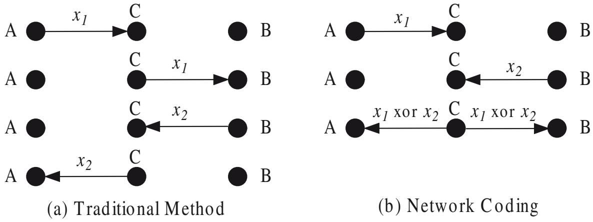


Fig. 1. Information Exchange in Wireless Networks

In this paper, we focus on intra-cluster information exchange in wireless ad hoc sensor networks by using Network Coding. Consider this kind of scenario that each node is a source that transmits information to all other nodes in the cluster. As energy efficiency is very critical to wireless ad hoc sensor networks, we are interested in the minimum amount of energy required to transmit one unit of information from all the sources to all receivers. Such all-to-all communication is traditionally used during routing discovery and routing update phases. Exchange of congestion control information and synchronization in distributed computation environments may require that some information from all the nodes be broadcast to all other nodes of the network. This kind of information exchange in a network is called all-to-all broadcast or gossiping [7]. Another important applications where information exchange is used are in the situation awareness problem [8][9] and personalized exchange [10]. More recently, it has been described as a key mechanism for application layer communication in intermittently connected ad-hoc networks [11].

The main contributions of this paper can be summarized as follows:

- So far as we know, it is the first time to consider the application of Network Coding in clustered wireless ad hoc sensor networks.
- we propose some approaches to achieve information exchange in cluster, which are easy to apply in practical networks.

- The network coding algorithm we proposed is novel and efficient for information exchange. Comparing to other methods, our algorithm significantly reduces the transmission number and so as to the energy and time cost.

The rest of the paper is organized as follows. In section 2, we will show cluster model for wireless ad hoc sensor network. Section 3 describes the algorithms for information exchange: the traditional ways and *network coding* approach. The simulation description and results are given in section 4. At last we will conclude the paper in section 5.

2 Cluster Model

In this paper, we do not concern about how the clusters are formed, but focus on how efficient the information are exchanged. We will formulate the intra-cluster information exchange problem by an ideal topology as shown in figure 2. We consider this kind of scenario: node $A_i (1 \leq i \leq 6)$ wants to broadcast its information(a sequence of packets $\{X_i(n)\}$) to node $A_j (1 \leq j \leq 6, i \neq j)$, and node $A_j (1 \leq j \leq 6)$ wants to broadcast a sequence of packets $\{X_j(n)\}$ with the same length of $X_i(n)$ to node $A_i (1 \leq j \leq 6, i \neq j)$, i.e., all the nodes in the cluster broadcast information to all the other nodes in the same cluster.

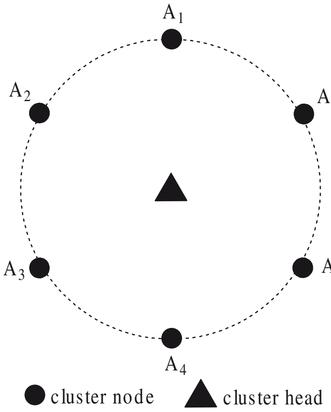


Fig. 2. Cluster Model

We assume that each node A_i can successfully broadcast one unit of information to all neighbors $N(A_i)$ within a given transmission range, through physical layer broadcast. The transmission range is the same for all nodes, while cluster head can reach every node in cluster by only one hop. For each node, there exists at least one other node, such that the two nodes transmission range covers the entire cluster. Take the Fig. 2 for example, node A_1 can communicate with node

A_2, A_3 and node A_7, A_8 directly; node A_3, A_4 and A_6, A_7 are in the transmission range of node A_5. Thus, the ranges of node A_1 and A_5 cover the whole cluster. Thus, minimizing the energy is equivalent to minimizing the number of transmissions required to convey a unit of information from all sources to all receivers.

More precisely, let T denote the total number of transmissions required to finish one unit of intra-cluster information exchange process, and let n denote the number of common nodes (cluster head is not included) in the cluster.

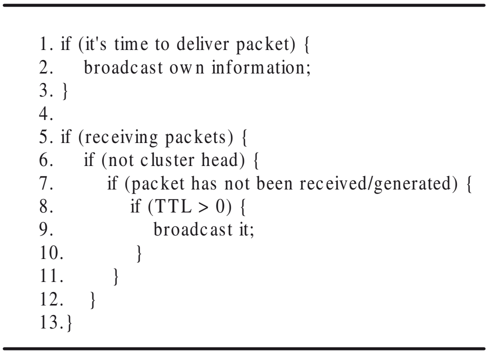
3 Algorithms

To solve the intra-cluster information exchange problem, there are many methods: flooding, routing, relaying and network coding. These algorithms are described in this section and the transmission numbers $T_{algorithm}$ are compared to evaluate the algorithm efficiency.

3.1 Flooding

Flooding is an old technique [12] that can also be used for routing in wireless ad hoc sensor networks. In flooding, each node receiving a data or management packet repeats it by broadcasting, unless a maximum number of hops for the packet are reached or the destination of the packet is the node itself. Flooding is a reactive technique, and it does not require costly topology maintenance and complex route discovery algorithms. On the other hand, the main disadvantage of the flooding algorithm is high energy consumption levels, which is an extremely important factor in ad hoc sensor networks.

The flooding algorithm for intra-cluster information exchange is shown in Fig. 3. The algorithm is explicit and we can analyze the transmission number T_{flood} refer to Fig. 2. Assume node A_1 broadcasts its information firstly, after rebroadcasting by node A_2, A_6, A_3 and A_5, the packet arrives node A_4. Though



```
if (it's time to deliver packet) {
    broadcast own information;
}

if (receiving packets) {
    if (not cluster head) {
        if (packet has not been received/generated) {
            if (TTL > 0) {
                broadcast it;
            }
        }
    }
}
```

Fig. 3. Flooding Algorithm for Intra-cluster Information Exchange

node A_4 is the last node receiving the packet, A_4 still needs to rebroadcast it. Because without global topology and schedule knowledge, the last node in flooding process still needs to broadcast one extra time. Then we can make out that to spread the information of node A_1, the 6 times broadcasts are needed. And now we can come to a conclusion that the transmission number of intra-cluster information exchange by flooding is n^2, i.e.,

$$T_{flood} = n^2. \tag{1}$$

3.2 Random Routing

As assumed above, each node does not know the global information about the cluster and can only communicate with its neighbors. Take the Fig. 2 for example, assume node A_1 wants to send information to node A_5, it does not know $A_1 \rightarrow A_6 \rightarrow A_5$ is the shortest path, and maybe it chooses the path: $A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4 \rightarrow A_5$. Without loss of generality, we assume the probability which path is chosen is 0.5. Thus the average transmission number for node A_1 to deliver its information to any node in cluster is $(n-1)/2$. Consequently, the transmission number of one node to spread its information to all the others is $n(n-1)/2$ and the transmission number of the whole information exchange is $n^2(n-1)/2$, i.e.,

$$T_{routing} = n^2(n-1)/2. \tag{2}$$

For simplicity, we just consider the transmission number of information deliver and do not take the routing discovery into account, which is complicated in such circular network. Thus, we get $T_{routing} > n^2(n-1)/2$.

3.3 Cluster Head Relay

There is another explicit but efficient algorithm for intra-cluster information exchange: relay information by cluster head. Every node sends information to the cluster head who broadcasts the packet to all the nodes in the cluster. Thus, the total transmission number is

$$T_{relay} = 2n. \tag{3}$$

This cluster head relay algorithm is very simple and effective. Each node just sends own information directly to the head without knowing any other information such as topology, geography, etc. The cluster head is responsible to exchange information between the cluster members. But the drawback of this algorithm is that the network traffics are not well balanced because the transmission number of cluster head is at least n, half of the total transmission number, which cause cluster head easy to die or to be rotated frequently.

3.4 Network Coding Algorithm

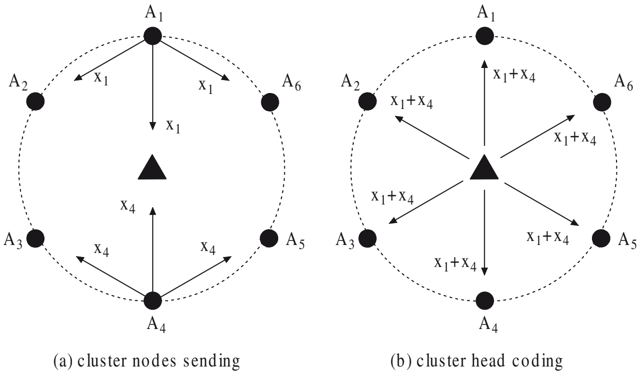
In this section, we propose a novel network coding algorithm for intra-cluster information exchange. In our algorithm, each step consists of two phases, which are described in Fig.4. In the first phase nodes transmit and cluster head receives. In the second, cluster head broadcasts and cluster nodes receive. The scheme operates as follows.

Algorithm NC:
Step 1:

- Phase 1: Two cluster nodes broadcast their information to their nearest neighbors, and the information will be heard by cluster head.
  (For example, in Fig.4(a), node A_1 and A_4 broadcast x_1 and x_4 respectively. And the cluster nodes, receive either x_1 or x_2, as assumed in section *Cluster Model*, while the cluster head receives both.)
- Phase 2: The cluster head simply *xor* the two packets and broadcasts it.
  (Take the Fig.4(b) for example, after receiving x_1 from node A_1 and x_4 from A_4, cluster head broadcasts $x_1 \oplus x_4$ to the whole cluster. Thus, each node in cluster can obtain x_1 and x_4.)

Step $k, k > 1$:

- Phase 1: Another two nodes broadcast their information.
- Phase 2: Cluster head adds and broadcasts the information; if there are cluster nodes having not broadcasted their information, then step $k - 1$, otherwise end the algorithm.



(a) cluster nodes sending

(b) cluster head coding

Fig. 4. Network Coding Algorithm for Intra-cluster Information Exchange

From the algorithm, we can see that every node in the cluster needs to broadcast its information only once and the cluster head just need broadcasts the $\frac{n}{2}$ coded packets. Thus, the transmission number of our network coding algorithm can be made out:

$$T_{NC2} = n + \frac{n}{2} = \frac{3}{2}n. \tag{4}$$

In section 2, we assume that for each node, there exists at least one node, and the transmission range of the two nodes can cover the whole cluster. But sometimes the condition can not be satisfied, and in our algorithm the cluster head will broadcast extra $\frac{n}{2}$ packets in case that some nodes have not sufficient information to decode the packets. For example, in Fig 3.4, if node A_6 is out of the transmission range of node A_1 and A_4, consequently A_6 will not obtain x_1 and x_4, but only receive $x_1 \oplus x_4$ from cluster head. Therefore, node A_6 cannot decode the information by itself. Thus the cluster head has to broadcast x_1 or x_4 to those nodes which have not efficient information to decode the incoming information. And the number of the extra packets needed to broadcast by cluster head is at most $\frac{n}{2}$. So, we can get:

$$\frac{3}{2}n \leq T_{NC2} \leq 2n = T_{relay}. \tag{5}$$

We can see from the inequation (5) that the upbound of the transmission number of our algorithm is equal to the *Cluster Head Relay Algorithm.*

4 Simulation

4.1 Environment Setup and Performance Metrics

We evaluate Information Exchange algorithms via simulation using NS2 [13] and compare the network coding algorithm (NC) to the flooding algorithm, random routing scheme and cluster head relay mechanism. In order to evaluate the performance, we compare the four algorithms with respects to the following metrics: residual energy and average time consumed in the information exchange process.

In our simulation environment, sensor nodes are distributed over a $200m \times 300m$ area, and the transmission range of each node is set to be $50m$. Without loss of generality, the cluster head (node 0) is located at the center of the simulation area. Two kinds of topology scenarios are simulated: one is the circular networks as shown in Fig. 2, and the other is *random topology*, i.e., cluster nodes are uniformly random distributed in the area. Each sensor has the same maximum battery energy capacity and for simplicity, the cluster head has the same maximum battery capacity as other sensors. In the simulation, all nodes exchange their data packets to each other, as described in section 1. Each data packet is set to the fixed length (64 bytes) and a clear channel is assumed. In each kind of simulated networks, five different numbers of deploying nodes, 6 nodes, 8, 12, 16 and 20 nodes are simulated to evaluate the performances under different densities. And totally, there are 20 different scenes (different topologies and packets generating sequences, excluding the partitioned networks) in each of these five cases.

4.2 Simulation Results

Fig. 5 and Fig. 6 show the residual energy after information exchange in circular network and random network respectively. We can see from the two figures that in both circular and random networks, network coding algorithm, cluster head

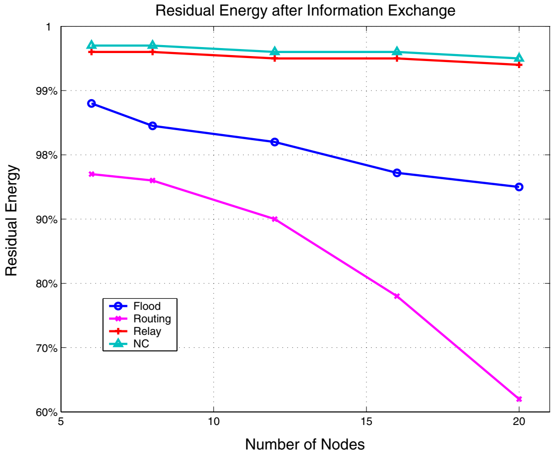


Fig. 5. Energy Consumption after Information Exchange (Circular Networks)

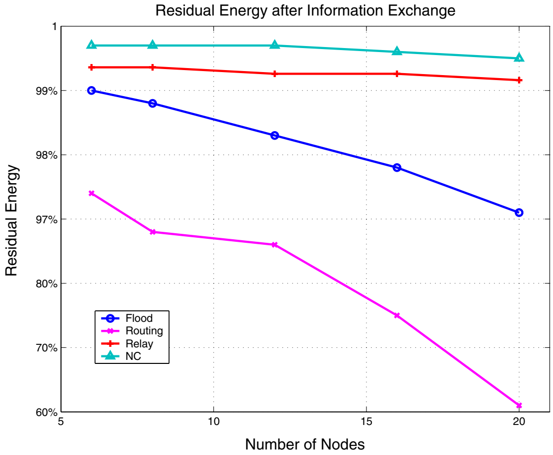


Fig. 6. Energy Consumption after Information Exchange (Random Topology)

relay and flood approach cost much less energy than routing scheme. It is because routing is a kind of unicast communication, i.e., each source node must generate single packet for each destination node. Among network coding, cluster head relay and flood algorithms, network coding costs the least energy, which accords with the equations (1)-(4). Also we can know from the graphs that the residual energy of the four algorithms reduces as the node number increases.

The other performance metric we focus on in this paper is the information exchange time. It is shown in Fig. 7 and Fig. 8 that network coding costs the least time in both circular and random networks while random routing costs most, which accords with the theoretical analysis in section 3. Although the performance of cluster head relay is better than flood and routing, it is still

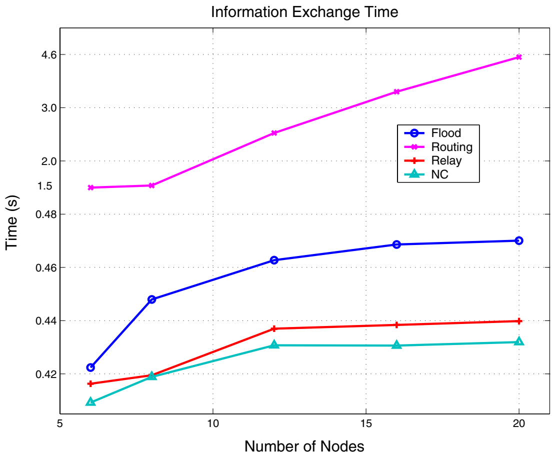


Fig. 7. Time Consumption after Information Exchange (Circular Networks)

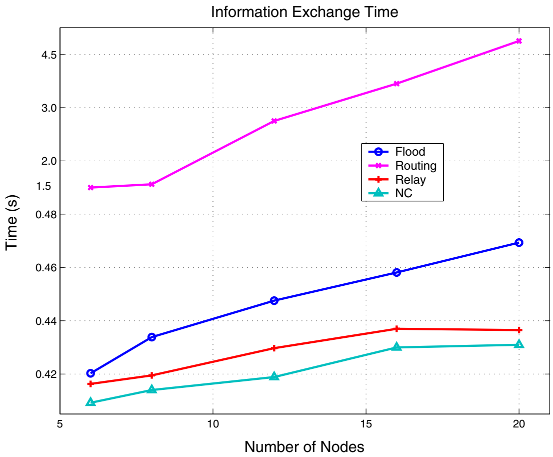


Fig. 8. Time Consumption after Information Exchange (Random Topology)

not as good as network coding. And the load-balancing in cluster head relay approach is not optimal, for the traffic load of cluster head is half of the total traffic, as analyzed in section 3.

We also compare the information exchange time of network coding algorithm between circular network and random topology. The result is shown in Fig. 9. From the graph, we can see that the required time of information exchange in random topology is less than in circular network. It is because that in random topology, the distances between nodes in cluster, especially the distance between cluster member nodes and cluster head, are not as large as in the circular network, which is convenient for network coding. And from the figure, we can find

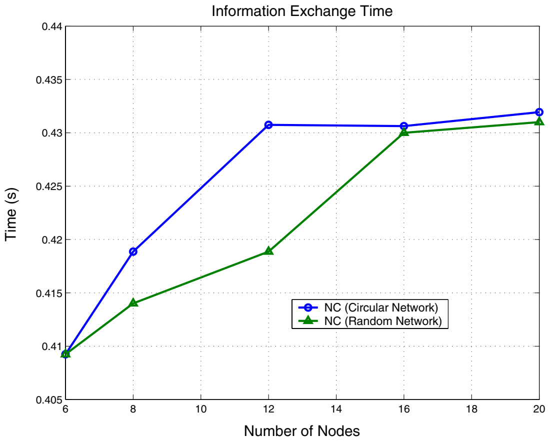


Fig. 9. The Comparison of Time Consumption after Information Exchange (NC Algorithm Between Circular Network and Random Topology)

that proper numbers of nodes, such as numbers from 8 to 16, are more suitable for using network coding in random topology.

5 Conclusions

In this paper, we identify intra-cluster information exchange in wireless ad hoc sensor networks as a new application scenario, and propose some novel solutions. After the theoretical analysis and simulations, we come to a conclusion that the cluster head relay approach and network coding algorithms perform much better than conventional solutions, such as routing and flooding, both in energy consumption and time cost. Further more, network coding algorithm performs best and can offer unique advantages. It is because Network coding algorithm, together with physical layer broadcast property offered by the wireless medium, can improve the efficiency in using resources. And the network coding algorithm we proposed is simple to implement and our simulation work indicates that there is a potential for significant benefits, when deploying network coding over a practical wireless environment. In the future, we will focus on MAC layer modification to develop more efficient network coding algorithms for intra-cluster information exchange.

References

1. R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. IEEE Transactions on Information Theory, 46(4):1204-1216, 2000.
2. S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear Network Coding. IEEE Transactions on Information Theory, 49(2):371-381, 2003.
3. R. Koetter and M. Médard. An Algebraic Approach to Network Coding. IEEE/ACM Transactions on Networking, 11(5):782-795, 2003.

4. T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros. The Benefits of Coding over Routing in a Randomized Setting. In Proceedings of the IEEE International Symposium on Information Theory, 2003.
5. S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial Time Algorithms for Multicast Network Code Construction. Submitted to IEEE Transactions on Information Theory, 2003.
6. Y. Wu, P. A. Chou, and S.-Y. Kung, Information exchange in wireless networks with network coding and physical-layer broadcast, in Proc. 39th Annu. Conf. Inf. Sci. Syst., Mar. 2005, [CD-ROM].
7. P Panchapakesan, D Manjunath, "Transmission Scheduling for Gossiping in Multihop Packet Radio Networks", IEEE International Conference on Personal Wireless Communications, 2000.
8. S Lindsey, CS Raghavendra, "Energy efficient all-to-all broadcasting for situation awareness in wireless ad hoc networks", Journal of Parallel and Distributed Computing, 2003.
9. Lindsey S, Raghavendra, C.S, "Energy efficient broadcasting for situation awareness in ad hoc networks", International Conference on Parallel Processing, 2001. 3-7 Sept. 2001 Page(s):149-155.
10. P Panchapakesan, D Manjunath, "Transmission Scheduling for Gossiping in Multihop Packet Radio Networks", IEEE International Conference on Personal Wireless Communications, 2000.
11. C. Diot, J. Scott, E. Upton, and M. Liberatore, The Haggle architecture, Intel Research Cambridge, Tech. Rep. IRC-TR-04-016, 2004.
12. Ahmed E. Kamal Jamal N. Al-Karaki. Routing techniques in wireless sensor networks: A survey. Wireless Communications, IEEE, 11:6C28, Dec. 2004.
13. http://www.isi.edu/nsnam/ns/

FNSCSDP: A Forward Node Selection Based Cross-Layer Service Discovery Protocol for MANETs

Zhenguo Gao[1], Yongtian Yang[2], Ling Wang[3], Jianwen Cui[1], and Xiang Li[2]

[1] College of Automaion, Harbin Engineering University, Harbin, 150001, China
gag@hrbeu.edu.cn
[2] College of Computer Science, Harbin Engineering University, Harbin, 150001, China
[3] College of Computer Science, Harbin Institute of Technology, Harbin, 150001, China

Abstract. Service discovery is expected to be a crucial feature for the usability of MANETs(mobile ad-hoc networks). However, service discovery protocols existing nowadays are generally separated from backbone construction schemes usually performing in network layer in MANETs. Recognizing this problem, we propose a FNSCSDP(Forward Node Selection based Cross-layer Service Discovery Protocol) for MANETs, which makes use of periodically hello packets used in backbone construction schemes to facilitate service discovery tasks. When forwarding service request packets, FNSCSDP deliberately selects some nodes that should forward the service request packet. Those nodes are selected basing on local topology information gathered from hello packets and history information piggybacking in service request packets. Hence, all nodes that are at most 2-hop away from the current node will be able to receive the service request packet. Simulations show that FNSCSDP outperforms a well-known service discovery protocol GSD in packet overhead and promptness.[1]

1 Introduction

MANETs (Mobile ad-hoc networks)[1] is a temporary infrastructure-less multihop wireless network that consists of many wireless mobile nodes. MANETs are mainly characterized by highly dynamic topology. Flexibility and minimum user intervention are essential for MANETs. Therefore, Service discovery, which allows devices to automatically locate network services with their attributes and to advertise their own capabilities to the rest of the network, is a major component of MANETs.

Service discovery was originally studied in the context of wired networks. Several different industrial consortiums and organizations were established to standardize various service discovery protocols, such as IETF's Service Location

[1] This work is supported by Fundamental Research Foundation of Harbin Engineering University under contact number HEUFT06009 and Foundation of Experimentation Technology Reformation of Harbin Engineering University. (No.SJY06018).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 220–231, 2006.

Protocol (SLP)[2], Sun's Jini[3], Microsoft's UPnP(Universal Plug and Play)[4], and IBM's Salutation [5]. However, these standardization endeavors do not applicable to MANET environments where self-configurability is the key.

Researches on service discovery protocols for MANETs adopt an abstract view on service discovery without paying particular attention to how the service types and attributes are defined. There are two basic methods to perform service discovery in MANETs: reactive and proactive [6][7]. In a reactive method, nodes reactively send out service request packets. If a node that supports the requested services receives the packet, it should generate a service reply packet and send it back to the source node where the request packet was originated. On the other hand, in a proactive method, nodes that need services passively learn about the available services by listening to service advertisements that are proactively generated by nodes that provide the services. These two basic methods are too rough to provide sufficient support to service discovery tasks [6].

In recent years, some protocols have been proposed to support service discovery targeted at MANETs, such as Konark[8], RICFFP[9], GSD[6], and Service Ring[10]. However, these protocols are generally separated from backbone construction schemes, which are widely used in MANETs. Hence, their service discovery protocols coexist with backbone construction schemes in MANETs, which results in serious collisions and latency of service acquisition.

Motivated by this observation, we propose a FNSCSDP(Cross-layer Forward Node Selection based Service Discovery Protocol) for MANETs, which benefits from periodically hello packets generally used in backbone construction schemes. In FNSCSDP, we extend the structure of hello packets used in backbone construction, and utilize a forward node selection based scheme to forward service request packets. FNSCSDP mainly works in application layer, while hello packets in backbone function works in network layer. Thus it is cross-layered.

The rest of the paper is organized as follows. In section 2, an overview of related works is given. In section 3, the operation of FNSCSDP is described in detail. In section 4, packet overhead of FNSCSDP and GSD are analyzed. In section 5, performances of FNSCSDP and GSD are evaluated through simulation. Finally, in section 6, we summarize the paper and present a conclusion.

2 Related Works

In service discovery protocols, the objective is to reduce service request packet redundancy while retaining service discoverability. According to the methods used to reduce packet redundancy, service discovery protocols can be classified into two classes: probability-based approach, semantic-routing-based approach.

2.1 Probability-Based Approach

In probability-based service discovery schemes, when receiving a request packet, each node that does not have matched services should forward the request packet with probability P. When P is 1, this scheme degenerates to flooding, which may lead to great packet redundancy. Flooding is used in Konark[8]. When P is less than 1, request packet redundancy will be reduced. However, request packet coverage will be reduced, which greatly reduces service discoverability. Hence, in some papers[9], some modifications are made: probability P varies along the travel of a request packet.

2.2 Semantic-Routing-Based Approach

In a semantic-routing-based approach, nodes can intelligently select forward nodes by inspecting service description semantics. Nodes that are not selected as forward nodes by the senders should not forward request packets. This approach is used in GSD[6], and Service Ring[10].

In GSD, Services are classified into several groups. A node's service group information is cached in its neighbors. When a request packet is to be forwarded, the node should intelligently select forward nodes basing on service group information cached. In Service Ring, nodes are organized into multi-layer hierarchical rings. Its peculiar structure restricts its applicable situations.

In probability-based schemes, service discoverability is not guaranteed, making semantic-routing-based schemes more preferable. However, the latter methods need to generate their own hello packet exchanging mechanisms, which is unnecessary since backbone construction schemes widely implemented in MANETs provide similar mechanism.

Recognizing this problem, we propose a Cross-layer Forward Node Selection based Service Discovery Protocol (FNSCSDP) for MANETs, which benefits from periodical hello packets in backbone construction function.

3 Protocol Description of FNSCSDP

3.1 Definitions

- **Definition 1 (Service):** Service is any hardware or software feature that can be utilized or benefited by any nodes.
- **Definition 2 (Service Description):** Service description is the information that describes a service's characteristics, such as its types and attributes, access method, etc.
- **Definition 3 (Server):** A server is a node that offers services to other nodes.
- **Definition 4 (Client):** A client is a node that needs services of other nodes.
- **Definition 5 (k-hop Neighbor Set):** If two nodes can communicate with each other directly, then they are neighbors. They are 1-hop away from each other. The k-hop neighbor set of node u is all nodes that are at most k hops away from node u, denoted as $N_k(u)$.The shortest hop length between two nodes u and v in a MANET is denoted as $d(u, v)$.
- **Definition 6 (Service Request Session):** When a node needs services from others, it can send out a service request packet. A node that offers matched services should response with service reply packets. Node with no

matched services should forward the request packet. All these corresponding packet transmissions, including request packets and reply packets, make up a service request session. To limit the spreading range of service request packets, the maximum hop that request packets can travel is restricted to a predefined value.

- **Definition 7 (Service Discoverability):** Service discoverability is the ability of finding services that meet requests.
- **Definition 8 (Forward Node):** A forward node should forward the request packet it received.

3.2 Data Structures

Hello Packet. Each node broadcasts hello packets periodically. A hello packet has the following structure:

[hello-packet]::=[packet-type][sender-id][service-count][service-list][neighbor-count][neighbor-list][lifetime]

[packet-type] filed indicates the packet type. [sender-id] indicates the sender of the hello packet. [neighbor-count] indicates the count of the sender's neighbors. [neighbor-list] contains the list of the sender's neighbors. [life-time] indicates the valid time of the information in the hello packet.

Neighbor List and Service Information Cache (NLSIC). Node that receives a hello packet can cache the information in the hello packet in NLSIC. A NLSIC entry has the following structures:

[NLSIC entry]::=[lifetime][sender-item][unpruned-neighbor-count][neighbor-count][neighbor-item-list]
[sender-item]::=[node-item];
[neighbor-item-list]::=[node-item]—[neighbor-item-list];
[node-item]::=[node-id][pruned];

[pruned] field has Boolean value, which is used in dominate node selection process shown in the following section. Other fields are the same to a hello packet. The entry will be removed after [time-out] expires.

Service Request Packet. A service request packet has the following structure:

[request-packet]::=[packet-type][source-id][request-id][request-description]
[visited-count][visited-list][receiver-count][receiver-list]
[hopcount]

[packet-type] indicates the packet type. [source-id] indicates the node that generates the request packet. [request-id] increases monotonically with each request packet from a node. A pair [source-id, request-id] uniquely identifies a service request session. [request-description] describes the needed service. [visited-count] indicates number of the nodes that the packet has visited, the visited node list is stored in [visited-list]. [receiver-count] indicates the number of forward nodes selected by the current node. [receiver-list] contains the forward node list. [hop-count] indicates hops that the packet can still travel. If the field is 0, the packet should be discarded.

Forwarded Packet Table (FPT). Each node maintains a FPT, which stores the ([source-id], [request-id]) pair of request packets that have been forwarded. FPT is be used in two tasks: 1) to check duplicated request packet, and 2) to reversely forward service reply packets toward the corresponding source node. A FPT entry has the following structure:

[FPT-entry]::=[source-id][request-id][predecessor-id]

[source-id] field indicates the corresponding source node. [request-id] field identifies different request packets from the same node. They are the same to a service request packet. [predecessor-id] field indicates the node from which the current node receives the corresponding request packet. [predecessor-id] is just the node that the reply packet should be forwarded to.

Service Reply Packet. When receiving a request packet, each node that provides matched services should respond with a reply packet. A service reply packet has the following structure:

[reply-packet]::=[packet-type][source-id][request-id][receiver-id][replier-id] [service-description]

[packet-type] indicates that the packet is a service reply. [source-id] identifies the node that generates the corresponding request packet. [receiver-id] indicates the node that the reply packet should be forwarded to. [replier-id] identifies the node that generates the reply packet. [service-description] stores the founded service's description.

3.3 Operations of FNSCSDP

In FNSCSDP, there are three basic operations: hello packet exchanging, service request packet routing and service reply packet routing.

Hello Packet Exchanging and Caching. Each node should broadcast hello packets periodically. A hello packet contains the sender's ID and its neighbor list. If the node provides any services, the service list should also be included. Node can cache the information in the hello packets received. By caching hello packets, a node maintains its local topology information. Before broadcasting new hello packets, a node should update its neighbor list based on its NLSIC.

Service Request Generating and Routing. When a node requests services from other nodes, it should first check its NLSIC to decide whether there are any matched services. If several matches are found, the service request is automatically satisfied. Otherwise, the node should construct a service request packet and broadcast it.

When a node receives a new request packet, it will insert a corresponding entry into FPT, then check whether there are any matched services provided by itself or in NLSIC. Here two cases are considered.

If it finds matched services, then it should reply with a service reply packet.

A node should forward a new request packet to other downstream nodes if the following conditions are met:

- the node fails to find any matched services;
- it is in the [receiver-list] field of the request packet, or the packet's [receiver-count] is 0;
- the packet's [hop-count] is great than 0.

Before forwarding the request packet, the node has to select some forward nodes based on both local topology information and the information piggybacked in the packet. The number of selected nodes are minimized how guaranteeing that the current node's 2-hop neighbor set be fully covered.

The pseudo code of selecting forward nodes is listed as follows:

```
Algorithm:                 Select forward nodes;
Input:       Req:          Request packet;
             NLSIC         Neighbor List and Service Info Cache;
             MY_ID:        ID of the current node;
             e:            NLSIC entry;
output:      Req:          Updated request packet.
Body Begin:
1. if (NLSIC is empty) Broadcast packet Req;
2. else {
    2.1 Reset all [pruned] fields to FALSE in NLSIC;
    2.2 Initiate all [unpruned-neighbor-count] to corresponding entry's [neighbor-
        count] value;
    2.3 Prune MY_ID from NLSIC;
    2.4 for (each node $u$ in Req.[visited-list]) {
        2.4.1 Prune each node in $N_2(u)$ from NLSIC;}
    2.5 for (each entry e in NLSIC){
        2.5.1 Prune e.[sender-item].[node-id] from NLSIC; }
    2.6 while(e∈NLSIC, such that (e.[unpruned-neighbor-count])<0){
        2.6.1 Select entry e such that e.[unpruned-neighbor-count] is maximum;
        2.6.2 Select e.[sender-item].[node-id] as a forward and insert it into
              Req.[receiver-list];
        2.6.3 Prune $N_1$(e.[sender-item].[node-id]) from NLSIC. }
    2.7 Insert MY_ID into Req.[visited-list];
    2.8 Broadcast packet Req;}
Body End
```

The definition of the function **Prune** in previous algorithm is as follows:

Service Reply Generation and Routing. As long as the node that received a request packet finds matched services (i.e., the services can be either provided by the node itself or are found in NLSIC), it should respond with a reply packet.

When a node receives a reply packet, it will do as follows:

If the current node equals the [source-id] of the reply packet, then the service request session is successfully completed.

Algorithm:		Prune a node from NLSIC;
Input:	*u:*	The node that will be pruned from NLSIC;
	NLSIC	Neighbor List and Service Info Cache;
output:		none.

```
Body Begin:
1. for (each entry e in NLSIC){
    1.1 for (i=0;i<(e.[neighbor-count]), i++){
        1.1.1 if (e.[neighbor-list][i].[node-id]==$u$ && e.[neighbor-list][i].[pruned]
                ==FALSE){
            1.1.1.1 e.[neighbor-list][i].[pruned]=TRUE;
            1.1.1.2 e.[unpruned-neighbor-count]–; }}}
Body End
```

If the current node is not the reply packet's [source-id], then it should forward the reply packet. It lookups the ([source-id], [request-id]) pair in FPT. If an entry is found, then the entry's [predecessor-id] is stored in the reply packet's [receiver-id] and then unicast the packet. If it failed to find a matched entry, the packet will be discarded.

4 Packet Overhead Analysis

In this section, we will analyze the packet overhead of FNSCSDP and GSD[6], which is a typical service discovery protocol for MANETs.

4.1 GSD

Two special mechanisms of GSD are peer-to-peer caching of service information and intelligent forwarding of service request packets. In GSD, services are classified into groups. A server should broadcast hello packets periodically. A hello packet contains the descriptions of the services provided by the server. If there are some servers in the sender's vicinity, then the group information of the services provided by these servers is also included into the hello packet. The hello packets are subsequently cached and forwarded for a limited hops.

When a request packet is forwarded, a copy of the packet will be forwarded to each potential node. A potential node is a node that has neighbours providing services that belong to the same group as the requested service. A request packet may be matched at potential nodes with high probability.

4.2 Packet Overhead

Packet transmissions in unit time include three parts: hello packets, request packets, and reply packets. That is,

$$P_{total} = f.P_{hello} + c.(P_{req} + P_{rep}) \tag{1}$$

where 1) f is the frequency of hello packet transmission, 2) P_{hello} is hello packet transmission number in one hello packet exchange cycle, 3) c is request session number in one time unit, 4) P_{req} and P_{rep} is the request and reply packet number in one request session, respectively.

Assume that there are n_s servers in a MANET. Request packets can travel q hops. Average node count of a k-hop neighbour set is $n_{k,hop}$. Hello packets in GSD can travel h hops. k_{GSD} and $k_{FNSCSDP}$ is the coefficient of request packets in GSD and FNSCSDP, respectively. k_{GSD} may be greater than 1.

Then in GSD,

$$\begin{aligned} P_{hello,GSD} &= n_{h,hop}.n_s \\ P_{req,GSD} &= n_{q,hop}.k_{GSD} \end{aligned} \tag{2}$$

In FNSCSDP,

$$\begin{aligned} P_{hello,FNSCSDP} &= n \\ P_{req,FNSCSDP} &= n_{q,hop}.k_{FNSCSDP} \end{aligned} \tag{3}$$

Since any hardware or software feature can be a service, server number in a MANET may be very large. Therefore, $P_{hello,FNSCSDP}$ is generally less than $P_{hello,GSD}$. $k_{FNSCSDP}$ is usually less than k_{GSD}, hence $P_{req,FNSCSDP}$ will always be less than $P_{req,GSD}$, which is confirmed in the following simulation studies. Hence, P_{total} in FNSCSDP is generally less than that in GSD.

Recall that main fields of the hello packets in FNSCSDP packets are as follows: [service-count], [service-list], [neighbor-count], and [neighbor-list]. The former two fileds is also contained in the hello packets (called as advertisement packets in GSD) of GSD, whereas the last two fields is small in size since that only the IDs of nodes are contained. Hence, when compared with GSD, size change of hello packets in FNSCSDP is trivial.

5 Simulation Study

5.1 Performance Metrics

Four performance metrics are considered in our simulations.

- **Number of Service Request Packets Per Session:** It measures the number of service request packets sent in one simulation. It reflects the efficiency the policy of forwarding service request packets.
- **Ratio of Succeeded Requests:** It is the number of service discovery sessions in which the client has received at least one successful reply packet. It reflects the effectiveness (service discoverability) of service discovery protocols.
- **Response Time (m/s):** It is the interval between the arrival of the first reply packet and the generation of the corresponding request packet. This metric is averaged over all succeeded service discovery sessions. It measures the promptness of service discovery protocols. It also reflects the average distance between clients and the corresponding first repliers.
- **Ratio of Succeeded Requests to Total-packet-number (Suc2Packet):** This metric is the ratio of Succeeded-SDP-number to the sum of service request packets and service reply packets. It reflects the efficiency of service discovery protocols.

5.2 Simulation Models

Simulation studies are performed using GloMoSim [11]. The distributed coordination function (DCF) of IEEE 802.11 is used as the underlying MAC protocol. Random Waypoint Model (RWM) is used as the mobility model.

In RWM mobility model, nodes move towards their destinations with a randomly selected speed $V \in [V_{min}, V_{max}]$. When reaching its destination, a node keeps static for a random period $T_P \in [T_{min}, T_{max}]$. When the period expires, the node randomly selects a new destination and a new speed, then moves to the new destination with the new speed. The process repeats permanently. In our simulations, $T_{min} = T_{max} = 0$,$V_{min} = V_{max} = V$.

5.3 Simulation Settings

Some basic parameters that are used in all the following simulations are set as shown in Table 1. Simulation scenarios are created with 100 nodes randomly distributed in the scenario area. At the beginning of each simulation, some nodes are randomly selected out to act as servers. These selected servers provide randomly selected services. During each simulation, 100 SDP sessions are started at randomly selected time by randomly selected nodes.

Table 1. Basic parameters

Parameters	Value	Parameter	Value
Scenario area	1000m×1000m	Number of service group	2
Node number	100	Maximum hop of request packets	3
Server number	50	Valid time of SIC item	21s
Simulation time	1000s	Number of service info in each group	5
SDP session number	100	Service advertisement interval	20s
radio range	150m	Maximum hop of advertisement packets	1
Wireless bandwidth	1Mb/s		

5.4 Simulation Results

In this section, we inspect the effect of node speed on the selected protocols through simulations. To do this, we run 4 simulation sets that use the 4 selected service discovery protocols, respectively. Each set includes 5 subsets of simulations, where $V = V_{min} = V_{max}$ and V is set to 0m/s, 5m/s, 10m/s, 15m/s, and 20m/s, respectively. Each subset consists of 50 similar simulations. Simulation results are averaged over 50 simulations. The results are shown in Fig. 1 to Fig. 3. In all these figures showing simulation results, error bars report 95% confidence.

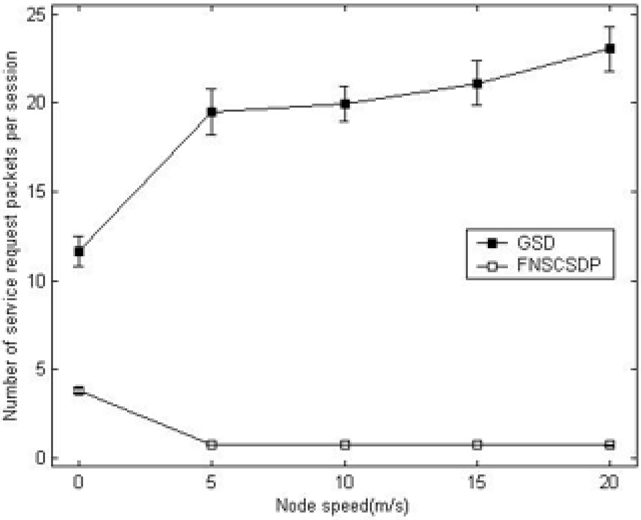


Fig. 1. Number of request packets per session vs. node speed

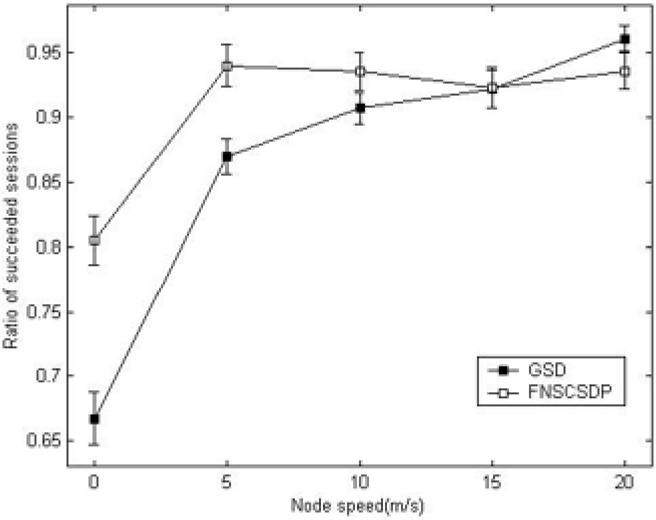


Fig. 2. Number of request packets per session vs. node speed

Fig. 1 shows that request packet overhead of FNSCSDP is much lower than that of GSD. The superiority of FNSCSDP in this metric becomes more remarkable at higher speeds. Hence, FNSCSDP is more efficient than GSD.

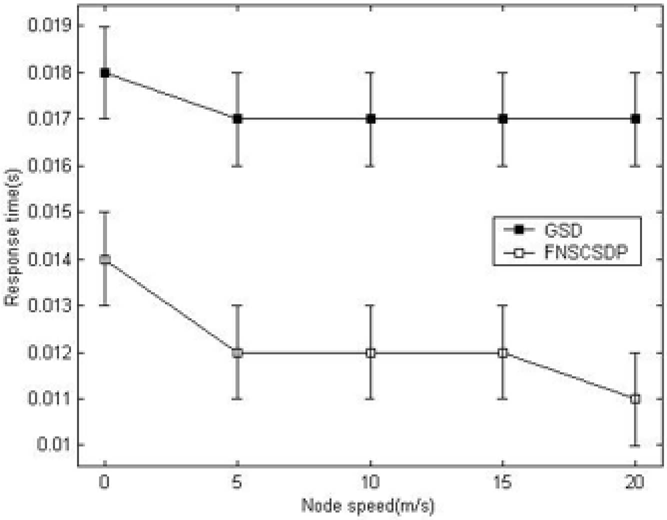


Fig. 3. Number of request packets per session vs. node speed

Fig. 2 shows that success ratio of FNSCSDP is about 20% greater than that of GSD when nodes are steady. As node speed increases, the superiority of FNSCSDP over GSD becomes more insignificant. Hence, FNSCSDP generally outperforms GSD in efficiency.

Fig. 3 shows that response time of FNSCSDP is about 2/3 of that of GSD. Therefore, FNSCSDP is more prompt than GSD.

Therefore, a conclusion can be made from the simulation results: FNSCSDP is generally more efficient, more effective, more prompt than GSD.

6 Conclusions

In existing service discovery protocols used in MANETs, service discovery functions are typically separated from the backbone construction functions that are performed in network layer. In this paper, we have proposed a new service discover protocol, referred as FNSCSDP, which makes use of periodical hello packets generated for backbone construction. In FNSCSDP, before forwarding service request packets, a node will select forward nodes. The number of selected nodes are minimized however guaranteeing that the current node's 2-hop neighbor set be fully covered.

Simulation results show that FNSCSDP outperforms GSD in terms of number of service request packets per session, ratio of succeeded requests, and response time.

References

1. IETF, Mobile ad-hoc network (MANET) working group. [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html
2. E. Guttman, C. Perkins, J. Veizades, and M. Day. Service location protocol, version 2. IETF RFC 2608, Jun. 1999
3. Sun Microsystems. Jini architecture specification. Nov. 1999
4. Microsoft Corporation. Universal plug and play: background. [Online]. Available: http://www.upnp.org/resources/UpnPbkgnd.htm
5. Salutation Consortium. Salutation Architecture Specification. [Online]. Available: http://www.salutation.rog/specordr.htm. 1999
6. D. Chakraborty, A. Joshi, Y. Yesha, and T. Finin. GSD: a novel group-based service discovery protocol for MANETs. Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, Sep. 2002, pp: 140-144
7. N. Klimin, W. Enkelmann, H. Karl, A. Wolisz. A hybrid approach for location-based service discovery in vehicular ad hoc networks. Proceedings of the 1st Intelligent Workshop on Intelligent Transportation, Hamburg, Germany, Mar. 2004, pp: 1-5
8. S. Helal, N. Desai, V. Verma, and C. Lee. Konark - a service discovery and delivery protocol for ad-hoc networks. Proceedings of the Third IEEE Conference on Wireless Communication Networks, New Orleans, USA, Mar. 2003, pp: 2107-2133
9. Z. G. Gao, X. Z. Yang, T. Y. Ma, and S. B. Cai. RICFFP: an efficient service discovery protocol for MANETs. Proceedings of the 2004 International Conference on Embedded and Ubiquitous Computing, Aizu-Wakamatsu City, Japan, Aug. 2004, pp: 786-795
10. M. Klein, B. K. Ries, and P. Obreiter. Service rings - a semantic overlay for service discovery in ad hoc networks. Proceedings of the 14th International Workshop on Database and Expert Systems Applications , 2003, pp: 180-185
11. Zhenguo Gao, Yunlong Zhao, Xiang Li, Shaobin Cai, Chunsheng Wang, Jianwen Cui. The Analyzation of the GloMoSim Wireless Network Simulator. Journal of System Simulation (Supplement), 2006, 18(8s): 672-675

Service Discovery Protocols for MANETs: A Survey

Zhenguo Gao[1], Yongtian Yang[2], Jing Zhao[2], Jianwen Cui[1], and Xiang Li[2]

[1] College of Automaion, Harbin Engineering University, Harbin, 150001, China
gag@hrbeu.edu.cn
[2] College of Computer Science, Harbin Engineering University, Harbin, 150001, China

Abstract. MANETs(Mobile Ad-Hoc Network) are temporary networks composed of many autonomous nodes. Service discovery is the technology of finding services matching one's needs in the network, which is crucial to the usability of MANETs. Many service discovery protocols for MANETs have been proposed, but no comprehensive comparative studies have been made. Hence, some typical service discovery protocols are analyzed and compared. The advantages and drawbacks of each protocol are analyzed. The paper provides a perspective overview on service discovery protocols for MANETs.[1]

1 Introduction

MANETs (Mobile Ad-Hoc Networks)[1] are temporary infrastructure-less multihop wireless networks that consist of many autonomous wireless mobile nodes. Service discovery is the technology of enabling a node to find services matching its needs in MANETs. Service discovery is a crucial feature for the usability of MANETs.

To search for a required service in a MANET, a node sends out a service request packet which will be forwarded by others. When receiving service request packets, every node with matched services responds with a service reply packet, which will be forwarded reversely to the source of the corresponding service request packet.

Many service discovery protocols for MANETs have been proposed, but no comprehensive comparative studies have been made. Hence, comparative study is performed in the paper.

The rest of the paper is organized as follows. In Section 2, some typical service discovery protocols for MANETs are explained in detail in subsections. In Section 3, comparative analysis about these typical protocols is performed. In Section 4, extensive simulations are performed and simulation results are analyzed. Finally, in Section 5, a conclusion is made.

[1] This work is supported by Fundamental Research Foundation of Harbin Engineering University under contact number HEUFT06009 and Foundation of Experimentation Technology Reformation of Harbin Engineering University. (No.SJY06018).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 232–243, 2006.

2 Service Discovery Protocols for MANETs

Though there are many serious challenges to the design of service discovery protocol for MANETs, such as limited bandwidth, dynamic topologies, variety of network size, limited physical resources, serious security threats, etc, many research efforts have been focused on different aspects of service discovery protocols for MANETs. References [2] [3] focus on protocol's security feature. Other efforts aim at saving packet overhead through different approaches: network layer-embedded approaches [4][5], cross-layer approaches [6], and application layer approaches.

In network layer-embedded approaches, such as [4][5], service requests information are enclosed in route discovery packets. Thus, services requests will be accomplished through a route discovery operation. In cross-layer approaches [6], application layer modules utilize network layer hello packets widely used in network layer protocols to facilitate its service discovery operation. Since that the procedure of service discovery is similar to that of route discovery in MANETs, application layer approaches can be adapted to other approaches easily.

To facilitate service discovery operation, nodes are organized into different structures. According to the structures, application layer approaches are classified into three classes: one-layer approaches, two-layer approaches, and multi-layer approaches.

2.1 One-Layer Approaches

In one-layer approaches [7] [14], all nods are in the same logical layer with the same role, as shown in Fig.1.

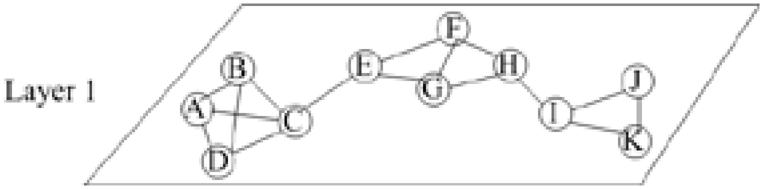


Fig. 1. Structure of one-layer approaches

FLOOD is the simplest and the most straight-forward approach to service discovery protocol for MANETs, where each node should forward a unduplicated request packets. FLOOD is widely accepted as the benchmark for performance evaluation of service discovery protocols. However, FLOOD is prone to flood storm problem [17]. Hence, FFPSDP(Flexible Forward Probability based Service Discovery Protocol)[10] and RICFFP(Reply Info Cache enhanced FFPSDP) [11] are proposed where the probability of nodes forwarding a new request packet decreases along with the travel of request packets. Although the two protocols reduce the risk of flood storm problem, FFPSDP and RICFFP's ability of finding services is also reduced.

GSD is a group-based service discovery protocol, whose salient characteristics are peer-to-peer caching of service advertisement packets and group-based intelligent forwarding of service request packets. Services are described using DAML+OIL and classified into groups. By sending and caching service advertisement packets periodically, each node knows the services provided by its neighbors and the groups of the services its neighbors have seen. When forwarding service request packets, instead of broadcasting the service request packet to all neighbours, GSD selectively forwards the packet towards those nodes that have seen services of the same group in unicast mode, named as candidate nodes.

However, GSD causes many unicast packets, which may overwhelm its the advantage of group-based intelligent forwarding scheme. Hence, CNPGSDP is proposed to improve GSD through two schemes: 1)several unicast packets is substituted by one broadcast packet that encloses all receivers; 2)removing some candidate nodes doomed not knowing about any matched services.

2.2 Two-Layer Approaches

In two-layer approaches [16] [23], some nodes are selected out according to variety of criterions to form up an upper logic layer, as shown in Fig.2.

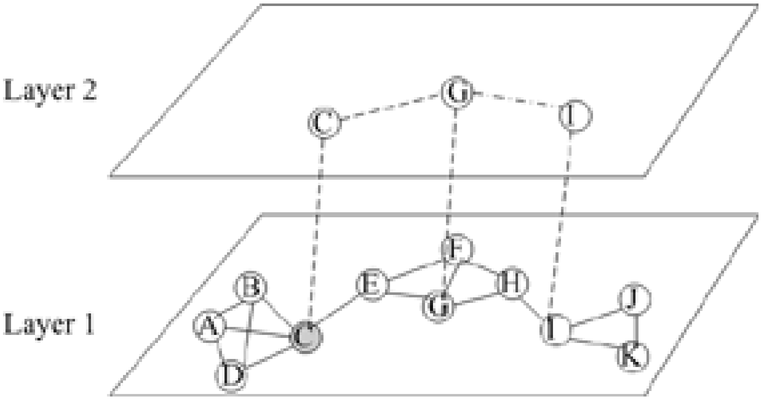


Fig. 2. Structure of two-layer approaches

Among these two-layer service discovery protocols, the protocol s in [16][19] construct backbones; nodes in [20][21] are organized into clusters; ProximityCAN [22] organizes nodes into 2-dimentional matrix structure; Lanes[23] organizes nodes into lanes.

DSDP selects out some nodes according to link stability and degree to form a backbone. Packet transmissions usually travel along the backbone. In this way, packet overhead will be reduced.

In VB, nodes are organized into clusters as follows. The node with smallest identity becomes the first cluster head, and then it randomly determines other *M*-1 cluster heads. Here *M*is a protocol parameter. Description of a service stored in three places: the node itself, the home cluster head, and the target cluster headed indicated by the hash result of the description. When requiring a service, a node A delegates the service discovery task to its home cluster head B.

In ProximityCAN, some node in physical space are selected as matrix nodes. Each matrix node supervises a rectangular area within the logic space of [0,1]× [0,1]. Hash functions are used to determine the matrix node that should store the description of a service. When requiring a service, the service request is hashed and routed in logic space to the matrix node indicated by hash results.

The basic idea of lanes[25] is to define a two-dimensional overlay structure. Nodes in one dimension form a lane, and each MANET contains multiple lanes. Nodes in the same lane share the same anycast address and they know about all services in the lane. Thus, service request packets only travel between lanes.

2.3 Multi-layer Approaches

In multi-layer approaches, nodes are organized into tree-like hierarchy structures. References [24] [25] propose several multi-layer hierarchy protocols, as shown in Fig.3.

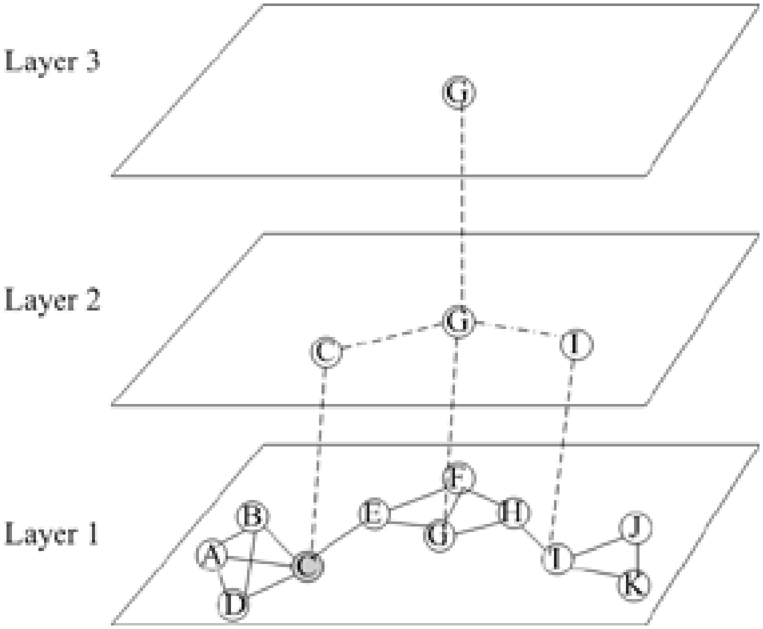


Fig. 3. Structure of multi-layer approaches

In service ring[24], children of a node are organized into rings. While in MultiLayerSDP[25], all nodes in a MANET form a normal DNS-like hierarchy structure.In ServiceRing, nodes are organized into rings according to the similarity of their services. In each ring, there is a SAP (Service Access Point) node. SAP nodes are organized into higher layer rings. Higher rings have their own SAP nodes and they form higher layers. With this mechanism hierarchy structure of arbitrary depth is built. When searching for a service, the service request is transferred up and down along the hierarchy rings.

Multi layer structure has good scalability but hard to maintain.

3 Comparative Analysis

The characteristics of these typical service discovery protocols for MANETs are listed in Table 1.

Table 1. Characteristics of typical service discovery protocols

	GSD	DSDP	VB	Proximity CAN	Lanes	Service Ring
Storage of service description	Packet spreading and cache	Home backbone node	Home backbone node, Hashed target	Home matrix node	All nodes in the same lane	Home SAP node
Request routing	Semantic routing	Topology routing	hashed target, Topology routing	Topology routing in logic area	Unicast between lanes	Semantic routing
structure	flat	backbone	cluster	matrix	lanes	tree-like
Robustness	Good	middle	middle	middle	middle	bad
Operation mode	Push-pull	Push-pull	Push-pull	Push-pull	Push-pull	Push-pull
Main overhead type	Advertisement packets	Structure maintenance	Structure maintenance	Structure maintenance, service description storing	Advertisement packet spreading within in lanes	Structure maintenance
Message overhead	middle	middle	middle	great	great	greatest

Hierarchical multi-layer structure, such as ServiceRing[24], is intrinsically of good scalability. But the highly dynamic nature of MANETs makes such a multi-layer structure hard and costly to maintain. So they are not suitable for MANETs. This has been proved by simulation results in [26].

Comparing to multi-layer structures, two-layer structures are simpler and easier to maintain.

In DSDP[16], packet transmissions are mainly routed along the backbones. Thus, request packet overhead is reduced, service response is prompted. However, backbone maintenance causes some packet overhead. Additionally, its service request packet forward policy does not benefit from the broadcast nature of wireless transmissions.

In VB[21], 1) cluster construction phase requires several loops of information exchanges; 2) storing service descriptions on the target cluster head causes much packet transmissions; 3) QoS determination procedure increases its packet overhead.

In ProximityCAN[22], storing service descriptions on target nodes causes much packet overhead. Adjacent points in logic space may be far away from each other in physical space. Hence, an optimal route in logic space may map to a quite worse path in physical space. Thus, more packets will be generated during request packet forwarding operation.

In Lanes[23], operations of service advertisement packet spreading and service request packet forwarding do not benefit from the broadcast nature of wireless transmissions.

Analysis above shows that DSDP[16] is superior to other two-layer protocols.

Compared to two-layer and multi-layer protocols, one-layer protocols, because of their intrinsically resistance to dynamic nature, are more suitable for highly dynamic MANETs. Reference [24] claims that semantic routing is a step in the right direction. GSD is just a semantic routing based protocol. However, it does not benefit from the broadcast nature of wireless transmissions either.

4 Simulation Study

4.1 Performance Metrics

Four performance metrics are considered in our simulations.

- **Number of Service Request Packets Per Session:** It measures the number of service request packets sent in one simulation. It reflects the efficiency the policy of forwarding service request packets.
- **Ratio of Succeeded Requests:** It is the number of service discovery sessions in which the client has received at least one successful reply packet. It reflects the effectiveness (service discoverability) of service discovery protocols.
- **Response Time (m/s):** It is the interval between the arrival of the first reply packet and the generation of the corresponding request packet. This metric is averaged over all succeeded service discovery sessions. It measures the promptness of service discovery protocols. It also reflects the average distance between clients and the corresponding first repliers.
- **Ratio of Succeeded Requests to Total-packet-number (Suc2Packet):** This metric is the ratio of Succeeded-SDP-number to the sum of service request packets and service reply packets. It reflects the efficiency of service discovery protocols.

4.2 Simulation Models

Simulation studies are performed using GloMoSim [27]. The distributed coordination function (DCF) of IEEE 802.11 is used as the underlying MAC protocol. Random Waypoint Model (RWM) is used as the mobility model.

In RWM mobility model, nodes move towards their destinations with a randomly selected speed $V \in [V_{min}, V_{max}]$. When reaching its destination, a node keeps static for a random period $T_P \in [T_{min}, T_{max}]$. When the period expires, the node randomly selects a new destination and a new speed, then moves to the new destination with the new speed. The process repeats permanently. In our simulations, $T_{min} = T_{max} = 0$,$V_{min} = V_{max} = V$.

4.3 Protocols in Comparison

We select several typical service discovery protocols for MAENTs and make comparative studies among them through extensive simulations.

As in many papers, FLOOD is selected as the benchmark for our simulation studies. One-layer protocols are more suitable for highly dynamic MANETs. Among one-layer protocols, GSD is more preferable for its interesting semantic routing policy. CNPGSDP improves GSD significantly by its excellent BSU (Broadcast Simulated Unicast) scheme and CNP (Candidate Node Pruning) scheme. Two-layer protocols are more suitable for more stable MANETs. Analysis in section 3 shows that DSDP is more preferable than other two-layer protocols. Hence, DSDP is selected. Hence, four protocols, FLOOD, GSD, CNPGSDP, DSDP, are implemented and compared in through simulation studies.

4.4 Simulation Settings

Some basic parameters that are used in all the following simulations are set as shown in Table 2. Simulation scenarios are created with 100 nodes randomly distributed in the scenario area. At the beginning of each simulation, some nodes are randomly selected out to act as servers. These selected servers provide randomly selected services. During each simulation, 100 SDP sessions are started at randomly selected time by randomly selected nodes.

Table 2. Basic parameters

Parameters	Value	Parameter	Value
Scenario area	1000m×1000m	Number of service group	2
Node number	100	Maximum hop of request packets	3
Server number	50	Valid time of SIC item	21s
Simulation time	1000s	Number of service info in each group	5
SDP session number	100	Service advertisement interval	20s
radio range	150m	Maximum hop of advertisement packets	1
Wireless bandwidth	1Mb/s		

4.5 Simulation Results

In this section, we inspect the effect of node speed on the selected protocols through simulations. To do this, we run 4 simulation sets that use the 4 selected service discovery protocols, respectively. Each set includes 5 subsets of simulations, where $V = V_{min} = V_{max}$ and V is set to 0m/s, 5m/s, 10m/s, 15m/s, and 20m/s, respectively. Each subset consists of 50 similar simulations. Simulation results are averaged over 50 simulations. The results are shown in Fig. 4 to Fig. 7. In all these figures showing simulation results, error bars report 95% confidence.

Fig. 4 shows the effect of node speed on the number of request packets per session. The number of request packets per session in CNPGSDP is the lowest among the 4 protocols. All other three protocols have almost the same value in this metric, and DSDP is a little superior to other two protocols. Number of request packets per session in CNPGSDP is about 7% to 23% of that in DSDP.

Fig. 5 shows the effect of node speed on ratio of succeeded sessions. The result shows that 1) DSDP has the lowest service discoverability; 2) two group-based protocols are superior to others, which is because that each server spread its services to neighbors.

Fig. 6 shows the effect of node speed on response time. GSD and CNPGSDP are more prompt than FLOOD and DSDP. CNPGSDP becomes more prompt as node speed increase, whereas GSD does not. This is because that 1) node's movement enlarges the spreading range of service advertisement packets, and 2) packet transmission in CNPGSDP is more effective because of its fewer packet transmissions.

Fig. 7 shows that CNPGSDP outperforms all other protocols distinctly. Suc2Packet of CNPGSDP is about 4 to 6 times of other protocols. The superiority of CNPGSDP becomes more significant as node speed increases.

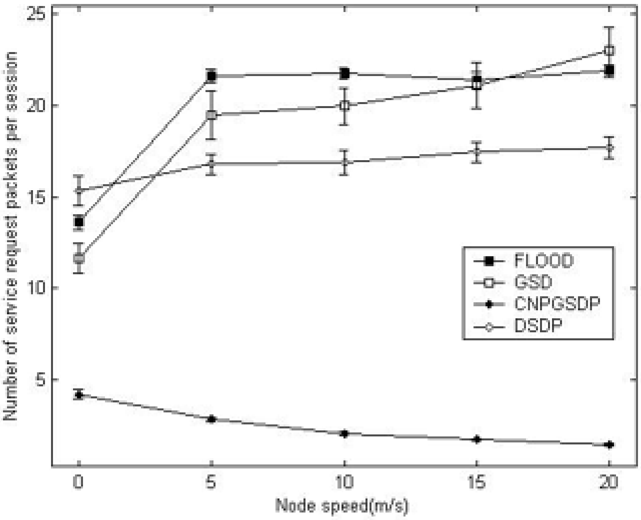


Fig. 4. Number of request packets per session vs. node speed

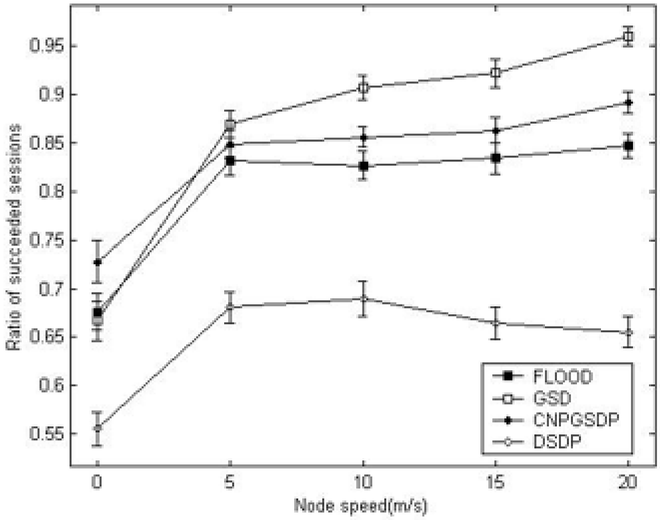


Fig. 5. Ratio of succeeded sessions vs. node speed

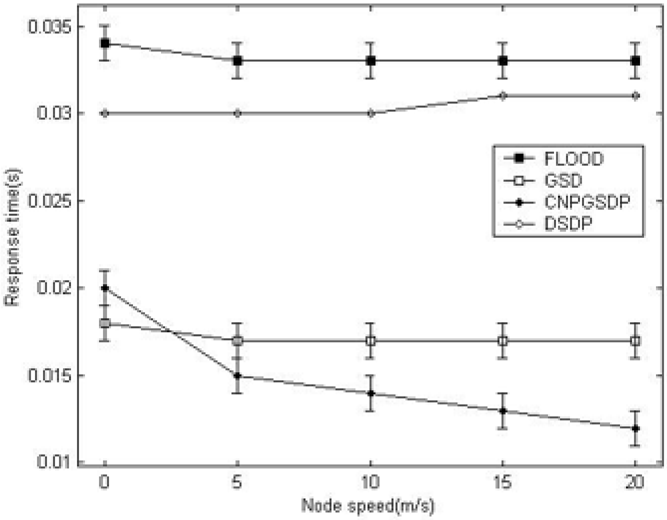


Fig. 6. Response time(s) vs. node speed

From above simulations, a conclusion can be made.

- CNPGSDP is generally the most effective, the most efficient, and the most prompt one among tested service discover protocols.
- the superiority of CNPGSDP is generally more significant with higher node speed.

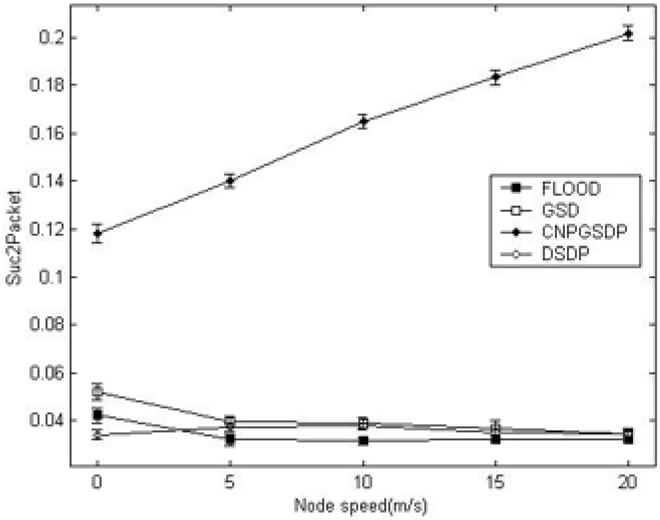


Fig. 7. Suc2packet vs. node speed

- DSDP is more suitable for MANETs with longer radio range and more servers.
- DSDP is almost the slowest protocol in all tested cases.

5 Conclusions

In this paper, existing service discovery protocols for MANETs are categorized. Several typical services discovery protocols representing these categories are introduced and analyzed in detail. Comparative analysis and simulation studies are then performed. Future works are prospected at last. From the paper, an overall perspective of service discovery protocols for MANETs can be obtained.

Simulation results show that CNPGSDP is almost the most preferable protocol for MAENTs. DSDP is more suitable for MANETs with longer radio range and more servers. However, DSDP is slower than other protocols.

References

1. IETF, "Mobile ad-hoc network (MANET) working group". [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html.
2. F. Zhu, M. Mutka, L. Ni, "PrudentExposure: a Private and User-centric Service Discovery Protocol." Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications, Orlando,Florida, 2004: 329-340
3. Y. Yuan, W. Arbaugh, "A secure service discovery protocol for MANET," Proc. 14th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Communication, 2003, vol. 1, pp. 502-506

4. C. N. Ververidis, G. C. Polyzos, "Routing Layer Support for Service Discovery in Mobile Ad Hoc Networks". Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications - Pervasive Wireless Networking Workshop, Hawaii, USA, 2005: 258-262
5. C. N. Ververidis, G. C. Polyzos, "Extended ZRP: Performance Evaluation of a Routing Layer based Service Discovery Protocol for Mobile Ad Hoc Networks". Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, San Diego, California, USA, 2005: 114-123
6. A. Varshavsky, B. Reid, E. D. Lara, "The Need for Cross-layer Service Discovery in MANETs". Technical Report CSRG-492, Department of Computer Science, University of Toronto, 2004
7. S. Helal, N. Desai, V. Verma, C. Lee, "Konark - a service discovery and delivery protocol for ad-hoc networks," Proc. 3rd IEEE Conf. Wireless Communication Networks, 2003, pp. 2107-2133
8. Z. G. Gao, X. Z. Yang, S. B. Cai, "Flexible forward probability based service discovery protocol for MANETs," Journal of Harbin Institute of Technology (Chinese), 2005, 37(9): 1256-1260
9. Z. G. Gao, X. Z. Yang, T. Y. Ma, S. B. Cai, "RICFFP: an efficient service discovery protocol for MANETs," Proc. Int'l Conf. Embedded And Ubiquitous Computing, 2004, pp. 786-795
10. Y. Sasson, D. Cavin, A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," Proc. 3rd IEEE Conf. Wireless Communication Networks, 2003, pp. 1125-1130
11. D. Chakraborty, A. Joshi, Y. Yesha, T. Finin, "GSD: a novel group-based service discovery protocol for MANETs," Proc. 4th IEEE Conf. Mobile and Wireless Communications Networks, 2002, pp. 140-144
12. D. Chakraborty, A. Joshi, Y. Yesha, T. Finin, "Towards Distributed Service Discovery in Pervasive Computing Environments", IEEE Transactions on Mobile Computing. To appear
13. O. Ratsimor, D. Chakarborty, A. Joshi, T. Finin, "Allia: Alliance-based Service Discovery for Ad-hoc Environments". Proceedings of the 2nd ACM International workshop on Mobile commerce, Atlanta, Georgia, USA, 2002: 1-9
14. Z. G. Gao, L. Wang, M. Yang, X. Z. Yang, "CNPGSDP: An Efficient Group-based Service Discovery Protocol for MANETs". Computer Networks. 2006, 29(12): 2433-2445
15. Y. C. Tseng, S. Y. Ni, Y. S, Chen, J. P. Sheu, The broadcast storm problem in a mobile ad Hoc network. ACM Wireless Networks. 2002, 8(2): 153-167
16. U. C. Kozat, L. Tassiulas, "Service Discovery in Mobile Ad Hoc Networks: an Overall Perspective on Architectural Choices and Network Layer Support Issues". Ad Hoc Networks. 2003, 2(1): 23-44
17. A. Greede, D. O. Mahony, A Service Driven Routing Protocol for Bluetooth Scatternets. Proceedings of the 5th European Personal, Mobile Communications Conference, Glasgow, Scotland, 2003: 307-311
18. N. A. Nordbotten, T. Skeie, N. D. Aakvaag, Service Discovery in Highly Dynamic Scatternets. Proceedings of the Workshop on Applications and Services in Wireless Networks, Bern, Switzerland, 2003: 211-220
19. N. A. Nordbotten, T. Skeie, N. D. Aakvaag, "Methods for Service Discovery in Bluetooth Scatternets". Computer Communications. 2004, 27(11): 1087-1096
20. J. Nuevo, J. C. Gregoire, Proposition of a Hierarchical Service Distribution Architecture for Ad Hoc Networks based on the Weighted Clustering Algorithm. Proceedings of the 5th European Wireless Conference, Barcelona, Spain, 2004

21. J. C. Liu, Q. Zhang, W W. Zhu, B. Li, ”Service Locating for Large-scale Mobile Ad Hoc Network”. International Journal of Wireless Information Networks. 2003, 10(1): 33-40
22. H. J. Yoon, E. J. Lee, H. Jeong, J. S. Kim, ”Proximity-based Overlay Routing for Service Discovery in Mobile Ad Hoc Networks”. Proceedings of the 19th International Symposium on Computer and Information Sciences, Lecture Notes in Computer Science, vol. 3280, 2004: 176-186
23. M. Klein, B. K. Ries, P. Oberiter, ”Lanes - a Lightweight Overlay for Service Discovery in Mobile Ad Hoc Networks”. Proceedings of the 3rd Workshop on Applications and Services in Wireless Networks, Berne, Switzerland, 2003: 101-112
24. M. Klein, B. K. Ries, P. Obreiter, ”Service rings - a Semantic Overlay for Service Discovery in Ad Hoc Networks”. Proceedings of the 14th International Workshop on Database and Expert Systems Applications, Prague, Czech, 2003: 180-185
25. M. Klein, B. K. Ries, ”Multi-layer Clusters in Ad-Hoc Networks - an Approach to Service Discovery” Proceedings of the 1st International Workshop on Peer-to-Peer Computing, Pisa, Italy, 2002: 187-201
26. M. Klein, M. Hoffman, D. Matheis, M. Mussig, ”Comparison of Overlay Mechanisms for Service Trading in Ad Hoc Networks”. Technical Report Nr. 2004-2, University of Karlsruhe, 2004
27. Zhenguo Gao, Yunlong Zhao, Xiang Li, Shaobin Cai, Chunsheng Wang, Jianwen Cui, ”The Analyzation of the GloMoSim Wireless Network Simulator”. Journal of System Simulation (Supplement), 2006, 18(8s): 672-675

A BDD-Based Heuristic Algorithm for Design of Reliable Networks with Minimal Cost

Gary Hardy[1], Corinne Lucet[1], and Nikolaos Limnios[2]

[1] LaRIA, CNRS FRE 2733, Amiens, France
[2] LMAC, EA 2222, Compiegne, France

Abstract. This paper describes a new algorithm based on the Binary Decision Diagrams (BDDs) to solve the reliable communication network design problem (RCND). In this NP-hard problem, a subset of communication links must be chosen such that the network cost is minimized given a network reliability constraint. This problem is closely related to the network reliability computation problem.

1 Introduction

Topological optimization of networks is an important problem in many real world fields such as telecommunications, electricity distribution, oil and gas lines and computer networking. It is of great importance in the design of communication networks when considering performance criterion such transmission delay or network reliability.

This paper describes an heuristic algorithm to solve the reliable communication network design problem (RCND). The aim is to design a communication network topology with minimal cost that satisfies a given reliability constraint. We assume that a set of perfectly reliable nodes and their topology are given, along with a set of possible communication links that connect them. Each link has a known reliability and cost per unit distance (commonly a link is assigned a weight which is used as its complete cost). The relevant reliability metric is *all-terminal network reliability* defined as the probability that every pair of nodes can communicate with each other. This means that the network forms at least a spanning tree. A trade-off between the investments and the quality of service provided to the users must be found. Both the network design problem and the network reliability computation, have been proven to be NP-hard [4,5].

In literature, this problem has been studied with exact enumerative methods [9] and heuristic methods [6,7,10,11]. The main contribution of this paper is based on the first use of BDD [1,2] in an approached resolution of the RCND problem. Contrary to the other methods that estimate all-terminal reliability using a Monte Carlo simulation approach, we use the exact BDD-based method we introduced in [8] that allows to compute the network reliability even for large scale networks with small linear-width. This method consists in encoding the network structure function into a BDD then the network reliability (among other reliability measures such as importance measures) is easily deduced from this data structure. Consequently, the BDD can be re-used and be seen as a powerful tool for designing highly reliable networks with our heuristic algorithm. Unfortunately, very few numerical results have been presented in literature, so we compare our algorithm with the best results obtained by genetic algorithm in [6,7].

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 244–255, 2006.

In the following section we give a short problem description. Next Section 3 briefly explains our exact BDD-based method for computing network reliability. The concept of importance measures (also known as sensitivity analysis), which rank network links according to their contribution to the network reliability, is presented in Section 4. A new importance measure dedicated to the RCND problem is also introduced. Then Section 5 presents a new algorithm for solving the network design problem given a cost constraint and a minimum network reliability constraint. Several benchmarks taken from [6,7] underline the effectiveness of our method. Finally, we draw some conclusions and outline the direction of future works in Section 6.

2 Statement of the Problem

For the RCND problem [3], a network topology with minimal cost must be found that satisfies a given reliability constraint. Two network reliability constraints are often considered:

- *2-terminal network reliability*
- *all-terminal network reliability* (or *overall network reliability*)

The 2-terminal reliability is the probability that two distinguished nodes s and t are connected by at least one path of operating edges in G. The all-terminal reliability is the probability that G remains connected, *i.e*, there exists at least one path made of functioning edges between each pair of nodes. This paper focuses on this latter network reliability. This RCND problem is defined as follows:

A network is modeled by an undirected graph $G = (V, E)$ with V its vertex set (representing sites) and $E \subseteq V \times V$ its possible edge set (representing the candidate links between the sites). We denote m the number of candidate edges ($m = |E|$).

The objective function of this problem is stated as:

$$\text{Minimize } C(G') = \sum_{i=1}^{m} c_i x_i$$
$$\text{subject to: } R(G') \geq R_0$$

where:

- x_i ($x_i \in \{0, 1\}$) is equal to 1 if the edge e_i exists in the solution and 0 otherwise.
- c_i is the cost of a link $e_i \in E$.
- $G' = (V, E')$ is a partial graph of G such that $E' = \{e_i \in E / x_i = 1\}$.
- $C(G')$ is the total cost of network G'.
- $R(G')$ is the network all-terminal reliability.
- R_0 is the minimum reliability requirement.

The objective function is the sum of the total cost for all chosen network links. In other words, the RCND problem consists in finding a partial graph $G' = (V, E')$ of G ($E' \subseteq E$) with minimal cost such that $R(G') \geq R_0$. We made the following assumptions:

- We consider the vertices as perfect.
- Each link can fail independently and randomly.
- The links have two states: either operational or failed.
- The cost c_i and the functioning probability p_i of each link e_i is known.
- Each link is bi-directional.
- At most one link is possible between each pair of nodes.

3 BDDs and Network Reliability

In this section, we introduce the notion of Binary Decision Diagrams and the link between the network reliability and its encoding by this data structure. From BDD representing the network structure function, the network reliability among other network reliability measures can be easily deduced.

3.1 Binary Decision Diagrams

A BDD is a data structure that is used to represent a Boolean function [12,13]. BDD is based on a decomposition of Boolean functions called *Shannon decomposition*. A Boolean function f can be decomposed in terms of a Boolean variable x as:

$$f = x.f_{x=1} + \bar{x}.f_{x=0}$$

where $f_{x=i}$ is f with x replaced by constant i ($i \in \{0,1\}$).

A Boolean function can be represented as a *rooted directed acyclic graph* which consists of decision nodes and two terminal nodes called *0-terminal* and *1-terminal*. Each decision node is labeled by a Boolean variable and has two child nodes called *low child* and *high child*. The edge from a node to a *low* (*high*) child represents an assignment of the variable to 0 (1). Such graph is called *BDD* if the variables occur in the same ordering on all paths and it is reduced according to two rules:

- Merge any isomorphic subgraphs.
- Eliminate any node whose two children are isomorphic.

A path from the root node to the *1-terminal* represents a variable assignment for which the represented Boolean function is true. Figure 1 illustrates the represention of Boolean function $(x_1 \Leftrightarrow x_3) \wedge (x_2 \Leftrightarrow x_4)$ by BDD.

3.2 All-Terminal Network Reliability Computation

Our network model is an undirected stochastic graph $G = (V, E, p, c)$. The graph is said stochastic because each link can fail, statistically independently, with known probability. We consider the vertices as perfect. The failure probability of a link e_i ($i \in \{1, \ldots, m\}$) is denoted q_i ($q_i \in [0,1]$) and its reliability p_i ($p_i = 1 - q_i$). Hence, p represents a probabilistic vector $(p_1, \ldots, p_m)$. Similarly, we note $p_{(0,i)}$ the vector $(p_1, \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_m)$. At each edge e_i is associated a cost c_i (we note $c = (c_1, \ldots, c_m)$). Let X_i be the binary random variable "state of the link e_i in G", defined by $X_i = 1$ if link e_i is operational, and $X_i = 0$ if link e_i is down.

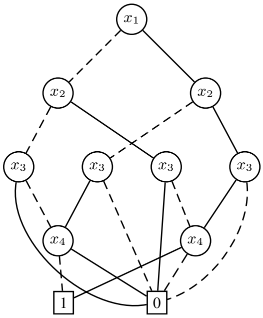


Fig. 1. Boolean function $(x_1 \Leftrightarrow x_3) \wedge (x_2 \Leftrightarrow x_4)$ representing by BDD. A *dashed* (*solid*) line represents the value 0 (1).

$X = (X_1, \ldots, X_m)$ is the *random network state vector*. Given a graph state $\mathcal{G} = (x_1, x_2, \ldots, x_m)$ ($x_i \in \{0, 1\}$), we consider the graph $G_{\mathcal{G}} = (V, E_{\mathcal{G}})$ defined by:

$$E_{\mathcal{G}} = \{e_i \in E, x_i = 1\}$$

The associated probability of $\mathcal{G}$ is defined as:

$$Pr(X = \mathcal{G}) = \prod_{i=1}^{m} (x_i.p_i + (1 - x_i).q_i)$$

The all-terminal network reliability is then given by:

$$R(G; p) = \sum_{\mathcal{G}/G_{\mathcal{G}} \text{ is connected}} Pr(X = \mathcal{G})$$

In order to compute the all-terminal network reliability, we use our method presented in [8]. In short, the network reliability is encoding by a BDD which represents the network structure function $\phi : \{0, 1\}^m \to \{0, 1\}$ defined as follows:

$$\begin{cases} \phi(X) = 1 & \text{if } G_X \text{ is connected} \\ \phi(X) = 0 & \text{otherwise} \end{cases}$$

Now we consider the reliability with respect to the structure function ϕ:

$$R(G; p) = \sum_{\mathcal{G}} Pr(X = \mathcal{G}).\phi(\mathcal{G})$$

This BDD structure is a compact and implicit representation of the entire set of the functioning and failing network states. Hence, it avoids huge storage for large number of networks states. As BDD is based on Shannon decomposition, the probability of the all-terminal connectivity is quickly computed in a recursive way:

$$\begin{aligned}\forall k \in [1 \ldots m] \ :& \\ R(G;p) &= Pr(\phi = 1) \\ R(G;p) &= Pr(x_k.\phi_{x_k=1} = 1) + Pr(\bar{x_k}.\phi_{x_k=0} = 1) \\ R(G;p) &= p_k.Pr(\phi_{x_k=1} = 1) + q_k.Pr(\phi_{x_k=0} = 1)\end{aligned}$$

Clearly, we can compute the reliability of the same network for different values of p_i. Once the BDD Φ encoding the network structure function is constructed, the network reliability can be exactly computed for all p_i (Fig. 2(c)) by applying Algorithm 1. That is to say that the network reliability can be re-evaluated without BDD construction each time the failure probability of one or many links is changed.

Our algorithm for encoding the network reliability by BDD and computing the all-terminal reliability runs in $O(m.w.B_w)$ where w is the *linear-width* of the graph [15] and B_w (Bell number) is the number of partitions of a set with w elements.

Others network reliability measures useful for the network reliability optimization could also be found. The next section emphazises the key role of the BDD structure in this process of optimization.

Algorithm 1. $Rel\_Net(\Phi, p)$

```
input:
 • $\Phi$: BDD encoding network reliability
 • $p$: probabilistic vector $(p_1, \ldots, p_i, \ldots, p_m)$
/* $p_i = Pr(x_i = 1) = 1 - q_i$ with $x_i$ top variable of $\Phi$ */
/* $\Phi = x_i.\Phi_1 + \bar{x_i}.\Phi_0$ */
output:
 • $R(G;p)$

if ($\Phi ==$ 1-terminal) then
   return 1
end if
if ($\Phi ==$ 0-terminal) then
   return 0
end if
if (entry $\{\Phi\}$ exists in hash table) then
   return corresponding value $prob$
else
   $P_1 = Rel\_Net(\Phi_1, p)$
   $P_2 = Rel\_Net(\Phi_0, p)$
   $prob = p_i * P_1 + q_i * P_2$
end if
insert entry $\{\Phi\}$ with corresponding value $prob$ in hash table.

return $prob$
```

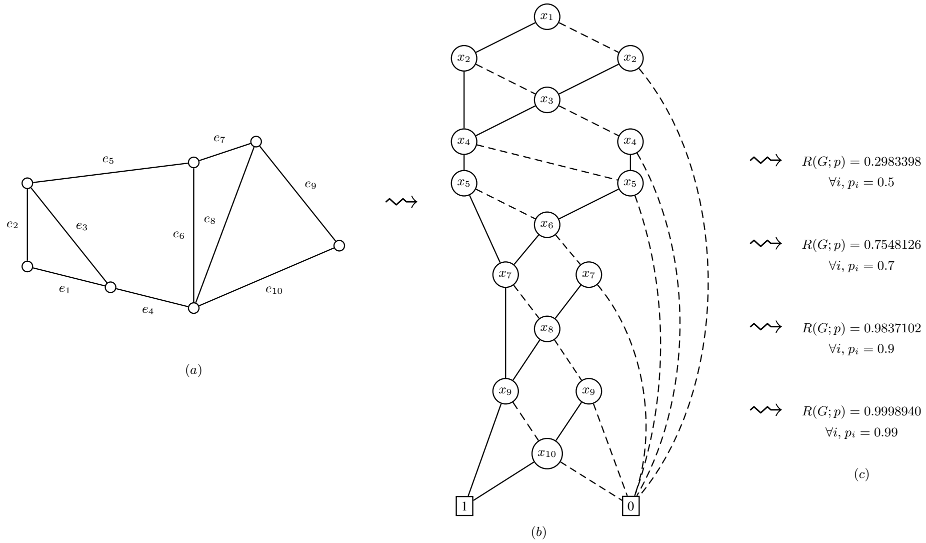


Fig. 2. *(a)* Network G *(b)* Structure function of G encoding by BDD. A *dashed* (*solid*) line represents the value 0 (1). *(c)* All-terminal reliability computation for a given p.

4 Importance Measures

4.1 Introduction

The importance concept was originally introduced by Zygmund Birnbaum in 1969 [14] who proposes quantitative ranking measures for components. The concept came from the fact that some of the components are more important than others in providing certain system characteristics. Importance measures are used to detect design weaknesses and component failures that are critical to the proper functioning of a system. They can assist in identifying the component whose improvement is most likely to yield the greatest improvement in system overall performance. Obviously, a communication network is seen as a system where components (*i.e* communication links) are subject to failure and can be improved in term of reliability. Since there are several possible interpretations of the relationship between a component event and a system event, various methods exist to compute the importance of components. The two most commonly used importance measures are:

- *Birnbaum importance measure*
- *Criticality importance measure*

Birnbaum [14] proposed the following importance measure for a component e_k:

$$I_k^B = R(G/e_k \textit{ is up}; p) - R(G/e_k \textit{ is down}; p)$$
$$I_k^B = Pr(\phi_{x_k=1} = 1) - Pr(\phi_{x_k=0} = 1)$$

This measure gives the contributions to the system reliability due to the reliability of the various system components. Components with the largest variation in reliability results have the highest importance. This measure is independent of the actual unavailability of component e_k, which can lead to assigning high importance measures to components that are very unlikely to fail and may be very difficult to improve. Therefore, to focus on components that are not only critical to the system reliability but also are more likely to occur, a modified importance measure known as *Criticality importance measure* is typically used to determine the next basic component to improve. This measure is defined as:

$$I_k^C = I_k^B * \frac{p_k}{R(G;p)}$$
$$I_k^C = I_k^B * \frac{Pr(x_k = 1)}{Pr(\phi = 1)}$$

for a component e_k of network G.

4.2 Network Importance Measure

Unfortunately, these previous measures are not intended for this network design problem. They are used to identify the weakest links in order to improve the overall network reliability. We want to provide a slightly different classification since the objective is to identify "superfluous" links with high cost. We have to propose a new measure relying on similar principles. The following facts must be taken into consideration:

- links have associated costs.
- we are only interesting in the degradation of network reliability.

In order to consider these parameters, we proposed this importance measure for ranking network links:

$$I_k^N = \frac{1}{c_k} * (R(G;p) - R(G/\ e_k\ \textit{is down};p))$$
$$I_k^N = \frac{1}{c_k} * (Pr(\phi = 1) - Pr(\phi_{x_k=0} = 1))$$

In other words, this reliability measure is the ratio of the degradation of the network reliability if link e_k is down and the cost of this link. This measure is inspired by the knapsack problem. This other combinatorial problem derives its name from the maximization problem of choosing possible essential items that can fit into one bag (of maximum weight) to be carried on a trip. For each item, its *"price per pound"* is computed, and we take as much of the most expensive items until the knapsack is full (or the items lack). Hence, the link with the smallest value has the best profit in the network degradation. Clearly, this measure can be computed for each link e_k by traversing twice the BDD structure ($I_k^N = \frac{1}{c_k} * (Rel\_net(\Phi, p) - Rel\_net(\Phi, p_{(0,k)}))$). We give the link classification using this network importance measure of network shown in Fig. 3.

e_k	p_k	c_k	I_k^N (rank)
e_1	0.90	89	0.0010736 (6)
e_2	0.90	89	0.0010736 (7)
e_3	0.90	179	0.0000785 (5)
e_4	0.80	159	0.0001196 (9)
e_5	0.95	284	0.0001913 (12)
e_6	0.80	79	0.0001156 (8)
e_7	0.95	189	0.0002346 (13)
e_8	0.80	79	0.0000274 (1)
e_9	0.90	89	0.0001499 (11)
e_{10}	0.90	179	0.0000667 (3)
e_{11}	0.80	79	0.0001370 (10)
e_{12}	0.90	89	0.0002977 (14)
e_{13}	0.90	79	0.0000768 (4)
e_{14}	0.80	159	0.0000624 (2)

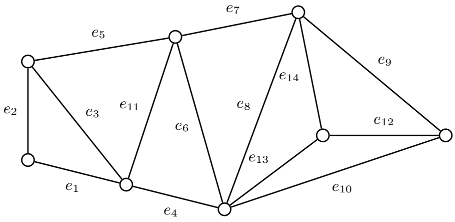


Fig. 3. *Network G.* $R(p; G) = 0.980763$

5 Description of the Approach

5.1 The *NDImprovement* Algorithm

The basic idea mainly relies on the BDD encoding the network structure function since both the overall network reliability and the network importance measure of a link are computed by using this structure. The *NDImprovement* algorithm (algorithm 2) is divided into two main steps:

- Starting from a maximal theoretical network, the algorithm tries to lower the network reliability to R_0 by removing unuseful edges (*i.e.* by fixing their functioning probability to 0) selected from the network importance measure (Section 4.2). This process is applied until reaching the minimal reliability constraint R_0. A link with a functioning probability fixed to 0 is considered as not present in the solution.
- Starting from this first solution, we try to find a better solution (i.e a network with a lower cost that still satisfies the network reliability constraint) by switching selected edges for previously discarded edges with lower cost.

This process is described in algorithm 2 (*NDImprovement()*). An example is shown on test problem in Fig. 3. The results obtained for different values of R_0 are shown in table 1. In 5 out of 6 cases, the optimal cost was found. Figure 4 illustrates two network topologies with minimal cost for 2 different reliability constraints.

Theorem 1. *The whole process, BDD generation plus algorithm NDImprovement, runs in* $O(m^3.B_w)$.

Proof. The algorithm for constructing the BDD structure runs in $O(m.w.B_w)$. *Rel_Net()* runs in $O(m.B_w)$ (the size of BDD is bounded by $m.B_w$). In *NDImprovement*, the computation of the network importance measures of m edges is done in $\theta(m^2.B_w)$. *Step 2* is executed at most m times. In *Step 3* at most m^2 edge switches could be done. Hence, the algorithm time complexity is bounded by $O(m^3.B_w)$.

Algorithm 2. NDImprovement()

```
input:
  • $G = (V, E, p, c)$
  • BDD $\Phi$ encoding the network reliability of $G$.
  • $R_0$: minimal all-terminal network reliability required
output:
  • partial network $G'$ (with edge set $E' \subseteq E$) represents the best solution found.
  • total cost $C_{best}$ associated.
Step 1 (Initialization)
 (a) $E' = E$
 (b) Compute $R_{max} = R(G, p)$ by applying Algorithm 1.
     $if (R_{max} < R_0)\ then$ the problem has no solution
 (c) $C_{init} = \sum_i c_i\ (i = 1, \ldots, |E|)$
     $C_{best} = C_{init}$
Step 2
 (a) Order links in $E'$ according to their network importance measure.
 (b) Select link $e_i$ which has the minimum measure such that $R(G'/e_i$ is down$) \geq R_0$.
 (c) $if$ such link does not exist $then$ go to Step 3 $else$
       ∗ $p_i = 0$
       ∗ $E' = E' \setminus e_i$
       ∗ $C_{best} = C_{best} - c_i$
       ∗ go to Step 2
Step 3
  • For each edge $e_i$ in $E'$:
       For each edge $e_{i'}$ in $E \setminus E'$ such that $c_{i'} < c_i$:
          if $R(G'/e_i$ down & $e_{i'}$ up$) \geq R_0$ and $R(G'/e_i$ down & $e_{i'}$ up$) \leq R(G')$:
             ∗ restore $p_{i'}$ from its original value
             ∗ $p_i = 0$
             ∗ $E' = E' \setminus e_i \cup \{e_{i'}\}$
             ∗ $C_{best} = C_{best} - c_i + c_{i'}$
```

5.2 Experimental Results

The test problems (fully-connected networks) are taken from [6,7] and the results obtained with *NDImprovement* are summarized in Table 2. The link costs were randomly

Table 1. Experimental results

R_0	C_{init}	C_{min}	C_{best}	$R(G)$	*CPU sec.*
0.95	2011	1674	1674	0.95065	0.03
0.92	2001	1415	1415	0.92164	0.03
0.90	2011	1384	1415	0.92164	0.03
0.85	2011	1256	1256	0.85083	0.03
0.80	2011	1200	1200	0.81304	0.03
0.70	2011	1066	1066	0.71777	0.03

Table 2. Experimental results of NDImprovement

$\|V\|$	$\|E\|$	p	R_0	C_{init}	C_{min}	C_{best}	R	$\|BDD\|$	CPU sec.
8	28	0.90	0.90	1343	208	218	0.9278960	2745	0.18
8	28	0.90	0.90	1351	203	213	0.9202432	2745	0.19
8	28	0.90	0.90	1352	211	211	0.9202432	2745	0.16
8	28	0.90	0.90	1452	291	300	0.9125905	2745	0.12
8	28	0.90	0.90	1263	178	181	0.9345921	2745	0.18
8	28	0.90	0.95	1343	247	259	0.9614724	2745	0.02
8	28	0.90	0.95	1351	247	253	0.9529587	2745	0.02
8	28	0.90	0.95	1352	245	245	0.9529587	2745	0.08
8	28	0.90	0.95	1452	336	351	0.9570147	2745	0.08
8	28	0.90	0.95	1263	202	202	0.9518108	2745	0.09
8	28	0.95	0.95	1343	179	179	0.9637055	2745	0.13
8	28	0.95	0.95	1351	194	196	0.9637055	2745	0.14
8	28	0.95	0.95	1352	197	197	0.9637055	2745	0.15
8	28	0.95	0.95	1452	276	280	0.9602138	2745	0.15
8	28	0.95	0.95	1263	173	184	0.9706888	2745	0.16
9	36	0.90	0.90	1859	239	244	0.9323920	10265	1.06
9	36	0.90	0.90	1897	191	194	0.9065639	10265	1.05
9	36	0.90	0.90	1828	267	273	0.9039811	10265	1.02
9	36	0.90	0.90	1749	171	183	0.9143124	10265	1.05
9	36	0.90	0.90	1678	198	198	0.9121600	10265	1.05
9	36	0.90	0.95	1859	286	286	0.9566703	10265	1.03
9	36	0.90	0.95	1897	220	237	0.9504716	10265	1.02
9	36	0.90	0.95	1828	306	306	0.9527789	10265	1.03
9	36	0.90	0.95	1749	219	219	0.9559816	10265	1.09
9	36	0.90	0.95	1678	237	239	0.9512120	10265	1.02
9	36	0.95	0.95	1859	209	209	0.9669353	10265	1.19
9	36	0.95	0.95	1897	171	171	0.9536669	10265	1.13
9	36	0.95	0.95	1828	233	249	0.9586425	10265	1.16
9	36	0.95	0.95	1749	151	177	0.9685938	10265	1.07
9	36	0.95	0.95	1678	185	206	0.9536669	10265	1.09
10	45	0.90	0.90	1803	131	131	0.9119878	39856	9.76
10	45	0.90	0.90	2155	154	154	0.9050143	39856	9.68
10	45	0.90	0.90	2546	263	263	0.9358917	39856	9.79
10	45	0.90	0.90	2517	293	309	0.9055179	39856	9.59
10	45	0.90	0.95	1803	153	164	0.9509778	39856	9.99
10	45	0.90	0.95	2155	197	205	0.9505284	39856	9.85
10	45	0.90	0.95	2546	291	309	0.9585558	39856	9.69
10	45	0.90	0.95	2517	358	366	0.9562390	39856	9.56
10	45	0.95	0.95	1803	121	127	0.9532522	39856	9.45
10	45	0.95	0.95	2155	136	144	0.9548279	39856	9.68
10	45	0.95	0.95	2546	245	256	0.9579791	39856	9.55
10	45	0.95	0.95	2517	268	277	0.9571913	39856	9.87

Table 3. Large networks

V	E	p	R_0	C_{init}	C_{best}	R	$\|BDD\|$	CPU sec.
12	66	0.6	0.98	17094	8237	0.9801417	673934	218
12	66	0.6	0.99	17094	9792	0.9901090	673934	192
12	66	0.6	0.999	17094	15357	0.9990244	673934	60
64	112	0.99	0.90	49896	14949	0.9004722	179410	104
64	112	0.99	0.95	49896	15840	0.9673246	179410	100
64	112	0.99	0.99	49896	19800	0.9902008	179410	92
81	144	0.99	0.90	70848	22906	0.9014549	797916	772
81	144	0.99	0.95	70848	26352	0.9511343	797916	721
81	144	0.99	0.99	70848	42563	0.9901553	797916	600

Table 4. Summary of our approach and comparison to LS/NGA

Problem			**LS/NGA**		**NDImprovement**	
$\|V\|$	$\|E\|$	*Search space*	*Mean % from optimal*	Mean CPU sec. (P 133MHz)	*Mean % from optimal*	Mean CPU sec. (P 3GHz)
8	28	$2.68 * 10^8$	0.889	118.75	0.0224	0.12
9	36	$6.87 * 10^{10}$	1.050	203.38	0.0343	1.07
10	45	$3.15 * 10^{13}$	1.094	458.93	0.0303	9.65

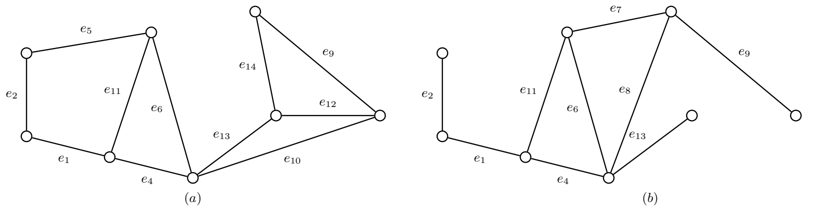


Fig. 4. Network topologies with minimal cost for $R_0 = 0.95$ *(a)* and $R_0 = 0.70$ *(b)*

generated over [0,100]. Details are available from the authors. C_{init} is the initial cost of the network G. C_{min} represents the minimal cost given the required reliability R_0. C_{best} is the best cost obtained with algorithm *NDImprovement*. In all the cases, a near-optimal solution is found. Our computations were made on a Pentium 3 GHz PC using C code. Table 4 displays a summary of the test problems comparing the performance of our method with the GA approach LS/NGA [6] in term of nearness to optimality and computational effort. For all instances, *NDImprovement* gives always a better solution than LS/NGA with a very improved gap. In addition, we proposed a set of larger networks in order to underline the effectiveness of *NDImprovement* (Table 3).

6 Conclusions and Future Work

This paper deals with the network design problem subject to reliability constraint. A method based on the Binary Decision Diagram for solving this problem was presented.

The first results are encouraging and showed that this method is intended to solve larger real word networks. Our future work will focus on similar problems as design of reliable networks with more or other constraints (such as delay or throughput) or the maximization of network reliability given a maximum budget.

Acknowledgment

The authors acknowledge financial support from the *Conseil Regional de Picardie*.

References

1. Bryant, R.: Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. ACM Computing Surveys **24**(3) (1992) 293–318
2. Akers, B.: Binary Decision Diagrams. IEEE Trans. On Computers **C-27** (1978) 509–516
3. Boorstyn, R., Frank, H.: Large-scale network topological optimization. IEEE Trans. on Reliability **25** (1977) 29–37
4. Garey, M., Johnson, D.: Computers and Intractibility: A guide to the Theory of NP-Completness. W. H. Freeman and Company, San Francisco, 1979
5. Ying, L.: Analysis method of survivability of probabilistic networks. Military Communication Technology Magazine **48** (1993)
6. Dengiz, B., Altiparmak, F., Smith, A.: Local search genetic algorithm for optimal design of reliable networks. IEEE Trans. on Evolutionary Computation **1**(3) (1997) 179–188
7. Cheng, S-T.: Topological optimization of reliable communication network. IEEE Trans. on Reliability **47**(3) (1998) 225–233
8. Hardy, G., Lucet, C., Limnios, N.: Computing all-terminal reliability of stochastic networks with Binary Decision Diagrams. Proc. 11th International Symposium on Applied Stochastic Models and Data Analysis, may 2005
9. Jan, R.-H., Hwang, F.-J., Chen, S.-T.: Topological optimization of a communication network subject to a reliability constraint. IEEE Trans. on Reliability **42** (1993) 63–70
10. Aggarwal, K., Chopra, Y., Bajwa, J.: Topological layout of links for optimising the overall reliability in a computer system. Microelectronics and Reliability **22** (1982) 347–351
11. Venetsanopoulos, A., Singh, I.: Topological optimization of communication networks subject to reliability constraints. Problem of Control and Information Theory **15** (1986) 63–78
12. Bryant, R.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans. on Computers **C-35**(8) (1986) 677–691
13. Akers, S.: Binary Decision Diagrams. IEEE Trans. on Computers **C-27**(6) (1978) 509–516
14. Birnbaum, Z.: On the importance of different components in a multicomponent system. in P. R. Krishnaiah, ed., Multivariate Analysis-II Academic Press, New York, (1998) 581–592
15. Bodlaender, H., Thilikos, D.: Computing small search numbers in linear time. Technical Report Technical Report No. UU-CS-1998-05, Dept. of Computer Science, (1998)

Coverage-Enhancing Algorithm for Directional Sensor Networks*

Dan Tao, Huadong Ma, and Liang Liu

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
School of Computer Science & Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China
tdfxy@vip.sina.com, mhd@bupt.edu.cn

Abstract. Adequate coverage is very important for sensor networks to fulfill the issued sensing tasks. In traditional sensor networks, the sensors are based on omni-sensing model. However, directional sensing sensors are with great application chances, typically in video sensor networks. Toward this end, this paper addresses the problem of enhancing coverage in a directional sensor network. First, based on a rotatable directional sensing model, we present a method to deterministically estimate the amount of directional nodes for a given coverage rate. We also employ Sensing Connected Sub-graph (SCSG) to divide a directional sensor network into several parts in a distributed manner, in order to decrease time complexity. Moreover, the concept of convex hull is introduced to model each sensing connected sub-graph. According to the characteristic of adjustable sensing directions of directional nodes, we study a coverage-enhancing algorithm to minimize the overlapping sensing area of directional sensors only with local topology information. Extensive simulation is conducted to verify the effectiveness of our solution and we give detailed discussions on the effects of different system parameters.

1 Introduction

Recently sensor networks have attracted tremendous research interests due to its vast potential applications [1, 2]. Conventional sensor networks often assume the omni-directional sensing model [3]. Actually, directional sensing range and sensors also have great application chances, typically in video sensor networks [2,4]. Potential applications of video sensor networks span a wide spectrum from commercial to law enforcement, from civil to military [2, 5-7]. However, many methods for conventional sensor networks is not suitable for directional sensor networks. Thus, the directional sensor network also demands novel solutions, especially for coverage control scheme.

In our best knowledge, a few papers have indeed studied the concept for video sensor networks. However, the work has mainly focused on data fusion [2, 4], typical

* This work reported in this paper is supported by the National Natural Science Foundation of China under Grant No.90612013, Beijing Natural Science Found under Grant No. 4062024, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No.20050013010 and the NCET of MOE, China.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 256 – 267, 2006.

applications [5], low power hardware platform support [6] and network architecture [7]. Authors of paper [8] have first proposed the concept of directional sensor network, and have studied on deployment policy and connectivity maintenance. However, some fundamental issues, such as coverage control aiming at directional sensing feature, have been left unaddressed. The feature directly affects the deployment of sensors and calls for novel coverage control schemes for sensor networks.

Coverage is a key issue of sensor network, and in most cases, "coverage" means area coverage [3]. Area coverage has been defined as the achievement of a static arrangement of nodes that maximize the detection rate of targets appearing in a given area. It's essential to maintain network coverage over the entire area as long as the network is alive, because a small unmonitored sub-area defeats the whole purpose of the network (for instance, an intrusion in that area may go undetected). This requires nodes to be spread as uniformly as possible over the entire sensing region with the minimum gaps (or uncover areas) [9], typically in video sensor networks with directional sensing feature.

In this paper, we mainly address the problem of enhancing area coverage in a directional sensor network. First, based on a rotatable directional sensing model, we design an exact estimate of the amount of directional sensors for a given coverage probability in a static directional sensor network. Second, we propose the concept of sensing connected sub-graph (SCSG) to divide a directional sensor network into several components in a distributed manner. Furthermore, we model each sensing connected sub-graph as a multi-layered convex hull set to address the problem of enhancing coverage by adjusting the sensing direction of directional sensor, thus minimize the overlapping sensing area of directional sensors.

The reminder of this paper is organized as follows. Related work is discussed in Section 2. In Section 3, we propose a novel rotatable directional sensing model and provide a method for estimating directional sensors amount for a given ratio of coverage. Section 4 details our coverage enhancing algorithm. A set of experimental results are presented in Section 5 and we conclude this paper in Section 6.

2 Related Work

While assumption of omni-directional sensing model has facilitated elegant properties of traditional sensor networks [8], video sensor is characterized by its directional and adjustable sensing region. These characteristics introduce fundamentally different properties in terms of network coverage targeted by this paper. Our focus is the coverage-enhancing problem for randomly deployed directional nodes.

In general, coverage can be achieved by designing density control mechanisms. That is, scheduling the sensors to work alternatively to minimize the waste of sensing resource due to the overlap of sensing area of sensor nodes [10-12]. However, as far as video sensor network is concerned, the cost of video node is much higher than that of traditional nodes. We have to balance the two important performance parameters *cost* and *coverage*. Hence, it is not practical for redundant video nodes to work alternatively while maintaining ideal area coverage. Traditional coverage-enhancing schemes are mainly classified into two categories [13]: one is redeploying additional nodes to ensure adequate area coverage in sensor networks [14]. The other is deploying partial nodes with autonomous position mobility to increase coverage dynamically [9, 15-17]. Although we can improve the area coverage through the mobility of nodes, the mobility occurs only over short distances. Furthermore, mobility easily causes the death of node, and alters the network topology.

The common assumptions of previous works are that sensor is omni-directional sensing. In [8], Ma et al. first employ a directional sensing model directly rooted from the concept of field of view in cameras and propose a systematic method for coverage of randomly deployed directional sensor networks. They are only concerned with the directional sensing feature of video nodes while overlooked another important feature of video nodes: the rotatable ability of sensing direction. Moreover, their algorithm focused on scheduling redundant directional nodes to work alternatively along with traditional coverage control methods. In this paper, we study the algorithm of improving area coverage by rotating sensing directions of directional nodes. To our best knowledge, this is the first time that a coverage-enhancing algorithm aiming at directional sensing model is adopted in sensor networks.

3 Rotatable Directional Sensing Model and Deployment Scheme

3.1 Rotatable Directional Sensing Model

Different from the conventional sensing models where the omni-sensing area centers on the sensor node, we propose a rotatable directional sensing model. Compared with the directional sensing model proposed in [8], this rotatable directional sensing model focuses on two distinct features of a video node. On one hand, the video nodes are constrained by the field of view. This means the sensing region of each video node is not omni-directional. On the other hand, the video nodes have the ability of rotating their sensing directions. That is, at a discrete time *t*, the sensing model of a video node can be regarded as a directional one. However, in a continuous period *T*, a video node has omni-directional sensing ability. Therefore, here we improve the 2-dimension sensing model as follows:

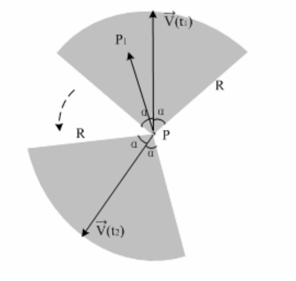


Fig. 1. Rotatable directional sensing model

Definition 1. A directional node *n* is a sector denoted by 4-tuple <*P*, *R*, $\vec{V}(t)$,*α*>, where *P* is the location of the directional node, *R* is the sensing radius. $\vec{V}(t)$ is the center line of sight of the camera's field of view, termed as sensing direction, which is effected by time *t*. And *α* is the offset angle of the field of view on both sides of $\vec{V}(t)$. Without otherwise specified, $\vec{V}(t) = (\vec{V_x}(t), \vec{V_y}(t))$ is of unit length, where $\vec{V_x}(t)$ and $\vec{V_y}(t)$ are the components along x-Axis and y-Axis, respectively.

Fig.1 illustrates the rotatable directional sensing model. Note that the conventional omni-sensing model is a special case of this new model when *α* is π.

The rotatable directional sensing model is a Boolean model. At time *t*, a point P_1 is said to be covered by the node *n* if and only if the following conditions are met:

1) $d(P,P_1) \leq R$, where $d(P,P_1)$ is the Euclidean distance between point *P* and P_1.
2) The angle between $\overrightarrow{PP_1}$ and $\vec{V}(t)$ is within [-*α*, *α*].

A simple method to judge if point P_1 is covered by directional node *n* is as follows: at time *t*, if $d(P,P_1) \leq R$ and $\overrightarrow{PP_1} \cdot \vec{V}(t) \geq \| \overrightarrow{PP_1} \| \, cos\alpha$ then P_1 is covered and otherwise not. An area *A* is covered by the node *n*, if and only if any point $P_1 \in A$, P_1 is covered by the node *n*.

3.2 Deployment Scheme

There are two kinds of deployment schemes in sensor network: deterministic deployment and random deployment. Deterministic deployment is often adopted for a specific purpose, such as security monitoring in a museum. In many situations (battlefield, forest, high-risk area), deterministic deployment is neither feasible nor practical. Random sensor deployment (such as throwing sensors from an airplane) is always believed to be one of the major advantages of wireless sensor networks over traditional wired sensor networks. In such randomly deployed networks, it is hard or impossible to 100% guarantee complete coverage of the monitored region even if the node density is very high. So, it is a practical issue to study how to guarantee the given coverage rate of sensor network for a targeted region. Assume that the area of the targeted region is *S*, and there are no two sensors located at exactly same position and sensing region. If the sensors are randomly deployed in the targeted region, and the locations of sensors obey uniform distribution. Because the directional sensor with the offset angle *α* covers the sensing area αR^2, the probability of covering the targeted region by each sensor is $\alpha R^2/S$. Therefore, after *N* directional sensors are deployed, the probability of covering the targeted region is represented in Equation (1) [8]:

$$p = 1 - (1 - \frac{aR^2}{S})^N \tag{1}$$

Thus, if the coverage rate of the targeted region is at least *p*, the number of deployed directional sensors should be as follows:

$$N \geq \frac{\ln(1-p)}{\ln(S-\alpha R^2) - \ln S} \tag{2}$$

4 Coverage-Enhancing Algorithm

In this section, we describe a coverage-enhancing algorithm to minimize overlapping sensing area among multiple neighboring directional nodes, thus to obtain the maximal area coverage with the given number of directional nodes. Our coverage-enhancing algorithm follows a three-step: 1) partitioning sensing connected sub-graphs (SCSG); 2) forming a multi-layered convex hull set in each SCSG; 3) rotating sensing directions of directional nodes.

4.1 Partitioning Sensing Connected Sub-graph

First, we try to partition sensing connected sub-graphs in directional sensor networks. Assume that a directional sensor network can be modeled as a sensing graph $G(V,E)$, where V is a set of nodes in the network, and E is the set of edges. For a pair of node $n_1,n_2 \in V$, the edge $(n_1,n_2) \in E$ if $d(n_1,n_2) \le 2R_s$. We define the maximal sensing connected sub-graph as follows:

Definition 2. Given a sensing connected sub-graph $G_1(V_1,E_1) \in G$, if for any node $n_i \in V-V_1$, no matter how the node n_i rotates, the sensing area of n_i and that of any node $n_j \in V_1$ has no overlapping area, then we call G_1 as the maximal sensing connected sub-graph[1].

Property 1. There are N_s sensing connected sub-graphs G_i $(1 \leqslant i \leqslant N_s)$ in a directional sensor network, if and only if the following conditions are met: $V(G_i) \cap V(G_j) = \Phi$ $(i \neq j$ *and* $1 \leqslant i,j \leqslant N_s)$ and $\sum_{i=1}^{N_s} V(G_i) = V$.

Partitioning a directional sensor network into several SCSGs is dividing and conquering a centralized issue into a distributed one, thus decreasing the time complexity. Moreover, in this way, we can easily detect coverage holes in the targeted region. Then, we can deploy additional directional nodes to reduce or eliminate coverage holes. This part of work is beyond the topic of the paper.

We take *FindSCSG* algorithm to search for all the sensing connected sub-graphs in a directional sensor network. Assume that network graph G be represented as the adjacency list. The algorithm takes use of depth-first search algorithm.

```
Algorithm FindSCSG (n)
/* n denotes nodes in a directional sensor network.*/
//use a visit flag array Visited
  visited[n] = TRUE; // the node n is visited
  n = *n.first; // take the first adjacent vertex
  While (n is not NULL) do //if there is adjacent vertex
  begin
    if (!visited[*n.vertex])//if the node is not visited
      FindSCSG (*n.vertex);
    n = *n.next;//take the next vertex adjacent to n
  end.
```

[1] Without otherwise specified, sensing connected sub-graph mentioned in this paper means the maximal sensing connected sub-graph, which is abbreviated as SCSG.

Definition 3. If the number of SCSGs N_s equals to 1, G is a complete sensing connected graph; otherwise, G is a partial sensing connected graph.

When the total number of deployed directional sensors is fixed, to some extent, the number N_s reflects the performance of area coverage in a directional sensor network. The greater N_s is, the worse the coverage rate becomes. In other words, the number of coverage holes will increase with the increasing of N_s for the directional sensor network. We take the sensing graph G as an example. In Fig.2(b), G contains 5 sensing connected sub-graphs. According to the Definition 3, G is a partial sensing connected graph.

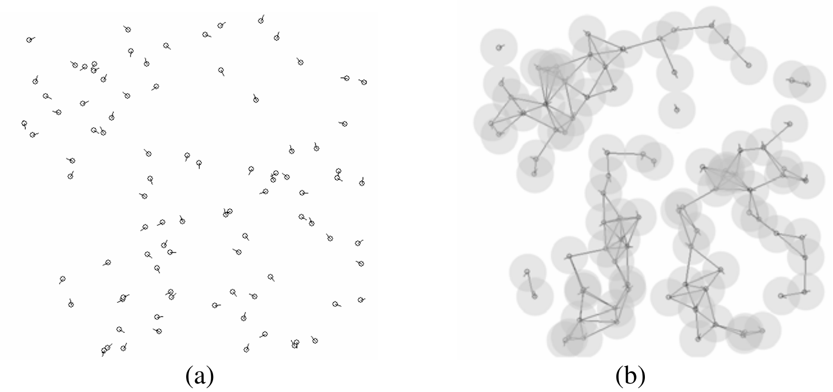


Fig. 2. Sensing connected sub-graph (SCSG)

4.2 Forming a Multi-layer Convex Hull Set

Once partitioning a directional sensor network into sensing connected sub-graphs, we decomposed a large task into several subtasks. Then, we can model each SCSG as a multi-layer convex hull set to optimize the process of increasing area coverage and decrease the time complexity especially.

Definition 4. Given a node set $V = \{n_{1,} n_2, \ldots, n_m\}$, the convex hull is the smallest convex polygon that contains all of the nodes of V.

Property 2. In a convex hull, each polygon vertex (node) is convex and its interior angle (2β) is less than π.

There are two reasons for using convex hull to solve the coverage enhancing problem. First, the convex hull is the most ubiquitous structure in computational geometry. We can always find a convex hull to contain all the nodes in a plane. Second, multi-layer convex hulls can efficiently partition a SCSG into multiple annular sub-areas to achieve extended coverage. Here, we utilize Graham algorithm [18], which is an optimal algorithm to generate convex hulls layer by layer.

In general, the number of convex hulls in a SCSG is more than 1. Therefore, we design a algorithm to form a multi-layer convex hull set, and this algorithm is described as follows.

```
Algorithm MultiLayer-Convex-Hull(V,num)
/* V denotes a set of points and num denotes the node number of
node set V.*/
k = 1, V' = V, num' = num
//k denotes the number of convex hulls in a SCSG.
begin
  While (V' is not NULL) Do
    Call Graham (V', num') to calculate the convex hull of node
    set V'. The vertex set of convex hull is V(k)={$v_1^k$, $v_2^k$ ,…, $v_{m_k}^k$ }
    and the edge set of convex hull is E(k)={$e_1^k$, $e_2^k$,…, $e_{m_k}^k$ }.
    V' = V'-V(k);
    n' = n'- $m_k$;
    k = k + 1;
end.
```

A case is also illustrated in Fig.3. We can observe that the multi-layer convex hull set contains four convex hulls from outer to inner. The four convex hulls divide the SCSG into the four annular sub-areas. In the next step, we will rotate the sensing directions to improve coverage quality of the directional sensor network.

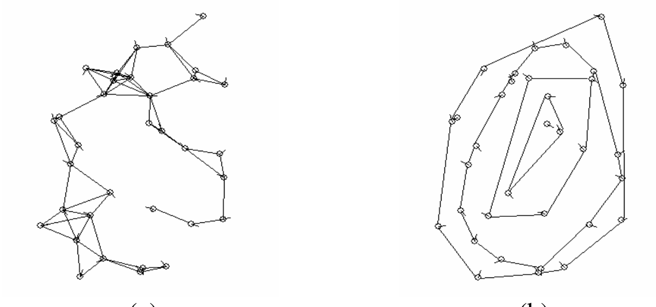

(a) (b)

Fig. 3. A multi-layer convex hull set

4.3 Rotating Sensing Directions of Directional Nodes

Once we form a multi-layer convex hull set in each SCSG accordingly, the sensing directions of directional nodes will be rotated to obtain the maximal sensing coverage region.

From the Fig.3 (b), we see that each directional node only belongs to one convex hull as a vertex. According to Definition 4, when we connect any two adjacent vertexes v_i^j and v_{i+1}^j (where v_i^j denotes the i^{th} vertex on the j^{th} layer convex hull), all the vertexes of this convex hull lie in one side of line $\overline{v_i^j v_{i+1}^j}$. Based on this property, we rotate the directional node v_i^j to make its sensing direction $\overrightarrow{V_i^j}(t)$ and the interior angle-bisector on the same line, as illustrated in Fig.4. In this way, we can achieve the less overlapping area between neighboring two directional nodes on the same convex hull. The rotation process can be executed on individual directional node with the knowledge of local neighboring topology.

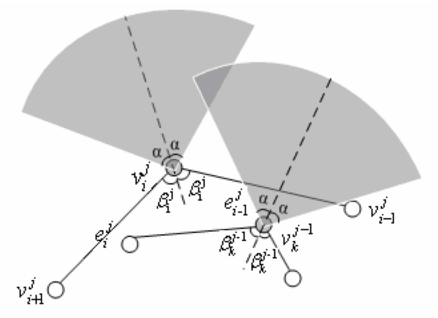


Fig. 4. Rotating the sensing directions of directional nodes

Take the vertex v_i^j as an example, v_i^j only need know the topology information of its adjacent two vertexes v_{i-1}^j and v_{i+1}^j. According to the edge information $e_{i\text{-}1}^j$ and e_i^j in the edge set E, v_i^j calculates its interior angle $2\beta_{\mathrm{i}}^{\mathrm{j}}$, thus rotates the sensing direction $\overrightarrow{V_i^j}(t)$ to the inverse direction of interior angle-bisector. It is clear that our method of rotating the sensing directions of directional nodes is simple and feasible.

4.4 Complexity

Our coverage enhancing algorithm mainly includes three steps. First, given n directional nodes, the depth-first search algorithm for finding sensing connected subgraphs takes $O(n^2)$ time. Then, a multi-layer convex hull set for each SCSG can be constructed in $O(km*logm)$ time using Graham algorithm, where k is the number of convex hull in a convex hull set and m is the number of node in a SCSG. In the third step, each directional node rotates its sensing direction by calculating corresponding interior angle-bisector in $O(n)$ time. Therefore, the total time of our coverage enhancing algorithm is as follows: $O(n^2)+O(km*logm)+O(n)= O(n^2)$.

5 Experimental Results and Performance Analysis

We studied the coverage rate of the target region of 500*500m^2 in our simulation. The number of randomly deployed directional sensors is varied from 0 to 200. The offset angel of directional sensor is varied from 0° to 90° (π/2), and the sensing radius is changed from 0m to 60m. The experiments have been executed in Senetest2.0 simulation platform we developed[2].

5.1 Case Study

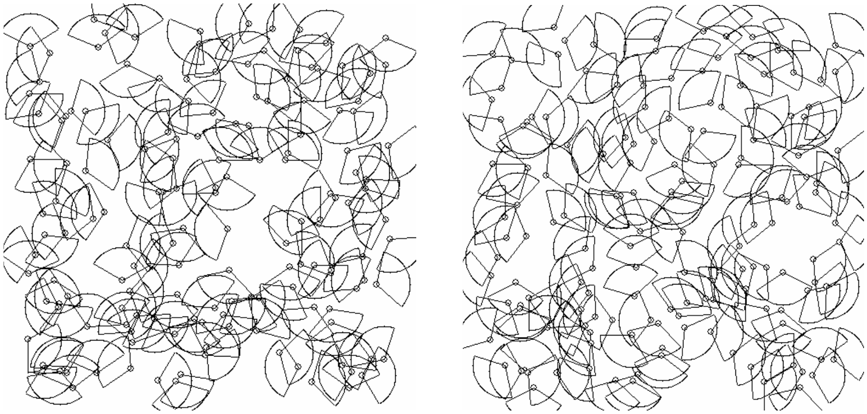
From the above analysis, it is clear that our coverage enhancing algorithm can reduce the coverage redundancy by re-adjusting the sensing directions of directional nodes to minimize the overlapping sensing area.

[2] Senetest2.0 is a simulation tool developed by C++, which is used to simulate the topology of sensor network, sensing completeness and communicating connectivity.

Here, we use a case to illustrate the effectiveness of our algorithm. In a 500*500m^2 field, we want to deploy the directional nodes with the sensing radius 50m and the offset angel 60º (π/3) for gathering the visual information. If the required coverage rate is at least 80%, we can calculate the number of node to be deployed as follows:

$$N = \frac{\ln(1\text{-}0.80)}{\ln(250000\text{-}0.33\pi * 50 * 50)\text{-}\ln(250000)} = 152$$

With re-adjusting sensing directions of nodes by our algorithm, the adjusted coverage rate *p'* is greater than the initial coverage rate *p*. Fig.5 illustrates the network coverage before rotation and after rotation for the case, where *p* in Fig.5(a) is 80.37% and *p'* in Fig.5(b) is 85.03%. It is obvious that the detected region can be covered more evenly by directional nodes in Fig.5 (b). To achieve the coverage rate 85.03% with the traditional method, we calculate more than 180 directional nodes to be deployed by Equation (2). According to our algorithm, we can save 28 directional nodes in the node deployment phase.



(a)Before rotation (b)After rotation

Fig. 5. The network coverage before rotation and after rotation

Here, we verify our theoretical analysis for the relationships among the initial coverage rate *p*, the coverage rate difference Δ*p* and the node number difference Δ*N*. From Equation (2), we can evaluate the number of node we can save (Δ*N*) from coverage rate *p* to *p*+Δ*p* as follows:

$$\vartriangle N = \frac{\ln(1-(p+\vartriangle p))-\ln(1-p)}{\ln(S-\alpha R_S^2)-\ln S} \tag{3}$$

Fig.6 shows us the relationship among *p*, Δ*p* and Δ*N* deduced from Equation (3). With the increase of *p* and Δ*p*, the value of Δ*N* will increase obviously. In our simulation experiment, when *p* approaches 80% and Δ*p* approaches 3%, the value of Δ*N* approaches 100. After comparison, we can conclude that the simulation results are a nice match for our theoretical results.

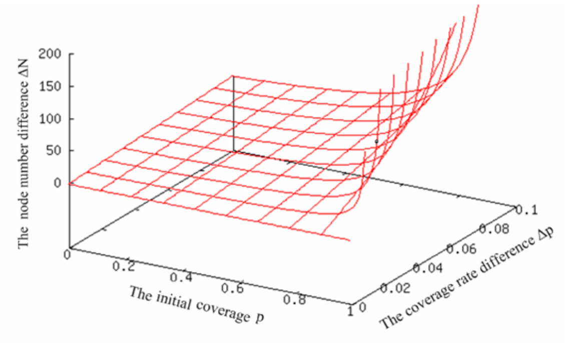


Fig. 6. The relationship among *p*, Δ*p* and Δ*N*

5.2 Simulations

In this section, through a set of simulation experimental results, we discuss how the parameters (the sensing radius *R*, the offset angle *α* and the number of directional node *N*) influence on the performance of coverage enhancing algorithm.

5.2.1 The Effect of Sensing Radius (*R*)

First, we examine the effect that sensing radius *R* makes to the improvement of coverage rate *p*. Let Δ*p*=*p*-*p'*. For each directional node's offset angle *α*=60°, we run the simulation on several different network densities, 50-node, 100-node, 150-node.

Seen from the Fig.7, we can get that *p* and *p'* make increases with an increase in the size of network. Clearly, some coverage performance improvement is achieved in a network by rotating sensing directions of directional nodes. The value of Δ*p* will increase with the increasing of *R*. For instance, in a 50-node network, *p'* can achieve about 6.21% more network coverage than *p* when *R*=40m, while 2.61% improvement is achieved when *R*=20m. However, once the value of *R* exceeds a threshold (≥ 45m in this experiment), Δ*p* turns to be inverse proportional to *R*. This is because, when the network density is fixed, the greater the sensing radius of a node is, the greater the possibility of neighboring nodes to form overlapping sensing area becomes. Without doubt, the overlapping sensing area will weaken the coverage improvement. In addition, some directional nodes in boundary area will cause area coverage loss.

5.2.2 The Effect of Offset Angle (*α*)

We evaluate the effect that the offset angle α improves the coverage rate *p*. Under the case *R*=40m, we run the simulation on three different network densities (50-node, 100-node, 150-node).

From the Fig.8, we can see that the value of *p* and *p'* are proportional to the scale of network (that is, network density), respectively. The value of Δ*p* will increase with the increasing of *α*. When α=60°, our coverage enhancing algorithm offers better coverage improvement than the case of α=30° in a 100-node network. However, once *α* exceeds a threshold (≥60° in this experiment), Δ*p* will decrease with the increasing of *α*. The intuitive reason for this result is similar to the relationship between Δ*p* and *R* described in section 5.2.1.

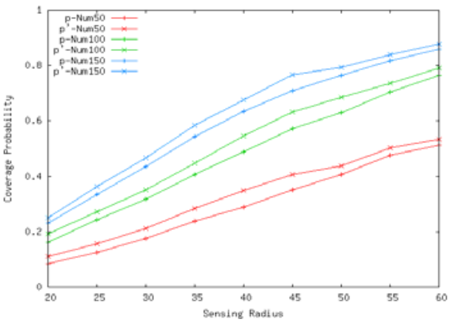


Fig. 7. The effect of the sensing radius *R*

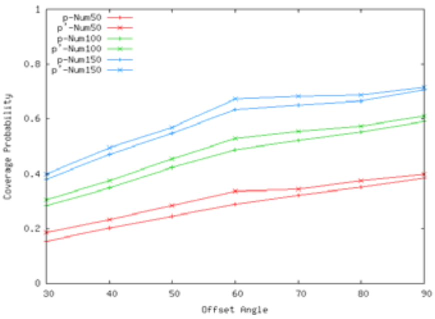


Fig. 8. The effect of the offset angle *α*

In the Fig.8, we also find that the greater *N* is, the less Δ*p* becomes. When the detected area is fixed, the increase of *N* causes the increase of network density, thus the probability of forming overlapping sensing area among directional nodes becomes greater. In this case, the directional nodes on the two neighboring convex hulls will cause much overlapping sensing area and weaken the performance of coverage improvement.

6 Conclusion

Different sensing model of directional sensor networks demands for efficient method for the node deployment and the coverage improvement. Motivated by this, this paper proposes a method for enhancing coverage rate of sensor networks based on a novel rotatable directional sensing model. By quantifying the requirements of deploying directional sensors for a given coverage rate, we can optimize the scale of node deployment. According to the characteristic of rotatable sensing directions of directional nodes, we propose a coverage enhancing algorithm to maximize network area coverage. First, we propose the concept of sensing connected sub-graph (SCSG) to partition a network into several parts in a distributed manner. Second, convex hull is introduced to model each SCSG. Our algorithm is proved to be effective through our experimental study. Furthermore, we analyze the effects of key parameters of directional sensors in determining the performance of coverage improvement.

References

1. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless Sensor Networks: a Survey, Computer Networks, vol.38, pp.393 - 422, 2002.
2. Huadong Ma, Yonghe Liu. Correlation Based Video Processing in Video Sensor Networks, IEEE WirelessCom2005, Jun.13-15,2005, Hawaii.
3. Ren Yan, Zhang Hang Si-Dong, et al. Theories and Algorithms of Coverage Control for Wireless Sensor Networks, Journal of Software, Vol.17,No.3, pp.422 - 433,March 2006.

4. Dan Tao, Huadong Ma and Yonghe Liu. Energy-efficient Cooperative Image Processing in Video Sensor Networks, 2005 IEEE Pacific-Rim Conference on Multimedia, LNCS, Vol. 3768, pp.572 - 583, Nov. 13-16, 2005. Korea.
5. Holman, R.; Stanley, J.; Ozkan-Haller, T. Applying Video Sensor Networks to Nearshore Environment Monitoring. IEEE Transactions on Pervasive Computing, Vol.2, Issue 4, pp.14 - 21, Oct.-Dec. 2003.
6. W. Feng, B. Code, E. Kaiser, M. Shea, and W. Feng, Panoptes: Scalable Low-power Video Sensor Networking Technologies, the 11th ACM International Conference on Multimedia, Berkeley, CA, Nov. 2003.
7. Purushottam Kulkarni, Deepak Ganesan, Prashant Shenoy and Qifeng Lu. SensEye: A Multitier Camera Sensor Network. ACM MM’05, Nov. 6-11, 2005, Singapore.
8. Huadong Ma, Yonghe Liu. On Coverage Problems of Directional Sensor Networks, International Conference on Mobile Ad-hoc and Sensor Networks. LNCS Vol. 3794, pp. 721 - 731, Dec 13-15, 2005, Wuhan.
9. Archana Sekhar, BS Manoj, C. Siva Ram Murthy, Dynamic Coverage Maintenance Algorithms for Sensor Networks with Limited Mobility, the Third IEEE International Conference on Pervasive Computing and Communications. pp.51 - 60, 2005.
10. JunLu, Tatsuya Suda, Coverage-aware Self-scheduling in Sensor Networks, 2003 IEEE 18th Annual Workshop on Computer Communications, pp.117-123, 20-21 Oct. 2003.
11. Benyuan Liu, Don Towsley. A Study of the Coverage of Large-scale Sensor Networks, the Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pp.475 - 483, 2004.
12. D.Tian and N.D. Georganas, A Coverage-preserving Node Scheduling Scheme for Large Wireless Sensor Networks, the First ACM International Workshop on Wireless Sensor Networks and Applications, pp.32 - 41, 2002.
13. Megerian S, Koushanfar F, Potkonjak M, Srivastava MB. Worst and Best-case Coverage in Sensor Networks. IEEE Trans. on Mobile Computing, 2005, 4(1):84 - 92.
14. A. Ghosh, Estimating Coverage Holes and Enhancing Coverage in Mixed Sensor Networks, the 29th Annual. IEEE International Conference on Local Computer Networks. pp. 68 - 76, 16-18 Nov. 2004.
15. A. Howard, M. Mataric, and G. Sukhatme. Mobile Sensor Network Deployment using Potential Fields: A distributed scalable solution to the area coverage problem. the 6th International Conference on Distributed Autonomous Robotic Systems, pp. 299 - 308, Fukuoka, Japan, June 2002.
16. S. Poduri and G. S. Sukhatme. Constrained Coverage in Mobile Sensor Networks. IEEE International Conference on Robotics and Automation, pp.40 - 50, New Orleans, LA, USA, Apr./May 2004.
17. G. Kesidis, T. Konstantopoulos, and S. Phoha, Surveillance Coverage of Sensor Networks under a Random Mobility Strategy, IEEE Sensors, October, 2003.
18. Graham R.L., An Efficient Algorithm for Determining the Convex Hull of a Planar Set, Info. Proc. Lett., pp.132 - 133, 1972.

Sensor Scheduling for *k*-Coverage in Wireless Sensor Networks

Shan Gao, Chinh T. Vu, and Yingshu Li

Department of Computer Science
Georgia State University
Atlanta, GA 30303, USA
{sgao, chinhvtr, yli}@cs.gsu.edu

Abstract. Some sensor network applications require k-coverage to ensure the quality of surveillance. Meanwhile, energy is another primary concern for sensor networks. In this paper, we investigate the Sensor Scheduling for k-Coverage (SSC) problem which requires to efficiently schedule the sensors, such that the monitored region can be k-covered throughout the whole network lifetime with the purpose of maximizing network lifetime. The SSC problem is NP-hard and we propose a heuristic algorithm for it. In addition, we develop a guideline for users to better design a sensor deployment plan to save energy by employing density control. Simulation results are presented to evaluate our proposed algorithm.

1 Introduction

Sensor networks which usually consist of a large number of sensors are attracting people's attentions. They can sense and collect information from all kinds of objects in the monitored area. Furthermore, they can process the gathered information and send it back to users. Therefore, they are being widely employed for military fields, national security, environmental monitoring, traffic control, health, industry, disaster prevention and recovery [1]. However, current sensor networks still have some limitations that prevent them from better serving the people. The limitations are as following: limited power at each sensor, limited communication ability, limited computation ability, limited wireless bandwidth, large number of nodes in a network, huge deployment area and infinite sensing data streams. These limitations bring a lot of challenging problems. In this paper, we address the k-coverage problem which requires that every point of the whole monitored area can be covered by at least k sensors at any time.

To deploy a sensor network, an aircraft may be used to spread the sensors into an area when ground access is not possible. This causes the lack of accurate placement of sensors, which will be compensated by deploying more redundant sensors. Another reason for deploying redundant sensors is to provide fault-tolerance, since sensors are prone to failures [1]. If any point in the monitored area is monitored by at least k sensors, proper operation of the network can still be ensured, even if some sensors fail. The required coverage level k may

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 268–280, 2006.

be different for different applications. In friendly environment such as home monitoring, k can be set to a small value, while in hostile environment such as battle fields, k should be set to a large value. Even for a single sensor network, k may be different. For example, for forest fire detection, k may be low in the rainy season, but high in the dry season. One may say that since much redundant sensors are deployed, the k-coverage problem can be easily solved. However, considering the power limitation of sensor networks, to make all sensors remain active greatly shortens the network lifetime. It is shown in [2] that each sensor spends $0.34W$ to $0.7W$ power when it is in the transmit, receive and idle states, however, only $0.03W$ in the sleep state. In addition, the lifetime of a battery discharging in short bursts with significant off-time is approximately twice as much as that in a continuous operation mode [3]. These facts indicate that a good active/sleep scheduling mechanism can dramatically extend network lifetime. Therefore, while maintaining the coverage level k, only a subset of the sensors is needed to be active at any time.

In this paper, our contributions are as following: i) we define the problem of Sensor Scheduling for k-Coverage (SSC) which is NP-hard; ii) we design a heuristic algorithm for the SSC problem which divides the sensors into subsets, such that a schedule can be worked out by activate these subsets successively to extend network lifetime; and iii) we propose a density control scheme for sensor deployment to reduce the number of unallocated sensors such that the network efficiency is improved.

2 Related Work

Recently, the coverage problem, a fundamental problem in sensor networks about how well an area is monitored by sensors, has attracted people's attentions. Basically, there are three kinds of coverage problems [6] which are target coverage problem, area coverage problem and breach coverage problem. The work in [7] $\sim$ [8] addressed the target coverage problem where the purpose is to cover all the targets. The work in [4], [5], [9] $\sim$ [11] addressed the area coverage problem where the purpose is to cover the whole monitored area. The breach coverage problem is addressed in [12] where the purpose is to minimize the number of uncovered targets. Some other work, [14] and [15], tried to find a path which is best or worst monitored by sensors and connects two given points inside or outside the surveillance area and this path can indicate the sensing ability of the sensor network in the best or the worst situation.

None of the above work considers the k-coverage requirement for the purpose of quality of surveillance. To the best of our knowledge, not much work address the k-coverage problem. Wang et al. [9] first studies this problem. The coverage levels of all the intersection points are determined through verifying the coverage degrees of the area. They proposed a localized heuristic for constructing a cover set (subset of sensors) that can provide k-coverage. However, the size of the obtained subset cannot be guaranteed to be small. In [5], the authors designed a greedy heuristic for the k-coverage problem and the size of their constructed cover set is within $O(logn)$ factor of the optimal. The main idea is to select a candidate path which has the maximum K-Benefit value. Both of these two work only consider constructing one cover set instead of dividing sensors into subsets such that each subset can provide k-coverage. In [13], the authors study the sensor deployment problem so that k-coverage can be guaranteed.

In [4], the coverage problem is formulated as a decision problem, whose goal is to determine whether each point in the monitored region is covered by at least k sensors. The main idea is to check the perimeter coverage level of each sensor. They prove that the whole monitored region is k-covered if and only if each sensor in the monitored region is k-perimeter-covered. Based on this work, we design in this paper a heuristic algorithm to divide the sensors into subsets and each of the subset can provide k-coverage, such that the network lifetime can be maximized.

The differences between these algorithms and ours are: 1) our algorithms provide solutions to k-cover the monitored area; 2) in our algorithms, k-coverage is 100% guaranteed; 3) there is no limitation on sensor's sensing range which could vary in a range instead of several fixed values; 4) our algorithms have no limitation on the number of sensors and the sensor positions.

3 Sensor Scheduling for *k*-Coverage

We consider a sensor network which monitors a two dimensional region and no two sensors are located at the same location. Every point in the region needs to be continuously monitored (covered) by at least k sensors. The *network lifetime* is defined as the total duration during which the whole region is k-covered. We assume the number of the deployed sensors is more than the required number of sensors that can provide k-coverage for the monitored region. To extend the network lifetime, instead of making all the sensors to be active throughout the whole network lifetime, a subset of the sensors can be turned on to provide k-coverage at any time, while the rest sensors are in sleep mode. We also assume the transmission range of a sensor is at least twice the sensing range of a sensor so that connectivity is also guaranteed within each subset [9]. All the sensors have uniform transmission range and sensing range. Then the problem of sensor scheduling for k-coverage can be defined as following.

Definition 1 ***Sensor Scheduling for k-Coverage (SSC)*****:** *Given a sensor network with n sensors that can provide k-coverage for the monitored region, schedule the activities of the sensors such that at any time, the whole region can be k-covered and the network lifetime is maximized.*

In [5], the authors consider the problem of constructing a single connected k-coverage set and this problem (CCP) is proved to be NP-hard. Therefore, the SSC problem is NP-hard since CCP is a special case of the SSC problem when the number of the constructed k-coverage sets is one. The scheduling decisions can be made at the Base Station (BS). The BS broadcasts the schedule to all the sensors so that each sensor can know when it should be active to monitor the region. To solve the SSC problem, we can divide the sensors into disjoint subsets. Each subset can k-cover the whole region, where k-cover indicates for every point in the monitored region, at least k sensors can cover this point. These subsets can be scheduled to be active successively. For each subset, its lifetime is decided by the sensor which has the least power. Therefore, the lifetime of the entire network highly depends on the number of subsets.

The following notations are used to formulate the SSC problem and to describe our algorithm.

- K: If all the sensors are active, any point in the monitored region can be covered by at least K sensors.
- k: k ($k \leq K$) is a user-specified parameter which specifies the required coverage level the sensor network must provide at any time.
- S: The set of all the sensors.
- m: All the sensors can be divided into at most m subsets and each subset can k-cover the monitored region.
- C_i: The ith subset, $1 \leq i \leq m$.
- cov_i: The coverage level of the C_i, which means any point in the monitored region is covered by at least cov_i sensors which belong to C_i.

Our goal is to construct as many subsets as possible such that i) each subset can k-cover the whole monitored region; ii) the network lifetime is maximized. Then the SSC problem is formulated as

$$\begin{aligned} &\text{Objective: } \text{Max m} \\ &\text{Subject to: } \bigcup_{1\leq i\leq m} C_i \subseteq S \\ &\qquad C_i \cap C_j = \emptyset, 1 \leq i, j \leq m, i \neq j \\ &\qquad cov_i \geq k, 1 \leq i \leq m \end{aligned}$$

4 Disjoint Cover Sets with Fixed Sensing Range

In this section, we present a greedy heuristic for the SSC problem. In [4], the authors proved that the entire monitored region is k-covered if and only if each sensor in the monitored region is k-perimeter-covered. k-perimeter-cover requires that any point on the perimeter of a sensor i be covered by at least k sensors other than sensor i. Based on this fact, we propose a greedy algorithm, *PCL-Greedy-Selection* (GS). We define the *Perimeter Coverage Level* (*PCL*) of a sensor a as the number of the sensors in the same set that cover any point on a's perimeter of the sensing area. The lower the PCL is, the smaller the node density (the number of nodes per unit area) is.

The main idea of GS is to iteratively construct subsets C_i by choosing sensors from the area with the lowest sensor density. When construct an individual C_i, the sensor with a smaller PCL value will be added to C_i at each step. In this way, we can include as less sensors as possible in C_i and these sensors are distributed in the area as widely as possible, such that more sensors can be left to join

Algorithm 1. PCL-Greedy-Selection(k, S)

```
Sort $S$ in non-decreasing order based on their $PCL$ values
while $S$ is not empty do
    $cov_i \leftarrow getCoverageLevel(C_i)$
    if $cov_i < k$ then
        $node \leftarrow$ the first sensor in $S$
        Add $node$ to $C_i$
        Remove $node$ from $S$
    else
        PruneGreedySelection($k, S, C_i$)
        Add $C_i$ to $C$
        $i++$
    end if
end while
output $C$
```

other subsequent subsets and the overlapped sensing regions in each subset are reduced as much as possible. This also indicates when construct a subset C_i, the area with smaller node density is taken care of with higher priority.

The GS is shown in Algorithm 1. The input includes k, a user-specified coverage level, and S, the set of all the sensors. The output is a collection of subsets C, and each subset can k-cover the whole monitored region. To justify if a subset C_i can k-cover the entire monitored region, we can use the method proposed in [4] and we call it $getCoverageLevel(C_i)$. Firstly, all the sensors in S are sorted in non-decreasing order based on their PCL values. Then sensors are added to a subset in a greedy manner. If at some iteration, the current subset C_i can provide k-coverage, a new subset C_{i+1} will be constructed in the same manner. GS stops when we can no longer construct a subset that can k-cover the whole monitored region.

Since each subset is constructed in a greedy manner, it is possible that there exist some redundant sensors in a subset. Therefore, after constructing a subset, we need to remove those redundant sensors and add them back to S so that they are still available to be added to the subsequent subsets. The algorithm to conduct this operation is $PruneGreedySelection$ which is described in Algorithm 2. In this algorithm, given a subset C_i, we check for each sensor in C_i to see if the removal of it will make cov_i smaller than k. If a sensor is redundant (after the removal of this sensor, cov_i is still no less than k), it will be added back to S.

In [4], the authors have shown the fact that if no two sensors are located at the same location, the whole monitored region is k-covered if and only if each sensor is k-perimeter-covered. Based on this fact, the correctness of our algorithm is guaranteed.

Theorem 1. *The time complexity of GS is $O(n^2 dlog(d))$. Here, n is the number of the sensors, and $d = max(d_1, ..., d_i, ..., d_n)$ where d_i is the number of neighbors of $sensor_i$.*

Algorithm 2. PruneGreedySelection(k, S, C_i)

```
for $j = 1$ to $|C_i|$ do
    $s_j \leftarrow$ the $j$th sensor in $C_i$
    Remove $s_j$ from $C_i$
    $cov_i \leftarrow getCoverageLevel(C_i)$
    if $cov_i \geq k$ then
        Add $s_j$ to S
    else
        Add $s_j$ back to $C_i$
    end if
end for
```

Proof. The time for sorting S is $O(nlogn)$. There are n iterations in the *while* loop. At each iteration, the main part that dominate the time complexity is *getAreaCoverageLevel* or *PruneGreedySelection*. The function, *getAreaCoverageLevel*, is proposed in [4]. Its time complexity is $O(|C_i|dlog(d))$, where $|C_i|$ is the size of a subset $|C_i|$. The time complexity for *PruneGreedySelection* is $O(|C_i|^2 dlog(d))$. Therefore, the time complexity of GS is $O(n^2 dlog(d))$. □

The number of the subsets constructed by GS decides the network lifetime. The following theorem gives the bound of the number of the constructed subsets in the ideal cases.

Theorem 2. *Given some sensors* K*-covering an area, if the sensors' sensing range is fixed and the constructed subsets are disjoint, the maximum number of subsets* m *is* $\lfloor \frac{K}{k} \rfloor$*, where K ($K \geq k$) is the minimum coverage level that the sensor network can provide if all the sensors are activate.*

Proof. If the minimum coverage level provided by a sensor network is K, there exists some point a in the monitored region such that there are K sensors that can cover a. After the first subset is constructed, there are $K - k$ candidate sensors that can cover a. By repeatedly constructing subsets, ideally at most $\lfloor \frac{K}{k} \rfloor$ subsets can be constructed so that each of them can k cover a, that is, to guarantee k-coverage for the whole monitored region. Thus, $\lfloor \frac{K}{k} \rfloor$ is the upper bound of m. The lower bound of m is $\lfloor \frac{K}{k} \rfloor$ too. To prove this, without loss of generality, we assume $m = \lfloor \frac{K}{k} \rfloor - \alpha$. Then, after allocating sensors into $\lfloor \frac{K}{k} \rfloor - \alpha$ subsets, the remaining sensors should be able to $(K - \lfloor \frac{K}{k} \rfloor k + \alpha k)$-cover the monitored area in ideal cases. Because $(K - \lfloor \frac{K}{k} \rfloor k) \geq 0$, the remaining sensors could construct α more subset(s). This leads to a contradiction. Thus, the lower bound of m is $\lfloor \frac{K}{k} \rfloor$ too. Based on the upper bound and the lower bound of m, we conclude that $m = \lfloor \frac{K}{k} \rfloor$. □

5 Density Control of the Sensor Deployment

From Theorem 2, we can see there is a linear relationship between K and the number of the constructed subsets. This is also validated through the simulation results in Section 6. As the network lifetime is decided by the number of the constructed subsets, to have a longer network lifetime, K should be larger which indicates the total number of the sensors should be larger. Another factor that may affect the network lifetime is the *sensor density* which is defined as the number of sensors in each unit area. Different areas in a monitored region have different sensor densities. From the simulation results, we found that there always exist some sensors that were not allocated to any subset which is a waste of resource. The waste is due to the difference between the sensor density of the area near the border of the monitored region and the sensor density of the area at the center of the monitored region. The unallocated sensors are usually the ones at the center of the monitored region. The sensors near the borders have smaller PCL values and the sensors at the center have larger PCL values. GS adds sensors to a subset beginning from the sensors with smaller PCL values. Thus, it is possible after all the sensors near the border have been added to some subsets, there still exist some sensors at the center and no more subsets can be constructed to provide k-coverage for the whole monitored region. Therefore, to extend the network lifetime, the PCL values of the sensors need to be balanced, so that the closer to the border the area is, the more sensors this area should have. To guarantee balancing the PCL values, the number of the neighbors of the sensors' close to the borders should be equal to the number of the neighbors of the sensors at the center. We derive a relationship between the sensor density of the area near the border and the sensor density of the area at the center in Theorem 3. We define a disk centered at c as D_c. The sensor density of D_c is denoted as ρ_c, and

$$\rho_c = \frac{number\ of\ sensors\ in\ D_c}{|D_c|},$$

where $|D_c|$ is the area of D_c.

Theorem 3. *Assume the sensor density at the center of the monitored region A is ρ_c. To guarantee the number of the neighbors of the sensors' close to the borders be equal to the number of the neighbors of the sensors at the center, for a point p whose distance to the border of A is r, the sensor density at p should be*

$$\rho_p = \frac{4\pi R_s^2}{4(\pi - \arccos\frac{r}{2R_s})R_s^2 + r\sqrt{4R_s^2 - r^2}}\rho_c$$

where R_s is the sensing range of a sensor.

Proof. Assume the sensor density in a disk is uniform. As shown in Fig. 1, D_c is the disk centered at c (center of the monitored region) with radius of $2R_s$ and D_p is the disk centered at p with radius of $2R_s$ minus area A. Since we assume the transmission range of a sensor is at least twice of the sensing range of a sensor to guarantee connectivity, the neighbors of the sensor located at c must be within D_c. We desire that the number of the neighbors of the sensor located at c is the same as that of the sensor located at p. This indicates that the sensor density in D_p and the sensor density in D_c satisfy the following

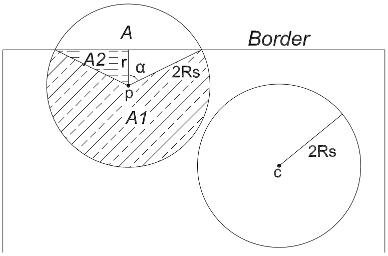


Fig. 1. Density computation

condition: $\rho_p = \frac{|D_c|}{|D_p|}\rho_c$. We know $|D_c| = \pi(2R_s)^2$ and $|D_p| = A_1 + 2A_2 = \pi(2R_s)^2\frac{2\pi-2\alpha}{2\pi} + 2(\frac{1}{2}r\sqrt{(2R_s)^2 - r^2}) = 4(\pi - \alpha)R_s^2 + r\sqrt{4R_s^2 - r^2}$, where A_1 is the area filled with dashed lines, A_2 is the area filled with dotted lines and $\alpha = \arccos\frac{r}{2R_s}$. Hence, $\rho_p = \frac{4\pi R_s^2}{4(\pi - \arccos\frac{r}{2R_s})R_s^2 + r\sqrt{4R_s^2 - r^2}}\rho_c$. □

Based on Theorem 3, users can develop a plan for deploying sensors such that any point in the monitored region may be covered by almost the same number of sensors. This scheme can reduce the number of unallocated sensors. In other words, the amount of wasted recourse can be minimized and the network lifetime can be further extended.

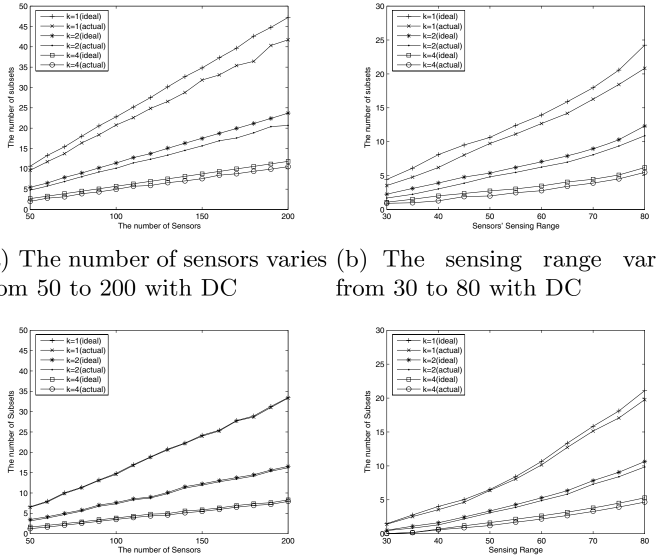
6 Simulation Results

In this section, we evaluate GS's performance by conducting simulations to measure the network lifetime in terms of evaluating the constructed subsets, the number of unallocated sensors, and the effect of density control mentioned in Theorem 3. Networks are randomly generated in a fixed region of 100 × 100. We assume the sensing area of a sensor is circular. Each set of experiments are conducted for $k = 1$, 2 and 4. All data are averages from 50 times experiments.

6.1 Performance of Greedy Selection

In this section, we study the performance of GS algorithm. The results of our algorithm are compared with the ideal case which is proved in Theorem 2.

Fig. 2(a) shows the comparisons between GS's results and the ideal results when the number of sensors varies from 50 to 200 and Density Control is applied. We can see the actual numbers of subsets are close to the ideal results. In fact, the ratio (actual/ideal) are between 85% and 90% stably. When the sensing range of the sensors varies from 30 to 80, Fig. 2(b) shows that the results of our algorithm are still very close to the ideal numbers. The ratio tends to be stabilized between 80% and 90%. Fig. 2(c) shows that our results are almost the same as the ideal results when Density Control is not applied. Actually the ratios are all above 90%, whereas it is not good enough. The reason is that the percentage



(a) The number of sensors varies from 50 to 200 with DC

(b) The sensing range varies from 30 to 80 with DC

(c) The number of sensors varies from 50 to 200 without DC

(d) The sensing range varies from 30 to 80 without DC

Fig. 2. Compare GS's results with the ideal results

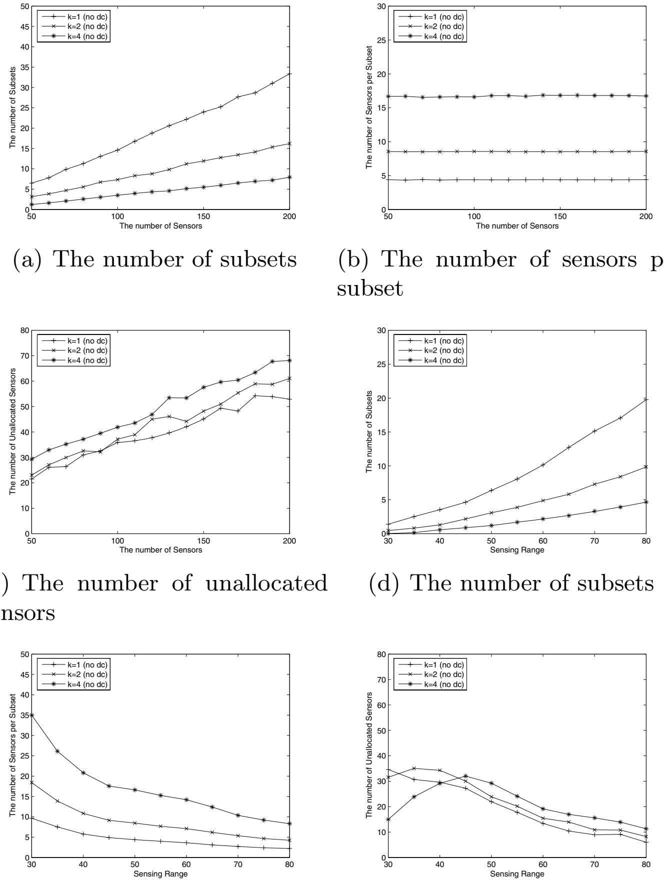
of the used sensors is quite low, only 61.34% on average. Due to low usage percentage, there are enough redundant sensors for constructing more subsets. By applying Density Control, the usage percentage is improved to 83.17% on average. There are less redundant sensors (potential *unallocated* sensors) left. The comparison when increasing sensors' sensing range without Density Control is shown in Fig. 2(d). All these results show that GS's results are stable and very close to the ideal results whatever changing the number of sensors or the sensors' sensing range.

6.2 Effect of Total Number of Sensors on Network Lifetime

The purpose of this set of simulations is to evaluate how the total number of deployed sensors affects the network lifetime. The sensing range of a sensor is set to 50.

Fig. 3(a) shows how many subsets can be constructed when the number of sensors ranges from 50 to 200. As shown in Fig. 3(a), the number of the constructed subsets increases linearly with respect to the network size. This fact is also validated by Theorem 2. It is shown in Fig. 3(b) that the number of nodes per subset keeps constant and this also consolidates with Fig. 3(a). As the coverage level k increases, the number of the constructed subsets decreases since more sensors are required for a subset.

Fig. 3(c) illustrates the number of the unallocated sensors. Around 33%∼40% sensors are not allocated on average. After studying the experiment data, we



(a) The number of subsets

(b) The number of sensors per subset

(c) The number of unallocated sensors

(d) The number of subsets

(e) The number of sensors per subset

(f) The number of unallocated sensors

Fig. 3. Effect of total number of sensors and the sensing range on the network lifetime. Density Control is NOT applied.

found that only a few small areas close to the corners of the region are not covered by these unallocated sensors. The reason is that the density of sensors close to the center of the region is larger than the one close to the corners and borders. Therefore, there are not enough sensors near the borders to form some subsets with the sensors at the center of the region. To solve this problem, we can reduce the number of unallocated sensors through density control which will be evaluated in section 6.4.

6.3 Effect of Sensing Range on Network Lifetime

The purpose of this set of simulations is to evaluate how the sensing range of a sensor affects the network lifetime. 50 sensors are deployed in the region.

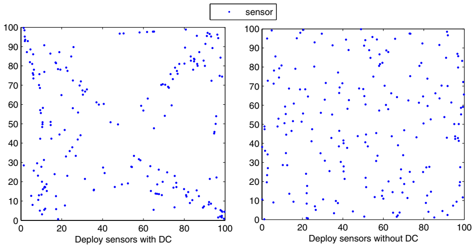


Fig. 4. Deploy sensors with and without DC

The sensing range ranges from 30 to 80. Fig. 3(d) shows that more subsets are constructed as the sensing range increases. Even one sensor can cover the whole area when the sensing range becomes very large. However, only those placed at the center of the region can solely k-cover the entire region except that the sensors have very large sensing range. Those sensors close to the corners and borders need other sensors' cooperation to k-cover the whole region. Thus, the curves of the number of subsets do not keep increasing. There is the maximum number of subsets which is shown in Theorem 2. Fig. 3(e) indicates that larger sensing range leads to fewer sensors in each subset. Larger sensing range also makes more sensors at the center of the region be used. In the previous simulation, they are unallocated sensors generally. With larger sensing range, they can provide the required coverage level without the help from the sensors close to the corners and borders. Fig. 3(f) validates this fact. On average, 39.27% sensors are not allocated into any subset.

6.4 Effect of Density Control on Network Lifetime

In this set of experiments, we apply Density Control (DC) for sensor deployment and evaluate its effectiveness. When deploy the sensors, we apply Theorem 3 to control the density of sensors in the monitored region. By employing DC, we can deploy more sensors in the areas close to the corners and borders such that all the sensors have almost the same number of neighbors. In other words, the monitored region is covered uniformly. The effect of DC is presented in Fig. 4. Due to the space limitation, only the numerical results are shown as following. In the simulations on increasing the number of sensors from 50 to 200, we observed that compared with the cases where DC is not applied, by employing DC, up to 74.6% (averagely 39.8%) more subsets can be constructed and there are up to 66% (averagely 60.6%) less unallocated sensors. In the simulations on increasing the sensing range from 30 to 80, compared with the cases where DC is not applied, by employing DC, we can obtain 39.68% more subsets on average and the number of the unallocated sensors is reduced 18.91% on average. Therefore, by deploying sensors more rationally, sensors are used more effectively.

7 Conclusion and Future Work

In this paper, we investigate a new SSC problem of scheduling sensors to provide k-coverage for a monitored region with the purpose of maximizing the network lifetime. We propose a heuristic algorithm to solve the SSC problem. In addition, we develop a guideline for users to better design a sensor deployment plan by employing density control. Theoretical analyses as well as simulation results are presented to evaluate our proposed algorithm.

We will further investigate the k-coverage scheduling problem with more constraints, such as connectivity, adjustable sensing range and communication range, bandwidth limitation, transmission delay requirement and *etc.* In addition, other non-greedy heuristics as well as distributed algorithms are also of our interest.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a survey, Computer Networks, 38:393–422, 2002.
2. V. Raghunathan, C Schurgers, S. Park, and M. B. Srivastava, Energy-aware wireless microsensor networks, *IEEE Signal Processing Magazine* 19:40–50, 2002.
3. L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino and R. Scarsi, A Discrete-Time Battery Model for High-Level Power Estimation, *Proceedings of DATE*, pp.35–39, 2000.
4. C. Huang and Y. Tseng, The coverage problem in a wireless sensor network, WSNA'03, San Diego, CA, Sep. 2003.
5. Z. Zhou, S. Das, and H. Gupta, Connected k-coverage problem in sensor networks, in Proceedings of the International Conference on Computer Communications and Networks, 2004.
6. M. Cardei and J. Wu, Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks, Journal of Computer Communications on Sensor Networks, 2004.
7. M. Cardei, M. Thai, Y. Li and W. Wu, Energy-Efficient Target Coverage in Wireless Sensor Networks, IEEE INFOCOM 2005, Miami, FL, Mar. 2005.
8. M. Cardei, D.-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, ACM Wireless Networks, 11(3):333–340, 2005.
9. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, SenSys'03, Los Angeles, CA, Nov. 2003.
10. H. Gupta, S. Das, Q. Gu, Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution, MobiHoc'03, Annapolis, MA, Jun. 2003.
11. Z. Abrams, A. Goel, S. Plotkin, Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks, Proc. of Third International Symposium on Information, 2004.
12. M. X. Cheng, L. Ruan, and W. Wu, Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks, IEEE INFOCOM 2005, Miami, FL, Mar. 2005.

13. S. Kumar, T.H. Lai and J. Balogh, On k-Coverage in a Mostly Sleeping Sensor Network, in Proc of the 10^{th} international Conference on Mobile computing and networking, Philadelphia, PA, USA, pp.144–158, 2004.
14. Q. Huang. Solving an open sensor exposure problem using variational calculus. Technical Report WUCS-03-1, Washington University, Department of Computer Science and Engineering, St. Louis, Missouri, 2003.
15. X. Li, P. Wan, and O. Frieder. Coverage in wireless ad hoc sensor networks. IEEE Trans. Comput., 52(6):753–763, 2003.

A New Media Access Control Protocol for Ad-Hoc Wireless Sensor Networks

Kai Chen, Fan Jiang, and Zongyao Tang

Department of Computer Science, University of Science and Technology of China
Hefei, Anhui 230027, P.R. China
ckg@mail.ustc.edu.cn

Abstract. This paper proposes a Media Access Control protocol,called CT-MAC, for ad-hoc wireless sensor networks. It is a contention-based TDMA scheme, which assimilates the valuable design philosophies of CSMA and TDMA while offsetting their weaknesses. Inspired by S-MAC[1], CT-MAC proposes a novel conception of contention/doze and communication/dormancy duty cycle. Unlike S-MAC where the durations of each cycle are fixed, CT-MAC makes these durations self-adaptive for energy efficiency. More importantly, CT-MAC originally adopts the protocol interference model to contend the noninterference channels for each time-slot, and extends the conventional IEEE 801.11 RTS-CTS scheme with stronger functions to successfully solve the exposed terminal problem, which is not well addressed before. We believe that the initial high overhead in the contention period would be eventually compensated by the improved throughput and energy efficiency subsequently. Simulations are performed among CT-MAC, S-MAC and simplified 802.11 DCF to demonstrate the efficiency and effectiveness of the proposed CT-MAC.

Keywords: Media Access Control; sensor networks; ad-hoc networks; energy efficiency; Quality of Service.

1 Introduction

One radio channel cannot be accessed simultaneously by two or more nodes that are in a radio interference range. Because neighboring nodes may cause conflict or signal interference at some nodes if transmitting at the same time on the same channel. Communication in wireless ad-hoc and sensor networks can, like most network communication, be divided into several layers. One of these is the Media Access Control(MAC) sub-layer. This sub-layer is described by a MAC protocol, which tries to ensure that no two nodes interfere with each other's transmissions, and deals with the situation when they actually do.

MAC protocols for wireless networks can be generally divided into CSMA-based(Carrier Sense Multiple Access) and TDMA-based(Time Division Multiple Access). CSMA-based protocols are popular for their simplicity, flexibility and robustness. They do not require strict synchronization or global information,

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 281–293, 2006.

and dynamic nodes joining and leaving are handled gracefully. They often adopt IEEE 802.11[2] RTS-CTS handshake to solve the hidden terminal problem and ensure the reliable communication. However, they scarcely consider the exposed terminal problem and use the channels insufficiently[3]. TDMA-based protocols are, on the other hand, naturally energy preserving in that they have a duty cycle built-in with an inherent collision-free nature, but they often have high complexity in design due to a non-trivial problem of synchronization. What's more, the known TDMA-based sensor protocols are always centralized, not adaptive or suffering the exposed terminal problem.

Careful investigation on the former protocols for ad-hoc wireless sensor networks motivates us to think deeply about how to design an efficient MAC protocol to meet the applications. A stand-alone CSMA or TDMA-based protocol cannot provide a satisfactory solution. Hence, we begin to explore a new MAC scheme that can embody the pick of the basket of the both design philosophies. Considering the characteristics and actual requirements of sensor networks, we are presenting our new CT-MAC, a contention-based TDMA protocol that combines the strong points of traditional CSMA and TDMA while offsetting their weaknesses. It uses CSMA as the baseline MAC scheme, and uses TDMA as a hint to enhance collision resolution. Briefly, CT-MAC is mainly based on a newly proposed contention/doze and communication/dormancy duty cycle for each time-slot, which is a little longer than the traditional TDMA slot. Relied on the protocol interference model[4] and extended RTS-CTS, non-interference links that can transmit simultaneously without causing collision are picked out and assigned concurrently in the contention period. These collision-free links can be used for transmission simultaneously in the subsequent communication period. Although a little higher overhead may be incurred at the beginning of each time-slot because of the extended RTS-CTS handshake and the longer contention period, we believe that such deficiencies will be eventually compensated by improved throughput and energy efficiency during the data transmission.

The rest of the paper is organized as follows: Section 2 formulates the foundational problems. Section 3 briefly reviews the related work and indicates the deficiencies. Section 4 presents the proposed CT-MAC protocol in detail. Then, section 5 shows its performance by comparing it with the traditional IEEE 802.11 DCF and S-MAC. Finally, section 6 gives concluding remarks and directions for future work.

2 Problem Formulation

2.1 Interference Models

We adopt the interference models[4] in our MAC design. They are very useful to address the exposed terminal problem and hence serve to enhance the spatial reutilization rate of wireless radio channels.

The Physical Model: Consider a wireless network consisting of a set of nodes. A direct communication link can be established between node i and j if the corresponding Signal-to-Noise Ratio(SNR) is greater than or equal to a certain threshold γ_0, that is, if

$$SNR(i,j) = \frac{P_i}{L_b(i,j)N_r} \geq \gamma_0 \tag{1}$$

where P_i is the transmitting power of node i, $L_b(i,j)$ is the path-loss between nodes i and j, and N_r is the effect of the thermal noise. There are some constraints and restrictions when assigning the spatial TDMA time slots. First, a node can only transmit or receive a packet in a time slot. Second, a node can only receive data from one other node at a time. Finally, a link is error-free only if the Signal-to-Interference Ratio(SIR) is greater than or equal to a certain threshold γ_1, that is, if

$$SIR(i,j) = \frac{P_i}{L_b(i,j)(N_r + \sum_{k \in N, k \neq i,j} \frac{P_k}{L_b(k,j)})} \geq \gamma_1 \tag{2}$$

where the term $\sum_{k \in N, k \neq i,j} \frac{P_k}{L_b(k,j)}$ is the accumulated interference from other nodes.

The Protocol Model: Under the protocol model, a transmission from node i to j is successful if for any other node k that is transmitting simultaneously,

$$d(k,j) \geq (1+\eta) \times d(i,j); \qquad for\ \eta > 0 \tag{3}$$

where $d(i,j)$ denotes the distance between node i and j. The parameter $\eta > 0$ models the situations where a guard zone is specified by the protocol to prevent a neighboring node from transmitting on the same sub-channel at the same time. It is well known that with a fading factor greater than 2, the protocol model is equivalent to the physical model, where each transmitter uses the same power[4]. The **equation(3)** is a very important criterion in design of CT-MAC.

2.2 Design Requirement of a "Good" Sensor MAC

In order to design a "good" MAC protocol for sensor networks, the following attributes must be considered specifically. The first and most important attribute is energy efficiency. We have to describe energy efficient protocols in order to prolong the network lifetime. Other attributes, such as scalability and adaptability, should also be paid enough attention due to the changing network topology. A well-defined MAC protocol should gracefully accommodate such volatile sensor networks. Although most former sensor MAC protocols suggest that the QoS metrics, such as throughput, delay, and bandwidth utilization, be the secondary factors in sensor applications, we believe that a really "good" MAC protocol should not trade so much QoS performance for its energy requirement, especially for some special sensor networks. Therefore, we are in pursuit of a "good" MAC protocol that satisfies both energy efficiency and QoS guarantee.

3 Related Work

There are increasing MAC protocols proposed for sensor networks. S-MAC[1], a well-known energy efficient protocol, is a CSMA-based scheme with listen and sleep duty cycle. As shown in Figure 1, each time-slot for S-MAC is divided into listen period and sleep period. Only in the listen period, are sensor nodes able to communicate with other nodes by control packets. By a SYNC exchange, all neighboring nodes synchronize together to maintain virtual clusters. And then, by the successful RTS-CTS exchange, two nodes communicate with each other, meaning that they can use their normal sleep time for data transmission. Other nodes will simply follow their sleep schedules to avoid idle listening.

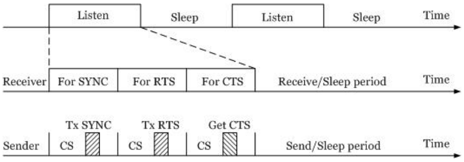


Fig. 1. The illustration of the communication process of S-MAC

T-MAC[5] improves S-MAC's energy usage by using a very short listening window TA at the beginning of each active period, and consumes one fifth the power of S-MAC under variable workloads. However, the adaptive scheme incurs new early sleeping problem because the synchronization of listen periods within virtual clusters is broken. B-MAC[6], a CSMA-based protocol for sensor networks, provides a flexible interface to obtain ultra low power operation, effective collision avoidance, and high channel utilization. Nevertheless, it still bases on the conventional RTS-CTS , and suffers the exposed terminal problem.

S-TDMA[7] has been dismissed as an impractical solution for sensor networks for its lack of scalability and adaptability to changing environments. However, it provides a precious idea of energy efficiency and collision-freedom. Recently, some proposals(e.g.,[8, 9]) are made for TDMA in sensor networks. Unfortunately, these protocols still fail to address the fundamental deficiencies that stay with TDMA. They often have high computational complexity in design due to a non-trivial problem of synchronization. Furthermore, most of them are always centralized, not adaptive or suffering the exposed terminal problem.

4 The Proposed Protocol

In this section, we will describe CT-MAC in detail. Firstly, we briefly formulate the motivation of our work. Secondly, we present the detailed design procedure.

4.1 Motivation

First of all, we analyze the behaviors of the network in Figure 2(a) under S-MAC. The network has six nodes: node 1, 3 and 5 have data to send to node 2, 4 and 6 respectively. Suppose that node 1 and 2 successfully exchange RTS-CTS at first and begin the data transmission, Figure 2(b) describes this process detailedly.

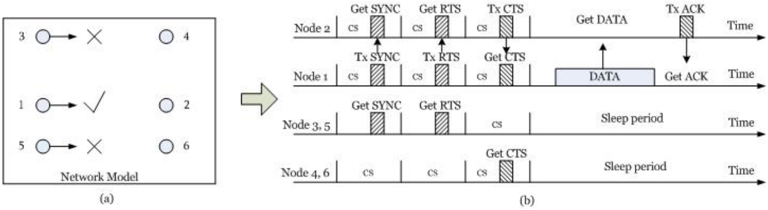


Fig. 2. A simple network model(a), and the behaviors of each sensor node in this network model under S-MAC(b)

We may find at least two drawbacks in Figure 2(b). First, upon receiving RTS from node 1, node 3 and 5 give up contending for their intending channels, although node 3 can transmit to 4 without interfering with the communication between 1 and 2 if we refer to equation(3). This is the exposed terminal problem and results in a waste of channels. Therefore, we should carefully reconsider traditional RTS-CTS when adopting the protocol model (equation(3)) for channel contention. Second, it is not optimal due to a fixed duty cycle to reduce the idle listening. The problem lies in that: when a node has no data to send, hence RTS-CTS exchange may not occur in the corresponding listen period, it still has to be awake on idle listening. This observation also motivates the idea that the nodes should go to sleep early even in the listen period when they are aware of that they have no data queued at the current time, or detect that their transmission will cause collision with the existent links.

4.2 Design of CT-MAC

CT-MAC is a contention-based TDMA protocol and is mainly based on the proposed contention/doze and communication/dormancy duty cycle. This cycle is inspired by S-MAC's periodic listen and sleep mode but more practical and efficient. As shown in Figure 3, CT-MAC uses the adaptive duty cycle and extended RTS-CTS to distributively contend channels for each time-slot. Moreover, it adopts a smart doze and dormancy mechanism to reduce energy consumption.

Neighbor Discovery and Virtual Clustering. We do not assume that a node has built-in knowledge of its one or two-hop neighborhood. This is because the connectivity of the network is dynamic, and so the topology is unpredictable. We think that the neighboring information can be obtained from upper layers such as routing protocols. When a node starts up, it first runs a simple neighbor discovery scheme where it periodically broadcasts a "ping" to its one-hop

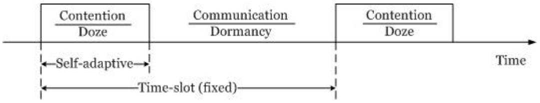


Fig. 3. The periodic contention/doze and communication/dormancy duty cycle of CT-MAC

neighbors to gather its one-hop neighbor list. The "ping" message contains the current list of its one-hop neighbors. In our implementation, each node sends a "ping" message at a random time. Through this process, each node gathers the information received from the "pings" from its one-hop neighbors that essentially constitute its two-hop neighbor information.

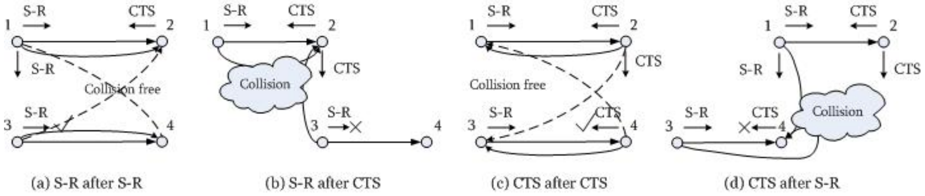
As a contention-based TDMA scheme, CT-MAC needs to maintain the time synchronization among neighboring domains. However, we do not require strict restriction as conventional TDMA, and it is impossible and unnecessary for large scale sensor networks. The time synchronization should be relatively looser and easy to carry out. We review the virtual clustering and synchronization maintenance techniques in S-MAC[1], and adapt them in our CT-MAC implementation.

Combination of SYNC and RTS. It is just a simple trick that we make a combination of SYNC packet and RTS packet. However, it is helpful to strive more time for channel contention and to save more energy. Note that if a sensor node wants to win the channel by sending SYNC firstly, this node is assumed to send RTS subsequently for data transmission. From this point of view, it is possible to combine SYNC and RTS to a single S-R packet. Such combination can be achieved by allowing a node to embed the RTS into its SYNC . It's very simple, and for simplicity and concision, we do not give the illustration here.

Extended RTS-CTS/DATA/ACK. CT-MAC adopts the philosophy of traditional CSMA-based protocols and extends conventional 802.11 RTS-CTS scheme with stronger contents for channel contention. It is important to note that the extended RTS-CTS handshakes operate under the protocol interference model (equation(3)). It can solve both the hidden terminal problem and the exposed terminal problem, and greatly enhances the channel utilization and other performances. We assume that the SYNC and RTS have been combined into an S-R packet. The scheme can be illustrated specifically as follows.

As a sender, the node can still send its S-R after it overhears an S-R or CTS if its intending transmission link is not interfered with that link. The interference can be detected by equation(3) based on its own two-hop neighborhood information, containing the node ID and location. For example, in Figure 4(a), node 1 first sends an S-R to node 2. This packet is overheard by node 3. After receiving the S-R, node 3 detects that the link form 1 to 2 does not interfere with the link from 3 to 4 when transmitting simultaneously. In other words, for specific $\eta_0 > 0$ these two links meet the requirements:

$$\begin{cases} d(1,4) \geq (1+\eta_0) \times d(3,4) \\ d(3,2) \geq (1+\eta_0) \times d(1,2) \end{cases} \tag{4}$$



(a) S-R after S-R (b) S-R after CTS (c) CTS after CTS (d) CTS after S-R

Fig. 4. The illustration of the extended IEEE 802.11 RTS-CTS handshakes

and then node 3 sends its own S-R to node 4 for channel contention. Again, in Figure 4(b), after receiving the S-R request packet from node 1, node 2 replies with its CTS to node 1. This packet is overheard by node 3, node 3 detects that the link from 1 to 2 causes interference with the link from 3 to 4 when transmitting simultaneously. This is because for specific $\eta_0 > 0$:

$$\begin{cases} d(1,4) \geq (1+\eta_0) \times d(3,4) \\ d(3,2) < (1+\eta_0) \times d(1,2) \end{cases} \tag{5}$$

so node 3 cancels its S-R sending. It is necessary to note that, in Figure 4(a), node 3 should wait until an interval of CTS time before sending out its S-R. This is necessary to avoid the collision between the CTS and the coming S-R. Provided that node 3 sends out its S-R towards node 4 immediately after receiving the S-R from node 1 and executing the interference verdict(4). This S-R may probably cause collision with the CTS from 2 to 1 on node 1 or 4. To avoid such collision, we may divide the contention period into several small phases, in which S-R is allowed to be sent anteriorly and CTS is allowed to be sent posteriorly to avoid above collision.

As a receiver, the node can still accept the S-R aiming to it and reply with its CTS after it overhears a CTS or S-R if its transmission link is not interfered with that link. Again, based on the two-hop neighborhood information, equation(3) provides the verdict criterion. As shown in Figure 4(c), after overhearing a CTS from node 2 and a latter S-R from node 3, node 4 detects that the link from 4 to 3 and the link from 2 to 1 does not cause interference when transmitting simultaneously because for specific $\eta_0 > 0$ these two links meet the requirements:

$$\begin{cases} d(4,1) \geq (1+\eta_0) \times d(2,1) \\ d(2,3) \geq (1+\eta_0) \times d(4,3) \end{cases} \tag{6}$$

and then node 4 accepts the S-R and reply with its CTS to node 3. Once more, in Figure 4(d), after overhearing an S-R from node 1 and a subsequent S-R from node 3, node 4 detects that the link from 4 to 3 causes interference with the link from 2 to 1 when transmitting simultaneously since for specific $\eta_0 > 0$:

$$\begin{cases} d(4,1) < (1+\eta_0) \times d(2,1) \\ d(2,3) \geq (1+\eta_0) \times d(4,3) \end{cases} \tag{7}$$

so node 4 cancels its CTS response to node 3.

Furthermore, to avoid the collision between DATA and ACK, we introduce a fixed ACK mechanism. In the fixed ACK time, the sender begins to receive ACK while the receiver starts sending ACK. In this way, the noninterference links may not cause collision between DATA and ACK. For example, in Figure 4(c), when nodes 1 and 3 receive their intending CTS packets in the contention period from nodes 2 and 4 respectively, they use the subsequent communication period for DATA and ACK transmission. It is important that nodes 2 and 4 should use the same fixed time to transmit ACK packets. Otherwise, if one link's DATA transmission time overlaps with another link's ACK time, it may cause collision on the DATA reception.

Advanced Idle Listening Avoidance. Unlike S-MAC, which has a fixed duty cycle, CT-MAC uses a smart and adaptive doze and dormancy mechanism to further reduce the idle listening. Firstly, these node-pairs that have successfully gained their intending channels earlier will turn off the radios to doze mode to save energy in the contention period. Secondly, CT-MAC has all nodes to enter into dormancy mode much earlier either because no data transmission is expected to occur or for the reason that these nodes cannot transmit in that period under the restriction of the protocol interference model (equation(3)).

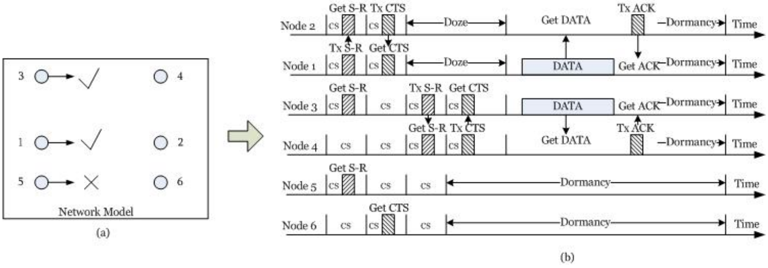


Fig. 5. The behaviors of each sensor node in model(a) under CT-MAC(b)

Distributed Channel Contention. Based on the introduced extended RTS-CTS/DATA/ACK scheme, CT-MAC is dedicated to detect the non-interference links distributively for each time-slot. Specifically, the detection can be illustrated in Figure 5. On the side of sender: First, node 1 intercepts no signal for a CS time and then sends out its S-R to node 2. Second, nodes 3 and 5 have overheard the S-R from node 1 before transmitting their own S-R packets, and then they execute the collision detection. Node 3 finds that the link from 3 to 4 does not interfere with the link from 1 to 2 when they transmit simultaneously, so it waits for an interval of CTS time and sends out its S-R to node 4. Node 5, on the contrary, finds that the link from 5 to 6 interferes with the link from 1 to 2, so it cancels its S-R transmitting. On the side of receiver: First, node 2 receives the S-R from node 1 to itself, since it has not overheard any CTS or S-R before, it directly sends out its CTS to node 1. When node 1 receives the

CTS, the link has been built up and these two nodes can turn off the radios for temporary doze within the contention period for energy conservation. Second, when node 4 receives the S-R from node 3, it has overheard a CTS from node 2 before. Node 4 detects that the link from 4 to 3 does not interfere with the links from 2 to 1, so it transmits back its CTS to node 3.

5 Performance Evaluation

The purpose of the experiments is to demonstrate the effectiveness and efficiency of the proposed CT-MAC and to compare it with traditional IEEE 802.11 DCF and the well-known S-MAC. We perform our experiments from both microview benchmark and macroview benchmark.

5.1 Microview Benchmark

For microview benchmark evaluation, we perform the tests on a simple topology of one-hop network as illustrated in Figure 6. Data flows pass through from source A to sink B, and from source C to sink D. We change the inter-arrival period of messages in order to estimate the performance under different traffic loads. In this experiment, the message inter-arrival period varies from 1 to 10 seconds, which means that a message is generated every 1(to 10) seconds by each source node. 20 messages with 100 bytes each are periodically generated to be transferred to each sink node. To measure the energy consumption, we define the energy model for transmitting, receiving, and sleep as $24.75mW$, $13.5mW$ and $15\mu W$ respectively. This is the same configuration as S-MAC[1]. For each traffic pattern, we have done 10 independent tests to measure three performance

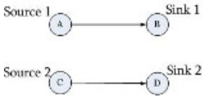


Fig. 6. One-hop symmetric network with 2 sources and 2 sinks

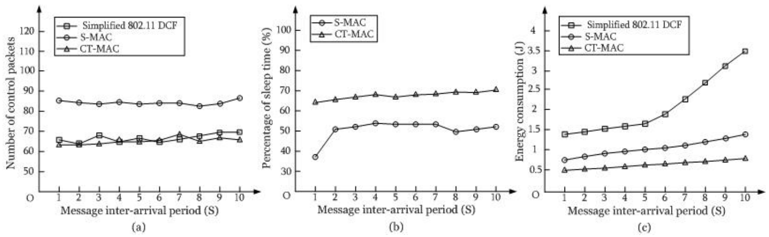


Fig. 7. The measured number of control packets(a), the percentage of sleep time(b), and the energy consumption(c)

metrics such as number of control packets, percentage of sleep (Here we use sleep to represent doze and dormancy of CT-MAC) time and energy consumption.

Figure 7(a) records the number of control packets (SYNC, RTS, CTS and ACK) transmitted and received at a source among 802.11, S-MAC and CT-MAC. In CT-MAC, when there is data traffic, RTS packet is piggybacked into SYNC packet as a combined S-R packet. Therefore, CT-MAC results in much smaller number of control packets than S-MAC, which is similar to 802.11. There are at least two advantages of it: First, less control packets contribute to reducing the control overhead wastage and lessen the probability of collision; Second, using combined S-R instead of SYNC and RTS can strive more phases for CT-MAC to content usable channels for each time-slot, and hence is very helpful to improve the channels utilization.

Figure 7(b) shows that the percentage of sleep time of S-MAC and CT-MAC. Since 802.11 has no sleep mode, it has not been shown here. Obviously, CT-MAC is constantly longer than S-MAC in sleeping. This is because CT-MAC is adaptive to the traffic information, and can intelligently adjust its modes to fit the current situation. Another reason lies in that CT-MAC has a smart "doze" mode, node-pairs that have successfully achieved the intending channels earlier may cleverly turn off the radios waiting for the subsequent communication. More sleep time indicates that CT-MAC is likely to consume less energy than S-MAC.

Just as what we expected anteriorly, CT-MAC has an optimal energy metric. According to Figure 7(c), when the message inter-arrival period is increased, CT-MAC uses less energy consumption than S-MAC. We think that this is because our scheme has an adaptive contention and communication duty cycle, each node can intelligently enter into energy conservation states, such as doze and dormancy, to reduce their energy consumption on idle listening. More importantly, under CT-MAC, the two channels can communicate simultaneously meaning that the exposed terminal problem has been avoided, hence the channel utilization rate has been doubled. This is helpful to save energy in the long run. Additionally, we consider that less control packets in CT-MAC may also contribute to the energy saving. Simplified 802.11 DCF, of course, consumes much more energy than S-MAC and CT-MAC. That is because 802.11 has no duty cycle and wastes much of its energy on idle listening.

5.2 Macroview Benchmark

For macroview benchmark evaluation, we have simulated a network with 100 nodes randomly distributed in a 200×200 meter square area. The communication range of each node is set identically as 40 meters. The coefficient η in equation(3) is set to 0.35. We use a different energy model from above, that is, for k-bit data that travels d-meter from node N_i to node N_j:

$$\begin{cases} E_{N_i}(k,d) = E_{elec} \times k + E_{amp} \times k \times d^2 \\ E_{N_j} = E_{elec} \times k \end{cases} \tag{8}$$

Here, $E_{elec} = 50nJ/bit$ and $E_{amp} = 100pJ/bit/m^2$ are used to denote energy dissipation for transceiver and amplifier respectively. We also change the payload of messages in order to evaluate the performance under different traffic loads. In this experiment, for simplicity, we use one-hop data byte-rate changing from 500*Bps* to 6000*Bps* in steps of 500*Bps*. Hence we do not consider the routing protocol, and in most cases, communication takes place roughly in a similar direction. For each payload, we have done 40 independent tests with random topologies to estimate three metrics: energy consumption, network throughput and end to end delay.

Figure 8(a) shows the per-byte energy consumption of different payloads. It is obvious that CT-MAC consumes energy more efficiently than S-MAC under all the payloads. We believe that CT-MAC has an adaptive duty cycle and uses the channels more sufficiently, so it uses less energy than S-MAC when transmitting the same amount of data. S-MAC, on the other hand, has a fixed duty cycle and suffers the exposed terminal problem. Therefore, it consumes much more energy both for the idle listening and for the suspended transmission. 802.11 has no sleep mode and always suffers idle listening, so it has the worst energy metric. Observe that the per-byte energy cost is higher when payloads are lower, we think that this is because all schemes suffer the idle listening deeply under light traffic loads. With the increase of the payloads, idle listening happens infrequently and per-byte energy cost becomes lower. To its extreme, when the payloads become too heavy, we conjecture that the energy cost becomes higher again because channels are limited and hence more data are stagnated at source nodes causing much energy wastage.

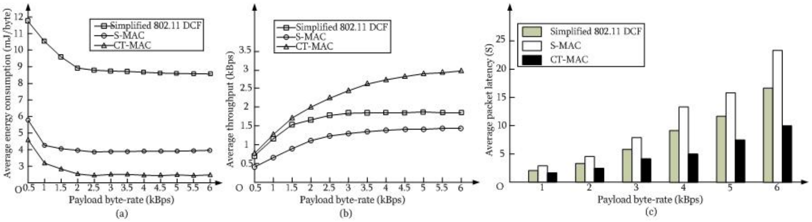


Fig. 8. The average energy consumption comparison(a), the average throughput comparison(b), and average packet latency comparison(c)

It can be seen clearly from Figure 8(b) that CT-MAC has an overwhelming predomination towards S-MAC and 802.11 in terms of average throughput especially under higher payloads. The reason lies in that CT-MAC extends the conventional RTS-CTS handshakes with stronger functions to avoid the exposed terminal problem. Many channels that cannot be used simultaneously in S-MAC or 802.11 can now be used to communicate simultaneously in CT-MAC. Therefore, channels are more saturated and sufficiently used in our CT-MAC. This directly leads to the improved network throughput. Furthermore, another merit that results from the full utilization of channels is that our CT-MAC has lower end to end delay. This can be seen through latency comparison in Figure 8(c). As is shown, CT-MAC has a visible superiority to S-MAC with respect to average packet delay. This is because data in our CT-MAC usually has more opportunities to be transmitted without causing interference, while in S-MAC these data transmission may be postponed to the latter periods due to the exposed terminal problem. Moreover, 802.11 has a lower delay than S-MAC because it does not trade latency for energy efficiency, no sleep mode results in lower transmission delay. However, 802.11 suffers the exposed terminal problem and does not use the channels sufficiently either. Hence, its average packet latency is higher than CT-MAC especially under heave payloads. We also realize that the saturated channel utilization may, on the other hand, increase the packet loss rate to some extent. However, we believe that such a deficiency will not degrade the overall performance of CT-MAC.

6 Conclusion and Future Work

In this paper, we have introduced CT-MAC for wireless sensor networks. It is a contention-based TDMA protocol mainly based on a contention/doze and communication/dormancy duty cycle. In contention periods, nodes are encouraged to contend for their intending channels based on the extended 802.11 RTS-CTS handshake, which can successfully solve both the hidden terminal problem and the exposed terminal problem. In communication periods, DATA are transmitted simultaneously with the relatively fixed ACK time to avoid the occurrence of collision between the DATA and ACK. Furthermore, CT-MAC has a smart doze and dormancy scheme to further reduce energy consumption on idle listening. Simulations are performed among CT-MAC, S-MAC and 802.11 from both microview and macroview benchmarks. The results confirm that CT-MAC is in the ascendant both in energy efficiency and in QoS optimization.

We do not compare CT-MAC with pro T-, B-MAC which are regarded as better schemes than S-MAC. This is because we regard S-MAC as a representative sensor MAC protocol and want to compare CT-MAC with such a foundational protocol. Comparison between CT-MAC and T-, B-MAC will be performed in future, and we are optimistic with the results since these two schemes suffer the exposed terminal problem either. One shortage of CT-MAC is that the extended RTS-CTS scheme is based much on two-hop neighborhood information. The reliance on the upper layer protocol to get such information may, to some extent, degrade the performance of CT-MAC. How to address this deficiency belongs to our future work.

References

1. W.Ye, J.Heidemann, and D.Estrin: An Energy-efficient MAC Protocol for Wireless Sensor Networks. Proc. of the 21st IEEE Infocom, June 2002.
2. LAN MAN Standards Committee of the IEEE Computer Society: Wireless LAN medium access control(MAC) and physical layer(PHY) specification. IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, 1997.
3. Vaduvur Bharghavan, Alan Demers, Scott Shenker and Lixia Zhang: MACAW: A Media Access Protocol for Wireless LAN's. ACM-SIGCOMM, 1994.

4. P.Gupta and P.R.Kumar: The capacity of wireless network. IEEE Transaction on information theory, 46(2), pp 388-404, March 2000.
5. T.van Dam and K.Langendoen: An adaptive energy-efficient mac protocol for wireless sensor networks. Proc. of the First ACM-SenSys, November 2003.
6. J.Polastre, J.Hill and D.Culler: Versatile Low Power Media Access for Wireless Sensor Networks. Proc. of the Second ACM-SenSys, November 2004.
7. R.Nelson and L.Kleinrock: Spatial-TDMA: A collision-free multihop channel access control. IEEE Transactions on Communications, vol.33, pp 934-944, 1985.
8. V.Rajendran, K.Obraczka, and J.J.Garcia-Luna-Aceves: Energy-efficient, collision-free medium access control for wireless sensor networks. Proc. of the First ACM-SenSys, November 2003.
9. J.Li and G.Lazarou: A bit-map-assisted energy-effcient MAC scheme for wireless sensor networks. In 3rd Int. Symp. On Information Processing in Sensor Networks(IPSN'04), pages 55-60, Berkeley,CA, April 2004.

Performance Evaluation of Binary Negative-Exponential Backoff Algorithm in IEEE 802.11 WLAN[*]

Hyung Joo Ki, Seung-Hyuk Choi, Min Young Chung[**], and Tae-Jin Lee

School of Information and Communication Engineering
Sungkyunkwan University
300, Chunchun-dong, Jangan-gu, Suwon, Kyunggi-do, 440-746, Korea
{ki0724, zealion, mychung, tjlee}@ece.skku.ac.kr

Abstract. IEEE 802.11 has employed distributed coordination function (DCF) adopting carrier sense multiple access with collision avoidance (CSMA/CA). To effectively resolve collisions, DCF uses binary exponential backoff (BEB) algorithm with three parameters, i.e., backoff stage, backoff counter and contention window. If a collision occurs, stations involving in the collision increase their backoff stages by one and double their contention window sizes. However, DCF with BEB wastes wireless resource when there are many contending stations. Therefore, in this paper, we propose a binary negative-exponential backoff (BNEB) algorithm which maintains a maximum contention window size during collisions and reduces a contention window size half after successful transmission of a frame without retransmissions. We also compare the performance of DCF with BEB to that with BNEB. From the results, BNEB yields better performance than BEB when the number of contending stations is larger than 4.

1 Introduction

The IEEE 802.11 medium access control (MAC) employs distributed coordination function (DCF) and point coordination function (PCF) [1]. DCF is a contention-based channel access function adopting a carrier sense multiple access with collision avoidance (CSMA/CA) for frame transmission during contention period. PCF is a centrally controlled channel access function and is based on a centralized polling protocol. Therefore, some bandwidth is wasted due to polling overheads and null packets that stations transmit to indicate they have no data to transmit [2]. Due to these problems, PCF is barely implemented in current products [3].

[*] This work was partially supported by a grant No.R01-2006-000-10402-0 from the Basic Research Program Korea Science and Engineering Foundation of Ministry of Science & Technology and by a grant under the Brain Korea 21 Initiative of the Korean Ministry of Education and Human Resources.

[**] Corresponding author.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 294–303, 2006.

In DCF, if channel is idle during distributed interframe space (DIFS), a station having frame(s) to transmit initializes its backoff stage to 0 and takes an minimum contention window size (CW_{min}). Then the station randomly selects a backoff counter from [0, CW_{min}]. The station decreases its backoff counter by one when the channel is sensed idle during a slot duration. At the beginning of an idle slot, the station whose backoff counter value is equal to zero starts to transmit a frame. If a collision occurs, stations involved in the collision increase their backoff stages by one and double their contention window sizes. If the station successfully transmits its frame, it resets backoff stage to 0 and contention window size to CW_{min}. In DCF, the more the number of stations uses wireless resource, the more collision occurrences are possible. To solve this problem, much research on the performance of IEEE 802.11 DCF has been studied [3]-[12].

Bianchi presented an analytical model using bi-dimensional Markov chain model and showed that the proposed model is very accurate [4][5]. With simple modification of Bianchi's model, Xiao showed the limits of throughput and delay of IEEE 802.11 DCF [6]. The fast collision resolution (FCR) uses less contention window size than DCF and exponentially decreases its backoff counter when consecutive idle slots are detected [7]. Although FCR can resolve collision faster than DCF, it has to be used with self-clocked fair queueing (SCFQ) algorithm to guarantee fairness. The gentle distributed coordination function (GDCF) proposed by Wang et al. [8], newly introduced a counter to measure the number of consecutive successful frame transmissions. In GDCF, stations decrease their backoff stages by one whenever the number of consecutive successful frame transmissions reaches the maximum number of permitted consecutive successes. Since GDCF uses the fixed number of permitted consecutive successful transmissions regardless of the number of stations, its performance depends on the number of contending stations. To solve this problem, the enhanced GDCF (EGDCF) used a consecutive success counter to represent the number of consecutive successful transmissions at the same backoff stage[9][10]. If the number of consecutive successful transmissions reaches maximum permitted value, stations decrease their backoff stages by one. Since the maximum permitted value of consecutive successful transmissions is assigned differently according to the stations' backoff stage, EGDCF has better performance than GDCF. However, EGDCF needs another counter like GDCF.

In this paper, we propose a simple and effective collision resolution algorithm called a binary negative-exponential backoff (BNEB) algorithm, and compare the performance of the proposed BNEB with that of binary exponential backoff (BEB) by mathematical analysis and simulations. From the results, BNEB yields better performance than BEB when the number of contending stations is larger than 4. Also we perform simulations to compare the performance of BNEB with that of BEB under the normal traffic condition. The rest of this paper is organized as follows. Section 2 explains BNEB algorithm. Section 3 illustrates an analytical model to evaluate the normalized throughput and MAC delay of BNEB under saturation condition. In Section 4, we verify our analytical model by simulations and compare the normalized throughput of DCF with BNEB to that with BEB under normal traffic condition. Finally, we conclude in Section 5.

2 BNEB Algorithm

The BNEB algorithm uses three parameters, backoff stage, backoff counter, and contention window. The roles of these parameters are similar to those in DCF with BEB. In DCF with BEB, contention window size becomes double whenever a collision is experienced, until it reaches the maximum contention window size (CW_{max}). However, in DCF with BNEB, contention window size initially sets to CW_{max} to reduce the probability that there are more than two stations selecting the same backoff counter value. When a frame successfully transmitted, BNEB decreases the contention window size by half to reduce the delay related to backoff time. Since BNEB introduces minus backoff stage to simply represent consecutive transmission successes, it uses two counters, backoff stage and backoff counter. The contention window size (W_i) at backoff stage i is decided as follows.

$$W_i = \begin{cases} CW_{max} + 1, & 0 < i \leq L, \\ max(2^i(CW_{max} + 1), CW_{min} + 1), & -m \leq i \leq 0, \end{cases} \tag{1}$$

where L is the maximum retry limit and m is the natural number that plays a role in assistance number.

Stations having frame(s) randomly select their backoff counter values from [0, $W_i - 1$] and decrease their backoff counter values by one whenever a slot is idle. A station starts to transmit its frame if its backoff counter value reaches zero. For the frame to be transmitted, the backoff stage is decided by both the previous backoff stage used for the previous frame and the result of its transmission, success or collision. If the previous frame was successfully transmitted at the backoff stage i, the station sets its backoff stage to 0 for $0 < i \leq L$. The station sets its backoff stage to i-1 for $-m < i \leq 0$. And the station sets its backoff stage to $-m$ if $i = -m$. If the transmission of the previous frame failed at the backoff stage i, the station sets its backoff stage to i+1 if $0 \leq i < L$. The station sets its backoff stage to 1 for $-m \leq i < 0$. And if the backoff stage is equal to L, the station drops its frame and then initializes its backoff stage to 0. Therefore, if a collision occurs, the station using BNEB algorithm can effectively resolve collision by using the maximum contention window size.

3 Analytical Model for BNEB

To evaluate saturation throughput and MAC delay of BNEB, we assume that there are n stations having frame(s) to transmit and each station has frame(s) after successful transmission. For a station, $s(t)$ is defined as the random process representing the backoff stage and $b(t)$ is defined as the random process representing the value of the backoff counter at time t. BNEB can be modeled as a bi-dimensional discrete-time Markov chain $(s(t), b(t))$. Fig. 1 illustrates the state transition diagram of the Markov chain of the BNEB.

Let $b_{i,j} = \lim_{t\to\infty} P\left\{s(t) = i, b(t) = j\right\}$ and p be the collision probability that a station experiences a collision in a slot. The state transition probability can be obtained as follows.

$$P\left\{i, j|i, j+1\right\} = 1,\ i \in \left[-m, L\right], j \in \left[0, W_i - 1\right]. \tag{2}$$

$$P\left\{0, j|i, 0\right\} = \frac{1-p}{W_0},\ i \in \left[1, L-1\right], j \in \left[0, W_0 - 1\right]. \tag{3}$$

$$P\left\{0, j|i, 0\right\} = \frac{p}{W_0},\ i = L, j \in \left[0, W_0 - 1\right]. \tag{4}$$

$$P\left\{i-1, j|i, 0\right\} = \frac{1-p}{W_{i-1}},\ i \in \left[-m+1, 0\right], j \in \left[0, W_{i-1} - 1\right]. \tag{5}$$

$$P\left\{i, j|i, 0\right\} = \frac{1-p}{W_i},\ i = -m, j \in \left[0, W_i - 1\right]. \tag{6}$$

$$P\left\{i+1, j|i, 0\right\} = \frac{p}{W_{i+1}},\ i \in \left[1, L-1\right], j \in \left[0, W_{i+1} - 1\right]. \tag{7}$$

$$P\left\{1, j|i, 0\right\} = \frac{p}{W_1},\ i \in \left[-m, 0\right], j \in \left[0, W_1 - 1\right]. \tag{8}$$

Thus, we can make the following relations through chain regularities.

$$b_{i,0} = (1-p)^{-i} b_{0,0},\ i \in \left[-m+1, 0\right]. \tag{9}$$

$$b_{i,j} = \frac{W_i - j}{W_i} b_{0,0},\ i \in [-m+1, 0], j \in [0, W_i - 1]. \tag{10}$$

$$b_{-m,0} = \frac{(1-p)^m}{p} b_{0,0},\ i = m, j = 0. \tag{11}$$

$$b_{i,0} = p^{i-1} b_{0,0},\ i \in \left[1, L\right]. \tag{12}$$

$$b_{i,j} = \frac{W_i - j}{W_i} b_{0,0},\ i \in [1, L], j \in [0, W_i - 1]. \tag{13}$$

$$\sum_{i=-m}^{l} \sum_{j=0}^{w_i - 1} b_{i,j} = 1. \tag{14}$$

From Equations (9)-(14), we can derive $\frac{1}{b_{0,0}}$.

$$\frac{1}{b_{0,0}} = \frac{(p+1) + \frac{1-p}{2}[W(\frac{1-p}{2})^m - 1]}{p(1+p)} + \frac{W+1}{2}\left(\frac{1-p^L}{1-p}\right).$$

Let τ be the probability that a station attempts to transmit a frame. Then we have

$$\begin{aligned} \tau &= \sum_{i=-m}^{L} b_{i,0} \\ &= \left(\frac{1}{p} + \frac{1-p^L}{1-p}\right) b_{0,0}, \end{aligned} \tag{15}$$

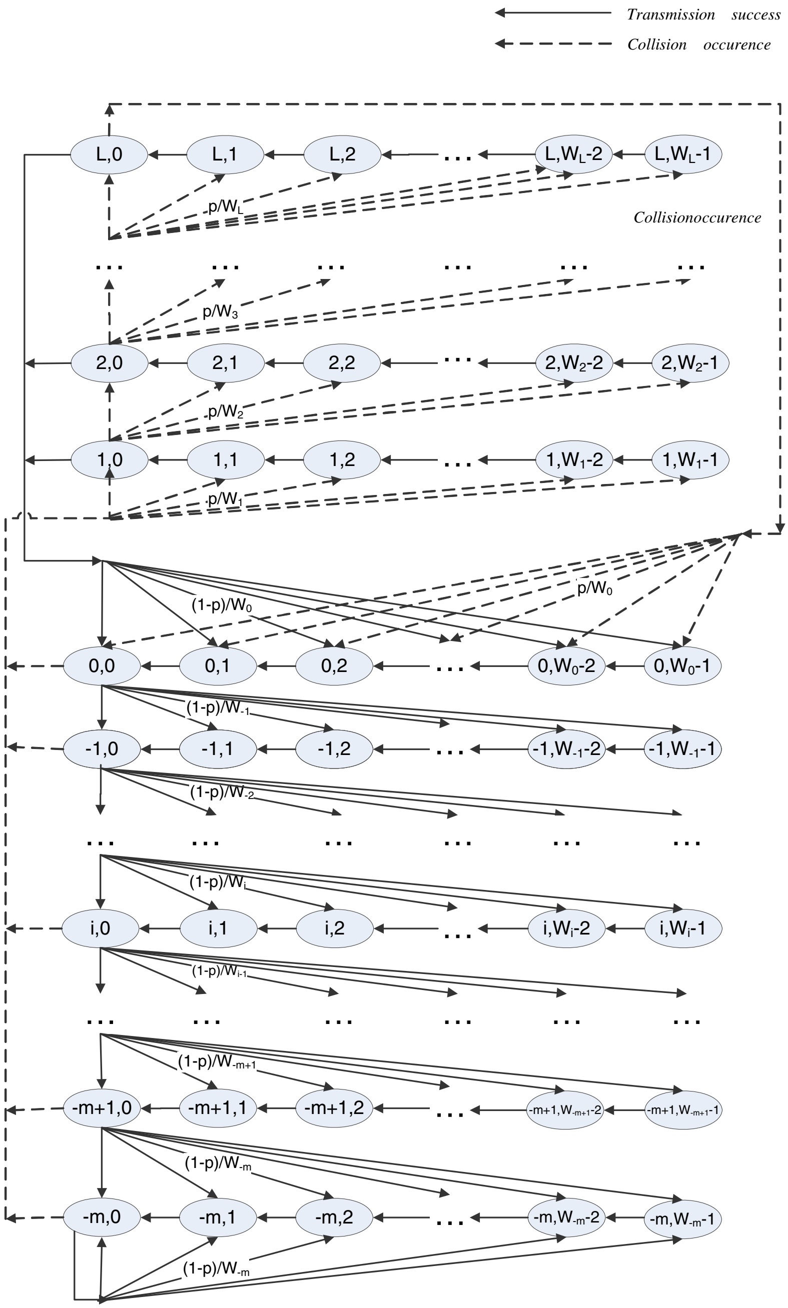


Fig. 1. Markov chain model of BNEB

and

$$p = 1 - (1 - \tau)^{n-1}. \tag{16}$$

The transmission success probability P_s is calculated as

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} \tag{17}$$

where P_{tr} is the probability that there are at least one transmission

$$P_{tr} = 1 - (1 - \tau)^n. \tag{18}$$

Let T_s be the mean time required for the successful transmission of a frame and T_c be the mean wasting time due to the collision of a transmitted frame. T_s and T_c are obtained as follows

$$T_s = DIFS + H + E[P] + 2\delta + SIFS + ACK \tag{19}$$

and

$$T_c = DIFS + H + E[P*] + \delta, \tag{20}$$

where $E[P]$ is the mean transmission time by successfully transmitted packet. $E[P*]$ is the mean transmission time of collided packet. SIFS represents a short interframe space and ACK denotes a transmission time of acknowledgment. $H(= PHY_{hdr} + MAC_{hdr})$ is a transmission time for PHY header (PHY_{hdr}) and MAC header (MAC_{hdr}). The parameter δ denotes a propagation delay.

We can obtain normalized throughput of BNEB under saturation condition as follows

$$S = \frac{P_s P_{tr} E[P]}{(1-P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}(1-P_s)T_c}, \tag{21}$$

where σ is the duration of a backoff slot.

To analyze the mean MAC delay of DCF with BNEB, we first calculate the mean sojourn time d_i at backoff stage i as follows.

$$d_i = (1-p)T_s + pT_c + [(1-P_b)\sigma + P_bT_b]\frac{W_i}{2}, \qquad i \in [-m, L], \tag{22}$$

where T_b and P_b denote the mean time required for the mean freezing time due to the busy channel and the probability that channel is busy, respectively. They are

$$T_b = \frac{(n-1)\tau(1-\tau)^{n-2}}{P_b}T_s + (1 - \frac{(n-1)\tau(1-\tau)^{n-2}}{P_b})T_c \tag{23}$$

and

$$P_b = 1 - (1 - \tau)^{n-1}. \tag{24}$$

The mean MAC delay time D_i when the previous frame was transmitted in state $(i,0)$ is given by

$$D_i = \begin{cases} d_i + pD_1, & -m \le i \le 0, \\ d_i + pD_{i+1}, & 1 \le i \le L, \\ d_i, & i = L. \end{cases} \tag{25}$$

Finally, the mean MAC delay D can be calculated as

$$D = \frac{\sum\limits_{i=0}^{m} b_{i,0} D_i}{\sum\limits_{i=-m}^{0} b_{i,0}}. \tag{26}$$

4 Performance Evaluation

To evaluate the performance of DCF with BEB and BNEB, we use the MAC parameters in Table 1. Normalized saturation throughput of DCF with BEB and BNEB is shown in Fig. 2. For BNEB, the normalized saturation throughput has a slight difference between mathematical analysis results and simulations for n=2 and n=3. However, for n=1 and $n \geq 4$, analysis results are close to simulations. For $2 \leq n \leq 4$, the normalized saturation throughput of BNEB is less than that of DCF with BEB. For $n > 4$, the normalized saturation throughput of BNEB is greater than that of DCF with BEB. The throughput difference between of BNEB and BEB remarkably increases as the number of stations increases. Because BNEB maintains maximum contention window size during collision occurrences, it can reduce the possibility of collision occurrence. Therefore, differently from DCF with BEB, the increment of the number of contending stations does not affect much the throughput of BNEB.

For each station, we perform simulations to compare the throughput of BEB with that of BNEB under normal traffic condition. Packets arrive by a poisson process with parameter λ. Fig. 3 shows the normalized throughput of DCF with BEB and BNEB varying packet arrival rates (λ). The possibility that stations have frame(s) for transmission increases as λ increases until $\lambda = \lambda_{sat}$. If the packet arrival rate is larger than the specific value (λ_{sat}), the probability that a

Table 1. MAC parameters for simulation

PARAMETER	VALUE
Packet payload	8184 bits
MAC header	272 bits
PHY header	128 bits
ACK length	112 bit + PHY header
Channel Bit Rate	1 Mbps
Propagation Delay	1 μsec
DIFS	128 μsec
SIFS	28 μsec
Slot Time	50 μsec
CW_{min}	31
CW_{max}	1023
m	5
L	7

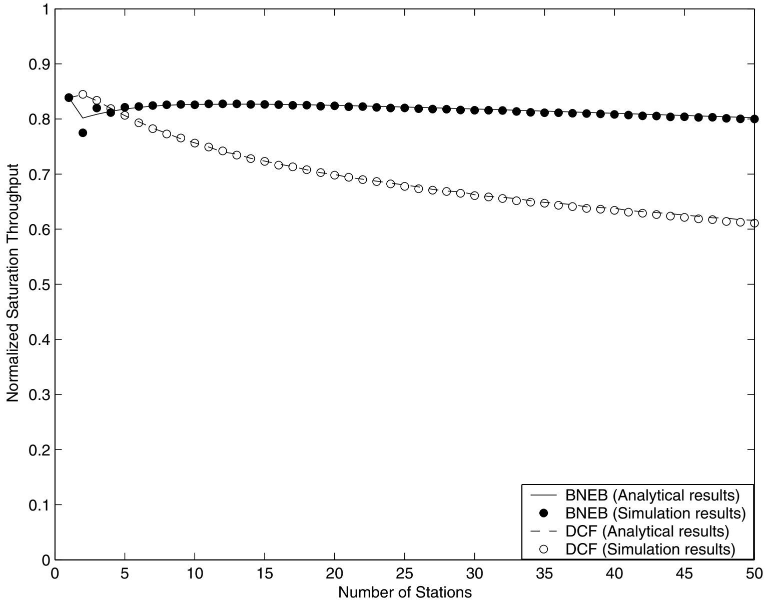


Fig. 2. Normalized saturation throughput of the DCF and the BNEB for variable number of stations (m=5, L=7, CW_{min}=31, CW_{max}=1023)

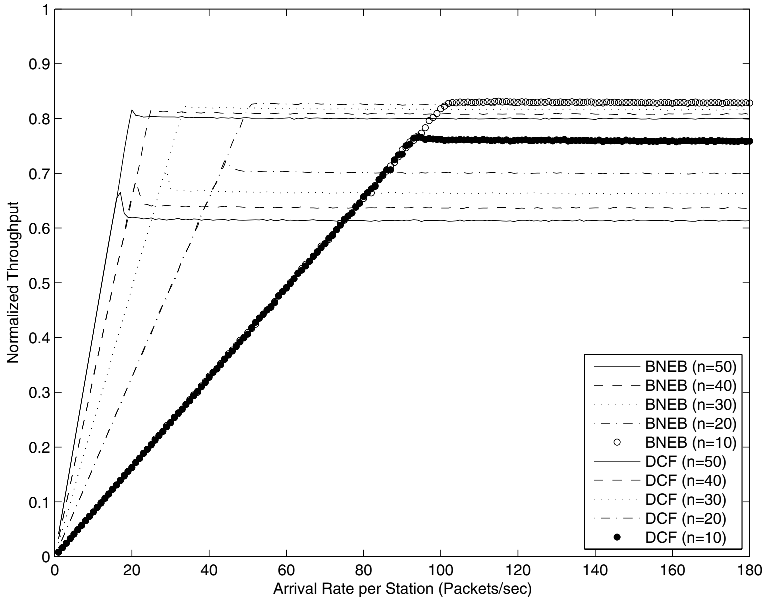


Fig. 3. Normalized throughput of the DCF and the BNEB varying packet arrival rates (m=5, L=7, CW_{min}=31, CW_{max}=1023)

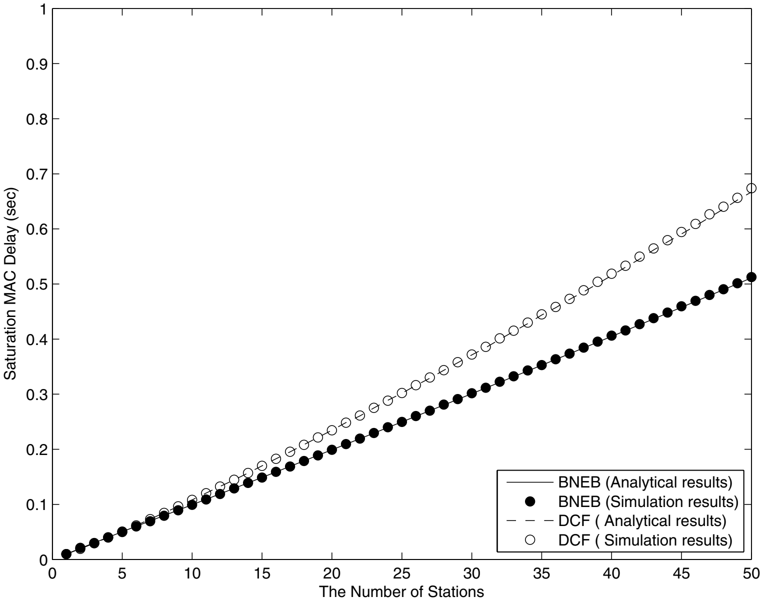


Fig. 4. Saturtion MAC delay of the DCF and the BNEB for variable number of stations (m=5, L=7, CW_{min}=31, CW_{max}=1023)

station has frame(s) for transmission is equal to 1. From the results, the throughput of DCF with BEB and BNEB linearly increases as λ increases until $\lambda = \lambda_{sat}$ and maintains constant for $\lambda \geq \lambda_{sat}$. The λ_{sat} of the BNEB is grater than the DCF with BEB. Therefore, the BNEB can serve more traffic than BEB.

Fig. 4 shows the saturation MAC delay of DCF with BEB and BNEB. The results show that the MAC delay of BNEB is lower than that of BEB. The saturation MAC delay of BNEB linearly increases as the number of contending stations increases. In addition, the difference of MAC delays between BNEB and BEB increases as the number of contending stations increases.

5 Conclusion

In this paper, we proposed a binary-negative exponential backoff algorithm to enhance the performance of DCF with BEB and verified our proposed algorithm via analytical model and simulations under saturation condition. From the results, BNEB can resolve collision more effectively than DCF with BEB. We also performed simulations to evaluate the throughput of DCF with BEB and BNEB under varying packet arrival rates. The results showed that BNEB yield better performance than BEB. For further studies, researches on analytical model of the BNEB are required to evaluate throughput and MAC delay under normal traffic condition.

References

1. IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ISO/IEC 8802-11: (1999(E)) Aug. 1999
2. Kanjanavapasti, A., Landfeldt, B.: An Analysis of a Modified Point Coordination Function in IEEE 802.11. Proceedings of PMRC 2003, Vol. 2, (2003) 1732-1736.
3. Xiao, Y.: Performance Analysis of Priority Schemes for IEEE 802.11 and 802.11e Wireless LANs. IEEE Transactions on Wireless Communications, Vol. 4, No. 4, (2005) 1506-1515.
4. Bianchi, G.: IEEE 802.11-Saturation Throughput Analysis. IEEE Communications Letters, Vol. 2, No. 12, (1998) 318-320.
5. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, (2000) 535-547.
6. Xiao, Y.,Rosdahl J.: Throughput and delay limits of IEEE 802.11. IEEE Communications Letters, Vol.6, No.8, (2002) 355-357.
7. Kwon, Y., Fang, Y., Latchman, H.: A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. Proceedings of IEEE INFOCOM, Vol. 2, (2003) 853-862.
8. Wang, C., Li, B., Li, L.: A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF. IEEE Transactions on Vehicular Technology, Vol. 53, No. 4, (2004) 1235-1246 .
9. Chung, M.Y., Kim, M.-S., Lee, T.-J., Lee, Y.: Performance Evaluation of an Enhanced GDCF for IEEE 802.11. IEICE Transactions on Communications, Vol. E88-B, No. 10, (2005) 4125-4128.
10. Kim, D.H., Choi, S.-H., Jung, M.-H., Chung, M.Y., Lee, T.-J., Lee, Y.: Performance Evaluation of an Enhanced GDCF under IEEE Normal Traffic Condition. Proceedings of IEEE TENCON, (2005) 1560-1566.
11. P. Chatzimisios, A. C. Boucouvalas, V. Vitsas: Effectiveness of RTS/CTS handshake in IEEE 802.11a wireless LANs. IEEE Electronics Letters, Vol. 40, No. 14, (2004) 915-916.
12. B. Raffaele, C. Marco: IEEE 802.11 Optimal performances: RTS/CTS mechanism vs. basic access. Personal, indoor and mobile radio communications, the 13th IEEE International symposium on, Vol.4, (2002) 1747-1751.

An Application-Aware Event-Oriented MAC Protocol in Multimodality Wireless Sensor Networks

Junzhao Du[1],* and Weisong Shi[2]

[1] Software Engineering Institute, Xidian University, Xi'an, Shaanxi 710071, P.R. China
[2] Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

Abstract. In this paper, we design and implement an application-aware, event-oriented MAC protocol (App-MAC) for event-driven multimodality WSN applications. We leverage the advantages of contention-based and reservation-based MAC protocols to coordinate the channel access, and propose channel contention and reservation algorithms to adaptively allocate channel time slots according to application requirements and current events status. To evaluate the proposed App-MAC, we have implemented App-MAC using Berkeley TelosB motes and compared with three state-of-the-art MAC protocols, i.e., S-MAC, TDMA, and TRAMA. We found that App-MAC outperforms other protocols tremendously, including decreasing the average event delivery latency from 3% to 75%, improving the channel utilization efficiency from 12% to 58%, while improving the energy consumption efficiency from 46% to 59%.

1 Introduction

Wireless sensor networks (WSN) is an emerging technology that has been widely used in many applications. In those event-driven WSN applications, e.g., environmental and habitat monitoring, heterogeneous sensor nodes are densely deployed in the sensing area. These sensor nodes are equipped with multimodality sensing devices, such as light sensors, video sensors (camera), and so on. To improve the collaboration and save energy, these sensor nodes located in a nearby area dynamically form a cluster. One of the sensor nodes with more computing power and energy supply can serve as the cluster head, while other sensor nodes act as the cluster members. Events happen spontaneously in a covered area of the cluster, some of them are urgent while others do not have timely delivery requirements. When an event happens, a group of nearby multimodality sensor nodes detect this event and generate variable-length event data to describe the same event from different viewpoints, e.g., image data from camera, sound data from ultrasound sensors, etc. These sensor nodes conduct a localized calculation to generate a unique ID of the event and to determine the priority of the event based on the space, time and data content of the event. After that, these sensor nodes transmit the event data to the cluster head. The cluster head usually performs data fusion or in-network aggregation, and sends the final decision to the sink node using multi-hop routing. To accomplish this, the cluster head needs to receive all of the event data from the event-correlated sensor nodes. In this paper, we assume the cluster heads of

* This work was done during the author's visit at Wayne State University.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 304–316, 2006.

different clusters can cooperate each other to avoid the inter-cluster interference and we investigate how to coordinate the channel access among the cluster members within one cluster, which report event data to the cluster head within only one hop. Hence, we have to address the following three requirements: (1) The urgent event data should be transmitted with higher priority and lower latency. The event delivery latency depends on the time when all the event data produced by different sensor nodes are transmitted to the cluster head; (2) To reduce the event delivery latency, the channel time slots should be fairly shared with the same prioritized events (i.e., inter-event fairness) and the event-correlated sensor nodes (i.e., intra-event fairness); and (3) We also need to improve channel utilization and save energy.

The requirements aforementioned pose challenge on the design of MAC protocol. Although many research efforts [1,2,3,4,5] address the channel access issues in WSN, these MAC protocols are either too general that ignore these special requirements or too specific that address one or two requirements while neglecting others. In this paper, we design and implement App-MAC in the TinyOS platform [6] using Berkeley TelosB motes [7] to address the application requirements and the event-oriented multimodality features of WSN. App-MAC supports prioritized event delivery, fairly shares channel time slots with the same prioritized events (inter-event) and these event-correlated sensor nodes (intra-event), improves channel utilization, and decreases energy consumption. To evaluate App-MAC, we first propose five performance metrics for event-driven multimodality WSN applications, then compare App-MAC with three state-of-the-art MAC protocols, namely S-MAC [5], TDMA [4], and TRAMA [3] with synthetic events. We found that App-MAC outperforms other MAC protocols significantly.

2 App-MAC Protocol Design

We propose an application-aware, event-oriented MAC protocol (App-MAC) in multimodality WSN. App-MAC combines the advantages of the contention-based and the reservation-based MAC protocols. Furthermore, App-MAC considers event-oriented and multimodality features and provides mechanisms to support application-specific requirements in WSN.

2.1 Protocol Overview

App-MAC assumes the time is slotted and every time slot is long enough to send one packet and tolerant of small time shift. Similar to the superframe structure of IEEE 802.15.4 [8], App-MAC divides the superframe into four parts, as illustrated in Fig. 1(a). The first part of the superframe is the beacon slot, which is always used by the cluster head to broadcast the beacon. The second part is the contention slots (CS), which is used by the cluster members to report the event information to the cluster head when they detect new events. During these slots, App-MAC employs the slotted CSMA-CA mechanism [9] for channel access. The following part is the reservation slots (RS). The cluster members leverage RS to transmit the event data to the cluster head. The final part is the inactive slots (IS). During these slots, all cluster members just go to sleep to save energy and cluster heads of different clusters make use of these time slots to exchange information to avoid inter-cluster interference. The CS and RS are further divided into subframes. Each subframe has variable time slots. One subframe of CS is used for those sensor nodes, which have specified sensor type and detect events with a specified priority. One subframe of RS is used for a specified sensor node to transmit the specific event data to the cluster head. The total number of time slots in the superframe is fixed and the applications can tune it according to their requirements. The number of time slots in CS and RS and the number of the subframe in CS and RS are dynamically adjusted by the cluster head through the proposed API (Section 3) according to the status of events. If more events happen spontaneously than expected, we increase the number of time slots in CS to reduce collisions. If only a small number of events happen spontaneously, we reduce the time slots in CS to improve channel utilization.

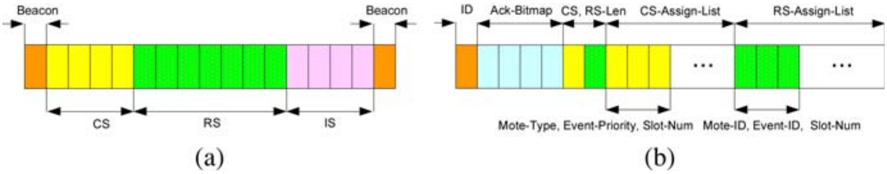


Fig. 1. (a) The superframe of App-MAC. (b) The beacon format of App-MAC.

Fig. 1(b) illustrates the beacon format. The cluster head uses beacons to synchronize with cluster members and announce the CS and RS assignments in this superframe. The first field of the beacon is beacon ID. Cluster members use beacon ID to identify beacon and prevent beacon mismatch. The second field is the acknowledgement bitmap. In the bitmap, one bit represents one time slot of the previous superframe. If the cluster head receives a packet in that slot, it sets the corresponding bit as 1, otherwise 0. Using this way, the cluster head accomplishes group acknowledgements to save energy. The following field describes the length of the CS and RS assignment lists in the beacon. Following this field are the CS and RS assignment lists. Every item of this list describes one subframe of CS or RS. Due to the limit of the packet size, there are fixed number of assignment items shared by CS and RS. Each CS assignment item contains the type of sensor node, the priority of event, and the number of time slots in this CS subframe. These information is used by App-MAC to choose sensor nodes, which have the specified sensor type and detect events with the specified priority, to compete for the time slots in this subframe. In extreme case, if we set the type of sensor nodes and the priority of events as 0, all sensor nodes with any sensor type can compete for these time slots to report any events. Each RS assignment item contains a node ID, an event ID, and the number of time slots in this RS subframe. These information is used by App-MAC to allocate these time slots for a specified node to transmit specified event data. The number of CS and RS assignment items, the content in every CS and RS assignment item are adjusted dynamically by App-MAC according to the WSN application requirements and current events status. The details of those algorithms about how to assign CS and RS will be depicted in the following.

We now describe the functionality of the cluster head and cluster members. Through the collaboration of the cluster head and cluster members, the event data are transmitted to the cluster head in order to meet those requirements we have identified in Section 1. The cluster head performs data fusion or in-network aggregation and forwards events information to the sink node using multihop routing. To do this, the cluster head should collect the event data from the cluster members. The cluster head broadcasts beacons periodically to poll the new events, acknowledges the packets transmitted by cluster members in the previous superframe, announces the CS and RS assignments for this superframe, and synchronizes with all cluster members. The main challenging work for the cluster head is to assign CS and RS. App-MAC has provided mechanisms to facilitate the CS and RS assignments, such as tuning the number of CS and RS assignment items and the content of these items. But how to tune these mechanisms to meet the application requirements is not non-trivial. In the following section, we design and implement CS and RS assignment algorithms to address this problem. We also provide interfaces (Section 3) to facilitate applications to assign CS and RS according to their special requirements.

The main function of cluster members is to detect the events, synchronize with the cluster head, and transmit the events data to the cluster head. To reduce energy consumption and improve channel utilization, the cluster members should follow the channel assignment from the cluster head. (Here we assume no malicious sensors exist.) The cluster members hear the beacon message and synchronize their time schedules according to it by adjusting their timers. They also update their event transmitting information according to the acknowledgement in the beacon. When one cluster member detects an event, it should report the event to the cluster head as soon as possible. To do this, it checks the CS assignment to confirm that it meets the requirements of the CS assignment to compete for these time slots. After that, it uses the CSMA-CA mechanism to compete for the CS slots. To reduce collisions, we also design a distributed algorithm to access the channel, which is described later. When a cluster member has reported event data to the cluster head, the cluster member checks the RS assignment and transmits the specified event data in the specified time slots in RS according to the RS assignment.

2.2 RS Assignment Algorithm

The objective of the RS assignment algorithm is to reduce event delivery latency, support prioritized event delivery, provide inter-event and intra-event fairness, improve channel utilization, and reduce energy consumption. This algorithm is executed by the cluster head. To conduct RS assignment, we assume the cluster head has collected some information about events detected by cluster members by previous events reporting during CS, as illustrated in Fig. 2(b). These events are stored in different queues according to their priority, e.g., P3 and P2. For every event, there is a queue to store information of sensor nodes, which have detected this event and reported the event information to the cluster head, e.g., Mote 1, 3, 4 and 5 are in the queue of Event 1. The information of sensor nodes includes the sensor type and the remaining packets for this event, e.g., Mote 1 has 5 packets. We assume that there are M RS assignment items available in this beacon and there are N time slots in RS. The RS assignment problem now is described as how to choose (at most) M sensor nodes from the current event queues and assign N time slots for them to transmit their remaining packets so as to minimize the event delivery latency, maximize the event and sensor fairness, minimize energy consumption, and maximize channel utilization. This problem is a complex optimization problem. If we know all events information, we can calculate an optimal solution using Integer Programming [10]. However, events happen continuously and spontaneously, it is very difficult to calculate an optimal solution. Thus, we propose to use heuristic method to solve this optimization problem. The basic idea is that we should provide higher priority to urgent events, share channel time slots with events and sensor nodes according to their requirements, and prevent starvation of events with low priority. We propose an algorithm, weight-based event selection in multi-priority queue (WMPQ), to assign RS. We first use the following equation to assign a weight to every event. $W_e = \alpha P_e + \beta T_e + \frac{\gamma}{M_e(\frac{K_e}{\lambda}+1)}$, where P_e is the priority of the event, T_e is the waiting time of the event, M_e is the current number of sensor nodes which has already reported this event, and K_e is the number of remaining packets of these sensor nodes. User can assign different value for α, β, γ, and λ, to tune the weight of these factors. For example, if the value of α is increased, the weight of the event with high priority is also increased. If we increase the value of β, we assign high weight to events which happen early. In general, if an event has less sensor nodes and less packets and it waits a long time for allocation, its weight will become large eventually. Finding a best combination of these parameters is our future work, which we believe is application specific.

WMPQ assigns all RS resources to one event to reduce the event delivery latency. There are two cases to handle. In case one, the number of nodes of this event is no larger than the number of RS assignment item (M), and the number of total packets of this event is no larger than the time slots in RS (N). In this case, the algorithm assigns all RS resources to this event. If there are free RS assignment items and free RS time slots, the algorithm assigns them to another event. In other case, either the number of nodes or packets is more than the corresponding RS resources. To improve channel utilization and to treat sensor nodes with fairness, the algorithm sorts the sensor nodes in this event according to their remaining packets, then picks up these nodes with larger packets, assigns all the RS assignment items to them, and distributes all the time slots in RS to these sensor nodes according to the proportion of their remaining packets. The evaluation results validate that this approach indeed achieves a better channel utilization and provides inter-event and intra-event fairness. Note that the algorithm also adapts to the lossy links. As time goes on, it assigns more slots for these sensor nodes with bad links. Fig. 2(a) lists the pseudo-code of the RS assignment algorithm. In this algorithm, M is all sensor nodes, which have reported events to cluster head; E is all events; P is all event queues with different priorities; mi is the number of the RS assignment items; ns is the time slots of RS. The algorithm builds multiple queues for events with different priorities, sorts every event queue according to the event happen time, sorts the nodes in the event according to their remaining packets, and assigns weight to every event. After that, it chooses the event with largest weight in the head of every queue and assigns RS resources to this event. This process continues until no RS resources to use or no event to assign.

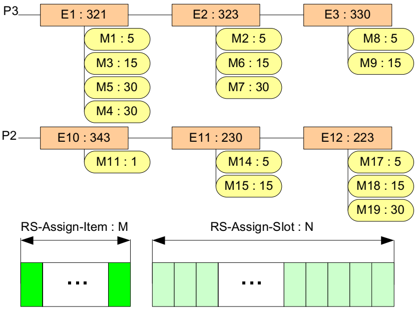
2.3 CS Assignment and Distributed CS Competition Algorithms

The objective of the CS assignment algorithm is to make the cluster members report the events as soon as possible, reduce collisions among the events reporting, improve

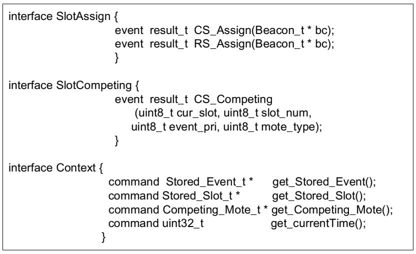
```
The RS Assignment Algorithm

Function RS-Assign ($M$[],$E$[],$P$[],$mi$,$ns$)
Begin Function
for all $m \in M \wedge e \in E \wedge p \in P$ do
    if $m \prec e$ then
        Enqueue(e,m)
    end if
    if $e \prec p$ then
        Enqueue(p,e)
    end if
end for
for all $p \in P$ do
    Sort(p)
end for
for all $e \in E$ do
    Sort(e)
    Weight(e)
end for
repeat
    $e \leftarrow MaxWeightInQueueHead(P)$
    $m_e \leftarrow MoteNumber(e)$
    $p_e \leftarrow PacketNumber(e)$
    if $p_e \leq ns$ then
        AssignAll($e$)
        $E \leftarrow E - e$
    else
        AssignProportion($e$)
    end if
    $mi \leftarrow mi - m_e$
    $ns \leftarrow ns - p_e$
until $mi = 0 \vee ns = 0 \vee Empty(E)$
End Function
```

(a)



(b)



```
interface SlotAssign {
                         event  result_t  CS_Assign(Beacon_t * bc);
                         event  result_t  RS_Assign(Beacon_t * bc);
                         }

interface SlotCompeting {
                         event  result_t  CS_Competing
                              (uint8_t cur_slot, uint8_t slot_num,
                             uint8_t event_pri, uint8_t mote_type);
                         }

interface Context {
                       command  Stored_Event_t *     get_Stored_Event();
                       command Stored_Slot_t *        get_Stored_Slot();
                       command Competing_Mote_t * get_Competing_Mote();
                       command uint32_t                get_currentTime();
                      }
```

(c)

Fig. 2. (a) The pseudo-code of the RS assignment algorithm in App-MAC. (b) An example to illustrate the RS assignment algorithm. (c) The interfaces provided by App-MAC for applications.

channel utilization, and save energy. Applications leverage the underlying mechanisms provided by App-MAC, i.e., adjusting the number of the CS subframe, the priority of events, the type of sensor and the number of time slots in this subframe, to meet the CS assignment requirements. We propose a algorithm to assign CS within one CS subframe. We fix the type of sensor, and the number of time slots in the CS subframe, and adaptively adjust the event priority. The basic idea of this algorithm is described as following. When the cluster head receives some events reporting, it records the priority of events. In the next beacon, the cluster head sets the event priority of the CS assignment item as the highest one. Using this way, in that CS, App-MAC filters out low priority events reporting. In the following beacon, the cluster head sets the event priority of the CS assignment item to next lower event priority to permit other nodes with low priority events report their events. This mechanism can reduce many collisions in the evaluation.

When cluster members have events to report to the cluster head, they should wait for the CS assignment from the beacon. We implement a distributed CS competition algorithm to make the nodes report events as soon as possible. This algorithm can reduce the collisions among the events reporting, and improve channel utilization and save energy. In this algorithm, cluster members first check if they meet the requirements of the CS assignment. If they can compete for the CS slots, cluster members use CSMA-CA to compete for the channel time slots. The cluster members, which have higher priority event or have large event data, will compete for the early time slots of CS. When other cluster members overhear an event report with a higher priority than that of their events, they delay the channel competition.

3 Protocol Implementation

We implement App-MAC in TinyOS platform [6]. The implementation is based on B-MAC [2], which provides bidirectional interfaces to implement other MAC protocols. The App-MAC provides three interfaces for applications to assign CS and RS according to the application requirements. Fig. 2(c) lists the interfaces provided by App-MAC to facilitate applications. The `SlotAssign` interface is used by the cluster head. Whenever the cluster head sends the beacon, it signals this event to applications, which can assign CS and RS based on their channel utilization policy and the events status. The RS and CS assignment algorithms are implemented using this interface. Other assignment algorithms can be easily integrated. The `SlotCompeting` interface is used by cluster members. In the beginning of every CS subframe, App-MAC signals this event to applications to inform the current information for this CS subframe, such as how many slots are there in this subframe and what is the sensor type and event priority for these slots. The distributed CS competition algorithm is implemented using this interface. App-MAC also provides the `Context` interface for applications to get the useful information, including the current reported events, the channel utilization, and the competing sensor nodes, and so on. More details can be found in the technical report version of this paper [11].

4 Performance Evaluation

We are now to evaluate the proposed MAC protocol. First, five performance metrics specially for WSN are proposed. Second, an intensive performance evaluation of App-MAC is conducted through empirical studies on eight Berkeley TelosB motes [7] with synthetic events. Finally, comprehensive comparison with three representative MAC protocols, i.e., S-MAC [5], TDMA [4] and TRAMA [3] is studied in the same context. The empirical study indeed shows the real-world effectiveness of App-MAC.

4.1 Performance Metrics

Given the inherent features of WSNs, e.g., multimodality, event-oriented, and prioritized events, we propose five new performance metrics in this paper.

Event Delivery Latency. We define *the event delivery latency* as the time period from the time the event happens to the time when all these sensor nodes finish transmitting the event data to the cluster head. Based on these data, we calculate *the average event delivery latency*.

Event Fairness Index. We propose *the event fairness index*, which indicates the *inter-event* fairness. The event fairness index is based on the event delivery latency of these events. $I_e = \frac{1}{n}\sum_{p=1}^{n}(\frac{1}{n_p}\sum_{i=1}^{n_p}\sqrt{(L_{p_i} - \overline{L_p})^2})$, where L_{p_i} is the event delivery latency of the ith event with priority of p, $\overline{L_p}$ is the average event delivery latency for those events with priority p, n_p is the total number of events with priority of p, and n is the total number of priority defined in the system. From this definition, we know that if the MAC protocol supports good inter-event fairness, the value of the event fairness index is small.

Sensor Fairness Index. We propose *the sensor fairness index*, which shows the *intra-event* fairness. We define the sensor fairness index based on the event data delivery latency of the event-correlated sensor nodes. $I_m = \frac{1}{m}\sum_{e=1}^{m}(\frac{1}{m_e}\sum_{i=1}^{m_e}\sqrt{(L_{e_i} - L_e)^2})$, where L_{e_i} is the event data delivery latency of the ith sensor node for event e, L_e is the event delivery latency of event e, m_e is the total number of sensor nodes for event e, and m is the total number of events. From this definition, we know that if the MAC protocol can allocate the channel time slots according to the requirements of sensor nodes, the value of the sensor fairness index is small.

Channel Utilization Efficiency. We define *the channel utilization efficiency* as the percentage of channel time slots that are used by sensor nodes to successfully transmit event data packets to the cluster head. If the MAC protocols use the channel time slots efficiently, the value of the channel utilization efficiency should be large.

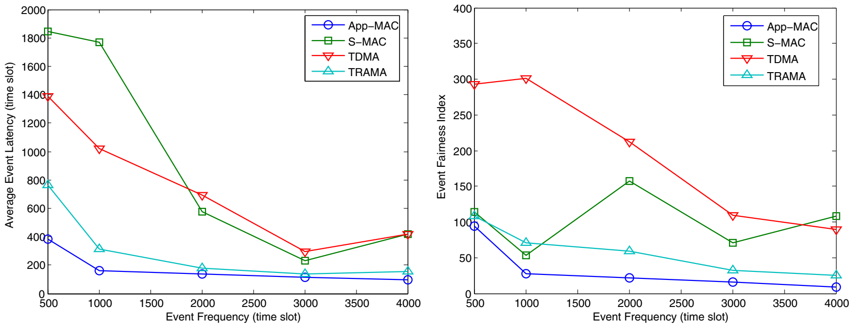
Energy Consumption Efficiency. We define *the energy consumption efficiency* as the ratio of the total energy consumption to the total packets produced by all events in the evaluation. If the MAC protocol is energy efficiency, the value of the energy consumption efficiency should be low. In this paper, we calculate energy consumption of the MAC protocols using the parameters provided by Berkeley TelosB motes [7].

4.2 Evaluation Setup

Many factors affect the performance of MAC protocols, including the cluster size, the link quality among the sensor nodes, the event frequency, the parameter of the MAC protocols, and so on. In this paper, we present the evaluation results with the cluster size as eight motes, which are deployed in the tables around our lab. The packet reception rate (i.e., link quality) among the motes are about 90%. We have randomly generated 15 events within the covered area. These events are fired within 500, 1000, 2000, 3000, 4000 time slots (randomly) to emulate different event frequency. One time slots is 100 ms. Therefore, the event frequency decreases while the time slots increasing. These events have three priorities with five events each. We have three sensor types in the system and different types of sensors produce 5, 15, and 30 packets for every event, respectively, to emulate the diversity of the event data. When an event happens, three or four nearby motes detect it and produce their event data. These motes include at least one mote of the three sensor types. All motes transmit at 0 dBm. For App-MAC, we fix the beacon size as 30 time slots. For the RS assignment algorithm, we fix the parameters for the weight calculation as 100 (α), 20 (β), 100 (γ), 5 (λ). For the CS assignment algorithm, we fix the CS assignment item as one and the default time slots of it as five. For S-MAC, the frame is 30 time slots and the duty cycle is 33%. Note that, S-MAC, TDMA and TRAMA are implemented by ourselves using TelosB motes based on their original ideas in order to compare them in the same context. We run every evaluation case five times and take the average value as the final results to report.

4.3 Evaluation Results

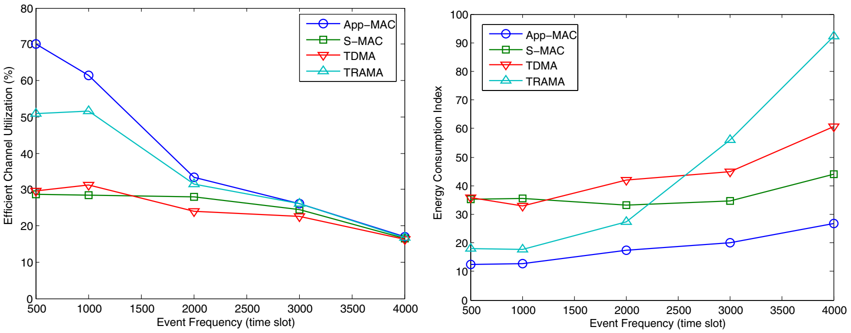
Event Delivery Latency. We first examine the average event delivery latency of these four protocols. Fig. 3(a) reports the average event delivery latency with event priority 1 varying with the event frequency. In this figure, the x axis is the event frequency, the y axis is the average event delivery latency. We compare App-MAC, S-MAC, TDMA and TRAMA in the same figure. For event priority 2 and 3, the similar results are shown. No mater in high event frequency or light one, App-MAC outperforms other MAC protocols, reducing the average event delivery latency about 35% to 75% (priority 1), 25% to 72% (priority 2), and 3% to 63% (priority 3). From these figures, we conclude that the average event delivery latency decreases with the event frequency reducing and App-MAC outperforms other protocols. We attribute this significant improvement to the fact that in App-MAC the CS assignment algorithm and the distributed CS competition algorithm can filter out lower prioritized events and reduce the collisions caused by spontaneous event reporting, and the RS assignment algorithm supports prioritized delivery of events and guarantees the urgent events transmitting without interrupting from others.



(a) (b)

Fig. 3. (a) The average event delivery latency of events with priority 1 vs. the event frequency. (b) The event fairness index vs. the event frequency.

Event and Sensor Fairness Index. The event fairness index shows the inter-event fairness, while the sensor fairness index indicates the intra-event fairness. Fig. 3(b) reports the event fairness index for all MAC protocols with different event frequency. In this figure, the x axis is the event frequency and the y axis is the event fairness index. The values of the event fairness index of App-MAC, S-MAC, TDMA and TRAMA are ranging from 8.4 to 94.4, 53.2 to 157.5, 89.9 to 301.3, and 25.4 to 107.9, respectively. According to the definition of the event fairness index, if the value is small, the MAC protocol supports inter-event fairness. App-MAC improves the event fairness index of 64% over S-MAC, 84% over TDMA, and 50% over TRAMA. The event fairness index decreases while the event frequency is reduced. From our evaluation, we found that App-MAC also improves the sensor fairness index about 94% over S-MAC, 98% over TDMA, and 83% over TRAMA. We attribute this to the fact that, using App-MAC, the CS assignment algorithm and the distributed CS competition algorithm provide the motes, which have same prioritized events, with the same probability to compete for CS, and the RS assignment algorithm fairly allocates RS for those motes.



(a) (b)

Fig. 4. (a) The channel utilization efficiency vs. the event frequency. (b) The energy consumption efficiency vs. the event frequency.

Channel Utilization Efficiency. For TDMA and TRAMA, when most motes have event data to transmit, channel utilization is high. With the time goes on, there are only few motes with event data to transmit, channel utilization keeps in a small amount and lasts for a long time. In S-MAC, motes compete for the channel time slots to send RTS packets, when one cluster member wins CTS, it sends DATA packets to the cluster head while receiving ACK packets from it. These mechanisms make the channel busy. However, the channel utilization efficiency is low. The mechanisms in App-MAC together reduce the collisions and increase channel utilization considerably. Fig. 4(a) shows the channel utilization efficiency of App-MAC, S-MAC, TDMA and TRAMA with different event frequency. In this figure, the channel utilization efficiency of App-MAC, S-MAC, TDMA, and TRAMA varies from 17% to 70%, 16% to 28%, 16% to 31%, and 16% to 51%, respectively. App-MAC improves the channel utilization efficiency 58% over S-MAC, 58% over TDMA, and 12% over TRAMA. From this figure, we see that the channel utilization efficiency decreases while the event frequency is reduced.

Energy Consumption Efficiency. The energy consumption of all protocols increases with the time and the increment ratio of them is different. Analytically, in S-MAC, motes with event data compete for the channel time slots and at most one of them gets the CTS packet. S-MAC always sends ACK to every received packet to solve the hidden terminal problem. These mechanisms consume more energy and extend the total time period of transmitting all the event data. In TDMA and TRAMA, sensor nodes transmit packets to the cluster head when they have event data, otherwise they just go to sleep. The cluster head in TDMA wakes all the time to receive packets from cluster members. TRAMA consumes less energy than that of TDMA using the weighted channel assignment mechanism. App-MAC consumes the least energy among all the protocols. It takes less time to finish transmitting packets, because App-MAC permits only part of motes to compete for the CS slots in its adaptive CS assignment algorithms. Also, App-MAC lets some motes to cancel competing for slots in its distributed CS competition algorithm. The adaptive CS and RS assignment algorithms also reduce idle listening and put sensor nodes to sleep as much as possible. Fig. 4(b) reports the energy consumption efficiency for App-MAC, S-MAC, TDMA and TRAMA in different event frequency. The value of the energy consumption efficiency of App-MAC, S-MAC, TDMA, and TRAMA varies from 12 to 26, from 33 to 44, from 32 to 60, and from 17 to 92, respectively. App-MAC improves the energy consumption efficiency 51% than S-MAC, 59% than TDMA, and 46% than TRAMA. To this end, we see that App-MAC is more energy efficiency.

5 Related Work

MAC protocols have been extensively studied in wireless networking, mobile ad-hoc networks, and wireless sensor networks, interesting readers please refer to [12] for a survey of those protocols. To our knowledge, App-MAC is the first MAC protocol that takes multimodality feature of sensors into consideration. Next, we group these efforts into two categories and discuss their relations with ours.

Contention-based MAC protocols. The IEEE 802.11 DCF [9] is an contention-based protocol, and it mainly builds on MACAW [13]. The basic idea of these protocols is for a sender to transmit a request-to-send (RTS) that the receiver acknowledges with a clear-to-send (CTS). If the RTS/CTS exchange is successful, the sender is allowed to transmit one or more packets. S-MAC [5], T-MAC [4] and B-MAC [2] are specially designed for WSN. S-MAC aims to energy conservation and self-configuration. In S-MAC, each mote goes to sleep for some times, and then wakes up and listens to see if any other wants to talk to it. All motes are free to choose their own listen/sleep schedules. The listen interval is divided into three parts, for SYNC, RTS, and CTS. Each part is further divided into many time slots for senders to perform carries sensing. B-MAC provides some interfaces for application to control the back off time when it initializes sending packet or when it detects collisions. App-MAC uses the contention-based protocol to report event information. In App-MAC, we design the mechanisms to reduce the collisions, e.g., filtering out some events with specified priority and some motes of specified sensor types. We also design CS assignment and the distributed CS competition algorithms to reduce collisions and save energy.

Reservation-based MAC protocols. TDMA [1] and TRAMA [3] are reservation-based protocols. In TDMA, time period is divided into frames that provide each node with a transmission slot over which it can transmit data without collisions. TRAMA is based on a distributed contention resolution algorithm that operates at each node based on the list of direct contenders and indirect interferences. They introduce energy-efficient collision-free channel access in WSN. IEEE 802.15.4 (Zigbee) [8] designs superframe structure, which comprises an active part and an optional inactive part. The active part is further divided into a contention access period (CAP) and an optional contention free period (CFP). App-MAC uses the reservation-based protocol to transmit event data. In App-MAC, we design a RS assignment algorithm to allocate channel time slots according to the application requirements and the event status. The RS assignment algorithm provides prioritized event delivery and supports motes and events with fairness.

6 Conclusion and Future Work

In this paper, we have designed and implemented an application-aware, event-oriented MAC protocol (App-MAC) for multimodality WSN. By using Berkeley TelosB motes, the evaluation results show that the proposed App-MAC is able to support prioritized delivery of events, provide inter-event and intra-event fairness, and improve channel utilization while reducing energy consumption. App-MAC outperforms three other MAC protocols, i.e., S-MAC, TDMA, and TRAMA, in term of the average event delivery latency, the event and sensor fairness index, the channel utilization efficiency and the energy consumption efficiency.

References

1. Arisha, K., Youssef, M., Younis, M.: Energy-aware tdma-based mac for sensor networks. In: Proceedings of the IEEE Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT'02). (2002)
2. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04). (2004)
3. Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J.J.: Energy-efficient, collision-free medium access control for wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03). (2003)
4. Dam, T., Langendoen, K.: An adaptive energy-efficient mac protocol for wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03). (2003)
5. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM'02). (2002)
6. Levis, P., Madden, S., Gay, D., Polastre, J., Szewczyk, R., Woo, A., Brewer, E.A., Culler, D.E.: The emergence of networking abstractions and techniques in tinyos. In: Proceedings of the 1st USENIX Symposium on Networked Systems Design and Implementation (NSDI'04). (2004)
7. Polastre, J., Szewczyk, R., Culler, D.: Telos: Enabling ultra-low power wireless research. In: Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN'05). (2005)
8. IEEE802.15.4/D18: Draft standard: Low rate wireless personal area networks (2003)
9. IEEE802.11: Part 11: Wireless lan medium access control (mac) and physical layer (phy) specification (1999)

10. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization : Algorithms and Complexity. Dover Publications (1998)
11. Du, J., Shi, W.: An application-aware event-oriented mac protocol in multimodalitywireless sensor networks. Technical Report MIST-TR-2005-010, Wayne State University (2005)
12. Kumar, S., Raghavan, V.S., Deng, J.: Medium access control protocols for ad hoc wireless networks: a survey. Elsevier Ad-Hoc Networks Journal (2004)
13. Bharghavan, V., Demers, A., Shenker, S., Zhang, L.: Macaw: A media access protocol for wireless lans. In: annual conference of the Special Interest Group on Data Communication (SIGCOMM'94). (1994)

Monte-Carlo Localization for Mobile Wireless Sensor Networks

Aline Baggio and Koen Langendoen

Delft University of Technology, The Netherlands
{A.G.Baggio, K.G.Langendoen}@tudelft.nl

Abstract. Localization is crucial to many applications in wireless sensor networks. This article presents a range-free anchor-based localization algorithm for mobile wireless sensor networks that builds upon the Monte Carlo Localization algorithm. We improve the localization accuracy and efficiency by making better use of the information a sensor node gathers and by drawing the necessary location samples faster. Namely, we constrain the area from which samples are drawn by building a box that covers the region where anchors' radio ranges overlap. Simulation results show that localization accuracy is improved by a minimum of 4% and by a maximum of 73%, on average 30%, for varying node speeds when considering nodes with knowledge of at least three anchors. The coverage is also strongly affected by speed and its improvement ranges from 3% to 55%, on average 22%. Finally, the processing time is reduced by 93% for a similar localization accuracy.

Keywords: Distributed localization algorithms, wireless sensor networks, mobility, Monte Carlo Localization, simulations.

1 Introduction

Many applications have a need for localization, be it for locating people or objects. Most of the time, data recorded from a wireless sensor only makes sense if correlated to a position, for example the temperature recorded in a given machine room or cold-store. Similarly, many end-user programs are location-aware, for example people would like to find the closest bus stop or mailbox, and emergency services need to localize persons to be rescued. In many cases, such as indoors, the Global Positioning System cannot be used. From now on, we will refer to a person, object or computer coupled with a wireless sensor to be localized as an *(unknown) node.*

This article presents a localization algorithm for wireless sensor networks specifically designed with mobility in mind. One important factor is to let the wireless sensors benefit from mobility and not only suffer from it. Literature [5,7,11,12,13,14] has shown that using mobile anchors in static wireless sensor networks helps improving the accuracy of the localization algorithm, as more nodes can benefit from the anchors' position broadcasts and as each node can

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 317–328, 2006.

hear more of these. Similarly, mobile sensors have a chance to get more information than in a fully static environment. The challenge, however, is that information in a mobile wireless sensor network gets invalidated more quickly if all the nodes are moving. Hu and Evans introduce a localization algorithm dealing with these different characteristics [9]. Their approach builds upon Monte Carlo Localization (MCL) methods used in robotics to locate a mobile robot. In this article, we present improvements to Hu and Evans' algorithm leading to better accuracy and lower computational cost when localizing nodes.

The remaining of this article is organized as follows. Section 2 presents some background information on localization mobile wireless sensor networks. Section 3 describes both our localization algorithm and Hu and Evans' algorithm it builds upon. Section 4 gives insight on the accuracy of the localization and efficiency of the algorithm. And finally, Section 5 concludes and gives some future work directions.

2 Dealing with Mobility

In the article [9], Hu and Evans present a range-free localization algorithm for mobile sensor networks based on the Sequential Monte Carlo method [4,8]. The Monte Carlo method has been extensively used in robotics [2,15] where a robot estimates its localization based on its motion, perception and possibly a prelearned map of its environment. Hu and Evans extend the Monte Carlo method as used in robotics to support the localization of sensors in a free, unmapped terrain. The authors assume a sensor has little control and knowledge over its movement, in contrast to a robot. A range-based version of the MCL algorithm has recently been proposed by Dil and al. [3].

Apart from the experiments with MCL, there are at the moment few localization protocols specifically designed with mobile wireless sensors in mind. Most of the papers presenting localization algorithms suggest that supporting mobility can be achieved by rerunning the localization algorithm after some time interval, either static or adaptable. While this is not optimal but feasible in some cases, the whole class of algorithms using information from distant nodes or iterative approaches will suffer from severe information decay. At the time the information reaches a distant node that wants to use it, it is very likely that the whole network configuration has changed. A node will therefore always calculate an inaccurate location, not due to the lack of information or to the intrinsic inaccuracy of the calculations it uses, but due to the way its localization algorithm gathers this information.

Mobility introduces a real-time component to the localization algorithms. Wireless sensor networks are usually considered delay-tolerant [6,11]. To the contrary, mobility makes a sensor network delay intolerant: information gathering and location calculation should happen in a timely manner, dependent on the speed of both the nodes and the anchors. This means that in a mobile wireless sensor network, methods relying on global knowledge such as calculating the number of hops or distances to all the anchors in the network are to be avoided.

Similarly, a mobile node cannot really benefit from iterative localization techniques where the location estimation is refined whenever a node receives more information from the network.

Besides possible information decay, a localization algorithm deployed in a mobile wireless sensor network should be able to cope with the temporary lack of anchors. In other words, the algorithms should be able to produce a location estimate in such conditions if the application layer has a need for it. In such cases, the location estimation could easily be tagged as uncertain, providing a mean for the application to assess how much the results of the localization algorithm should be trusted.

We believe mobility should be taken into account directly when designing new localization algorithms. A wireless sensor should benefit from mobility and exploit it to improve the efficiency of localization or get a better accuracy of its position estimates. The algorithms based on MCL are offering such guarantees. In the following, we build upon the range-free Monte Carlo Localization algorithm proposed by Hu and Evans [9] and show that by improving the way the anchor information is used, we can improve both the accuracy and the efficiency of the algorithm.

3 Localization in Mobile WSNs

This section first describes the basic MCL algorithm and then introduces our extensions.

3.1 Monte Carlo Localization

In [9], Hu and Evans define their localization algorithm as follows. The time is divided into discrete intervals. A sensor node relocalizes in each time interval. During the localization-algorithm initialization phase, a sensor picks a random set of N samples $L_0 = \{l_0^0, l_0^1, ..., l_0^{N-1}\}$, i.e. random localizations within the deployment area. From then on, the two steps, *prediction* and *filtering*, repeat. During the prediction step, a node picks random locations within the deployment area, possibly constrained by its maximum speed and the previous location samples. At time t, a sensor node thus generates a new set of samples L_t based on the previous set L_{t-1}. In practice, given a location l_{t-1}^i from L_{t-1}, a random location l_t^i is chosen from the disk of radius v_{max} around l_{t-1}^i, v_{max} being the maximum speed of a node. During the filtering phase, all impossible locations l_t^i are removed from the new set of samples L_t. The filtering occurs by using the position information obtained from both the one-hop and two-hop anchors. The one-hop-anchor group is composed of the anchors the sensor node heard directly. These anchors are assumed to be in the radio range r of the sensor node. The two-hop-anchor group is composed of anchors the sensor node did not hear itself but its one-hop neighbors did. These anchors are assumed to be in the range $2r$ of the sensor node but not within a radius r. In other words, MCL makes use of negative information. Note that this usually leads to an improved localization accuracy in an obstacle-free deployment area but is quite risky otherwise.

Note that after the filtering step, there may be less samples in the set than desired. The prediction and filtering process thus repeats until the desired number of samples is reached. The location estimate of a sensor at time t is the average of all the possible locations from the sample set L_t.

3.2 Monte Carlo Localization Boxed

Despite being quite accurate, especially in low-anchor configurations, MCL's efficiency can be improved. Drawing samples is a long and tedious process that could easily drain a lot of energy from a sensor node. Furthermore, the way MCL makes use of anchor information leaves room for improvement. Our version of the Sequential Monte Carlo Localization called Monte Carlo Localization Boxed (MCB) uses steps similar to those of MCL. The major differences lie in the way we use anchor information and the method we use for drawing new samples (see [1] for algorithmic details).

The original MCL algorithm uses information about one-hop and two-hop anchors at filtering time only, for rejecting impossible samples. In MCB, we use the information about the anchors heard to constrain the area from which the samples are drawn, method which we explain below. Reducing the area to sample from has two consequences. First, we draw good samples more easily and thus faster. Drawing good samples means that we have to reject samples less often in the filtering phase, reducing thereby the number of iterations the algorithm needs to fill the sample set entirely. The second consequence is implementation dependent. Unlike the pseudo-code shown in [9], the implementation of MCL sets a bound on the number of times a node can try to draw samples if its sample set does not contain the required number of samples yet. This boils down to avoiding that the algorithm loops endlessly if no valid sample can be drawn for a given configuration. In [9], Hu and Evans selected a sample-set size N of 50. A node tries at most twice 10,000 times to draw a sample. This happens once with a strict speed condition, drawing new samples from the disk of radius v_{max} around the old sample, and a second time with a relaxed speed condition, drawing new samples from the disk of radius $v_{max} + delta$ around the old sample. Drawing samples with a relaxed speed constraint only happens if the sample set is not full after the first series of 10,000 draws. After the 20,000 attempts, the sample set may still be not full, having less than 50 good samples. MCL does not try to fill the sample set any further. In MCB, we make sure that the sample set is as full as possible by drawing samples that do not have to be filtered and therefore do not require a redraw. In most cases, the sample set is full well before 10,000 tries. Experiences have shown that 100 attempts is ample enough to fill the sample set entirely. By ensuring that the sample set is full in 50 to 100 draws, a node can save precious battery power. Filling the sample set whenever possible also has a positive influence on localization accuracy over time.

The method used for constraining the area from which MCB draws samples is as follows. A node that has heard anchors – one-hop or two-hop anchors – builds a box that covers the region where the anchors' radio ranges overlap. In other words, this box is the region of the deployment area where the node is

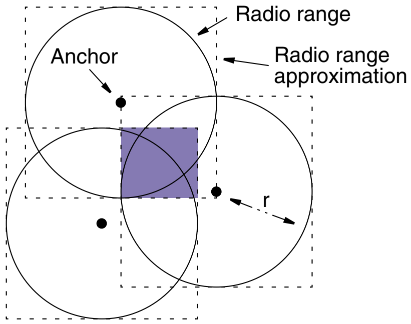


Fig. 1. Building the anchor box

localized. We call such a box the *anchor box*. Figure 1 shows an example of an anchor box (shaded area) in the case where three one-hop anchors were heard. For each one-hop anchor heard, a node builds a square of size $2r$ centered at the anchor position, r being the radio range. Building the anchor box simply consists in calculating coordinates (x_{min}, x_{max}) and (y_{min}, y_{max}) as follows:

$$x_{min} = \overset{n}{\underset{j=1}{max}}(x_j - r) \qquad x_{max} = \overset{n}{\underset{j=1}{min}}(x_j + r) \tag{1}$$

$$y_{min} = \overset{n}{\underset{j=1}{max}}(y_j - r) \qquad y_{max} = \overset{n}{\underset{j=1}{min}}(y_j + r) \tag{2}$$

with (x_j, y_j) being the coordinates of the considered anchor j and n being the total number of anchors heard. When considering two-hop anchors, we replace r by $2r$ in the above formulas.

In addition, in the simulation, the box-building algorithm cares for inconsistent or out-of-range boxes. In other words, for boxes where the minimum value x_{min} or y_{min} is larger than its respective maximum value x_{max} or y_{max}, the box is reset either to a box with one-hop anchors only, or to the whole deployment area. In the case where values are outside of the deployment area, for example x_{min} is negative, we reset the value to the coordinate of the border, in our example 0.

Once the anchor box is built, a node simply has to draw samples within the region it covers. Since the anchor box is a bare approximation of the radio range of the anchors, we keep a filtering step, as in the original MCL. And as in the original MCL, the prediction and filtering steps repeat until the sample set is full or until the maximum number of tries is reached.

Building an anchor box as described above is used in the case where the sample set is empty, for example at initialization time. In the case where we already have samples, the bounding box is built with an additional constraint, namely, for each old sample l_{t-1}^i from the sample set L_{t-1}, we build an additional square of size $2 * v_{max}$ centered at the old sample as follows:

$$x_{min}^i = max(x_{min}, x_{t-1}^i - v_{max}) \tag{3}$$

$$x_{max}^i = min(x_{max}, x_{t-1}^i + v_{max}) \tag{4}$$

$$y_{min}^i = max(y_{min}, y_{t-1}^i - v_{max}) \quad (5)$$

$$y_{max}^i = min(y_{max}, y_{t-1}^i + v_{max}) \quad (6)$$

where (x_{t-1}^i, y_{t-1}^i) are the coordinates of sample l_{t-1}^i. This updated box, which we call *sample box*, delimits per old sample the area a node can move in one time interval at maximum. Whenever a node has an initialized sample set but heard no anchor, we build the sample box solely based on the maximum node speed and the old samples. Box building remains a sequential process, where the anchor box is build first – and saved for subsequent uses – and updated independently for each old sample, creating thereby the sample box from which the new samples are effectively drawn.

Besides building anchor and sample boxes for drawing new samples, MCB tries to make the best possible use of all information a node received. This influences the localization algorithm in two ways. First, during the initialization phase or whenever the sample set becomes empty, MCB allows a node to use two-hop anchor information even if it has heard no one-hop anchor. Where the original MCL makes use of two-hop-anchor information only in combination with one-hop-anchor information during the filtering phase, MCB allows a node to use all information it got both at prediction and filtering time. This means that a node that heard only two-hop anchors can still draw samples using these and produce a location estimate.

Second, whenever a node has heard anchors and has an already initialized sample set but has failed to fill it (entirely) with new samples, MCB reverts to solely drawing new samples from the anchor box. In other words, the sample boxes are not used anymore. Not being able to fill the sample set typically happens when too many old samples are inconsistent with the current connectivity and speed constraints. To counter old sample inaccuracy and draw new valid samples, the algorithm would need to let the node travel a too long distance, i.e. more than what could be covered with speed v_{max} in one time interval, to finally meet the connectivity constraints. In such a case, MCL would try to draw new samples with a relaxed speed constraint $(v_{max} + delta)$. Drawing solely from the anchor box in MCB is equivalent to relaxing the speed. The advantage, however, is that no *delta* for the allowed speed increase has to be chosen in advance as it is the case with MCL.

4 Evaluation

In the following, we present the results of the simulations of three localization algorithms. We reused and extended the simulator used in [9]. First, we ran MCL as specified in [9] and presented above. MCB was implemented as described in Section 3. In addition, in order to compare with a well-known, simple and efficient localization algorithm, we chose to run the Centroid [10] algorithm in our simulations. Centroid calculates the position of an unknown node by computing the averages of all the x and all the y coordinates of the anchors heard.

The selected localization algorithms have been tested with simulated mobile wireless sensor networks. In the following, we assume a number of nodes and anchors deployed in an obstacle-free area of 500x500 units. We thus allow all algorithms to use negative information (see [1] for more on this topic). Both the nodes and anchors are mobile. The anchors know their location a priori, for example by using GPS. The radio range r is set to 100 units for both the anchors and the nodes.

A simulation run consists in feeding the simulator with a set of parameters such as the number of nodes in the network, the number of anchors, the maximum speed at which they move, the degree of irregularity used to model the radio communication. Time is discrete in the simulator. The speed of a node thus represents the distance in "units" a node can move per "time unit". For each selected maximum speed, the simulator generates a number of random network configurations, in our case 20. For each distinct network configuration, we simulate 200 time units. The first 100 units, the nodes move without localizing. For each subsequent time unit, the nodes first localize and then move. In other words, the time freezes and we localize the whole network using a snapshot. There is no movement while the nodes are localizing. This means that message transfer is instantaneous and that the received anchor locations are still accurate when a node receives them. As such, the simulation results represent a best-case scenario where no inaccuracy is introduced due to ongoing movement, communication delays, message loss or collisions, or other anchor-location inaccuracies (i.e. GPS error). As in [9], we use a modified random waypoint mobility model [16] where each node can vary its speed at each time step before it reaches its destination. The pause time is set to 0 and the minimum node speed is set to 0.1 to avoid speed decay [17]. The average node speed is close to $v_{max}/2$. In the following, speed is expressed as a multiple of the radio range r. Finally, as suggested in [9], we use a sample set of 50 location estimations.

To analyze the simulation results, we use the following metrics. First, we analyze the localization error. As done in [9], the localization error is calculated by measuring the distance between the real location of a node and its estimated location. Second, we consider the coverage of the different algorithms, that is the percentage of nodes that were able to calculate a location estimate. Third, we compare the processing times necessary for purely running each algorithm, thus excluding potential communications to gather anchor locations. Detailed network characteristics as well as other simulation results can be found in [1].

4.1 Localization Error

Figure 2 shows the localization error for all nodes, including the non-localized nodes, that is the nodes that are placed in the middle of the deployment area because they were not able to compute a location estimate (see [1] for more simulation results). Nodes can be non-localized for several reasons. First, they heard no anchor. This is typically the case with Centroid as it cannot produce a location estimate if no anchor is heard. In the case of MCL, this can happen at the beginning of the deployment when there is no previous sample set to build from. Second, in the case of MCL, a node that has heard anchors can sometimes still be non-localized. This happens when the algorithm is not able to fill the sample

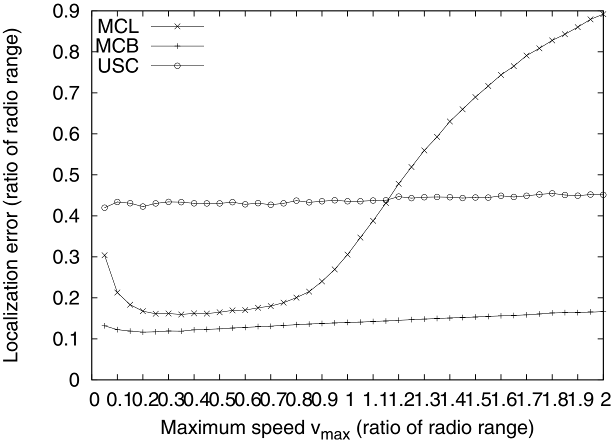


Fig. 2. Localization error (including the non-localized nodes)

set rapidly enough: the maximum number of random draws has been reached and the new sample set is still empty. This can be the case when the region to draw from is large and the area where the anchors' radio range overlap is small in comparison. Not being able to localize a node with anchors can also happen when the sample set becomes empty for some inconsistency reasons. Inconsistencies in a node's sample set generally occur after a period during which the node has heard no anchor. The new location estimations produced recursively from the old sample set gradually become less accurate as time passes and still no anchor is heard. Once an anchor is heard again, it can occur that all the new samples are rejected because they do not meet connectivity and speed constraints. Not being able to localize a node when anchors were heard is clearly unacceptable as it leads only to wasting energy and should be prevented as much as possible.

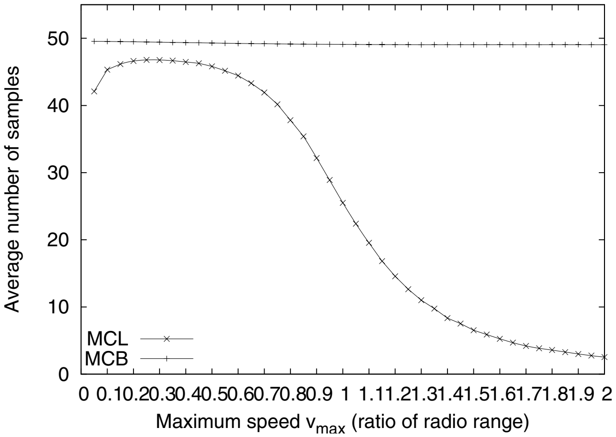


Fig. 3. Average number of samples

Figure 2 shows that MCL is rather sensitive to slow and high speeds while the curve for medium speeds, i.e. for a node moving at maximum between 20% and 70% of the radio range during one time interval, remains rather flat. The localization error for MCB as well as that of Centroid (USC) are rather independent of the node maximum speed and only show a slight deterioration of the accuracy as the maximum speed increases.

The behavior of the MCL localization error with respect to the maximum speed has several causes. Slow motion gives less chances to a node to hear anchors. More precisely, the average number of anchors heard remains quite stable as the speed varies, however, the time a node can remain anchor-less is on average longer. The reason for MCL's loss in accuracy for slow speeds is thus as follows: the slower an anchor-less node moves, the less chance it has to encounter a new anchor quickly since the whole network moves only in small steps at each time interval.

At slow speeds, nodes are thus more often producing location estimates without being able to use anchor locations. This increases the inaccuracy of the set of samples over time. In the worst case, nodes are localized in the middle of the deployment area if no valid sample can be drawn once an anchor is heard again. This effect is also noticeable in the coverage results (see [1]). The negative effect of slow motion was also observed by Hu and Evans in [9] for MCL.

For larger values of v_{max}, such as 0.8r and above, the motion of the nodes allows them to hear anchors more often and this limits the decay of the sample set. However, since the distance a node can travel in a time unit is larger, the area from which the random samples are drawn also increases. This affects the accuracy in a negative way. While the average number of anchors a node hears remains rather stable and the average number of anchor-less time intervals decreases as the speed increases, the average number of valid samples MCL is able to draw for high speeds considerably decreases. The coverage MCL achieves for high speeds is also decreasing. The impact of the maximum speed on MCL is thus purely due to the way the algorithm produces its samples. Hu and Evans also noticed this increase in inaccuracy in [9]. They did not provide a detailed study of the coverage and average number of samples though.

The behavior of MCB is not as dramatically affected by the maximum node speed as it is the case with MCL. The main reason is that the average number of samples the MCB variants can draw is rather stable with respect to speed as shown in Figure 3. This improves the coverage and the overall accuracy. This behavior is due to the more efficient way MCB draw samples.

4.2 Coverage

We studied the percentage of nodes that could be localized, i.e. for which a location estimate was produced, independently of how many anchors they heard. The coverage of Centroid is on average 96.62%. That of MCL is 92.13% on average, ranging from 96.87% (speed 0.45r) to 86.44% (speed 1r) down to 44.81%

(speed 2r). This comes from the fact that MCL is not able to draw enough good samples from a large draw area in which the overlap of the anchors' radio range is small. This occurs in general with high maximum node speeds. In the worst case, the new sample set remains empty leading to a non-localized node. The average coverage for MCB is extremely stable with respect to maximum node speed and stays around 99.98% (variation starts from the third decimal place).

4.3 Processing Time

Another factor positively influenced by the way MCB draws samples is the processing time, that is to say, the time needed by the algorithm to produce a location estimate. We consider here only the computation time and not the time needed for communicating (gathering anchor positions, listening to neighbors and forwarding anchor messages). Communication time is network-dependent and is identical for all the MCL variants. Only Centroid communicates less as it does not consider the two-hop anchors. We measured the processing time through simulation on a PC.

The processing time of the MCL variants depends on several factors. First, as the maximum number of samples N in the set grows, more samples have to be drawn and processing time also increases. There is of course a trade-off between the maximum number of samples in the set, the accuracy of the localization and the processing time. In [9], Hu and Evans provided an analysis of the impact of the maximum size of the sample set. We obtained similar results with our algorithm though the maximum number of samples can more easily be reduced with MCB than with MCL.

Second, the maximum number of random draws the algorithm is allowed to make also has an influence on the processing time. In the original MCL implementation, the maximum number of draws per sample is set to two times 10,000 draws for 50 samples, once with the maximum speed v_{max} and a second time with the relaxed maximum speed $v_{max} + delta$. More precisely, MCL allows two times 10,000 draws with an uninitialized sample set, and two times 200 draws per sample (maximum 50 samples) with an initialized sample set. With similar loop values, MCB is 40% to 50% faster while its accuracy and coverage are better than that of MCL and its average number of samples was higher (i.e. the sample set was full more often). These tests and those that follow were conducted using a 200 by 200 units deployment area with one unknown node and 32 anchors. The radio range was set to 50 units and the maximum speed to 50 units per time interval.

Thanks to its simplicity, Centroid performs much faster than any MCL variants. It runs in 0.0095% of the time needed by MCL and 0.0168% of the time needed by MCB when both MCL and MCB are using the original random-draw parameters (10,000 draws with an uninitialized sample set, 200 draws with an initialized sample set, 50 samples in the set at maximum).

Next, we compared MCL and MCB processing times for identical localization error when excluding the non-localized nodes. To get an identical accuracy, we varied the loop boundaries of both MCL and MCB. We kept the maximum number of samples unchanged (50 samples). In the case of MCL, we let the algorithm draw samples 1,000,000 times, once with the maximum speed v_{max}, and a second time with the relaxed maximum speed $v_{max} + delta$. For an initialized sample set, we allow 20,000 tries twice for each of the 50 samples. With MCB, we use a maximum of 100 draws for an uninitialized sample set. For an initialized sample set, MCB uses 50 draws from the sample box and, in the case of a partially full new sample set, it allows at maximum 100 extra draws from solely the anchor box. At speed 1r, MCL was able to produce 48.7227 samples on average and MCB 48.2153. The accuracy was 13.9% of the radio range for MCL and 13.8% for MCB. The coverage was 98.38% for MCL and 99.96% for MCB. The relative processing time was 100% for MCL and 6.238% for MCB. This demonstrates the fact that MCB is much faster than MCL for a similar localization accuracy. Even with a slightly lower average number of samples, the coverage of MCB is better than that of MCL.

5 Conclusion and Future Work

Localization in wireless sensor networks is a topic that has received much interest in the past years. Most proposed algorithms concentrate on static networks of sensors with either static or mobile anchors. The problem of localizing nodes in a mobile wireless sensor network has not yet received much attention although mobility needs to be taken into account at design time.

In this paper, we presented a localization algorithm that builds upon Hu and Evans' findings [9] and that makes Monte Carlo Localization more lightweight for use in wireless sensor networks. By making better use of the information a node gathers (one-hop and two-hop anchors) and by restricting the area a node has to draw samples from to a (small) box, we improve the whole process of localizing. The results of simulations of our algorithm, called Monte Carlo Localization Boxed, show that it allows a node to get an improved accuracy at a reduced cost. Most importantly, it ensures that a node having heard anchors will be localized and it will not pay a high price in term of processing time and energy expenditure because of the inefficiency of the localization algorithm (random draws). Our simulation results also show that the overall coverage of the localization algorithm is improved by ensuring that the sample sets are full as often as possible.

In the future, we are planning to deploy MCB on a test network of wireless sensors and study the behavior of the algorithm in a real-life setting. We will also make several extensions to the protocol so that it can benefit from extra information on the sensors' mobility patterns and mobility-pattern variability. This encompasses maintaining knowledge about sensors' speed and direction, possibly using additional equipment such as accelerometers and deploying it in heterogeneous networks using a mix of both mobile and static anchors.

References

1. A. Baggio. Monte-Carlo localization for mobile wireless sensor networks. Technical Report PDS-2006-004, Delft University of Technology, June 2006.
2. F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA99)*, Detroit, Michigan, USA, may 1999.
3. B. Dil, S. Dulman, and P. J. M. Havinga. Range-based localization in mobile sensor networks. In *Third European Workshop on Wireless Sensor Networks*, volume 3868 of *Lecture Notes in Computer Science*, pages 164–179, Zurich, Switzerland, Feb. 2006. Springer.
4. A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
5. P. Dutta and S. Bergbreiter. Mobiloc: Mobility enhanced localization, Dec. 2003.
6. K. Fall. A delay-tolerant network architecture for challenged Internets. In *ACM SIGCOMM*, pages 27–34, Karlsruhe, Germany, Aug. 2003.
7. A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem. Distributed online localization in sensor networks using a moving target. In *Third international symposium on Information processing in sensor networks (IPSN)*, pages 61–70, Berkeley, California, USA, Apr. 2004.
8. J. E. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 4(6):555–563, July 1970.
9. L. Hu and D. Evans. Localization for mobile sensor networks. In *Tenth International Conference on Mobile Computing and Networking (MobiCom'04)*, pages 45–57, Philadelphia, Pennsylvania, USA, Sept. 2004.
10. D. E. N. Bulusu, J. Heidenmann. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.
11. P. N. Pathirana, N. Bulusu, A. V. Savkin, and S. K. Jha. Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Transactions on Mobile Computing*, 4(3):285–296, May–June 2005.
12. R. Peng and M. L. Sichitiu. Localization of wireless sensor networks with a mobile beacon. In *First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, Fort Lauderdale, FL, USA, Oct. 2004.
13. N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *INFOCOM 2005*, Miami, FL, USA, Mar. 2005.
14. K.-F. Ssu, C.-H. Ou, and H. C. Jiau. Localization with mobile anchor points in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, pages 1187–1197, May 2005.
15. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1–2):99–141, May 2001.
16. J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *IEEE INFOCOM*, San Franciso, CA, USA, March–April 2003.
17. J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *ACM MobiCom*, pages 205–216, San Diego, CA, USA, Sept. 2003.

Novel Sink-Oriented Approach for Efficient Sink Mobility Support in Wireless Sensor Networks

Jeongsik In[1], Jae-Wan Kim[1], Kwang-il Hwang[1],
Jihoon Lee[2], and Doo-Seop Eom[1]

[1] School of Electrical Engineering, Korea University, 1, 5-ka, Anam-dong,
Sungbuk-ku, Seoul 136-701, South Korea
{windie, kuzzang, brightday, eomds}@final.korea.ac.kr
[2] i-Networking Lab, Samsung Advanced Institute of Technology, San 14-1,
Nongseo-Ri, Kiheung-Eup, Yongin, Kyungki-Do 449-712 Korea
vincent.lee@samsung.com

Abstract. Mobile sinks pose several challenges for network protocol design. While moving, a sink should continuously update its topological position information in the sensor nodes to maintain paths. This may require large signaling overhead, resulting in excessive energy consumption. Various schemes have been proposed to reduce the path management overhead. In many of these schemes, the sinks only maintain paths from active sources to reduce the overhead. While reducing the path management overhead, this approach introduces other problems. In this paper, a novel path management scheme, Net Cast Routing (NCR), is proposed. NCR provides ceaseless connection from every sensor node to every sink in an efficient manner. Since it does not distinguish active sources from other sensor nodes, its performance is not affected by the number or movement of the sensing targets. In addition, unlike multicast based schemes [2,3,4], NCR does not require any subscription procedure to active sources.

Keywords: sensor networks, routing, mobility.

1 Introduction

Wireless sensor networks (WSNs) are a promising research area, encompassing a wide range of applications. WSNs are typically comprised of large numbers of small nodes with sensing, processing, and communication capabilities. The nodes sense the environment and cooperate with each other to respond to users' queries. The sensing targets, i.e. stimuli, of WSNs can be various in characteristics. The network and application models are diverse. Sinks (nodes which gather sensing data) may be stationary or moving while receiving data or waiting for a query response.

Mobile sink can extend the application area of WSNs significantly. Several schemes [1,2,3,4,5] have been proposed for efficient sink mobility support in previous work. These schemes have advantages in some scenarios, and disadvantages in others, as discussed in Sect. 2.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 329–340, 2006.

In this paper, a novel sink mobility supporting scheme, named Net Cast Routing (NCR), is proposed. NCR is a sink-oriented scheme, in the sense that path setup and maintenance is performed on per sink basis, ignoring individual sources. It does not distinguish sources from the other sensor nodes. Thus does not require per source overhead. That is, the increase in the number of sources or movement of a stimulus does not incur additional overhead. In addition, every sensor is always connected to the path to the sinks. Thus any sensor node can transmit data to its sinks as soon as it begins to generate the data, without requiring any subscription procedure.

A key problem in taking a sink-oriented approach is the excessive per sink overhead required to maintain a path from every sensor node to every sink. NCR adopts two component schemes, the agent-based data gathering scheme and the net casting scheme, to reduce the path maintenance overhead. The agent-based data gathering scheme maintains paths with very low overhead, by not attempting to optimize the paths. However, the resulting inefficient routing paths increase energy consumption in data forwarding, which may cancel the effect of reduced overhead. The net casting scheme is an efficient path optimization scheme, designed to be used with the agent-based data gathering scheme. This exploits the position information and the overhearing capability of the wireless nodes to maximize signaling efficiency. By combining the two schemes, NCR effectively reduces path maintenance overhead, without significantly degrading path optimality, thus reducing overall energy consumption.

In Sect. 2, discussion on related work is presented. In Sect. 3, the proposed scheme is described in detail. In Sect. 4, performance analysis is presented along with simulation results, and Sect. 5 concludes.

2 Related Work

Directed Diffusion (DD) [1] represents early work on data dissemination on WSNs with mobile nodes. It is very flexible for maintaining paths. However, the flexibility is obtained from the periodic data flooding from every source, which can significantly degrade energy efficiency.

In TTDD [2], a source places the information about its data on the *dissemination node*s, which is selected among the sensor nodes to form a grid structure. A mobile sink can establish a data receiving path from the source through the nearest dissemination node. In SEAD [3] and ONMST [4], a multicast tree is constructed for each source to multicast the data to each sink. The tree is reconfigured as the source or the sinks are moving. The approach the above three schemes follow can be entitled *source-oriented multicast* approach. In this approach, a sink should subscribe to the source to join the multicast tree of the source. The information about the available sources is provided by dissemination nodes [2] or by separate index servers [3,4,6]. This approach has two major disadvantages. First, sinks are required to query dissemination nodes or index nodes periodically, even when they are not moving. There is no other way for the sinks to be aware of every occurrence of a new source. Second, path management overhead increases as the number of sources increases or the stimuli frequently change their positions. In contrast, the proposed scheme is not affected by the number or the changes of sources.

Hybrid Learning-Enforced Time Domain Routing (HLETDR) [5] does not require subscription procedure. Nor does the overhead increase with the number of sources. However, these merits rely on the important condition that the movements of the sinks should have spatiotemporal patterns which can be learned by the sensor nodes over some time period. This requirement can seriously limit the application of the scheme.

3 Net Cast Routing (NCR)

NCR is designed for efficient sink mobility support in large scale networks. A large number of sensor nodes deployed in a vast area on a 2-dimensional space are considered. Since these nodes communicate through short-range radios, long range data delivery is achieved by multi-hop packet forwarding. In addition, each sensor node is assumed to know its position.

NCR can be separated into two component schemes: the agent-based data gathering scheme, and the net casting scheme. The agent-based data gathering is a low overhead path maintenance scheme. It, however, is not a novel concept. A similar concept, with different terminologies, is adopted by mobile IP, TTDD [2], and SEAD [3], as a part of each protocol. The philosophy is the same: to make a stationary node represent a mobile node to the network, or a part of the network, so that other nodes can transmit packets to the representing stationary node, instead of directly to the mobile node. The position of the mobile node need not be propagated over the entire network, even if the node is moving, since the representing node is stationary. The representing node knows the method to deliver packets to the mobile node. Home agent in mobile IP, primary agent in TTDD, and access node in SEAD are the representing nodes. In our scheme, the representing node is named just an *agent*.

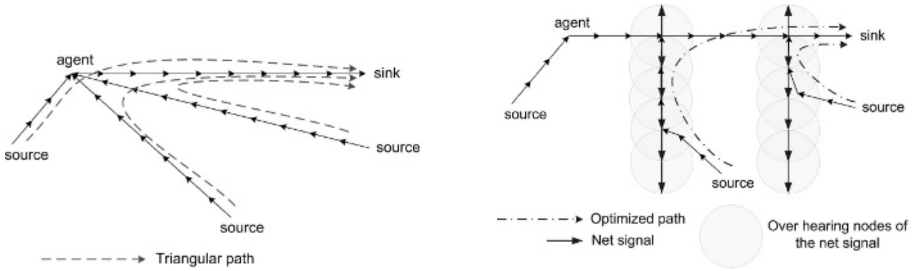
The main disadvantage of the agent-based scheme is degraded routing optimality, known as the triangular routing problem. That is, every packet is delivered to the sink via the agent, even if the source and the sink are very close to each other and the agent is far. The detouring path increases energy consumption by increasing the number of packet forwarding. The paths can be optimized by electing a new agent near the sink and updating the query over the entire network, so that the nodes can forward packets to the new agent. Unfortunately, this process consumes significant node energy by network-wide flooding of signaling packets, and thus cannot be performed frequently. The energy inefficiency due to a detouring path increases with the data rate of the path. Thus, to be feasible for high data rate applications, this scheme should be supported by an efficient route optimization technique.

The net casting is a route optimization scheme to solve the triangular routing problem of the agent-based data gathering scheme. It can effectively reduce the length of severely detouring paths with low signaling overhead, exploiting the geographic position information and overhearing capability of the nodes. The concept of net casting is illustrated in Fig. 1. To simplify the illustration, it is assumed that the nodes are deployed so densely that they appear to be continuous. To shorten the detouring paths, *net* signals are propagated along straight lines across the network. The lines are placed between the agent and the sink. The nodes around the lines become net members by receiving or overhearing the net signal. Net members around one of the lines comprise a net. When a data packet is caught by one of the net members on its way to the agent, it is forwarded along the net to take a shorter path than the original triangular path, as shown in Fig. 1 (b). This can be imagined as a fish swimming towards a light, caught in a net, and then drawn by a fisherman on a ship. The interval between the adjacent nets can be adjusted, in terms of the tradeoff between the amount of signaling overhead and the optimization level. If the sink changes its moving direction, a new net can be cast along a different direction. The direction of a net is selected from a pre-determined set of directions. The number of the possible directions also affects the amount of overhead and the path optimization level. A net is a dynamic entity, which is created and removed as the sink moves. When the sink is moving toward the agent, the nets left behind the sink are removed.

In the following two subsections, the agent-based data gathering scheme and net casting scheme are described in detail.

3.1 Agent-Based Data Gathering

Agent-based data gathering is an overhead reduction scheme for sink mobility support in WSNs. When a sink has a new query, it appoints its closest neighbor as its agent. A neighbor of a node is defined as another node which is within the direct communication range of the node. To find the distance from each neighbor, the sink requests HELLO messages from its neighbors. The sink transmits the query to the agent, so that the agent can flood the network with the query on behalf of the sink. During the flooding process, each node in the network can set up a shortest path to the agent, named an *agent path*, using the classic distance vector algorithm [7]. The query carries the required information for this as well



(a) Triangular routing problem (b) Net casting scheme

Fig. 1. Simplified concept of the net casting scheme

as information for the application layer. When a node has data matching the query, it forwards the data to the agent.

A path from the agent to the sink, named a *relay path*, is maintained in order for the sink to receive data continuously while moving. When the sink moves out of the radio range of its agent, it again selects the closest node among its current neighbors. The sink designates the selected node as its *immediate relay* (*immed-relay*) by transmitting a *relay path setup* (RPSP) message to the node. Then, the message is delivered to the agent along the existing agent path. The relay path is set up along the reverse direction of the path which the RPSP message is transmitted along. A node on a relay path is defined as a *relay.* When a relay has data for the sink, it forwards the data along the relay path.

When the sink moves out of the radio range of the current immed-relay, a new immed-relay is elected by the sink, and a new relay path is set up in the same way as above. When the agent receives the RPSP, it examines whether there is an old relay path for the same sink. If so, the agent transmits a *relay path clear* (RPCL) message along the old relay path. Parts of the new relay path and the old relay path can overlap. To distinguish between the old relay path and the new relay path, each relay path has a sequence number, *rp-seqno.* Every new relay path is given an rp-seqno one greater than that of the previous relay path, by the sink. When a relay receives a new RPSP message, it does not remove the state for the old relay path. The state for the old relay path is maintained until the relay receives an RPCL for the path. RPSP and RPCL messages include the rp-seqno of the relay path which they are creating or removing.

3.2 Net Casting

The purpose of net casting is to mitigate the triangular routing problem, induced by agent-based data gathering. To describe the net casting procedure, an analogy can be used, considering a net as a straight line, named a *net line.* Then, creating a net can be considered as drawing a straight line across the sensor field. A net line can take one direction selected from a set of pre-defined directions. Along with the average interval between the adjacent net lines, the number of directions of the net lines is a parameter determining the trade-off point between the amount of the overhead and the path optimization level. One-directional case is described below, and multi-directional cases can easily be extended from this description.

One Directional Case. In one directional case, one coordinate axis is chosen by the sink so that the sensor field is divided into stripes, as illustrated in Fig. 2. The widths of the stripes are identical and defined as a *net-space.* A stripe in Fig. 2 is a set of positions that have the same *index.* The index of a position is defined as $index = \lfloor coordinate/net\text{-}space + 0.5 \rfloor$, where *coordinate* is the coordinate of the position with respect to the axis. Every node has an index according to its position. The sink attempts to draw one net line on each stripe between itself and the agent, except on the stripe with index 0. The axis is determined when the first net line is drawn. When the first net is not yet

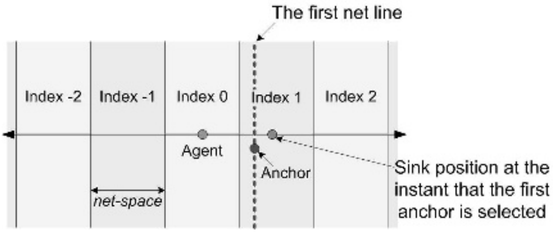
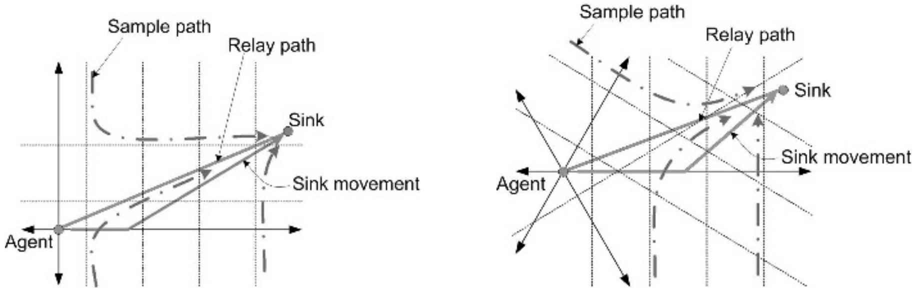


Fig. 2. Selection of the first net line in the one directional case

drawn, the sink takes a temporary axis to determine the index of the immed-relay whenever a new immed-relay is selected. The temporary axis is chosen so that the axis passes through the positions of the agent and the sink. The agent becomes the origin of the axis and the direction of the sink is defined as the positive direction. If the index of the new immed-relay is 0, the immed-relay is considered to be sufficiently close to the agent. Thus, no net line has to be drawn and the temporary axis is discarded. If the index is greater than 0, the first net line is drawn passing through the immed-relay position, perpendicular to the axis, as illustrated in Fig. 2. At the same time, the temporary axis becomes permanent. The indexes of nodes are computed with respect to this permanent axis. A new net line is drawn whenever an absolute value of the index of a new immed-relay is greater than that of the previous immed-relay, or the sign of the index of the immed-relay is changed. Thus new net lines are drawn as the sink moves away from the agent. The average space between two adjacent net lines is determined by the net-space. A net line always passes through the corresponding immed-relay and is perpendicular to the axis.

Multi-directional Case. In a multi-directional case, multiple axes are used. Naturally, the number of the axes is the same as the number of the possible directions. A node has multiple indexes each corresponding to each axis. Net creations for different directions are performed independently of each other. Thus, decisions for creating a net in a multi-directional case can be made in the same way as in one directional case, if the axes for the net lines are defined. In a multi-directional case, the directions of the axes are determined so that the angles between any two adjacent axes are identical. The origin is at the agent position. Thus, if one of the axes is determined, the others are also uniquely determined. The sink chooses an axis when it creates the first net line in the same way as in one directional case. Then, the other axes can be determined uniquely depending on the total number of the axes. The set of possible directions is not changed until a new agent is elected. When a new agent is elected, the relay path and all the nets of the sink are removed as the query is propagated over the network. Examples of net lines of two directional and three directional cases are illustrated in Fig. 3, along with some sample paths.



(a) 2 directional case

(b) 3 directional case

Fig. 3. Simplified concept of the net casting scheme

Net Cast. The net line is a virtual entity, which does not physically exist. Creation, or cast, of a real net is initiated by a new immed-relay on a request from the sink when the immed-relay is elected. The net is created according to the calculated position of the net line. The sink examines, for each axis, whether a new net is required to be cast whenever a new immed-relay is elected. If the sink decides new nets are required, it includes the information of the new nets to be created in the RPSP message to the new immed-relay. The immed-relay calculates the positions of the required net lines based on its position and the information included in the RPSP message. The RPSP includes the following information in addition to the information required for relay path setup: a list of directions of the net lines to be created, a unique *net-id* for each net, and indexes of the new immed-relay. In this information, only the directions of the nets are required to determine the positions of the nets. The net-id and indexes are later used to maintain the nets.

A net is composed of an anchor, knots, and regular net members. The anchor initiates the creation and destruction of the net. Thus, the anchor of any newly created net is always the immed-relay initiating creation of the net. The anchor can be changed later as the relay path changes. A net is created by propagating a *net cast* (NCST) message along the vicinity of the corresponding net line. A node receiving the message becomes a knot and forwards the message for propagation. For convenience, the anchor is referred to as a kind of knot. A node which overhears the message becomes a regular net member. The NCST message forwarding rule is as follows. The node, having the message, projects itself on the net line, and moves the projection along the line towards the desired direction by a pre-determined distance. The node forwards the message to the node closest to the moved projection. If no node is closer to the projection, the node discontinues forwarding of the message. During NCST propagation, paths from the net members to the anchor, named *net paths*, are set up along the reverse path of NCST. The anchor plays the role of a gateway from the net path to the relay path.

Net Management. A net is owned by one of the relays. The relay which is the owner of a net is either the anchor or a neighbor of the anchor of the net. A net which cannot be owned by any relay will be destroyed. When a node becomes a relay, i.e. by receiving an RPSP message, it examines whether it is a member of an existing net. If so, it becomes owner of the net as follows: If it is the anchor of the net, it continues to be the anchor. Otherwise, if it is a knot, it becomes a new anchor for the net. If it is a regular net member, it elects a new anchor among its neighboring knots, by transmitting an *elect anchor* (EANC) message. Note that a regular net member has at least one knot of the same net within its radio range. In the event the anchor is changed, the new anchor transmits a *net refresh* (NREF) message to the old anchor along the existing net path. When receiving or overhearing the NREF message, net members change the net path to the new anchor, and the old anchor resigns the anchor role. To distinguish an old NREF and a new NREF, each NREF message has a sequence number. If a node overhears an NREF message with the same sequence number from two or more knots, it processes the first and ignores the others. Since an NREF message is only generated when a new relay path is setup, the rp-seqno of the relay path can be used for this purpose. Note that the net-id is not changed until the net is removed. However, the rp-seqno of a net member is changed whenever it is receives a new NREF message. It is possible that two or more relays could be members of the same net. In this case, it should be made impossible for two or more relays own the same net. It is desirable that the relay closest to the sink should own the relay. To guarantee this, a relay includes the list of the nets it owns in the RPSP message. When a node becomes a relay, it does not attempt to own the nets included in the list.

Net Destruction. Each individual net is cleared independently of each other as when created. When a relay resigns the relay role, it transmits a *dismiss anchor* (DANC) message to the anchor of each net it owns. The message includes the rp-seqno of the old relay path and the net-id of the net to be cleared. If the relay itself is the anchor of the net, it processes the message instead of transmitting it. It is explained that before an old relay path is cleared, a new relay path is set up, and the nets intersecting the new relay path are refreshed by the new owner relays. The refreshed net members are given the rp-seqno of the new relay path. The rp-seqno of the same net can be different for each member, because a part of the net (i.e. from the old anchor to the new anchor) is refreshed. However, the current anchor always has the latest rp-seqno. When a node receives a DANC, if it is not the anchor for the net indicated by the message any more or if its rp-seqno is greater (i.e. newer) than that of the message, it ignores the message since the net is owned by a newer relay. Otherwise, it transmits a *net clear* (NCLR) message to each neighboring knot. The message is propagated to each end of the net along the reverse of the net path. A node receiving or overhearing an NCLR, clears the state for the net.

Data Packet Routing. When a node has a packet for some sink, it routes the packet as follows, consulting the entry for the sink in the routing table. If it is a relay, it forwards the packet to the next hop node of the relay path. Otherwise, if it is an anchor for a net, it forwards the packet to the relay owning the net.

Otherwise, if it is a member of a net, it forwards the packet to the next hop node of the net path. In this case, if it belongs to two or more nets, it selects the net with the largest index in absolute value. If it is neither a relay nor a net member, it forwards the packet to the next hop node of the agent path.

3.3 Unreliability Problem of Overhearing

In the proposed scheme, it is assumed that the MAC protocol of the node can detect transmission failure for unicast packets using data/ack exchange. Furthermore, the failed packet can be retransmitted as required. If pre-defined times of retransmissions are failed, the receiving node is considered as failed or out of the transmission range. In NCR, node failure is detected by transmission failure detection, without soft state maintenance requiring periodic refreshment signal. Thus, unicast signaling can be considered as reliable, provided that the receiver is working and is within range of the transmitter. However, signaling using overhearing is inherently unreliable. In NCR, regular net members overhear NCST, NREF, and NCLR messages. If some nodes failed in overhearing an NCST or an NREF, the routing paths including the nodes may become longer than they should be. It is not a critical problem. However, the effect of missing an NCLR message can be critical. A node failing in overhearing an NCLR message would attempt to forward packets along a net path, even though the net does not exist. The receiving node will attempt to forward packet along the agent path. This can create a routing loop. The key to the solution is the fact that a regular net member always forwards data packets to a knot and knots always have correct information of the net. When a node forwards a data packet along a net path, it sets the flag in the packet header, indicating that the packet is forwarded along the net path and includes the corresponding net-id. If the receiving node is not a knot of the net specified by the packet, it transmits a *net remove* (NREM) message to the sending node. On receiving or overhearing the NREM message, a node removes the information of the net specified in the message. The NREM message is not forwarded to other nodes.

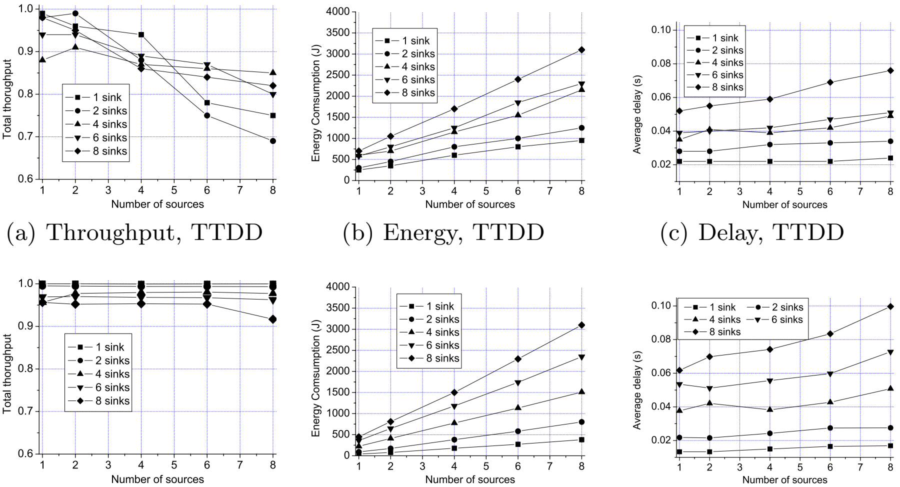
4 Performance Analysis

In this section, the performance of NCR is examined by simulation experiments, using Network Simulator 2 (NS2). The performance metric is throughput, delay, energy consumption, and overhead. In the experiments, default wireless communication modules of NS 2.27 are used with their default parameters. 802.11 DCF is used as the underlying MAC protocol.

First, the performance of NCR is compared with that of TTDD, to estimate the overall relative performance of NCR for various numbers of stationary sources and moving sinks. Figure 4 presents the performances of TTDD and NCR in the same scenario. The scenario and parameters are the same as those used in [2]. The important parameters are as follows: 200 sensor nodes are deployed on a $2\times2\text{km}^2$ field. The transmission rate of the physical link is 1 Mbps; source data rate is 1 packet/sec; data packet size is 64 bytes; control packet size is 36 bytes; and the sink movement follows the standard random Way-point model with maximum speed 10m/s and pause time 5s. The net-space value is set to 250m and the number of directions is set to 3, heuristically. The corresponding graphs are shown in the same scale to facilitate the comparison.

Significant differences can be observed in the throughputs. We observed that the throughput is dominated by the handoff scheme between the old immed-relay and the new immed-relay. This is because there is negligible congestion in the scenario used. Furthermore, a lost packet is retransmitted up to 4 times when using NS2 default parameters. In our implementation, the distance between the sink and the current immed-relay is used for handoff decisions. The sink examines its distance from the current immed-relay once per second and initiates a new relay path setup if the distance is greater than 90% of the nominal radio range. Using this scheme, the sink can set up a new relay path, with high probability, before it moves out of the radio range of the old immed-relay. Thus all the major sources of packet loss is removed. This scheme may not always be applicable in practice because of irregular channel changes in space and time. The overhead required for periodic examination of the sink position is not considered either. Nevertheless, the results in this paper are valid since the detailed handoff decision scheme is independent of the proposed routing scheme. The energy consumption and delay performance shows that the performance of NCR is comparable to that of TTDD for a small number of stationary sources.

Figure 5 shows the impact of the number and the speed of the sinks. In this experiment, 2000 sensor nodes are deployed on a 5×5km^2 field. One of the sensor nodes generates data packets at a rate of 0.1 packet/sec. The simulations are



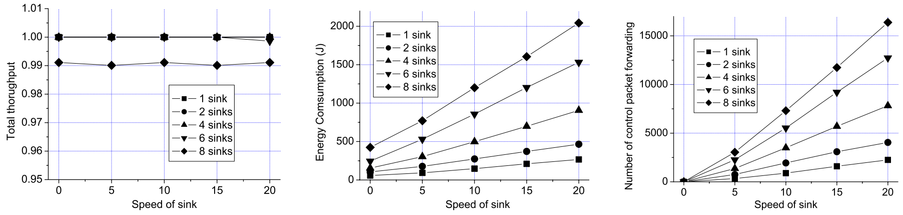
(a) Throughput, TTDD (b) Energy, TTDD (c) Delay, TTDD

(d) Throughput, NCR (e) Energy, NCR (f) Delay, NCR

Fig. 4. Performance comparison between TTDD and NCR

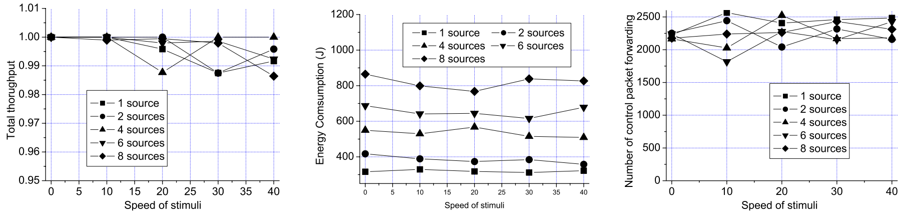
performed for 400 seconds and each result is averaged over 10 random topologies. The number and the speed of the sink vary, as shown in the graphs. A sink moves toward random destination point at a constant speed. On arriving at the destination point, it selects another random destination point and continues to move. The other parameters are the same as those in the previous scenario. The results show that energy consumption and overhead increase with the number and the speed of the sinks. There is no overhead when the sink is not moving. The throughput and delay does not significantly change with the sink number and speed. High throughput in a relatively high sink speed (20m/s) can be obtained by frequent handoff decision, i.e. once per second, as explained above. If handoff decision overhead is taken into account, this performance may not be obtained. The result show that NCR might provide high throughput with ideal handoff scheme.

Figure 6 shows the impact of the number and the speed of the stimuli. A number of stimuli are moving around the field at a constant speed. The number and the speed of the stimuli vary as shown in the graphs. The existence of a stimulus is detected by the sensor node closest to the stimulus, and reported to the sink moving at a speed of 20m/s, once every second. This models the cooperative detection of the target. However, details of the cooperation are not modeled, since they are independent of the routing scheme. The throughput, delay, and the overhead do not change with the number and speed of stimuli. Only the energy consumption increases with the number of stimuli due to the increased total source data rate.



(a) Throughput (b) Energy (c) Signaling

Fig. 5. Impact of the number and the speed of the sink



(a) Throughput (b) Energy (c) Signaling

Fig. 6. Impact of the number and the speed of the stimuli

5 Conclusion

In this paper, we have described NCR, a routing algorithm for large scale WSNs with mobile sinks. By combining low overhead path maintenance scheme and efficient route optimization scheme, NCR can efficiently provide ceaseless data forwarding paths for every sensor node, without severely detouring paths. In addition, NCR is not affected by the characteristics (number, distributed area, speed, and duration) of stimuli. These merits enable NCR to cover a wide range of applications.

Acknowledgments. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. Mobile Computing and Networks, 2000, pp. 56-67.
2. Ye, F.,Luo, H., Cheng, J. Lu, S., Zhang, L.: A two-tier data dissemination model for large-scale wireless sensor networks. Mobile Computing and Networks, 2002, pp. 148–159.
3. Kim, H., Abdelzaher, T., Kwon, W.: Minimum-Energe Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks. Sensys, 2003, pp.193–204.
4. Zhang, W., Cao, G., Porta, T.: Dynamic Proxy Tree-Based Data Dissemination Schemes for Wireless Sensor Networks. IEEE International Conference on Mobile Ad-hoc Sensor System, 2004, pp. 21–30.
5. Baruah, P., Urgaonkar, R., Krishnamachari, B.: Learning-Enforced Time Domain Routing to Mobile Sinks in Wireless Sensor Fields. IEEE International Conference on Local Computer Networks, 2004, pp.525–532.
6. Zhang, W., Cao, G., La Porta, T.: Data Dissemination with Ring-Based Index for Sensor Networks. IEEE international conference on Network Protocol, 2003, pp. 305-314.
7. Perkins, C., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. ACM SIGCOMM, August 1994, pp. 234-244.

Reduction of Signaling Cost and Handoff Latency with VMAPs in HMIPv6*

Jongpil Jeong, Min Young Chung**, and Hyunseung Choo

Intelligent HCI Convergence Research Center
Sungkyunkwan University
440-746, Suwon, Korea +82-31-290-7145
{jpjeong, mychung, choo}@ece.skku.ac.kr

Abstract. In this paper, we propose cost-reduced binding update scheme (CRBU) which further reduces signaling traffic for location updates by employing virtual mobility anchor point (VMAP) on top of overlapped MAP in Hierarchical Mobile IPv6 (HMIPv6). This proposed scheme significantly improves performance compared to HMIPv6, in terms of binding update rate per user and average handoff latency. Also it makes the mobile nodes (MNs) moving around the boundary access routers (ARs) of adjacent MAPs and they become to move within a VMAP. It is certain that our scheme uses the network resources efficiently by the removal of global binding updates for MNs in boundary ARs inside MAPs. And we explain an analytic model for performance analysis of HMIPv6 networks, which is based on the random walk mobility model. Based on this analytic model, we formulate binding update cost and analyze it as the probability that an MN stays in the current cell (q). As a result, our proposed scheme can greatly reduce the packet loss and delay by eliminating Inter-MAP handoff.

1 Introduction

MIPv6 [1,2] is the main protocol supporting IP mobility. This protocol provides connectivity with the Internet from the MN's movement. However, MIPv6 has weak points, such as handoff latency resulting from movement detections, IP address configurations and location updates which are unacceptable in real-time application. The purpose of HMIPv6 [3] is to reduce the amount of signaling to correspondent nodes (CNs) and the home agent (HA). However, this does not satisfy the requirements of real-time services which are susceptible to delay because HMIPv6 also uses MIPv6 for Inter-MAP handoff. Furthermore, HMIPv6 is managed by several MAPs to solve the Single Point of Failure (SPOF) and bottleneck state of traffic. Thus, Inter-MAP handoff is increasingly expanding.

In this paper, the Inter-MAP handoff scheme is proposed in order to improve HMIPv6 performance. This scheme is based on a virtual layer employing

* This research was supported by Ministry of Information and Communication, Korea under ITRC IITA-2005-(C1090-0501-0019).

** Corresponding author.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 341–352, 2006.

the VMAP (virtual MAP). The virtual layer consists of virtual MAPs, each of which is managed by a VMAP. This proposed scheme significantly improves performance compared to HMIPv6, for location update rate per users. The proposed scheme allows the MNs moving around the boundary ARs of adjacent MAP's to move within either a VMAP or an overlapped region. This greatly reduces the packet loss and delay, due to the Inter-MAP handoff not occurring.

This paper is organized as follows. In Section 2, we review the analytic models for location update, the basic operation of HMIPv6 and its MAP discovery. The motivation of this work and the new proposed scheme are presented in Section 3 based on HMIPv6. In Section 4, the performance of the proposed scheme is evaluated. Finally, this paper is concluded in Section 5.

2 Related Works

This section provides a brief overview of analytic approaches for performance improvement of mobile network. Xie *et al.* propose an analytic model for regional registration [4], which is a derivative of a hierarchical mobility management scheme [5]. The proposed analytic model focuses on the determination of the optimal size of regional networks, given the average location update and packet delivery costs. In this study, the existence of one-level regional networks is assumed where there is a single Gateway Foreign Agent (GFA). In addition, Woo proposed an analytic model, in order to investigate the performance of Mobile IP regional registration [6]. In [6], Woo measured registration delay and CPU processing overhead loaded on the mobility agents to support regional registration. Although this model is a well-defined analytic model, it is based on MIPv4 and not MIPv6. Furthermore, in this study, a spatial-oriented Internet architecture is used for performance analysis. Currently, the Internet is based on the spatial-oriented location area model, which specifies that the distance between two end points situated on the Internet is unrelated to the geographic locations of these two points. However, the ARs used in next-generation wireless and mobile networks may utilize a cellular architecture, in order to maximize utilization of the limited radio resources [7]. Therefore, it is more appropriate to analyze network performance in the context of IP-based cellular networks. Recently, in [11], Jeong *et al.* show that the total handoff latency of Intra-/Inter-MAP handoff is not occurred by removing the ping-pong effect of the Inter-MAP handoff using the virtual MAP.

HMIPv6 [3] is the extension of MIPv6 and IPv6 neighbor discovery protocols to support local mobility. Therefore, the introduction of a hierarchy only makes sense if the MAP is located between the MN and HA/CNs. It also reduces the signaling traffic for handoff due to transmitting a binding update (BU) regardless of the number of CNs. MN receives the Router Advertisement (RA) including the MAP information from AR. This creates the Regional Care-of-Address (RCoA) and On-link Care-of-Address (LCoA) with received MAP and AR's subnet prefix, registering them to MAP and HA/CNs. In the case of moving the ARs in the same MAP, MN only registers a changed LCoA to MAP. The MAP intercepts

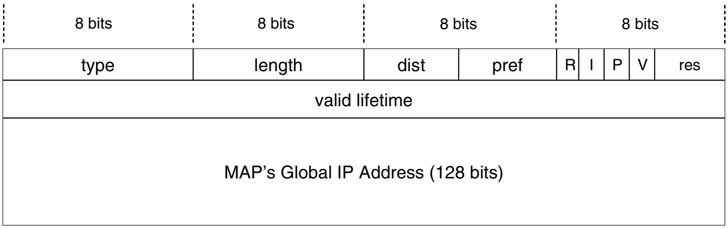


Fig. 1. Message format of MAP Option

all packets directed to MN in MAP, performing the local HA's role that encapsulates and delivers them. In HMIPv6, an MN entering a MAP domain will receive RA message containing information on one or more local MAPs. Then, the MN selects the most suitable MAP by a number of criteria (distance, mobility, preference, *etc.*). However, the question of how to select a MAP is the beyond the scope of this paper. We simply assume that a specific MAP selection scheme is used and that the MN sends a BU message to the selected MAP. The MN can bind its current location (*i.e.*, LCoA) with an address on the MAP's subnet (*i.e.*, RCoA). Acting as a local HA, the MAP will receive all packets on behalf of the MNs and will decapsulate and forward them to the MN's current address. If the MN changes its current address within a local MAP domain, it only needs to register the new address to the MAP. The RCoA does not change as long as the MN moves within the MAP domain. It offers the MN's mobility transparency to the CNs.

The MAP option newly reconfigures for MN to recognize the MAP, to configure the RCoA and to receive the necessary information of basic HMIPv6 operation. The length of preference field is 4-bit and indicates the MAP preference. It also consists of a valid life-time field, a distance field, a global IP address field and several flags. MAP discovery is the procedure of discovering the MAP for ARs and acquiring the MAP subnet prefix. It is possible to pre-configure the routers for MAP options from MAP to MN over the specified interfaces. All ARs in the same MAP are configured for receiving MAP options with the same MAP address.

3 Cost-Reduced Binding Update Scheme (CRBU)

The proposed scheme is configured to receive MAP options from all the adjacent MAPs, for ARs in the boundary area of its MAP. In addition, when an MN is connecting with a boundary area's AR in the overlapped MAP, it should perform MAP switching prior to the Inter-MAP handoff. This MAP switching performs identical procedures to Inter-MAP handoff operation. However, neither packet loss nor additional handoff latency occurs because of MAP switching on receiving the packet from previous MAP. One of important facts motivating the proposed design is that the cost of the location update for HA is much greater than that of MAP. A disadvantage is that since every location request in addition to the

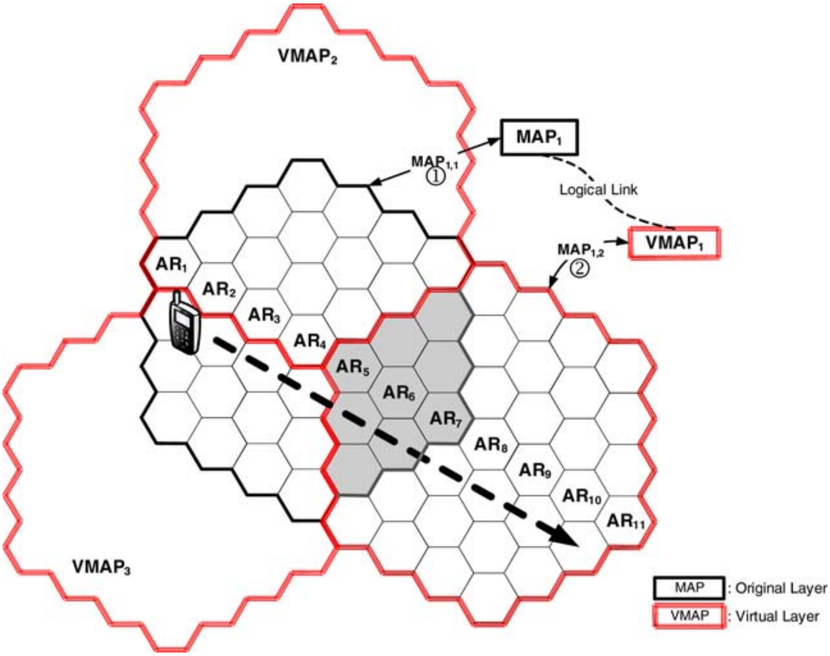


Fig. 2. The proposed virtual layer structure

location registration, is serviced through a HA, in addition to the HA being overloaded with database lookup operation, the traffic on the links leading to the HA is heavy. Therefore, the traffic required for updating HA should be minimized, and the objective adopted in the proposed scheme is to distribute the signaling traffic heading to HA and MAPs.

Our proposed location management scheme introduces the concept of virtual layer as presented in Fig. 2. Layer-2 MAPs represented as bold-faced solid lines are in a virtual layer. This scheme combines three neighboring MAPs as an expanded cluster in the original layer, an expanded MAP. As previously mentioned, each MAP has an associated VMAP. This original layer of MAPs is called Layer-1 and the expanded MAPs overlap each other. It is important to note that another parts of the area exists, using bold-faced dashed lines, which is in a virtual layer and are called in Layer-2. Each MAP of Layer-2 also has a VMAP. Each virtual MAP, which is of equal size, is laid upon the center of the three combined MAPs of each expanded MAP. As a result, the activity of MNs around the boundary ARs of adjacent MAPs increasingly moves within either a virtual MAP or an overlapping region. VMAPs, which manage the original layer, take charge of the management for the entire area. However, MAPs, which manage the virtual layer, take charge of the management for the partial areas. In what follows, $MAP_{i,j}$, the MAP_i of Layer-j, is denoted. The proposed structure effectively avoids the ping-ponging effect, occurring when a mobile user travels

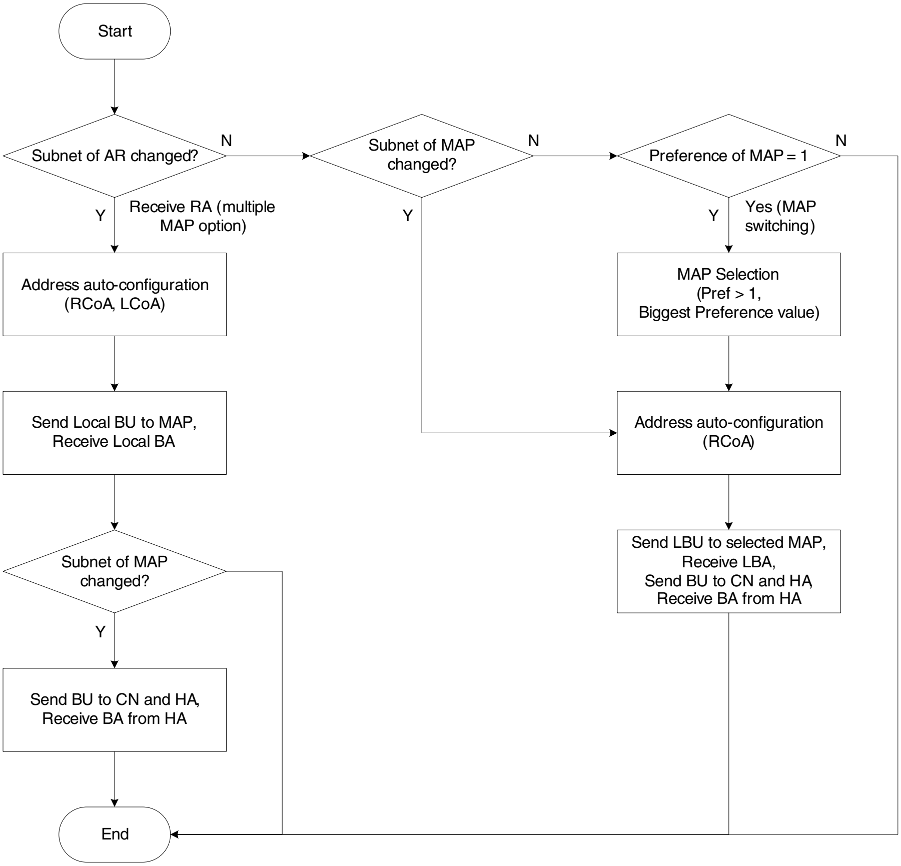

Start
Subnet of AR changed?
N
Y
Receive RA (multiple MAP option)
Address auto-configuration (RCoA, LCoA)
Send Local BU to MAP, Receive Local BA
Subnet of MAP changed?
Y
Send BU to CN and HA, Receive BA from HA
End
Subnet of MAP changed?
N
Y
Preference of MAP = 1
N
Y
Yes (MAP switching)
MAP Selection (Pref > 1, Biggest Preference value)
Address auto-configuration (RCoA)
Send LBU to selected MAP, Receive LBA, Send BU to CN and HA, Receive BA from HA

Fig. 3. Flowchart of MN's operation

along the boundary of two adjacent MAPs and distributes the location update signaling traffic over many ARs using the virtual layer.

The proposed scheme can be implemented by assigning an unique ID to each MAP of Layer-1 and 2. It is important to note that the proposed scheme covers the service area with homogeneous MAPs. The original MAPs are partially overlapped with the MAPs of the virtual layer, and expanded clusters combine three neighboring MAPs in the original layer so that they overlap each other. The ARs of layer-1 used to be entirely in one, two, or three MAPs, and the ARs of layer-2 managed by two or three MAPs. Even though each AR belongs to one, two, or three MAPs, the mobile user in an AR registers with only one MAP at any moment. The preference field of MAP options is used to cache the location information of each ARs in the MAP domain. If these are delivered the preference values to all ARs of the same MAP domain, they have exactly the same value. As AR's own location is close to the boundary AR, the preference value is reduced by 1 and then delivered to MN over the RA. MN recognizes that it arrived at the boundary ARs of MAP, using the preference field value from AR.

If two MAPs' domains overlap other boundary ARs, AR_5, AR_6 and AR_7 receive all the MAP options that are composed of MAP_1 and $VMAP_1$. The preference field of MAP options is used for caching AR's location information in the MAP domain and AR transmits the preference value through Router Advertisement (RA) to MN after decreasing by 1. MN is going to obtain a response that is the value of a preference field, and then obtains boundary ARs. In Fig. 2, when AR_6 is moved to an area of AR_7, Intra-MAP handoff occurs and receives a packet coming from CN through MAP_1 and AR_7 (path 1). AR_7 is located at the outer AR in the overlapping area, so MN is going to turn into its own MAP from MAP_1 to $VMAP_1$. MN transmits Local Binding Update (LBU) to $VMAP_1$, then obtains an Ack, and transmits BU again to HA/CNs. After CN receives BU, it transmits data packets with Ack to MN through AR_7 and $VMAP_1$ (path 2). AR_7 does not initiate Inter-MAP handoff if it does not move to AR_5, because AR_7 is located inside of the domain.

Fig. 4 depicts the message flows for the procedures of improving handoff performance by switching the MAP in advance when the MN moves AR_7 to AR_8. As we mentioned earlier, AR_7 and AR_8 receive all MAP options from MAP_1 and $VMAP_1$, preference value for MAP_1 is 1 and that for $VMAP_1$ is 3. The MN in the MAP_1 is recognized in boundary ARs, and then it selects the new MAP using a MAP selection algorithm. Then, it performs the MAP switching to the selected MAP. Also, since the MN has already switched from the MAP_1 to the $VMAP_1$ when the MN receives the RA moving from AR_7 to AR_8, the MN's handoff is completed by the MN and it sends only local binding update message to the $VMAP_1$.

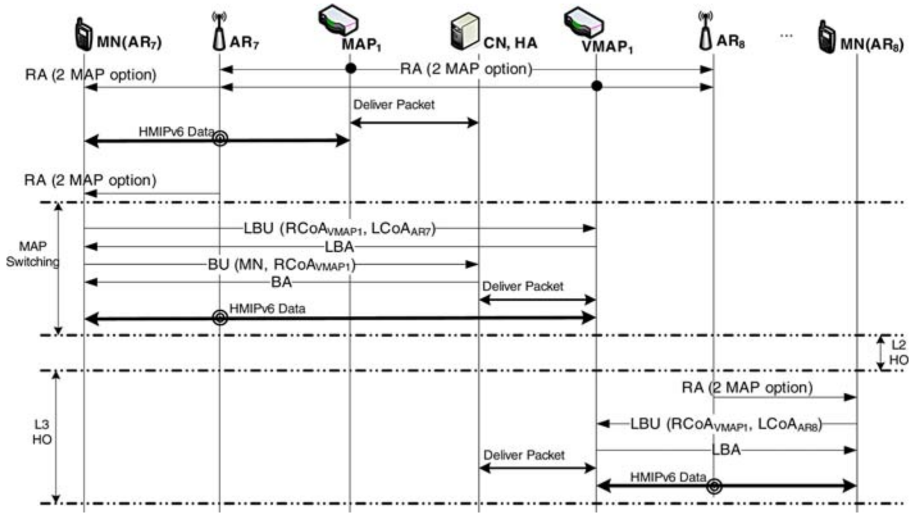


Fig. 4. Message flows

4 Performance Evaluation

This section is organized as two sub-sections. In section 4.1, a formula of average Inter-MAP handoff latency and total average handoff latency is simply presented. In section 4.2, we present an analytic model of HMIPv6 based on cellular networks and analyze the impact of the probability that an MN stays in the current cell (q) on the binding update cost for the proposed scheme.

4.1 Total Average Handoff Latency Comparison

To compare Inter-MAP handoff performance between the proposed scheme and HMIPv6, we consider an one-dimensional network topology with only 11 ARs, *i.e.*, AR_1, ... ,AR_{11} (See Fig. 2). AR_5, AR_6, and AR_7 have received all MAP options from MAP_1 and $VMAP_1$. As already mentioned, if the mobile node moves AR_6 to AR_7, and processes the local binding update to MAP_1 in AR_7, then the mobile node in AR_7 is going to process MAP selection algorithm and MAP switching. Parameters for performance evaluation are defined as $80ms$ for L2 handoff latency for wireless LAN, $20ms$ for receipt of RA for mobility detection, $2ms$ for D_W(wireless part delay), $10ms$ for D_L(wired part delay), and $50 \sim 130$ ms for D_{CN}(delay among MAP and CN). In this case, for performance comparison the following is assumed. First, On-link CoA Test and Return Routability Test are compared. It does not consider operation time for security in local BU and global BU. Second, it does not consider Duplicated Address Detection (DAD) operation in Address Auto-configuration (AA). Third, if CN delivers packets to MN directly without the HA, MN would transmit a binding update to CN first, after checking the local BU. Therefore, if a mobile node does not require the response acknowledgement message, the BA will not return and will transmit data packets with the request. Fourth, CN always transmits packets for moving MN during performance analysis. Fifth, Intra-MAP handoff and MAP switching are always completed during mobile node and stays with a single AR.

In case of moving AR_6 to AR_7, HMIPv6 is $2 \cdot (D_W + D_L)$ for Intra-MAP handoff latency, proposed scheme is identical. In case of moving AR_7 to AR_8, HMIPv6 is $2 \cdot (2D_W + 2D_L + D_{CN})$ for Inter-MAP handoff latency, the proposed scheme is $2 \cdot (D_W + D_L)$ for Intra-MAP handoff latency. Inter-MAP handoff latency is identical with Intra-MAP handoff latency for the proposed scheme, these are always the same values regardless of D_{CN} value. Otherwise, HMIPv6 handoff latency is larger increasing to D_{CN}, because the Inter-MAP handoff latency time should process a local binding update and additional BU to CNs.

The proposed scheme is only able to improve performance of Inter-MAP handoff. Therefore, it is required to grasp all Inter-MAP handoff probability, in order to decide improvement of the proposed scheme exactly. Given the MAP's radius, Inter-MAP handoff probability is obtained through mobility modeling, and the general formula of average Inter-MAP handoff latency is created after multiplying it by the handoff latency value. In addition, in the same manner, the average Intra-MAP handoff latency is obtained, the sum of the two values and the general formula of total average handoff latency are then obtained. For the proposed scheme, Inter-MAP and Intra-MAP handoff latency value are input parameters in this general formula. Average Inter-MAP handoff latency and total average handoff latency can be calculated, comparing these values.

For modeling of border crossing to be used for handoff procedures [8], the following is assumed.

1. MN's moving direction is distributed equally to $[0,2\pi)$ in AR.
2. Residence time is when MN remains in AR, and has negative exponential distribution.
3. All ARs have identical shape and size, form an adjacent area with each other, and are approximately a circular shape.

The border-crossing rate can be calculated as $V_{MAP} = \frac{\pi V}{4R_{MAP}}$ when MN moves away from MAP [9], where V is MN's average movement velocity and R_{MAP} is the radius of the circular MAP. Similarly, the border-crossing rate when MN moves away from AR, is $V_{AR} = \frac{\pi V}{4R_{AR}}$, where R_{AR} is the radius of the circular AR. Accordingly, the MN always passes by AR when it crosses MAP and the rate of crossing only AR's border in the same MAP, is calculated [8] as $V_{AR\_in\_MAP} = V_{AR} - V_{MAP}$. The Inter-MAP handoff probability is $\alpha = \frac{R_{AR}}{R_{MAP}} = AMR$, where AMR is the ratio for the radius of MAP and AR [11]. Similarly, the average Intra-MAP handoff probability is $\beta = \frac{R_{MAP}}{R_{AR}} - 1$.

For the topology shown in Fig. 2, the Average Inter-MAP handoff latency is found by multiplying the average Inter-MAP handoff probability by Inter-MAP handoff latency, and the average Intra-MAP handoff latency is achieved similarly. Therefore, the total average handoff latency (L_{T_Av}) is the sum of average Inter-MAP handoff latency (αL_R) and average Intra-MAP handoff latency (βL_A). Handoff latency for the HMIPv6 and proposed scheme are compared in Table 1.

For HMIPv6, the average Inter-MAP handoff latency increases in proportion to the D_{CN} value, however, the proposed scheme always has the same average Inter-MAP handoff latency for all D_{CN} values. As reviewed earlier, in the case of the Inter-MAP handoff of HMIPv6, the Local BU and Global BU are performed respectively. However, in the case of the proposed scheme, a local BU is always required, this local BU does not change latency according to D_{CN}, which is the distance between MAP and CNs.

AMR's value is the ratio of MAP's radius to AR's radius, actually the average Inter-MAP handoff probability is based on the formulas as mentioned earlier. When the AMR value is close to 1, the Inter-MAP handoff probability is greater

Table 1. Handoff latency comparison for HMIPv6 and proposed schemes

-	HMIPv6	Proposed
αL_R	$AMR(2D_{CN} + 48)$	$24AMR$
βL_A	$24(\frac{1}{AMR} - 1)$	$24(\frac{1}{AMR} - 1)$, $(\varTheta L_R = L_A = 2D_W + 2D_L)$
L_{T_Av}	$AMR(2D_{CN} + 48) + 24(\frac{1}{AMR} - 1)$	$24AMR + 24(\frac{1}{AMR} - 1)$

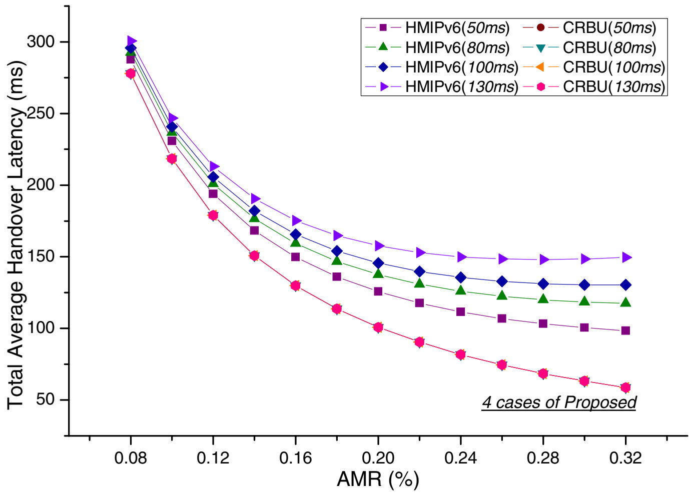


Fig. 5. Total average handoff latency

that that for HMIPv6, and the Inter-MAP handoff latency also increases. Therefore, when the AMR value is close to 1, there is significant Inter-MAP handoff latency improvement in the proposed scheme.

As presented in Fig. 5, when AMR is 0.32, the average handoff latency improve between 40.3% and 55% in accordance with D_{CN}. However, when AMR is 0.08, it is only improved between 5.4% and 9.3%. It rapidly increase the Inter-MAP handoff probability when there are too many ARs in MAP.

4.2 Binding Update Cost Analysis

We assume each domain managed by a MAP has the same number of rings, R. The innermost cell 0 is called the center cell; cells 1 form the first ring around cell 0 and so forth. A ring r is composed of $6r$ cells except the ring 0 with one cell. For network topology with R=3 as shown in Fig. 2, AR_4 is the innermost cell, six cells including AR_3 and AR_5 around AR_4 form the first ring, and the third ring consists of 18 cells including AR_1 and AR_7.

In terms of the user mobility model, we use the random walk mobility model [10] that is appropriate for pedestrian movements. In the random-walk mobility model, the next position of an MN is equal to the previous position plus a random variable whose value is drawn independently from an arbitrary distribution. In addition, an MN moves to another cell area with probability $1 - q$ and stays in the current cell with probability q. In the cellular architecture, if an MN is located in a cell of ring r $(r > 1)$, the probability that a movement will result in an increase $(p^{+}(r))$ or decrease $(p^{-}(r))$ in the distance from the center cell is given by

$$p^{+}(r) = \frac{1}{3} + \frac{1}{6r}, \quad p^{-}(r) = \frac{1}{3} - \frac{1}{6r} \tag{1}$$

We define the state r of a Markovian chain as the distance between the current location of the MN and the center of the location area. This state is equivalent to the index of a ring in which the MN is located. As a result, the MN is said to be in state r if it is currently residing in ring r. The transition probabilities $\delta_{r,r+1}$ and $\kappa_{r,r-1}$ represent the probabilities at which the distance of the MN from the center cell increases and decreases, respectively. They are given as

$$\delta_{r,r+1} = \begin{cases} (1-q) & \text{if, } r = 0 \\ (1-q)(\frac{1}{3} + \frac{1}{6r}) & \text{if, } 1 \leq r \leq R, \end{cases} \tag{2}$$

$$\kappa_{r,r-1} = (1-q)(\frac{1}{3} - \frac{1}{6r}) \quad \text{if, } 1 \leq r \leq R, \tag{3}$$

where q is the probability that an MN stays in the current cell.

Fig. 6 shows a state diagram for random walk mobility model [10]. We denote $\pi_{r,R}$ as the steady-state probability of state r within a MAP domain consisting of R rings.

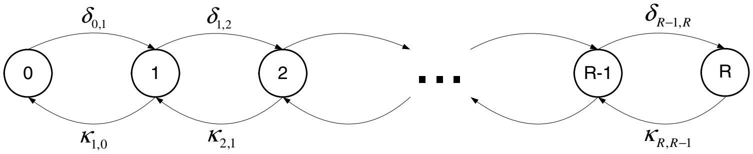


Fig. 6. State diagram for random walk mobility model

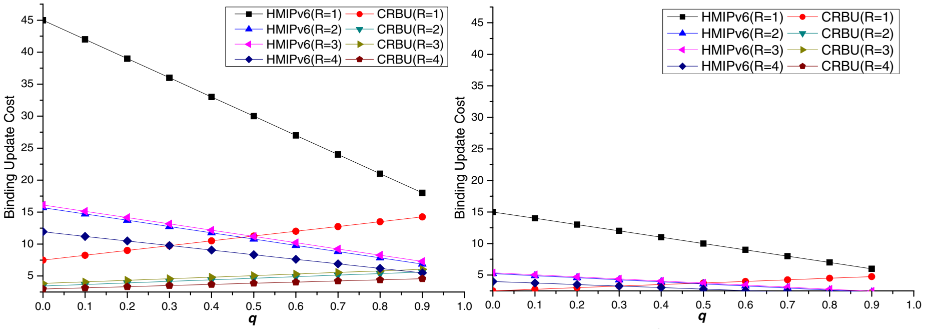
Using the transition probabilities in Equations (2) and (3), $\pi_{r,R}$ can be expressed in terms of the steady state probability $\pi_{0,R}$ as

$$\pi_{r,R} = \pi_{0,R} \prod_{i=0}^{r-1} \frac{\delta_{i,i+1}}{\kappa_{i+1,i}} \quad \text{for } 1 \leq r \leq R. \tag{4}$$

With the requirement $\sum_{r=0}^{R} \pi_{r,R} = 1$, $\pi_{0,R}$ can be expressed by

$$\pi_{0,R} = \frac{1}{1 + \sum_{r=1}^{R} \prod_{i=0}^{r-1} \frac{\delta_{i,i+1}}{\kappa_{i+1,i}}}. \tag{5}$$

In HMIPv6, an MN performs two types of binding update procedure: *global binding update* and *local binding update*. Global binding update is a procedure that an MN registers its RCoA with the CNs and the HA. On the other hand, if an MN changes its current address within a local MAP domain, it only needs to register the new address with the its MAP. Local binding update refers to this registration. C_g and C_l denote the binding update costs in global binding



a) T=1. b) T=3.

Fig. 7. Binding Update Cost for $C_l = 20$ and $C_g = 100$

update and local binding update, respectively. In IP networks, the signaling cost is proportional to the distance between two network entities. Thus, the binding update cost of the global binding update is larger than that of the local binding update. For the simplicity of analysis, we assume that the global and local binding update costs are constants.

Since every ring R is faced with six different ring Rs, the probability that an MN located in a specified ring R moves to the adjacent R rings can be calculated as

$$P_{R,outer} = (1-q)(\frac{1}{3} + \frac{1}{6R}) \quad \text{for } R \geq 1. \tag{6}$$

According to the mobility model, the probability that an MN performs the global binding update is $\pi_{R,R} \cdot P_{R,outer}$. In other cases except this event, the MN performs only the local binding update procedure. Let T be the average cell residence time. Then, the average binding update cost per unit time for HMIPv6 is

$$C^{HMIPv6}_{Binding\_Update} = \frac{\pi_{R,R} \cdot P_{R,outer} \cdot C_g + \pi_{R,R} \cdot (1 - P_{R,outer}) \cdot C_l}{T} \tag{7}$$

The proposed scheme can eliminate the global binding update cost for MNs in boundary ARs of MAPs, because of adapting Inter-MAP dynamic switching method using virtual layers. For the proposed scheme, the average binding update cost per unit time is obtained as

$$C^{CRBU}_{Binding\_Update} = \frac{\pi_{R,R} \cdot (1 - P_{R,outer}) \cdot C_l}{T} \tag{8}$$

Fig. 7 shows the binding update cost as q (the probability that an MN stays in the current cell) is raised. As mentioned above, the cell residence time is the period that an MN stays in a cell area. Thus, as the average cell residence time of an MN increases, the MN performs less movements and the binding update cost per unit time decreases. Therefore, the MN with a large q refer to the static MN. Namely, the MN performs less movements and requires less binding update cost. In Fig. 7, the binding update cost of the ring size of 1 is larger than that of the ring size of 4. This is because an MN located in the MAP domain with small ring size is more likely to perform global binding update procedures.

5 Conclusion

In this paper, an efficient location update scheme employing a partial virtual layer to reduce the update signaling traffic in HMIPv6, is proposed. The system has a two-layer architecture and is configured by homogenous MAP's. Conceptually, the proposed scheme is a combination of grouping, overlapping, and local binding update in MAP. This scheme yields significant performance improvement over the fully virtual layer scheme in terms of the average location update rate per user. Moreover, the new method offers considerable enhancement in utilizing the network resources which otherwise will be wasted by mobile users causing frequent binding update in HMIPv6. The signaling traffic concentrated on boundary ARs in the HMIPv6 is also distributed to many ARs. Also, we modeled binding update cost in HMIPv6 using the random walk mobility model, and analyzed the impact of the probability that an MN stays in the current cell (q) on the binding update cost for HMIPv6 and the our CRBU scheme.

References

1. D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004.
2. I. F. Akyildiz *et al.*, "Mobility Management in Next-Generation Wireless Systems," Proceedings of the IEEE, Vol. 87(8), pp. 1347-1385, August 1999.
3. H. Soliman and K. E1-Malki, "Hierarchical Mobile IPv6 mobility management(HMIPv6)," IETF RFC 4140, August 2005.
4. E. Gustafsson, A. Jonsson, and C. Perkins, "Mobile IPv4 Regional Registration," IETF Internet-Draft, November 2005.
5. J. Xie and I. F. Akyildiz, "A Distributed Dynamic Regional Location Management Scheme for Mobile IP," IEEE Transactions on Mobile Computing, Vol. 1, No. 3, July 2002.
6. M. Woo, "Performance Analysis of Mobile IP Regional Registration," IEICE Transactions on Communications, Vol. E86-B, No. 2, February 2003.
7. S. Pack and Y. Choi, "A Study on Performance of Hierarchical Mobile IPv6 in IP-based Cellular Networks," IEICE Transactions on Communications, Vol. E87-B, No. 3, pp. 462-469, March 2004.
8. Z. D Wu, "An Efficient Method for Benefiting the New Feature of Mobile IPv6," Proceeding of IASTED, pp. 65-70, October 2002.
9. J. H. Schuringa, "Performance Modeling of Location Management Strategies in Mobile Networks," Master Thesis, Dept. of Computer Science, University of Twente, 1995.
10. I. F. Akyildiz and W. Wang, "A Dynamic Location Management Scheme for Next-Generation Multitier PCS Systems," IEEE Transaction on Wireless Communications, Vol. 1, No. 1, January 2002.
11. J. Jeong, M. Y. Chung, and H. Choo, "Reducing Location Update Cost Using Multiple Virtual Layers in HMIPv6," LNCS 4239, pp. 357-367, 2006.

Adaptive Geographically Bound Mobile Agents

K. Tei[1,3], Ch. Sommer[3,5], Y. Fukazawa[1],
S. Honiden[3,2], and P.-L. Garoche[3,4]

[1] Waseda University, Japan
[2] The University of Tokyo, Japan
[3] National Institute of Informatics, Japan
[4] IRIT – ENS Cachan, France
[5] ETH Zurich, Switzerland

Abstract. With the spread of mobile phones, the use of Mobile Adhoc NETworks (MANETs) for disaster recovery finally becomes feasible. Information retrieval from the catastrophic place is attended in an energy-efficient manner using the Geographically Bound Mobile Agent (GBMA) model. The GBMA, which is a mobile agent on MANETs that retrieves geographically bound data, migrates to remain in a designated region to maintain low energy consumption for data retrieval, and provides location based migration scheme to eliminate needless migration to reduce energy consumption. In the data retrieval using the GBMA model, survivability of the agent is important. In a MANET, a GBMA with retrieved data may be lost due to its host's death. The lost of the agent causes re-execution of the retrieval process, which depraves energy efficiency. We propose migration strategies of the GBMA to improve its survivability. In the migration strategies, the selection of the next host node is parameterized by node location, speed, connectivity, and battery level. Moreover, in the strategies, multiple migration trigger policies are defined to escape from a dying node. We present the implementation of migration strategies and confirm the achievements with several simulations. This finally leads to the adaptive Geographically Bound Mobile Agent model, which consumes even less energy.

1 Introduction

Catastrophes, be they natural, like tsunamis or cyclones, or human, like industrial accidents or terrorism acts, do not only hurt people, they also destroy (often completely) the infrastructure needed by the rescue team. In case of huge fires or cyclones even satellites cannot give any information about the area inside the disaster zone. However, information sources inside the zone might still be intact and could give precious data. Means to access these information sources need to be investigated in order to retrieve their content in crisis situations. Even after a major catastrophe, when communication infrastructure might be down, many communication devices (like cell phones or PDAs) would still be functioning. The authority managing rescue should get access to these devices to be capable of offering communication solutions into the zone. This hypothesis is not unreasonable as rescue operations are often coordinated by government or military

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 353–364, 2006.

organizations, which could have the authority to switch communication devices into *ad-hoc* mode. Since most of them can be programmed to function as hosts of an ad-hoc network, assuming that this switch could be triggered by sending a specific message is reasonable as well. This provides a Mobile Ad-hoc NETwork (MANET) [14], ready to support emergency communication.

In mobile hosts, the energy available has to be handled with care, especially in the post-disaster scenario. Tei *et al.* [11,12,13] proposed to use a mobile agent model that gathers and aggregates location-specific data from the catastrophic zone in an energy-efficient manner instead of querying every resource separately. This paper addresses several improvements of the model to improve energy-efficiency. In particular, its major contributions are:

– We define a new mobile agent model: the adaptive Geographically Bound Mobile Agent moves according to its geographical location and the dynamics of the MANET topology.
– We propose a more realistic approach to deal with post-disaster scenarios, and thus, with specific properties of associated MANETs.
– This adaptive mobile agent model is simulated in several scenarios and outperforms the standard GBMA model.

A short overview about the GBMA model is given in Section 2. The improvements with implementation are presented in Section 3, affirmed by the simulation results in Section 4, and concluded in Section 5.

2 Related Work

In the MANET, hosts are moving around freely and their directions and speeds can hardly be predicted (especially in the post-disaster scenario considered in this work). Research by Aschenbruck *et al.* [1,2,3] addresses the modeling of real moving habits after a disaster. They build their model by the use of real data from firemen. However, it is still not clear how to simulate such movements appropriately. Johansson *et al.* [9] propose a mobility model for the post-disaster scenario.

Reducing the amount of data transferred in a MANET is one way to improve the energy-efficiency. The in-network aggregation used in [5,10] eliminates redundant data or aggregates data by intermediate nodes between a data source and its data sinks. The in-network aggregation is done according to data reduction code statically deployed in the intermediate nodes. Therefore, the data reduction operators in these works are very simple, such as calculating the maximum, average, or summation.

Application-specific data reduction can further improve this approach. The mobile agent model [16] provides the means to deploy application-specific code dynamically. The mobile agent [16] is a software entity that can migrate independently among hosts in a mobile network in order to complete the task assigned by the remote observer. It migrates based on its own needs and choices. Because of the mobility, the new computing model reduces network load, enhances communication efficiency, and adapts dynamically to the changing network environment in distributed or mobile computing. It migrates with the application-specific code and state to continue its task after migration.

In a context of data retrieval of mobile agents in a MANET there are three kinds of cost: data retrieval cost, migration cost, and software execution cost. The data retrieval cost is the total amount of energy consumed for communication to retrieve the data. Intuitively, it depends on the distance between the mobile agent's host node and data sources. The migration cost is the total energy consumption for transfer of agent program codes and its execution state to migrate the agent. It depends on migration frequency and the amount of data that the agent has collected already. The software execution cost is the total amount of energy consumed during computations of agent execution. The software execution cost is relatively small compared with communication cost such as the data retrieval cost and the migration cost [8]. In this paper, we focus on the data retrieval cost and the migration cost.

Using a mobile agent model for data retrieval from a sensor network was used in [7,15,17]. Agilla [7] is a mobile agent middleware for sensor networks and realizes dynamic injection of data aggregation code to reduce data retrieval cost. Wu *et al.* [17] proposed the computation of the optimal route for a mobile agent in a sensor network. The route computation utilizes signal strengths of nodes to maintain low energy consumption for the agent migration. These works only support proactive migration to retrieve information from various data sources, but they do not support reactive migration to maintain low data retrieval cost for each data source. Without reactive migration, the agents cannot maintain low data retrieval cost while its data retrieval, due to the node mobility. Tseng *et al.* [15] proposed a location tracking protocol with a mobile agent in a sensor network. The agent migrates among sensors to stay near the moving target. It can adapt its location in response to the change of the target location, but it does not consider the migration cost.

As we aim to retrieve information from a distant area, computations performed close to the information host can reduce the amount of data to transfer and the data retrieval cost significantly, allowing a longer life time to mobile nodes of the MANET. Moreover, the computations should stay near the data sources if all nodes move freely. In [11,12,13], Tei *et al.* introduce the GBMA model that aggregates the data retrieved from different areas and supports reactive migration in response to node-mobility. The GBMA gathers and aggregates location-specific data from the catastrophic zone in an energy-efficient manner instead of querying every resource separately by flooding the network.

The observer defines the *target zone* as the geographic region he is interested in. The GBMA then retrieves the location-specific data from nodes in the target zone. Intuitively, the GBMA migrates to remain near data sources to maintain the low data retrieval cost. Figure 1 shows an example of the behavior of the GBMA.

As an important part, they proposed a migration scheme by defining the *expected zone* to eliminate needless migrations. The expected zone uses a rectangular geographic region as a trigger for migration, which is adjusted dynamically

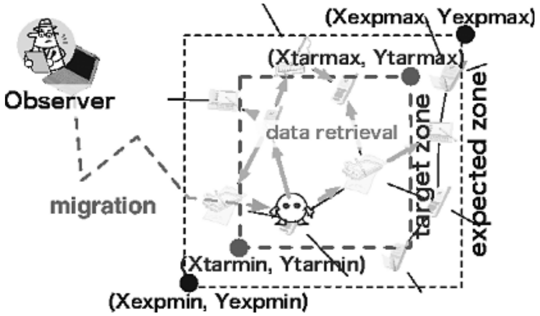


Fig. 1. Example of the GBMA behavior

using the host node's speed. When the node speed is quite low, the migration frequency of the GBMA will hardly increase if the expected zone is narrow, but when the node speed is high, the optimal expected zone will become wide. However, the GBMA model does not consider a possible loss of the agent. Therefore, its cost will increase in a MANET where nodes are down due to battery lost, because its survivability will deprave. In this paper, we propose migration strategies to improve the agent's survivability.

3 The Migration Strategy

In this section we develop the migration policy of the adaptive Geographically Bound Mobile Agent (aGBMA) model to improve its survivability. The survivability is an important factor to reduce the total energy consumption, because the observer has to launch the data retrieval again if an agent is lost. Of course, the relaunch consumes a lot of energy. Therefore, the improvement of the survivability highly improves the energy efficiency.

We first introduce the target zone center **c**, set initially by the observer[1]. It is assumed that data sources are concentrated at several points not known at first. Such data sources could be part of a sensor network like fire door sensors or smoke detectors, or they could represent a security room with all camera information.

Thus, we propose to adapt the virtual center's position according to the amount of data received from different information sources. The change of the center might result in a migration of the agent (*cf.* Figure 2).

Hosts are able to determine their location **s** and the agent can compute the distance δ to the target zone's center **c**. The movement **v** of a host can be determined easily using two snapshots of its location for time t and t', namely **s** and $\mathbf{s}'$.

Furthermore, factors like connectivity (the number of reachable hosts ν) and power (expected remaining battery β after task) are relevant, because in the

[1] This can be done automatically using the coordinates of the target zone.

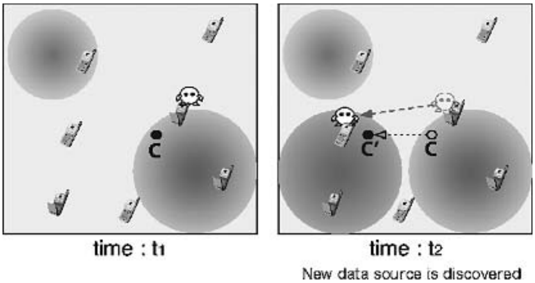


Fig. 2. Moving the center towards data sources

worst-case, the agent process has to be restarted and recomputed; this results in high additional energy consumption (considering the post-disaster scenario, loss of time and information might be even more disastrous).

The migration policy consists of two main mechanisms: the node selection to define the target of the migration and the migration triggers defining the conditions to migrate.

3.1 Node Selection

In order to choose the host as the best node available, in the aGBMA model, a quality value $\mathcal{Q}$ is defined for each mobile host node h. This value depends on four parameters (explained in detail afterwards): the escape speed ρ, the distance δ to the data center, the connectivity ν and the battery power β.

$$\mathcal{Q}(h) = \frac{\beta_h^{\omega_\beta} \cdot \nu_h^{\omega_\nu}}{\delta_h^{\omega_\delta} \cdot \rho_h^{\omega_\rho}}$$

The weights ω_β, ω_ν, ω_δ and ω_ρ can be chosen according to the scenario.

Data center. The aim of the agent is to collect data in the target zone. The initial work relies on the assumption that the center of the target zone is the most important point. Therefore, migration triggers depended on the distance to this center. The new approach takes care of the real position of the data inside the target zone. In the current model, the data can be retrieved by nodes of the MANET at some fixed locations of the target zone. When the aGBMA arrives in the target zone, its knowledge is the geographical center of the zone only. This is taken as the first virtual center. When receiving data messages, the agent discovers data sources and updates its approximation of the virtual center position.

The virtual center is defined as the barycenter of the data sources according to the amount of data received . Its position is determined by approximating positions of data sources and by using the amount of real data received from each location. This real data is evaluated by removing redundant data [2]. This

[2] Redundant data is defined as data that came from the same source and was given to the first MANET node within some time interval.

permits to deal with the case when a data source has many neighbors and a large amount of redundant data is received by the agent.

The position of each data source itself is evaluated according to the total amount of data received, including (sic!) redundancy. This allows to increase the accuracy of the approximation of the data source positions. The source's identifier as well as information about the node that first received the message (receiving time and current position) is included as additional information into data messages. Computing the barycenter out of received data allows to give an approximation of the location for each data source.

Escape speed. The node movement speeds are important factors influencing its quality. A fast node might soon leave the zone. We do not directly use the node's speed but rather its escape speed ρ from the center $\mathbf{c}$, because a fast node running around the center is not leaving the zone and thus, remains a good candidate for being a host node. The escape speed is the projection of the speed vector on the axis from the center to the node location (*cf.* Figure 3). Each node periodically determines its location $\mathbf{s}'$ and stores it together with its previous one $\mathbf{s}$. The aGBMA retrieves these two location tuples from nodes and calculates their escape speed.

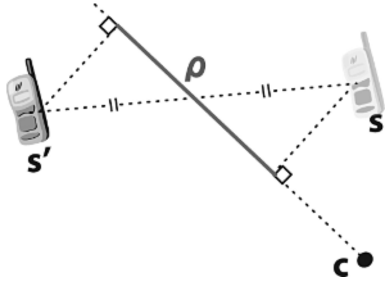


Fig. 3. Escape speed ρ

Connectivity and Battery power. Connectivity ν is the number of one-hop neighbors and indicates the likelihood of isolation. Each node periodically sends "hello" messages to one-hop neighbors and counts the nodes around. Battery β is the remaining amount of node energy and indicates the likelihood of battery exhaustion. The aGBMA retrieves ν and β from surrounding nodes.

3.2 Migration Triggers

We propose three migration triggers that will be evaluated in the simulation section.

Extension of the Expected Zone evaluation. The migration policy of the earlier GBMA model relies on the distance to the center of the target zone, the threshold was determined by both a coefficient α and the speed of the host. The extension of this trigger uses $\alpha \cdot \rho$, considering that only the escape speed ρ matters.

Quality threshold. This trigger relies on the current quality $\mathcal{Q}_c$ of the host node. Periodically, the agent checks its quality and stays with a certain probability if $\mathcal{Q}_c \leq \mathcal{Q}_i/q_t$, where $\mathcal{Q}_i$ denotes the quality of the node when the agent arrived and q_t denotes a threshold value, typically 2.

Both. This trigger conjugates the two preceding propositions. The agent migrates with a certain probability if its host has a bad quality but it moves as soon as the host leaves the expected zone according to the new criteria.

4 Simulation

In this section we compare the performance of the aGBMA model with the earlier GBMA model and a direct P2P approach. Though more sophisticated, the implementation of the aGBMA has almost the same size as the GBMA (about 30 kB). Experiments were computed using the simulator implemented on the Scalable Wireless Ad-hoc Network Simulator (SWANS) [4]. The weights $\omega_\beta = 0.6, \omega_\nu = 0.2, \omega_\delta = 1.5$ and $\omega_\rho = 1.5$ allow to tune the quality threshold and were chosen according to the scenario and to simulation during pre-experiments.

We also evaluated the performance of the three different types of migration triggers.

In these simulations, the 14^2 nodes are initially distributed on a grid covering 1000 m × 1000 m described by the coordinates $(0, 0)$ and $(1000, 1000)$. Each node is equipped with both

- an IEEE 802.11b wireless device allowing to communicate with other nodes within its range and
- a GPS receiver with which the location can be determined.

This last assumption could be relaxed considering a relative location as sufficient to determine if the node is going away from the center. This information could be obtained from:

- another node equipped with a GPS receiver;
- a data source with location information (like a static sensor network node);
- other nodes with relative position information.

Such solutions are not considered here for the sake of brevity.

We implemented, in the SWANS framework, a mobility model slightly extending the mobility model of a disaster area described in [9], to represent a simple but more realistic mobility model according to our post-disaster scenario. We introduce two kinds of nodes: walking nodes (speed: 1-10 m/10 sec) and running nodes (speed: 1-20 m/10 sec). Observe that the fast nodes in our mobility

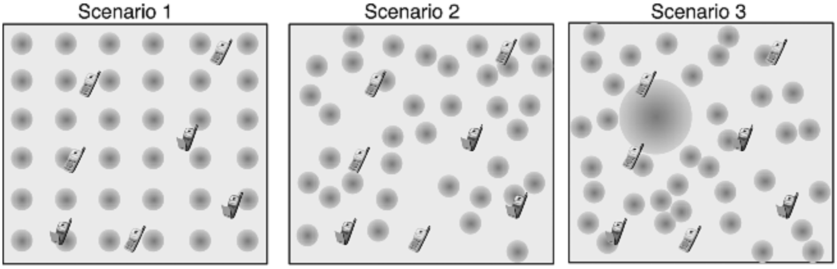


Fig. 4. Three scenarios

model are running victims and not rescue staff with cars as in the post-disaster mobility model of [9]. The agent is able to migrate between these groups. The battery level of each node is chosen randomly between 0 and 10 W. 10 W is enough energy to run the complete data retrieval in this simulation. All nodes consume energy according to the energy consumption model described in [6], *i.e.* when they send or transmit data messages, or when they receive data from information sources.

The simulations were run for three different scenarios (*cf.* Figure 4):

- In the first, 36 data sources are placed homogeneously on a grid. They represent a regular fixed sensor network composed of exit doors or movement detectors. Such sensors deliver a small amount of information within a short range but are highly autonomous. This kind of sensor can be used for one week with its own battery.
- The second scenario uses the same kind of data sources but placed randomly in the target zone.
- The third is built upon the second. We add a special source that delivers a lot of data but has only limited autonomy; for example the security room of the building. It contains a lot of information but is not able deliver it for hours without power supply.

In these experiments, the normal sensors provide 10 kB of data to nodes located within 6 m and for an infinite number of times during the simulation. The special source provides 500 kB of data to nodes located within 20 m but at most six times.

The target zone is a square region defined by the coordinates $(650, 650)$ and $(750, 750)$. The observer is located at $(200, 200)$ and does not move. Both agents, GBMA and aGBMA, are located at the observer node initially and migrate to a node in the target zone. After one hour, the observer sends result queries to its agent and receives the results.

The expected zone parameter α used for the GBMA and the aGBMA with the corresponding triggers is 250, which is optimal, as reported in [11].

Based on these three scenarios, we evaluated the performance of the GBMA model, and that of the aGBMA model with the quality migration trigger (aGBMA-Q), the expected zone migration trigger (aGBMA-E), and with both triggers (aGBMA-B). In the case of the aGBMA model, all nodes periodically

Table 1. Results of scenario 1 to 3

	Energy consumption (W)			Amount of information			Number of migrations			Number of agent loss		
Scenario	1	2	3	1	2	3	1	2	3	1	2	3
P2P	289	287	351	5.44	5.22	5.08	N. A.			N. A.		
GBMA	48.4	48.8	118	4.25	4.58	1.87	2.2	2.15	1.77	6.25	10.4	29.8
aGBMA-Q	92.1	88.1	154.1	4.64	4.51	2.91	5.91	5.78	5.84	2.22	2.22	15.6
aGBMA-E	89.9	88.6	145.3	4.84	4.44	3.26	2.78	2.56	2.72	4.44	2.33	28.3
aGBMA-B	93.4	94.3	167.6	4.53	4.94	3.21	6.34	6.43	6	2.22	2.13	18.8
aGBMA-Q - ν	49.1	48	127	4.35	4.82	2.82	7.2	6.39	6.41	0	0	6.82
aGBMA-E - ν	58.2	50.8	115.4	4.98	4.51	2.67	2.44	2.4	2.64	2.08	2.33	16.7
aGBMA-B - ν	53.6	53.5	130.1	3.85	4.34	2.83	7.6	7.04	7.68	2.13	2.17	12.5

(every 30 sec) send a "hello" message to count their neighbors, but that might be too costly. Therefore, we also evaluated the performance of the aGBMA model without using the connectivity argument ν. We run 50 simulations in each case.

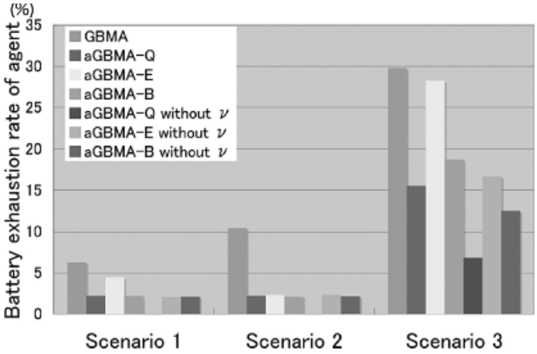


Fig. 5. The rate of agent loss

In these experiments, we compared the aGBMA models and the GBMA model considering energy efficiency and survivability. Table 1 shows the total energy consumption, the total amount of information retrieved and the number of migrations in each scenario. An agent can be lost if its current host node is down due to battery exhaustion. We adopt the percentage of agents lost while data retrieval due to battery exhaustion of its host node as a metric for survivability of agents. Figure 5 shows the percentage of lost agents. Moreover, from the point of energy efficiency, we introduce the amount of energy consumed per retrieved information as a metric for energy-efficiency. Figure 6 shows the amount of energy consumed per retrieved information.

In all scenarios, the overhead of using ν is not negligible. Even in the third scenario with one big data source, using ν is relatively expensive. The aGBMA model without ν provides more survivability. This can be explained by the

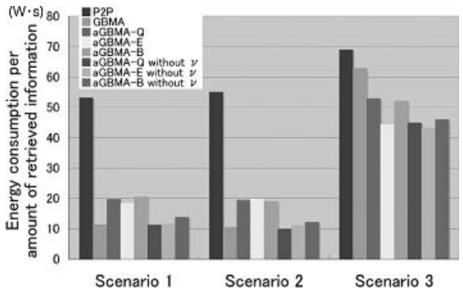


Fig. 6. Energy consumption per amount of received information

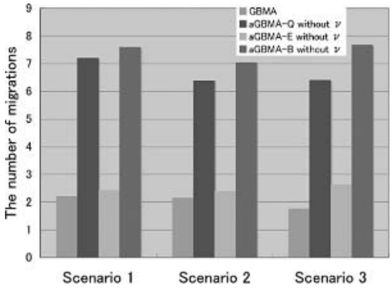


Fig. 7. Migration rate

"hello" message's overhead. We therefore consider in the following the migration triggers without ν.

Comparison to the GBMA model. The loss rate of the GBMA is bigger than that of the aGBMA, particularly in the third scenario, because the GBMA does not use the battery level neither as node selection parameter nor as migration trigger, whereas the aGBMA uses the battery level at least as node selection parameter. When network load is high, taking care of the battery improves the survivability significantly. From the energy efficiency point of view, considering the third scenario, the aGBMA consumes less energy than the GBMA, because the aGBMA selects a host node to maintain low data retrieval cost using ρ and δ. The GBMA does not care about the location of data sources and the amount of data, it only selects its host node according to its position from the geographical center and cannot maintain low data retrieval cost in a heterogeneous situation. The aGBMA uses the new virtual center for data retrieval determined by the approximated location of data sources and their amount of information and thus adapts well to heterogeneous situations. Compared to the basic P2P approach, the strength of the GBMA and the aGBMA concepts is obvious at first sight.

Comparing migration triggers. Among the different aGBMA models, the aGBMA-Q and the aGBMA-B provide better survivability than the aGBMA-E, because they use the battery level as migration trigger. In scenario 1 and 2, the aGBMA-Q provides best energy efficiency because it is hardly lost. It remains to investigate about the characteristics of scenarios 1 and 2 because the simulation results encountered are quite similar. In scenario 3 however, the agent migration cost is more expensive, according to the increased amount of received data. The aGBMA-Q migrates more frequently than the aGBMA-E (*cf.* Figure 7). Therefore, in scenario 3, the aGBMA-Q provides worse energy efficiency than the aGBMA-E (though, the parameter of the aGBMA-Q is not optimized). The aGBMA-B provides the worst energy efficiency among the three. With its two migration triggers, it migrates quite frequently. This leads to a non-negligible overhead. Thus, using both migration triggers is not efficient.

5 Conclusion

The Geographically Bound Mobile Agent was improved by using a far more adaptive migration mechanism. Node speed and movement direction was taken into account, as well as the definition of a virtual data center. Furthermore, a sophisticated node selection strategy prevents from choosing a poor node. Considering the remaining energy in a node's battery has resulted in a higher survivability of the mobile agent and the overall energy efficiency shows to be better than using the classical GBMA model. The quality value is useful for decisions. Even a not optimal aGBMA model (parameters were selected according to pre-experiments) outperformed the classical but optimized GBMA model considering energy-efficiency.

Our future research plans are twofold: in addition to simulations we aim to further improve the aGBMA model as follows.

- At the moment, the aGBMA performs many unnecessary migrations. We want to avoid these migrations using optimal parameters, which are to be defined.
- We also plan to consider node isolation (in the post-disaster scenario, a node can be separated from the other nodes of the MANET with high probability). The neighborhood value ν has shown to be too costly to compute compared to its use; we aim to use more sophisticated algorithms for this problem.
- Furthermore, migration cost depends on the size of the agent, the migration probability should therefore depend on the amount of data retrieved so far.
- To address the post-disaster scenario adequately, the mobile agent might send major chunks of data back to its observer (if the position is known) in order to provide valuable information as fast as possible. We have to find the optimal size of such data chunks without loosing the efficiency of the mobile agent approach.
- Moreover, if the area is quite large, multiple cooperating agents could be deployed.

Besides these optimizations, the post-disaster mobility model used is far from realistic. We slightly extended the random walk model by simulating two kinds of nodes, but the mobility model needs further improvements including but not limited to repulsion points (for example a fire in real world), group interactions and node disappearance.

References

1. N. Aschenbruck, M. Frank, and P. Martini. Statistical analysis of traffic measurements in a disaster area scenario considering heavy load periods. In *Proc. of the 2nd International Workshop on Wireless Ad-hoc Networks*, 2005.
2. N. Aschenbruck, M. Frank, P. Martini, and J. Tölle. Human mobility in MANET disaster area simulation - a realistic approach. In *Proc. of the 4th International IEEE Workshop on Wireless Local Networks*, pages 668–675, 2004.

3. N. Aschenbruck, M. Frank, P. Martini, and J. Tölle. Traffic measurement and statistical analysis in a disaster area scenario. In *Proc. of the 1st Workshop on Wireless Network Measurements*, 2005.
4. R. Barr, Z. Haas, and R. van Renesse. Scalable wireless ad hoc network simulation. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*, pages 297–311, 2005.
5. Guanling Chen and David Kotz. Policy-driven data dissemination for context-aware applications. In *Proc. of 3rd IEEE International Conference on Pervasive Computing and Communications*, pages 283–289. IEEE Computer Society, 2005.
6. L. Feeney. An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mobile Networks and Applications*, pages 239–249, 2001.
7. C. Fok, G. Roman, and C. Lu. Rapid development and flexible deployment of adaptive wireless sensor network applications. In *24th International Conference on Distributed Computing Systems*, pages 653–662, 2005.
8. Adam Woliszz Jean-Pierre Ebert, Brian Burns. A trace-based approach for determining the energy consumption of a wlan network interface. In *Proc. of European Wireless 2002*, pages 230–236, 2002.
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 *MOBICOM*, pages 195–206, 1999.
10. Joseph M. Hellerstein Samuel Madden, Michael J. Franklin and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *Proc. of the Fith Annual Symposium on Operating Systems Design and Implementation (OSDI)*, pages 131–146, May 2002.
11. K. Tei, Y. Fukazawa, and S. Honiden. An adaptive location-based reorganization scheme for geographically bound mobile agent in mobile ad-hoc networks, 2006. submitted.
12. K. Tei, Y. Fukazawa, S. Honiden, and N. Yoshioka. Geographically bound mobile agent in MANET. In *MobiQuitous*, pages 516–518, 2005.
13. K. Tei, N. Yoshioka, Y. Fukazawa, and S. Honiden. Using mobile agent for location-specific data retrieval in MANET. In *The IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM '05)*, 2005.
14. C. Toh. *Ad Hoc Wireless Networks: Protocols and Systems.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
15. Y. Tseng, S. Kuo, H. Lee, and C. Huang. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. volume 47 of *The Computer Journal*, pages 448–460, 2004.
16. J. White. Mobile agents white paper. 1994.
17. Qishi Wu, Nageswara S.V. Rao, Jacob Barhen, S. Sitharama Iyengar, Vijay K. Vaishnavi, and et al. On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Transactions on Knowledge and Data Engeneering*, 16(6):740–753, 2004.

Gradient-Driven Target Acquisition in Mobile Wireless Sensor Networks

Qingquan Zhang[1], Gerald Sobelman[1], and Tian He[2]

[1] Department of Electrical and Computer Engineering,
University of Minnesota, Twin cities, USA
[2] Department of Computer Science and Engineering,
University of Minnesota, Twin cities, USA
{zhan0511, sobelman}@umn.edu,
tianhe@cs.umn.edu

Abstract. Navigation of mobile wireless sensor networks and fast target acquisition without a map are two challenging problems in search and rescue applications. In this paper, we propose and evaluate a novel Gradient Driven method, called GraDrive. Our approach integrates per-node prediction with global collaborative prediction to estimate the position of a stationary target and to direct mobile nodes towards the target along the shortest path. We demonstrate that a high accuracy in localization can be achieved much faster than other random work models without any assistance from stationary sensor networks. We evaluate our model through a light-intensity matching experiment in MicaZ motes, which indicates that our model works well in a wireless sensor network environment. Through simulation, we demonstrate almost a 40% reduction in the target acquisition time, compared to a random walk model, while obtaining less than 2 unit error in target position estimation.

Keywords: Wireless Sensor Network, Navigation, Localization, Probabilistic Model, Rescue.

1 Introduction

Wireless sensor networks have gained extensive attention in many applications such as tracking, differentiated surveillance, and environment monitoring [1,2,3]. Moreover, the hybrid systems of mobile objects (e.g. Robots) and sensor networks create new frontiers for civilian and military applications, such as search and rescue missions in which the background environments are inaccessible to humans. A heterogeneous searching team consisting of robots and a wireless sensor network has greater advantage, considering its distributed computation and navigation capability achieved through the cooperation of embedded wireless sensor networks.

Although the applications of mobile sensor networks keep diversifying, several underlying capabilities remain fundamental and critical. In this work, we focus on the target acquisition – finding the locations of stationary targets using mobile sensor nodes. The challenging problem we address in this work is *to navigate*

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 365–376, 2006.

a team of mobile sensor nodes toward the stationary targets fast and accurately while consuming the least amount if energy and other resources. In this emerging research arena, most research groups employed static wireless sensor networks to navigate the mobile sensor nodes. Tan in [4] used distributed static sensor networks to collect the data and execute local calculations to generate a path for a mobile sensor network to move toward the goal. Although the in-network calculation implemented in that project was quite efficient in creating the shortest routing path, the additional requirement of a stationary distributed sensor network sets a barrier for rescue applications, because of the high cost to cover a large geographic area with a large number of sensors. Other research groups [5] proposed gradient methods in which the mobile wireless sensor nodes move toward the gradient direction assuming that targets carried the most intensive strength of interested signals. However, in all of their implementations, the assistance of a stationary wireless sensor network was assumed to be available in generating a local signal distribution map. A probabilistic navigation algorithm is presented in [6], where a discrete probability distribution of vertex is introduced to point to the moving direction. This algorithm computes the utilities for every state and then picks the actions that yield a path toward the goal with maximum expected utility. The shortcoming of this method is that it requires the arrival of a mobile sensor node to localize the target position and significant communication overhead is introduced by the iteration process.

2 Contribution

In this paper, we propose to compensate those deficiencies by incorporating a prediction model of real-time processes into a mobile sensor network sensing and navigation architecture. We are interested in the mutually beneficial collaboration of the algorithms described above but seek to reduce the costs and provide faster target localization. *The novelty of our approach is the seamless integration of a per-node prediction model with a global prediction model.* The per-node prediction model guarantees that a mobile node can acquire the position of a target alone, while the global prediction significantly reduces the navigation overhead and time, if collaboration among the nodes is available. Specifically, the main contributions of our prediction models are:

- Our model provides more meaningful description of individual sensor readings in term of accuracy and confidence.
- Our model works with a single mobile sensor node as well as a swarm of mobile sensor nodes. In the latter case, the sensor nodes have the ability to share local information in order to draw a global picture, which helps each sensor node to acquire the target along a significantly shorter path.
- The in-network prediction algorithm enables faster yet accurate target position acquisition: sensor nodes would be required to reach the target only when the model prediction is not accurate enough to satisfy the requirement with an acceptable confidence. This allows a significant reduction in navigation energy.

The remainder of this paper is organized as follows. Section 3 defines the assumptions. Section 4 overviews the design. In Section 5, we present the in-network per-node prediction model. Section 6 describes target acquisition in the context of global prediction and the corresponding mobile sensor node navigation protocol. Section 7 presents empirical data obtained from the MICAZ platform as well as simulation results. Finally, in Section 8, we present our conclusions and future works.

3 Assumptions

Our design is based on two assumptions: network connectivity and the self-localization of mobile wireless sensors.

- **Connectivity:** First, wireless sensor nodes in the network are assumed to be able to ensure connectivity. Individual mobile sensor nodes deployed in large area is likely to lose connection to a central base station, if the routing information is not updated. Therefore, it is desirable to maintain connections across a team of mobile sensor nodes while minimizing power consumption and allowing the sensor nodes to achieve their individual goals.
- **Node Self-Localization:** The second assumption hinges on the localization availability for a mobile wireless sensor network. If a mobile sensor node enters an unknown area, it must be able to specify its own location dynamically without a map. This location can be obtained either through GPS such as used in ZebraNet [7] and VigilNet [3]. It can also use a dynamic localization scheme [8] that adjusts the estimated location of a node periodically based on the recent observed motion.

4 Overview of Prediction Model

The objective of our GraDrive target acquisition scheme is to predict the location of stationary targets within allowable uncertainty (or a confidence level) dictated by a rescue plan. To illustrate the design of GraDrive, we start our description with a rescue scenario shown in Fig. 1. Here we note that our method is independent of this rescue application and can be applied in other scenarios as well.

- **Objective:** The control center (base) disseminates a search objective to a mobile sensor network with two parameters, *error tolerances* and *confidence level* of the target, specifying the quality of target acquisition. For example, the objective would be locating a target within 2 meters with at least 95% confidence. The tolerance levels for each mobile sensor nodes can vary correspondingly in case different nodes are designed for different purposes.
- **Individual Prediction Model:** Once the search objectives are received by the mobile nodes, individual node decides their most efficient way to locate the potential target with the requested confidence *individually*, using the

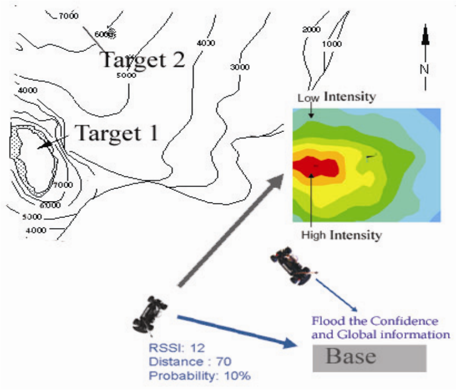


Fig. 1. The Architecture schematic of GraDrive

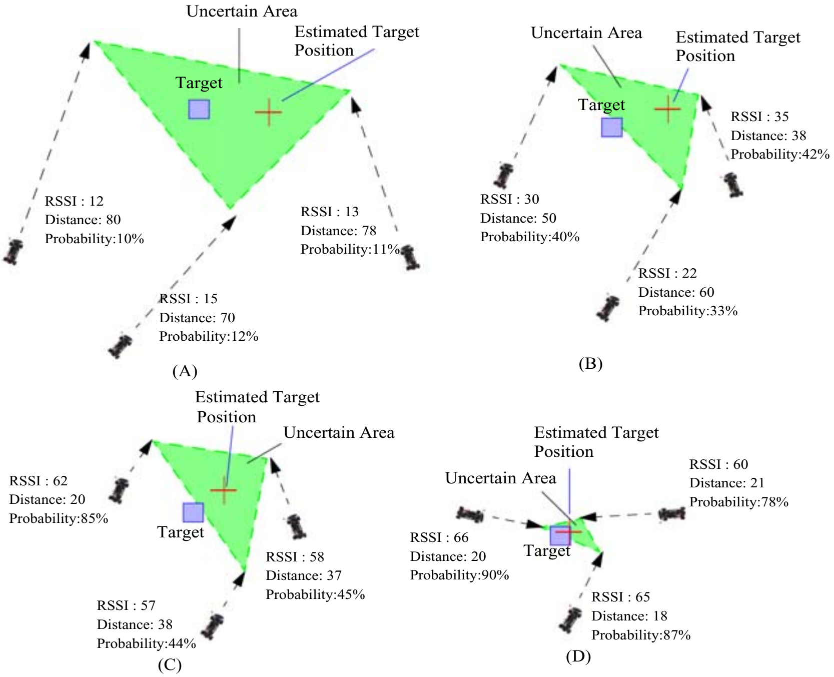


Fig. 2. Collaborative Prediction Scheme of GraDrive

per-node prediction model. It starts to move toward the direction in which it anticipates the fastest path to reach the confidence.

- **Collaborative Prediction Model:** In addition to its own plan and navigation, sensor nodes also report back to a base station, where all the individual nodes' readings and plans are collected and computed to create a global map and an uncertainty area. If computation results show probability increase by certain interval, e.g.5% to its previous state computation, the base station will disseminate the global prediction value over the network so that each sensor nodes in network can update their model. In other words, the prediction result based on collaborative information overrules the results from the individual prediction model.

As demonstrated in Fig. 2 from (A) to (D), the individual sensor node continuously predicts the target position with increasing probability and move toward the target, the uncertainty area where the target is located shrinks through collaboration among mobile sensor nodes. If collaborative probability calculated reaches the dictated objective, a success of rescue plan is achieved. The position it reports is the exact target position specified. Compared to other static sensor node navigation plans, the prediction results computed by our model still provide considerably more information than MobileRobot [6] and SafeRobot [9].

5 Gradrive Model Details

In this section, we formally describe our per-node prediction model to estimate the position of a stationary target with certain confidence. This per-node prediction model forms the basis for global collaborative prediction described in Section 6. We note even though we consider an unknown area with multiple targets, the searching for separate targets is independent to each other as long as the field (RSSI) generated by one target doesn't overwhelm that generated by others. Therefore in the remaining of paper, we focus on only single target acquisition problem.

5.1 Prediction Problem Formulation

Conventionally, we begin with a value-prediction problem, which creates a Received Signal Strength Indicator $F(\theta)$ over a parameter set θ. For example, if $\theta = (d, t, v)$, RSSI is related to d, the distance between a mobile sensor node and the target, t, the time of sampling, and v, the speed of mobile sensor nodes. This model can be established by getting consecutive sensing readings (system states) when a mobile sensor node moves. Typically, the number of parameters in θ is much less than the number of states collected and changing one parameter changes the estimated value of many states. To approximate our model appropriately, we seek to minimize the mean squared error over some distribution, P, of the inputs.There are generally far more states than components in θ. The flexibility of the function estimator is thus a scarce resource. Better approximation at some states can be gained generally only at the expense of worse approximation at other states.

5.2 Distance Prediction Model

In GraDrive, we extend the familiar one-dimensional normal probability density function known as Gaussian distribution to two variants multidimensional distribution. The predicted distance from sensor nodes' current position to predicted target position can be queried or estimated from the model. The multidimensional Gaussian distribution function over two attributes, trust interval and RSSI, can be expressed as a function of two parameters: a 2-tuple vector of means, μ, and a 2 × 2 matrix of covariances,Σ. Further, we assume the trust interval set by a rescue team is independent of the RSSI received, which means the trust interval of the predicted distance estimation T_i to the mean of historic results μ doesn't change dynamically along the searching process. The two dimensional distribution can be separated for description purposes. Without loss of generality, it is assumed that the predicted distance d is disproportional to RSSI, that can be expressed as $d = r_1/RSSI + r_2$, where r_1 and r_2 are two adaptive parameters that can be determined before the searching process. We note here other RSSI attenuation models can be used here as well without invaliding our approach. We then use historical data or experience data to construct the models, providing r_1 and r_2 at each RSSI value appropriately. Besides offering the predicted distance, a probability model associated the d is also constructed to provide confidence of the prediction, e.g. given a predicted distance of 2 feet, the confidence for this prediction is 95%. The models must be trained before it can be used, a general limitation for probabilistic model. The accuracy of the model, therefore, relies on the accuracy of data used to train it. Once the initial model is constructed, each sensor nodes can query the predicted distance map from saving model and come up with a confidence value. One distribution of the distance d against the confidence p over one RSSI is a Gaussian distribution. Suppose that rescue team have set a trust interval of T_i, given the distribution of distance over one RSSI, we can get the points d_i that satisfied that $P(d_i) - P(u) <= T_i$. Here we emphases that if the trust interval is too small, the amount of data needed to train the model will increase exponentially.

5.3 Signal Strength Distribution Prediction Model

Besides obtaining the distance d information based on measured $RSSI$, we can further refine the RSSI distribution Model. This distribution model can then be used to navigate the mobile sensor network toward the target at a shortest path. The central element in our approach is to construct a prediction model that represents attributes as accurate as possible in a mobile sensor network. As we discuss above, if the predicted RSSI distribution function depends on parameters including distance d and confidence or probability p, the function can be expressed as $F(d, p)$ considering d and p's distribution are independent. If we do the Tylor expansion on function F, a polynomial function of attributes d and p is achieved, shown as

$$F(d, p) = f(d_0, d_1, d_2 ...)f(p) \tag{1}$$

where d_i is the function of distance variable d. To reduce the computation energy consumption, only second order polynomial is considered in our case, which offers a 3-tuple vector of $D = [d_0, d_1, d_2]$:

$$\begin{aligned} d_0 &= \mathrm{c}_0 \\ d_1 &= 1/(d + c_1) \\ d_2 &= 1/(d^2 + c_2) \end{aligned} \tag{2}$$

where c_1, c_2, c_3 are constants used to avoid singularity when $d = 0$. Now we can define our gradient distribution function into a simple format as:

$$F = D \bullet A \bullet p \;\; where \; A = [a_0, a_1, a_2] \tag{3}$$

Equation 3 is our probabilistic gradient distribution prediction function for attributes of d and p. suppose that each sensor nodes observe the value of attribute D_j to be d_j, we now input sensing reading into a vector of D_j. Thus the vector D is extended as a matrix.

If enough sensing samplings are provided, we can apply non-linear Least Square Fitting to estimate the parameters A. For nonlinear least squares fitting to our undetermined parameters, linear least squares fitting may be applied iteratively to a literalized form of the function until convergence is achieved. Since we can anticipate the power type of fit and have decided initial parameters chosen for our models, the nonlinear fitting has good convergence properties.

In general, the computation of the matrix does cost a large amount of the wireless nodes' energy. The solution in GraDrive is to simplify the prediction distribution function as above, given that prediction function computation can be distributed over the network with collaboration of its neighbors or the data to be delivered back to a base station where stronger computation ability and energy are normally not limitations. If this is the case, the base station creates a gradient distribution map globally using a weighted average method as a function of probability and predicted distribution. This kind of global information is sent back to each individual node involved in application.

6 Target Localization Using the Collaborative Prediction Model

Based on the per-node prediction model, the mobile sensor nodes can infer the position of target (x, y) and the associated confidence value p. This information is then used to perform global predictions. Specifically, we propose to use a probability-weighted average model for global collaborative prediction, due to its high efficiency and low cost characteristics. The simple rational behind our method is that the sensor nodes having a higher probability are much closer to the intended target.

Generally, if the predicted target location provided by sensor nodes $n_1, n_2, ...,$ n_k, are $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$ associated with probability value $p_1, p_2, ...p_k$. The estimated position of the target is given as:

$$X = \frac{\sum_{i=1}^{k} p_k x_k}{\sum_{i=1}^{k} p_k} \; Y = \frac{\sum_{i=1}^{k} p_k y_k}{\sum_{i=1}^{k} p_k} \tag{4}$$

6.1 Collaborative Navigation and Prediction Protocol

With per-node and global prediction models established, we are now ready to describe how the sensor nodes navigate using these two models.

Initially, sensor nodes enter an intended region with certain moving speeds, moving directions and trust intervals. It should be noted that different mobile sensor nodes could have different moving speed or initial moving direction. After the entrance, the mobile sensor nodes continue to detect the RSSI in its sensing range. The detected RSSI readings are an important input for training the model it is assigned initially. Thus they use a default navigation plan, which is to keep moving forward unless they detect a smaller sensing reading. During the moving process, nodes themselves perform per-node prediction calculation to construct the local RSSI map as described in Section 5.3. Meanwhile, the sensor nodes estimate their distance to the target position according to the sensing RSSI, randomly pick one prediction within its trust interval. The predicted target location information is forwarded back to a base station. To prevent excessive energy consumption in communication, the frequency of updates can be specified in advance. As long as the global picture is not available, individual sensor nodes navigate according to the per-node prediction model. However, if the base station notifies the sensor nodes that it has constructed a global RSSI distribution with certain confidence, each sensor node will combine the information with its current model together and change its direction toward the gradient direction. This process will be repeated until the target position has been discovered locally or at the base station within acceptable confidence.

6.2 Default Navigation Plan When Global Prediction Unavailable

If initially there is no global picture constructed by the base station with acceptable confidence, or if there is only one separated node in the network for rescue plan, or if the network is partitioned or unable to deliver the data, the mobile sensor nodes fall back to the per-node prediction model. Given its current sensing reading, it compares with previous readings stored in memory at each motion step. After getting a smaller sensing reading, it rotates 90 degrees clockwise. The reason for that is that the target position is most likely located perpendicularly to its previous moving direction.

7 Experiments and Simulation

7.1 Model Matching Experiment

In order to verify the feasibility of the proposed prediction model and parameter-fitting algorithms, we have prototyped a light sensing system based on Berkeley MICAZ modes. Even though it is stationary, the prediction model and parameter-fitting algorithms can still be verified at the base station site which can be transferred to individual sensor nodes and implemented. Light signal strength is used as an example of RSSI to feed the model. One laptop equipped with motherboard acts as the base station. A lamp works as a target and a series of sensor nodes are deployed as shown in Fig. 3. The sensor nodes detect the sensing reading and exchange the readings to their neighbors. The base station calculates the parameters for the sensor nodes by using the least square fitting method. Fig. 4 shows one set of data fitted by the prediction model. The distance between two adjacent sensor nodes is equal and unified for matching purpose. Since the received signal strength is not an accurate measurement, probability approximation model comes into play. From the matching results, it is shown that the least square method tries to reduce the deviation among the sensing data collected. Other sets of data can also be collected and used to train the model before it can be applied into the mobile sensor scenario.

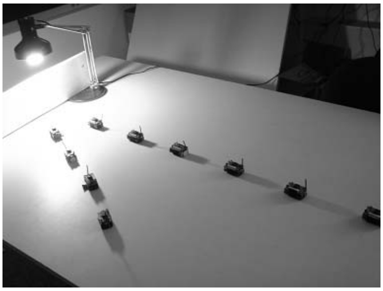


Fig. 3. Model fitting experiment with light as source of signal and using Micaz nodes in array to sense the signal strength

7.2 Simulation Setup

We have developed a program to verify the advantage of using our prediction model to locate the target in a faster approach. In our simulation, a 200×200 m^2 area is regarded as an unknown space with a target located at the center and a distribution along the diameter is defined. Essentially, it would be any random distribution that having a gradient toward the center. Each distance unit is represented as the smallest unit that the mobile sensor nodes can travel each time during simulation. The navigation algorithm is used to simulate the mobility of objects. Initially, the mobile nodes are located at the edges of the area. The initial direction is randomly picked by each mobile sensor node. If some sensor nodes move outside the simulation region, they bounce their moving direction back into simulation area. Under simulation, each mobile sensor node moves at a constant speed in integer multiples of 1m/s. After each time unit (1 second in our case), a node determines their next moving direction according to our algorithm.

7.3 Delay in Target Acquisition

We first experiment on comparing our algorithm (w/o global distribution calculation option) against Random Way Point Model. The simulation results (Fig. 5)

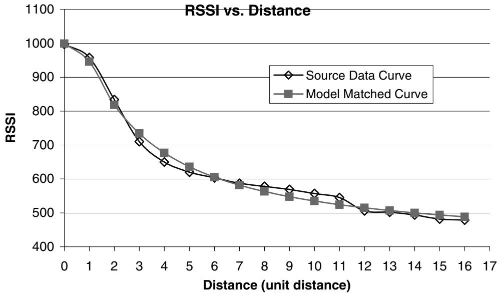


Fig. 4. The predicted model with real sensing data

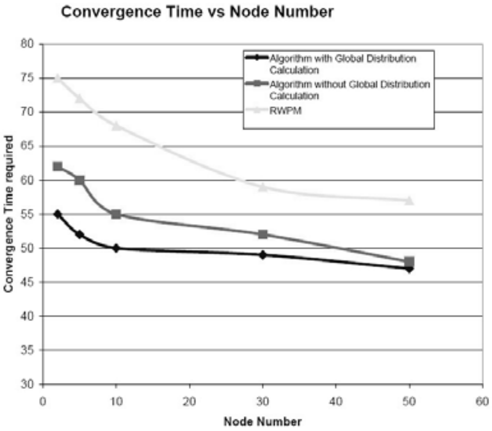


Fig. 5. Convergence time with node number for different models

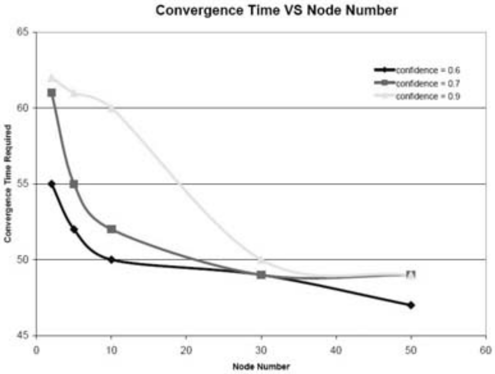


Fig. 6. Convergence time with Node number under different required confidence level

suggest that even without a global distribution calculation mode turning on, our default algorithm (rotating 90 degree counterclockwise) still provides 30% faster estimation than the random way method. If global calculation mode is on, then

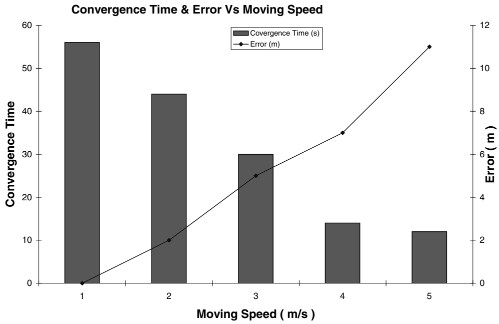


Fig. 7. Convergence time and Accuracy with different moving speeds of mobile sensor nodes

initially the sensor nodes still use default plan, but if the global signal strength distribution is available, it moves faster than the default algorithm.

7.4 Impact of the Confidence *p*

We also compare the impact of different required confidence level on the convergence time as shown in Fig. 6. It is clear that if the required confidence level goes beyond 90%, it will take much longer to simulate simply because it requires at least 2 nodes to get very close to target position. It is reasonable to choose a relative high confidence level e.g. 80% in order to balance accuracy and time cost. 0

7.5 Impact of the Target Speed

In Fig. 7, we further investigate the relationship between the moving speed of sensor nodes and prediction accuracy of target location. The convergence time correlated directly with moving speed of each sensor node since the average time for sensor nodes to get closer to target is reduced. However, the accuracy of prediction gets worse if the speed increases because the minimum deviation for the prediction is increased as well. Therefore the error continues to grow in the prediction as node moves faster from its original location. In the situation of high speed, accuracy error larger than 10 units is shown. To protect against inaccuracies in the prediction model of mobile sensor nodes, a user must set a limit for moving speed of sensor nodes.

8 Conclusion and Future Work

In this paper we present a probabilistic prediction model for dynamic target localization and evaluation of the localization algorithm. Our model does not require any known map to determine the positions of potential targets. Also the proposed gradient driven algorithm leads to a 40% reduction in time compared to that of a random working model. The relationship between sensor density and convergence time can be used as a reference of consideration for doing planning of such a mobile sensor network. Even though the computation power could be large, the error of the predicted target position can reach to almost zero and in a short time (about only 47sec). As future work, we would like to implement our algorithm on off-the-shelf hardware platforms. We would also need to design a speed self-adjusting algorithm so that the sensor node has the ability to trade off performance and cost.

References

1. A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, " A Wireless Sensor Network for Target Detection, Classification, and Tracking," *Computer Networks (Elsevier)*, 2004.
2. J. Liu, J. Reich, and F. Zhao, "Collaborative In-Network Processing for Target Tracking," *J. on Applied Signal Processing*, March 2003.
3. T. He, S. Krishnamurthy, J. A. Stankovic, and T. Abdelzaher, "An Energy-Efficient Surveillance System Using Wireless Sensor Networks," in *MobiSys'04*, June 2004.
4. J. Tan, A. Verma, and H. Sawant, "Selection and navigation of mobile sensor nodes using a sensor network," 2006.
5. I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis, "Distributed communication algorithms for ad hoc mobile networks," *J. Parallel Distrib. Comput.*, vol. 63, no. 1, pp. 58–74, 2003.
6. M. Batalin, G. Sukhatme, and M. Hattig, "Mobile robot navigation using a sensor network," in *IEEE International Conference on Robotics and Automation*, 2004.
7. P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," in *Proc. of ASPLOS-X*, October 2002.
8. S. Tilak and et al., "Dynamic localization control for mobile sensor networks," in *Conference Proceedings of the 2005 IEEE International Performance, Computing and Communications Conference*, 2005.
9. H. H. Gonzalez-Banos and J. C. Latombe, "Robot navigation for automatic model construction using safe regions," in *ISER '00: Experimental Robotics VII.* London, UK: Springer-Verlag, 2001, pp. 405–415.

Performance Study of Robust Data Transfer Protocol for VANETs

Mooi Choo Chuah and Fen Fu

Computer Science and Engineering Department
Lehigh University, PA 18015, USA
{chuah, fef205}@cse.lehigh.edu

Abstract. Vehicular Ad-hoc Networks (VANETs) have emerged as a new network environment for intelligent transportation systems. In this paper, we focus on traffic monitoring (TM) and roadside message transfer (RMT) applications. The TM application (TMA) allows drivers to query traffic conditions at some distance ahead of themselves so that they can make decisions on route changes. The RMT application (RMTA) allows data messages to be delivered between roadside entities e.g. emergency messages, via the moving vehicles. We design a robust data transfer protocol (RDTP), and evaluate its effectiveness on the two applications with various vehicular density and vehicular speed. Our study shows that our protocol achieves comparably accurate speed estimate and higher query success rate with lower control overhead than VITP, an existing protocol designed for TMA. It also achieves higher data throughput and lower delivery latency than another existing approach for RMTA.

1 Introduction

In the near future, many vehicles will be equipped with computing technologies and wireless communication devices. Thus, intelligent transport systems (ITS) [1],[2] can become a reality very soon. Such systems can enable a wide range of applications e.g. emergency message dissemination, real-time traffic condition monitoring, collision avoidance, and real-time route scheduling. Traditional ITSes often rely on certain infrastructures e.g. installing roadside traffic sensors (or cameras) and having such sensors report the data to a central database via cellular networks. Users can query the aggregated information via cellular networks. However, such traditional systems are expensive since sensors need to be installed on every road in which the system is going to be used. In addition, such systems are not scalable due to their centralized design.

Vehicular ad hoc networks based on short-range wireless communications (e.g. IEEE 802.11) has emerged as the preferred network design for intelligent transportation systems. The Federal Communications Commission (FCC) has recently allocated 75 MHz in the 5.9 GHz band for licensed Dedicated Short Range Communication (DSRC) [3] for vehicle-to-vehicle and vehicle-to-infrastructure communications. Unlike infrastructure-based networks, VANETs are constructed

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 377 – 391, 2006.

on-the-fly and do not require any investment except for the wireless network interfaces which may be a standard feature in next-generation vehicles.

An important problem to be solved for VANETs is the scalable and efficient dissemination of information among vehicles. In some applications, the information is disseminated proactively using broadcast (push model), while in others, the information is obtained on-demand (pull model). Different applications may require different dissemination models. In this paper, we consider two types of applications. The first application is real-time traffic monitoring [4]. A user behind a vehicle can issue a query to find out the average vehicular speed at a certain distance ahead of the vehicle. Such information allows a driver to determine if there is a traffic jam ahead so that he can take an alternate routes to avoid the congestion area. The second application is data transfer between two road-side entities [5]. There are several motivations for this second application. For example, an accident occurs at some point on the highway and a data relaying scheme can facilitate transfer of emergency data (e.g. medical information) between a vehicle at the site of an accident and a base station located further down the highway. Data relaying over vehicular networks reduce the number of road-side base stations that need to be installed and also reduce the cost of backhaul network that connect these road-side entities.

The organization of the paper is as follows. In Section 2, we review related work and in Section 3, we state the problem that is being addressed in this paper. In Section 4, we present our data transfer protocol for supporting these two applications in vehicular adhoc networks. In Section 5, we describe our simulation model. Then, we present and discuss the simulation results we obtained. We conclude with discussions on some future work in Section 6.

2 Related Work

A good data dissemination algorithm must address the unique characteristics of the network in which it will operate. Some unique characteristics of a vehicle-to-vehicle (V2V) network [9] include: (1) constrained movement, largely due to fix roadway geometry, (ii) rapidly changing topology due to high mobility, (iii) frequent partition due to the high mobility speeds and the number of vehicles that support V2V communications, (iv no significant power constraints, (v) unreliable communication channels. These properties make VANETs different and affect the design of data delivery protocols for such networks. Much work on the intelligent vehicular system (ITS) focus on designing protocols for collision warning systems in a VANET [6],[7],[8]. The collision warning system designed in [8] computes the time to collision (TTC) and the time to avoidance (TTA) using relative vehicular velocity and location information. A warning is issued to other vehicles if the TTC is less than TTA plus a tunable constant. In the Vehicular Collision Warning Communication (VCWC) protocol designed in [7], the authors adjust the transmission rate of emergency warning messages so that such warning messages do not overwhelm a VANET. These papers deal more with disseminating emergency warning messages to vehicles and do not support applications that use query/response type of transactions.

The two most relevant papers to our work are [4], and [5]. In [4], the authors describe an information transfer protocol called VITP for vehicular computing. VITP allows users to issue location-aware requests to obtain traffic information ahead of the drivers. VITP provides syntaxes that allow users to set return conditions of their issued queries e.g. return the average speed of 10 vehicles which are 500 m ahead of a driving vehicle. We will describe the data transfer protocol used in [4] in more details when we discuss the data transfer protocol we design in Section 4. In [5], the authors propose several protocols to allow roadside entities to use passing by vehicles as relaying nodes to transfer messages between them. The main feature of their protocols is to use a single hop data transfer approach. The source selects a nearby node as a relay node. The relay node will carry the transferred messages until it reaches a location that is within the transmission range of the destination. The different protocols described in [5] only differ in how the nodes who hear the solicit message from the source respond so that with high probability a higher priority node will send its response first and be selected as the relay node. In this paper, we design a multihop data transfer protocol to handle both types of applications. We compare our approach with the single hop approach described in [5] and the VITP protocol described in [4].

3 Problem Statement

In this paper, we design a data transfer protocol that can support two types of vehicular services. In the first type of service, a driver is able to issue a query to determine the traffic condition of a certain road segment ahead of himself. Such information allows him to make decision whether or not he wants to take an alternate route to reach his destination. The second type of application we want to support is the data transfer between roadside entities. The difference between these two types of applications is that we have query/response transactions in the first type of application but the second type of application is merely a data delivery transaction. We refer to the first type of the application as the Traffic Monitor Application (TMA) and the second type of application as the Roadside Message Transfer Application (RMTA).

For TMA, the information requested by the driver of a vehicle can be computed out of the data available on vehicles and roadside facilities located in the road segments specified by the driver e.g. the traffic flow on a road segment can be derived by estimating the average speed of vehicles moving in that road segment for a short period of time. To derive such information, the user's inquiries need to be translated into a series of location-sensitive queries. Each of these queries should be forwarded to the desired location of interest via the vehicular adhoc network. Upon arrival at its destination area, the vehicles in the relevant area must collaborate to generate relevant replies. The reply from each vehicle needs to be aggregated hop by hop as the reply traverses back to the querying node. The querying node will have to do the final computation.

For RMTA, the source needs to find an appropriate forwarder. Each intermediate forwarder in turns need to find an appropriate next hop node to carry the messages to the final destination. A trade-off can be made between using more hops (and hence each node may end up spending more energy) or incurring more delay for delivering a message. We argue that since there is no tight energy constraint in vehicles, a multihop approach is more useful.

To design a vehicular infrastructure that is capable of supporting TMA and RMTA, we need to have (a) a vehicular information transfer protocol, (b) a lightweight software component that implements the message syntax for query/response transactions and the information transfer protocol, and (c) the location encoding scheme to allow users to specify their location-aware queries. The message syntax defined in VITP [5] can meet our needs so we intend to use a modified version of this VITP syntax to support both TMA and RMTA. We, however, do not use the VITP for data transfer. Instead, we will be using a scheme we design called Robust Data Transfer Protocol (RDTP) for both applications. We will elaborate on the differences between VITP and RDTP in Section 4.

4 Robust Data Transfer Protocol (RDTP)

In this section, we describe the data transfer protocol we design for supporting TSA and RMTA. We refer to our scheme as the robust data transfer protocol (RDTP). First, we use two scenarios to illustrate how RDTP works. Then, we provide some pseudo codes for RDTP.

4.1 Traffic Monitoring Scenario

In Figure 1, we use a TMA scenario to explain the robust data transfer protocol we design and discuss how it differs from VITP [4]. We first describe how VITP works in a TMA scenario before we describe how RDTP works. In Figure 1, the source, src, broadcasts a query message to find the average speed of the vehicles in the query region. In VITP, this query message will be delivered via geographical routing protocol towards the query region. It is assumed that every node running VITP periodically issues a beacon announcing its location and speed. Based on the beacons heard, each node can maintain a neighbor list. Thus, a node that relays a query knows how to pick the next node to relay the query. Once the query reaches the query region, each node inside the query region that hears such a query will compute an average speed estimate based on the average speed, the counter value in the received query and its own speed information. If this node happens to be the first node in the query region that receives the new query, then it merely enters its own speed, and sets the counter value to one. Each intermediate node will pick another node in the query region to forward this query as long as the counter limit is not exceeded. Once the counter value reaches the count limit set in the original query, the last node that receives this query will generate a reply. This reply will be relayed back to the node that issues the query. Since the querying node may have moved during this period, one may have to use flooding to deliver the query reply once the reply has reached an area closer to where the querying node is originally thought to be located.

In RDTP, the source sets a timer to wait for the replies after issuing a query message. This timer is set to Time_Per_Hop*2*Max_Hops where Time_Per_hop is the time taken to transmit a query by one hop and Max_Hops is the number of hops the source expects it takes for the query to reach the end of the query region. Each node that receives a query message checks to see if it needs to forward the query. For example, in Figure 1, nodes 1,2,3 and 4 receive a query message from the source, src. However, node 4 will discard the query since it is farther away from the query region than its parent (the node from which it hears the query). Node 3 also discards the query since it is traveling in the opposite direction to the querying node. So, only nodes 1 and 2 will forward the query. They each sends a short reply to the source to indicate that the source is their parents. Then, they each updates the TTL in the query packet and starts a timer which is equal to Time_Per_Hop*2*(Max_Hops-TTL+1) to wait for a query reply. Such settings allow the timer at nodes 1 & 2 to expire before their parent node. In addition, they record who their parent (the node from which they first hear the query) is. Each node will only relay the query that it has heard the first time.

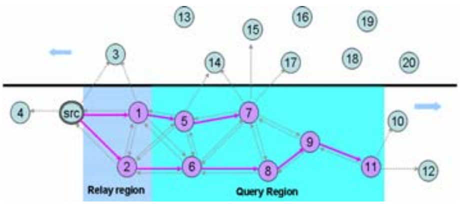


Fig. 1. Source issues query

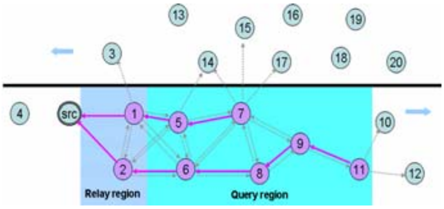


Fig. 2. Parent nodes aggregate response

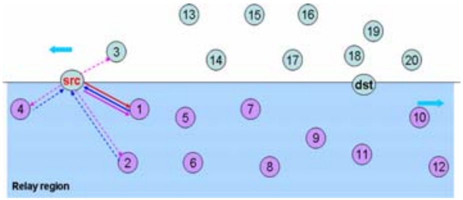


Fig. 3. Source solicits data forwarders

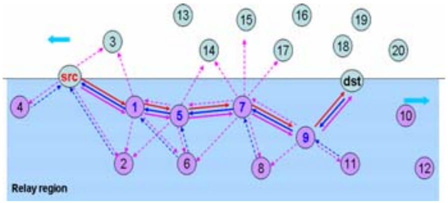


Fig. 4. Multihop forwarding for RMTA

Each node within the query region that receives the query will make a note that it needs to include its own information and process any query reply it will receive in the future. Such query forwarding continues until the query reaches some nodes that are near the end of the query region e.g. node 11. Nodes in the end zone (within a distance d=k*transmission_range of the end of the query region where k is set to 0.6) will generate a reply immediately to their parents after receiving the query. Other intermediate nodes will generate a reply only if reply timer expires or upon receiving replies from all children. We explain in Figure 2 how this takes place using the example we show in Figure 1.

Node 11 will generate a reply immediately after receiving the query since node 11 is in the end zone. When node 9 receives the reply from node 11 (as shown in Figure 2) node 9 will add its own speed, recompute the average speed, increment a counter before sending back the updated message to its parent. Node 7 may hear node 9's reply but since node 9's reply is unicasted to its parent which is node 8, node 7 will not utilize this information. Since each node has a unique parent, we ensure that each vehicle's speed information is only utilized once. Eventually, all replies will reach the query node and the query node performs the final computation to get a final answer.

To minimize the number of nodes within the query region to be involved in the query reply process, we allow each node in the query region to relay the query probabilistically. That way, not every node within the query region needs to be involved. The trade off is the accuracy of the estimated average speed of vehicles in the query region.

4.2 Roadside Message Transfer Scenario

In Figure 3, we show two roadside entities separated by a distance of L m. To select a forwarding node, the source (denoted as src in Figure 3) broadcasts a forwarder solicit request message which contains information about the intended destination. Any node that hears this message issues a forwarder solicit response message including information about its own speed and location. The source waits for a certain period of time, denoted as TIMER for replies from all nearby nodes. Then, the source computes a metric that reflects the contact duration it will have with any responding node. Based on this information, the source will select one node as the data forwarder and start sending data messages to that node. In Figure 3, we show that the source selects node 1 as the forwarder. The node selected as the forwarder transmits a data acknowledgement packet periodically. This message not only helps the source to know what messages to re-transmit but also allows the source to know when its link with the selected forwarder will be broken. Once the source has lost connectivity with a selected forwarder, the source will repeat the forwarder selection process again.

In Figure 4, we show that each forwarder (e.g. nodes 1, 5, 7 and 9) repeats the forwarder selection process to select the next-hop forwarder until the message reaches the destination (denoted as dst).

The delivery approach in RDTP differs from the data delivery approach used in [5]. In [5], once the source has selected node 1 as the data forwarder, node 1 will travel towards the destination after receiving messages from the source. When node 1 is within the transmission range of the destination e.g. it hears the periodic beacons sent by the destination), then it will deliver the picked up messages to the destination. So, we see that tradeoffs are made to deliver the message faster in RDTP by incurring extra transmissions to deliver the message faster. To reduce the transmission overhead, RDTP allows the vehicles to reply probabilistically to the forwarder solicit messages rather than always reply to such messages.

4.3 Description of Robust Data Transfer Protocol

Programs 1 & 2 show the pseudo code for a common routing module and two application specific modules.

Program.1. Pseudo code for common route module

```
distribution_type= 1(unicast),2(broadcast)
msg_type         = 1(query),2(reply),3(unicast),
                   4(solicit),5(data),6(net_control)
dest_type        = 1 (unicast),2(area),3(broadcast)
destination      = unique_destination_IP or target_area

send_message(distribution_type, msg_type,
             dest_type, destination)
{
    if (distribution_type==broadcast)
         transmit_message(broadcast, broadcast_addr);
         // destination description is in msg payload
    else if (distribution_type==unicast)
                next_hop=select_forwarder(destination);
                transmit_message(msg_type, next_hop);
}

select_forwarder(destination)
{
    if (forward_solicit_flag(destination)==TRUE)
         send(broadcast, solicit, broadcast_addr);
         // issue neighbor solicit message
         start solicit_timer;
         while(solicit_timer does not expire)
                 add response node to list;
                 wait for more responses;
         next_hop=forwarder_selection(dest_addr);
         forward_solicit_flag(destination)=FALSE;
    return next_hop(destination);
    // if next-hop info is available, return right away
}

receive_message(msg_recvd)
{
    if (msg_recvd.distribution_type==broadcast)
         if (msg_recvd.msg_type==query)
                 TMA_application(msg_recvd);
         else if (msg_recvd.msg_type==solicit)
                 RMTA_application(msg_recvd);
    else if (msg_recvd.msg_type==unicast &&
             msg_recvd.destination==my_id)
         if (port==TMA)
                 TMA_application(msg_recvd);
         else if (port==RMTA)
                 RMTA_application(msg_recvd);
    else if (msg_recvd.msg_type==unicast)
```

```
            next_hop=select_forwarder(
                         msg_recvd.destination);
            send_message(unicast, msg_recvd.msg_type,
                         msg_recvd.dest_type, next_hop);
    }
```

For the TMA scenario (shown in Program 2(a), the source issues a query, sets a timer and waits for the replies. The source sets the maximum number of vehicles' information, cnt_limt, that it will wait before generating an aggregate average speed estimate. Thus, either the timer expires or enough replies arrive to allow the source to generate an answer. As for intermediate nodes, they will only respond to any query that they receive the first time. In addition, they will check to see if they need to perform any actions upon receiving a new query (e.g. check if they are traveling along the right direction etc). If they need to respond, then, they re-broadcast the TMA_Query_Request, waits to count the number of downstream nodes that consider themselves as parents. If the node that receives a new query happens to be in the end zone, then it will generate a TMA_Query_Response with its own speed and sets cnt to 1. This node will send the response to its parent. Any intermediate node will wait for sufficient number of replies (or the reply timer expires depending whichever happens first) before generating a response back to its parent.

Program.2. (a) Pseudo code for TMA

```
TMA Source Node:
{
    send_message(broadcast, query, area, target_area);
    //issue a query
    start reply_timer;
    while (reply_timer not expired and reply_cnt<CNT)
          upon receiving TMA_Reply
                 record (speed, cnt) from TMA_Reply;
                 update reply_cnt (query_id);
    compute aggregated speed;
}

TMA Intermediate Node:
{
    while (1)
       upon receiving TMA_Query
          if (check_new_request(msg_recvd)==TRUE)
               // only process new requests
               in_target_area=check(msg_recvd.
                            target_area, cur_loc);
               if (eligibility_relay_check()==TRUE)
                       // need to relay the query?
                       cache_prev_hop(query_id);
                       send_message(broadcast,query,
                                    area, target_area);
```

```
                      start reply_timer;
              else if (check_end_zone()==TRUE)
                     //reply if near end of query zone
                     generate TMA_Reply(own speed,1);
                     send_message(unicast, reply,
                         unicast,prev_hop(query_id));
      upon receiving TMA_Reply
         record (speed, cnt) from TMA_Reply;
         update reply_cnt(query_id);
              if (reply_timer expires)
                     if (in_target_area==TRUE)
                           include own speed;
                           reply_cnt(query_id)++;
                     compute aggregated speed;
                     generate TMA_Reply(aggregated
                                  speed,reply_cnt);
                     send_message(unicast,reply,
                          unicast,prev_hop(query_id));
}
```

For the RMTA scenario, the source issues a forward solicit request. Upon receiving several responses, the source will select an appropriate forwarder (e.g. the one that has the longest contact duration among those nodes who respond). Each intermediate node that carries some messages in turn repeats this forwarder selection process until the message is delivered as shown in the pseudo code in Program 2 (b).

Program.2. (b) Pseudo code for RMTA

```
RMTA Source:
{
   if (forwarder_solicit_flag(destination)==FALSE)
       forwarder_solicit_flag(destination)=TRUE;
   send_message(unicast, data, unicast, destination);
}

RMTA Intermediate Node:
{
   while(1)
       upon receiving RMTA_forwarder_solicit request
          if (eligibility_check()==TRUE)
              generate RMTA_forwarder_solicit response;
              forwarder_solicit_flag(prev_hop)=FALSE;
              send_message(unicast, solicit, unicast,
                           prev_hop);
       upon receiving RMTA_forwarder_selection
          issue heartbeat to prev_hop;
          // heartbeat interval is long enough to avoid
          // high overhead but fast enough to pick a
```

```
            // new forwarder if the existing one leaves
        upon receiving Data
            send_message(unicast, data, unicast,
                             destination);
}

For all forwarders:
{
     listen for forwarder heartbeat;
     upon detecting loss of forwarder
          forwarder_solicit_flag(node_id)=TRUE;
}
```

5 Simulation Study

In this section, we first describe the simulation model that we implement using ns-2[10]. For TMA, we simulate a scenario similar to what is described in [4]. We use IEEE 802.11 radio with a peak data rate of 11 Mbps in our simulator. The transmission range is set to 250m. Our simulator allows us to change the road length, the average gap distances between vehicles, the number of lanes etc. In our simulator, once a vehicle leaves the road, a new vehicle enters the road. The speed of the vehicle is chosen uniformly between 10 and 30 m/s (thus the average speed of the vehicles is 20 m/s). The simulation time is set to 500 seconds. The query segment is fixed at 800 m. Each vehicle will respond to the query message with probability *p* (with default value of *0.67).*

The metrics used for the TMA scenario are (i) response time – this is the average time of a successful query/response transaction, (ii) the dropping rate – this is the percentage of unsuccessful queries, (iii) accuracy – this measures how close the estimated average speed is to the actual average speed of the vehicles in the region of interest, and (iv) efficiency [4] which measures the percentage of the number of exchanged query messages that were actually employed in calculating a result over the total number of query messages exchanged both in routing and inside the target location. The efficiency metric reported in [4] does not include the hello messages sent by the nodes so we include a new metric called control overhead per query. In Sections 5.1, we discuss the performance of RDTP and VITP as the query distance is varied. Section 5.2 discusses the performance of RDTP and VITP as the vehicle density is varied. In Section 5.3, we study the impact of varying the response probability on the data delivery performance.

For RMTA, we simulate a scenario similar to the one described in [5]. We have a highway of length 5000 m with three lanes. In the first scenario, we vary the speed from 40 mph to 70 mps. In the second scenario, we fix the speed to 55 mph and vary the vehicular density. In both scenarios, we compare our approach with the best approach described in [5]. The metrics we use for RMTA is the achievable throughput and the average message delivery latency. Section 5.4 discusses the performance comparison when we vary the free flow velocity and Section 5.5 discuses the performance comparison when we vary the vehicular density. Since we do not have access to the simulators in [4] & [5], the numbers we report for these protocols are extracted from the plots in their papers.

5.1 Effects of Query Distance D on TMA Performance

In Figure 5, we plots the response time versus the query distance for both VITP and RDTP. For the VITP plot, we use the results when the count limit is set at 20. The response time increases with the query distance for both protocols. The RDTP achieves better response time than VITP until a query distance of 4000 m after which VITP performs slightly better. With RDTP, each hop has a response expiry time of 5 ms so the minimum query response time is 5ms*number of hops traveled. With increasing query distance, the query response time increases. But with VITP, there is no minimum response expiry time so as the query distance increases, its query response time will be better than RDTP. Figure 6 plots the accuracy of the query results for both protocols. Our results indicate that RDTP can achieve similar accuracy as VITP. The key thing to note is that RDTP achieves similar performance results with much lower control overhead (refer to Table 1). Each vehicle only sends forwarder solicit request message when it has been chosen as a forwarder and hence it needs not send hello messages periodically.

Table 2 reports the dropping rates for different query distances. The forward dropping rate corresponds to the fraction of queries that are dropped due to the failure of query delivery while the backward dropping rate corresponds to the fraction of failed queries due to failure of having the query response delivered to the source. From Table 2, we see that both dropping rates increase with query distance for both protocols. RDTP has lower dropping rates than the VITP since RDTP indirectly uses multipath information. The lower dropping rates translate to higher query success rate for RDTP.

5.2 Effects of Vehicle Density on TMA Performance

Next, we study the impact of vehicular density on the performance of RDTP. In this set of experiment, we fix the query distance to 2000 m and change the vehicle density by changing the gap between consecutive vehicles on the same lane from 50 to 200 m. Table 3 tabulates the forward/backward dropping rates, accuracy of speed estimation and the packet utilization of RDTP and VITP (only for some statistics that are reported in [4]). The results show that the efficiency and the accuracy drop with increasing vehicular gap. The degradation in efficiency with increasing vehicular gap distance is more significant with VITP than with RDTP.

The response time (see Figure 7) increases with the gap for both protocols. Even though the response time for RDTP is slightly better, and we expect the query success rate for RDTP to be higher than that for VITP. From Table 3, we see that the forward and backward dropping rates of RDTP increase with increasing gap distance.

5.3 Impact of Response Probability *p* on the RDTP Performance for TMA

In this section, we vary the values of the response probability p and see how it impacts the RDTP performance for TMA and the accuracy of speed estimates. Table 4 summarizes our results. The results indicate that setting p to 0.67 gives an accuracy close to what is achieved using a p value of 1.

5.4 Effect of Free Flow Velocity on RDTP Performance for RMTA

For the first set of experiments on RMTA, we fix the vehicular density to 2000 vehicles/hour and vary the free flow velocity. In Figure 8, we plot the achievable throughput versus free flow velocity for both protocols. Our results show that RDTP can achieve higher throughput than Protocol 3(b) (the best protocol in [5]). Table 5 shows the utilization of both protocols. RDTP achieves higher utilization. We also plot the message delivery latency of both protocols in Figure 9. Our results show that RDTP achieves lower message delivery latency than Protocol 3(b).

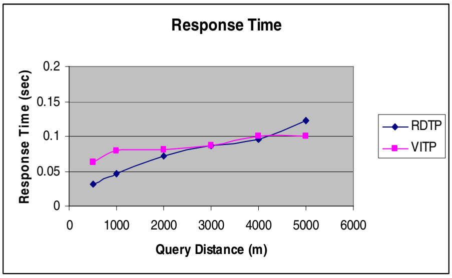


Fig. 5. Response time vs. query distance

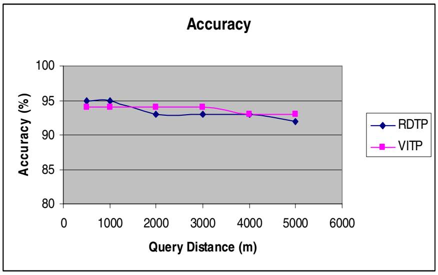


Fig. 6. Accuracy vs. query distance

Table 1. Control overhead for VITP and RDTP

query distance (meter)	control overhead (per second)	
	VITP	RDTP
500	750	9.0
1000	750	12.5
2000	750	15.4
3000	750	21.8
4000	750	26.2
5000	750	36.7

Table 2. Dropping rates vs. query distance

query distance (meter)	forward dropping (%)		backward dropping (%)	
	VITP	RDTP	VITP	RDTP
500	12	12	1	2
1000	18	17	1	3
2000	36	21	2	3
3000	51	26	3	5
4000	61	30	4	8
5000	66	37	4	10

Table 3. RDTP/VITP performance with different gaps

gap between consecutive nodes (meter)	packet utilization (%)		control overhead (message/second)		forward dropping rate (%)	backward dropping rate (%)	accuracy (%)
	VITP	RDTP	VITP	RDTP	RDTP	RDTP	RDTP
50	45	29	1500	19.2	10	2	95
100	14	24	750	15.4	21	3	93
150	8	15	500	12.7	28	5	92
200	6	12	375	9.1	33	5	90

Table 4. RDTP performance with TMA application vs. response probability

response probability	forward dropping rate (%)	backward dropping rate (%)	result accuracy (%)	packet utilization (%)	response time (second)	control overhead (msg/sec)
0.30	21	3	86	11	0.072	12.6
0.50	21	3	90	18	0.072	13.9
0.67	21	3	93	24	0.072	15.4
1.00	21	3	95	36	0.072	15.8

Table 5. Utilization: 2000 vehicles per hour

average velocity (mph)	protocol 3b	our protocol
40	0.91	0.95
45	0.91	0.96
50	0.90	0.95
55	0.91	0.98
60	0.89	0.97
65	0.85	0.98
70	0.88	0.97

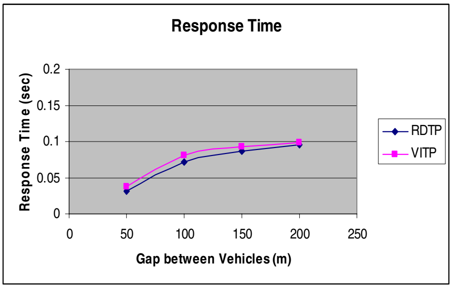


Fig. 7. Response time vs. vehicle gap

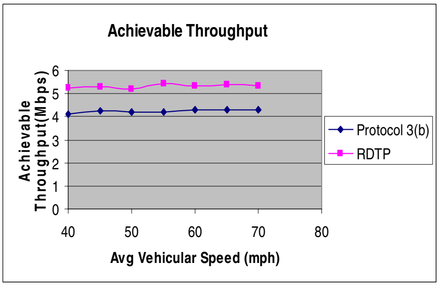


Fig. 8. Throughput vs. velocity

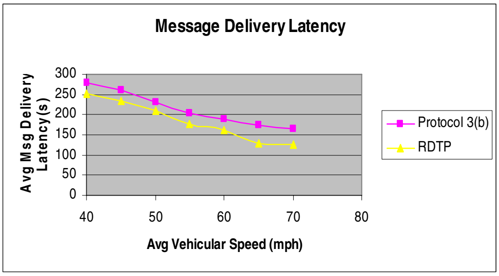


Fig. 9. Average latency vs. speed

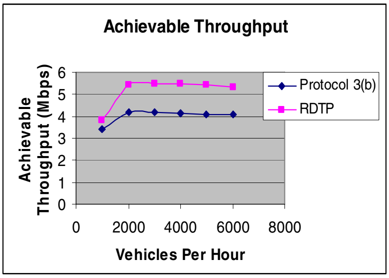


Fig. 10. Throughput vs. density

5.5 Effect of Vehicles Per Hour on RDTP performance for RMTA

Next, we fix the vehicle speed to 55 mph and vary the traffic levels in the roadway by varying the vehicles per hour. We plot the achievable throughput results for Protocol 3(b) in [5] and RDTP in Figure 10. Again, our results show that RDTP can achieve higher throughput than Protocol 3(b).

6 Conclusions

In this paper, we have described a robust data transfer protocol (RDTP) that we have designed to support two approach can provide comparably accurate speed estimate and higher query success rate with lower control overhead and response time than VITP. For RMTA, our scheme can achieve higher throughput than what can be achieved with the best scheme reported in [5]. This is just a preliminary work. We are in the process of developing a prototype with our data transfer protocol. In our prototype, we intend to have a voice-activated proxy that allows a driver to search for an alternate route upon finding congested area. We intend to carry out some field tests around Lehigh campuses. We also intend to add location-based broadcast messages into our system for business advertisement. Our goal is to design an intelligent transportation system that can benefit local community around our campuses.

References

1. The FleetNet Project, “www.et2.tu-harbug.ed/fleetnet”
2. http://www.its.dot.gov/index.htm.
3. http://www.leearmstrong.com/DSRC/DSCRHomeset.htm.
4. M. D. Kikaiakos etc, “VITP: An information transfer protocol for vehicular computing”, VANET 2005.
5. B. Petit etc, “Protocols for Roadside-to-Roadsie Data Relaying over Vehicular Networks”, Proceedings of IEEE WCNC, April 2006.

6. H. Wu, R. M. Fujimoto, R. Guensler, M. Hunter, “MDDV: Mobility Centric data Dissemination Algorithm for Vehicular Networks”, ACM workshop on vehicular adhoc networks, Oct, 2004
7. X. Yang, J. Liu, F. Zhao, N. Vaidya “A Vehicle-to-Vehicle Communicaiton Protocol for Cooperative Collision Warning”, ACM workshop on vehicular adhoc networks, Oct, 2004
8. R. Miller, Q. Huang, “An adaptive peer-to-peer collision warning system”, IEEE Vehicular Technology Conference (VTC), 2002
9. J. Tian, K.Rothermel, ”Building large peer-to-peer systems in highly mobile ad hoc networks: New challenges?” Technical Report, University of Stuttgart, 2002.
10. Network simulator ns-2, www.isi.edu/nsname/ns.

A Multi-layer Approach to Support Multimedia Communication in Mesh Networks with QoS*

Chungui Liu[1], Yantai Shu[1], Lianfang Zhang[1], Zenghua Zhao[1],
and Xiang-Yang Li[2,**]

[1] Dept. of Computer Science, TianJin University, PRC.
cgliumail@163.com,
{ytshu, lfzhang, zenghua}@tju.edu.cn
[2] Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA
xli@cs.iit.edu

Abstract. It is challenging to support multimedia transmissions over wireless networks, especially, wireless mesh networks, due to some natural resource constraints of wireless networks. In this paper, we investigate in detail some possible improvements on a number of layers to enable the multimedia transmission over wireless networks with QoS support. We implement all our protocols in some test-beds to study their real time performances. We first study a number of improvements of some existing routing protocols to support multimedia transmissions. Some new admission control and rate control mechanisms are studied and their performance gains are verified in our experiments. In our new cross-layer adaptive rate control (CLARC) mechanism, we adaptively change the video encoder's output bit rate based on the available network bandwidth to improve the quality of the received video. We design and implement a campus test-bed for supporting multimedia traffics in mobile wireless mesh networks. We also design a mobile gateway protocol to connect the MANET to Internet and a wireless LAN management protocol to automatically manage WLAN to provide some QoS.

1 Introduction

A number of protocols have been proposed to improve the throughput, reliability, security, or to reduce the delay or the energy consumption for wireless networks (*e.g.*, mesh networks). However, there are still a great number of technical challenges left. In particular, real-time multimedia traffic such as voice and video typically have high data rate requirements and stringent delay constraints, whereas wireless nodes generally have limited resources (especially in bandwidth). Networking tasks such as routing also become demanding due to the lack of an infrastructure and the frequent topology changes. Various interesting problems in supporting multimedia communication over wireless networks has previously been studied, *e.g.*, [2, 3, 4, 6, 7, 8]. To support the multimedia traffic with QoS guarantee, we need to investigate possible improvements on a number of layers. Adaptive link layer techniques (*e.g.*, [9, 10]) can be used to adjust the capacity of individual wireless links to support delay-constrained traffic, possibly

* This work was supported in part by NSFC under grant No. 60472078 and 90604013.
** This work is Partially supported by NSF CCR-0311174.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 392–403, 2006.

in multiple service classes. A congestion-optimized routing algorithm (*e.g.*, [6]) will provide multiple paths to real-time multimedia streams and will intelligently provide the reliability guarantee. An adaptive encoding method (*e.g.*, [11, 12]) that dynamically adjusts its bit-rate to the available bandwidth will improve the quality of the received video. An adaptive transmission rate control (*e.g.*, [13]) will also alleviate the link congestion in bandwidth limited wireless networks. So far, the majority of research on supporting multimedia over mesh networks has been done by simulations. Test-bed implementations are now typically sought to validate a proposed protocol. Current successful test-bed implementations (*e.g.*, [14, 15]) did not address the bandwidth demanding multimedia stream. Here we propose a multi-layer approach with test-bed implementation to support multimedia stream over mesh networks with QoS provision.

We choose a loosely coupled cross-layer design so a certain protocol independence between different networking layers are preserved. Unlike the majority of previous studies, we demonstrate the effectiveness of our methods via extensive simulations and, more importantly, via extensive experiments in a test bed. Our experiments show that our proposed new schemes indeed improve the QoS performance for multimedia traffics in wireless mesh networks. Our test-bed implementations systematically investigate how different techniques should interact with each other and how they should evolve over time to continuously adapt to the changing wireless environment and traffic demands. The major obstacle to support multimedia stream over wireless mesh networks is the unsteadiness of wireless links: the link capacity fluctuates over time and space in wireless networks. Thus, a fixed path is often not a good choice for multimedia stream. In this paper, we investigate several different routing approaches to enhance the multimedia experience over wireless mesh networks. As a running example, we identify several possible weaknesses (*e.g.*, misjudging link failures, non-negligible overhead caused by extensive ACK packets) of a routing protocol in supporting multimedia stream. For DSR, we revise the existing ACK mechanism to reduce the possible large ACK overhead while keep a certain level of transmission reliability.

The second main research is dynamic rate control for multimedia encoding. QoS is extremely difficult to achieve in wireless networks mainly due to the fluctuation of link bandwidth over time and space. For simple file services, we can dynamically adjust the sending rate to reflect the current available path bandwidth to the receiver. However, multimedia stream poses another difficulty: when the encoding rate is fixed, reducing the sending rate (*i.e.*, the number of frames sent per second) by the sender will greatly reduce the perceived multimedia quality if the sending rate is below a threshold. In this paper, we propose the cross-layer adaptive rate control (CLARC) that dynamically adjusts the encoding at the sender site, in addition to adjust the sending rate. We designed a mechanism (checking the number of ACKs received in a certain time period) to let the sender to dynamically adjust its encoding rate based on the path bandwidth. Our extensive test-bed experiments show that it greatly improves the perceived quality by the receiver. In our experiments, we assume that the sender (or several senders) will capture video (and audio) and then send the multimedia to Internet via a wireless gateway.

The rest of the paper is organized as follows. In Section 2, we study a number of possible improvements over some existing routing protocols to enhance their performances. In Section 3, we study how to use rate adaptive video transmission to improve the QoS performance. In Section 4 we briefly review the related works for multimedia communication over wireless mesh networks. We conclude our paper in Section 5.

2 Improved Routing Protocols

2.1 DSR Improvement to Support Multimedia

Dynamic Source Routing (DSR) [1] uses of hop by hop ACK mechanism to monitor and control link failure on route layer. It will regard the link as failing one, if the source hasn't received ACK for certain packets in some given time. Congestion, hidden terminal or channel instability can cause ACK lost, which will result in misjudging link failures by DSR. Therefore, a flooding route maintenance procedure will be initiated, which will further increase the delay. We first analyze problems existing in DSR-ACK mechanism, and illustrate the role ACK plays in performance improvement.

Link Failure Misjudgement: Judging link failures correctly is very important to the routing protocols' performance improvement. We conducted extensive experiments under different traffic loads using the original DSR. When the traffic is light, the DSR's old ACK scheme, that is, to acknowledge every data packet, can monitor the link status effectively and detect link failures in time. However, as the traffic load becomes heavier, the frequency of switching to new routes and initiating route discovery increases greatly, because DSR implementation mistakenly thought that the link was broken. And, to some extent, the upper layer application even cannot work at all due to the frequent route switching. To find out the reason, we deleted ACK mechanism from the code, and found that video transmission was much steadier in the same environment. In other words, in this case, DSR's assumption of link failure is not correct.

Analyzing the results of our experiments, we have found reasons of link failure misjudgement. When traffic is light, ACK_RTT (the time between the data packet is sent out and ACK is received) is small, within 10ms. As traffic load becomes heavier, ACK_RTT increases to a maximum of 5s! In DSR, ACK timeout value is a constant. So when the traffic is heavy enough, ACK_RTT will be greater than ACK timeout. So no matter how many times the data packets are retransmitted, the ACK will time out, which results in the DSR's assumption of link failure. The queuing time of ACK packets in upstream/downstream nodes' MAC layer takes up major part of ACK_RTT. So ACK_RTT changes with traffic load fluctuation. To judge link status correctly, dynamic ACK timeout value should be taken instead of fixed one. Futher, ACK could get lost just because of channel competition or disturbing signals, while the link still remains.

Route Maintenance Overhead: Bandwidth of wireless ad hoc networks is limited, thus, route maintenance overhead should be as small as possible. DSR protocol requires the last hop returns an ACK for every data packet in order to monitor the link status. Although ACK packets are as small as tens bytes, they also compete for channel usage and queue in the MAC layer. We have compared effective bandwidth of DSR without ACK and that with ACK, and found that the latter is only 70% of the original. ACK packets take up as much as 30% of the bandwidth. So, sending too many ACK packets decreases throughput and affects the transmission of regular traffic.

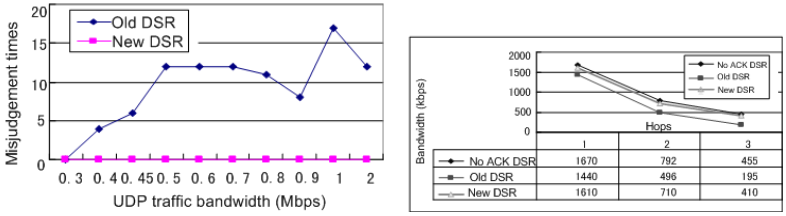
To do route maintenance, there are two other schemes. The first one removes ACK completely and sets upstream nodes into promiscuous mode to determine whether the downstream nodes have received and forwarded the packets or not. The second one uses MAC layer ACK to check whether the packets are received. The first one uses less bandwidth but it imposes great computation and power consumption burden on nodes. The second requires modifying drivers of wireless cards, which decreases the DSR protocol's generality. Thus, to decrease overhead caused by ACK in DSR protocol, we rely on decreasing the number of ACK packets sent. Our modified route maintenance mechanism and dynamic ACK algorithm can not only determine link failures more correctly, but also decrease overhead. The effective bandwidth is also increased. Compared with the bandwidth of DSR implementation without ACK, the effective bandwidth of UDP traffic has reached more than 98% of the latter, while the old DSR only gets 70% of the bandwidth of the latter.

Our Mechanism: Next we present a *dynamic ACK algorithm* to alleviate the problem stated above. This algorithm lets the upstream node adjust the number and time interval when packets need to be confirmed according to the link status. A downstream node will only confirm the packets that want to be confirmed. When the network load is light, the ACK can return in a few milliseconds, while it may be up to several hundreds in high load network. Because the difference is so large, the ACK overtime should change with the network situation. In new algorithm, we estimated next ACK overtime by the latest received ACK_RTT (denoted as LAST_ACK_RTT). Normally, the next ACK's return time will not be more than 2 times of LAST_ACK_RTT. Since the network load changes gradually, ACK_RTT will also fluctuate as the load changes. To reflect this gradual changing progress, we let ACK_TIMEOUT= $\frac{1}{2}$·ACK_TIMEOUT+$\frac{1}{2}$· 2·LAST_ACK_RTT.

In our new protocol, when a node sends packets, it will require ACK once in a certain time period, denoted by NEED_ACK_TIME, which also fluctuates as follows when the network load changes. Here NEED_ACK_TIME=JIFFIES(t)+LAST_ACK_RTT, where t is the current system time. When the network load is light, the ACK_RTT and LAST_ACK_RTT will be small, which will result in too many ACKs and consequently reduce the network throughput. Thus, we set a lower bound ACK_RTT_BOTTOM on them. Furthermore, in our modified DSR algorithm, only when k continuous ACKs are all over time (we set $k = 3$ in our experiments), the protocol will invalidate the route. Combining the dynamic DSR-ACK mechanism, this method can judge the route status more exactly.

Experiment Results on Improved DSR: We conducted extensive experiments to compare the performances of our modified routing protocol with the original DSR protocol. Our experimental data show that the DSR protocol with dynamic ACK scheme improves network bandwidth and the judgment of the link-state, and reduces maintaining cost. The traffic sending time of each experiment is 10s. The unit of ACK_RTT is millisecond (ms). ACK_RTT_BOTTOM is 100ms in the improved DSR.

The accuracy of link-state monitoring: To ensure the data's universality and comparability, all ACK_RTT data were selected from the source nodes of the 3-hop traffic.



(a) Link failure misjudgement

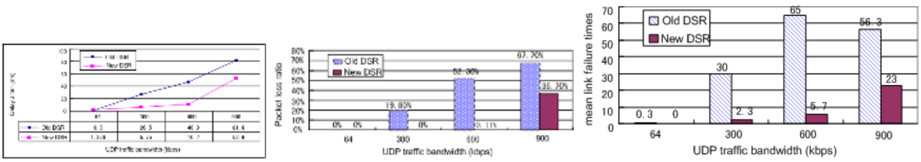
(b) bandwidth comparison

Fig. 1. Performance comparison of original DSR and modified DSR

Figure 1 (a) shows that perceived link-failure times that the improved DSR protocol has is much less than that of the original DSR. The stability of the improved DSR protocol is much better than the original one. It shows that the improved DSR protocol has fewer mistakes in judging the link failures. Each point in the figure is the mean perceived link-fail times computed with all node pairs in the network. The physical layer bandwidth of the middle nodes in the experiment is 2Mbps.

We compare the maximal path effective bandwidth between the original DSR protocol, the improved DSR protocol, and the DSR without ACK. We can find out the trend of the maximal effective bandwidth when the hop-count increases. To observe the transformation when effective bandwidth decreases with the hop number, we set bandwidth of all nodes to 2Mbps. Since in practice the hop account barely exceeds 3 in this experiment, we collect data of 1-hop, 2-hops and 3-hops. From Figure 1 (b) we can see that, when the hop number increases on a DSR path, the maximal path effective bandwidth decreases no matter the upper layer protocol is TCP or UDP, and no matter which kind of DSR protocol is used. We can also see from these figures, no matter how many jumps, the maximal path effective bandwidth achieved by improved DSR protocol is larger than the original DSR protocol. When the hop number increases, the advantage of the improved DSR protocol becomes more obvious.

Performance for Multimedia Traffics: We then conducted experiments of multimedia traffic to compare and analyze the performance of the DSR and improved DSR protocol. UDP traffic of different rate is used to simulate real voice and video flows. To measure the delay of packet delivery, ping packets of the same size as the multimedia traffic packets are sent simultaneously with the simulated voice and video flows. Then the destination node of UDP also takes part in the channel competition and consequently it has an impact on the sending of the simulated video traffic. This impact should be more remarkable when the traffic load of the link is heavy. The bandwidth range of voice traffic varies from 8kbps to 64kbps, and the video traffic needs at least 100kbps. To fit multi-hop transmission in ad hoc networks, MP4live is used to gather the video and the sending rate is between 100kbps and 450kbps. Moreover, considering the situation that two video flows will be sent simultaneously at one node, the node will have a load of 900kbps in the worst situation. In this group of experiments, Iperf is used to send UDP traffic from the source node to the destination node at a rate of 64kbps, 300kbps, 600kbps and 900kbps respectively. At the same time of sending the UDP traffic, ping packets of 1400 bytes, almost the same size as UDP packets, are sent.



(a) delay (b) delay jitter (c) packet loss ratio

Fig. 2. Performance comparison for multimedia traffics

Every reported experimental result of this group is the average value of three repeated experiments. In each experiment, the UDP traffic lasts for 60s. In all the experiments conducted here, the bandwidth of the three participating nodes is 2Mbps.

Figure 2 (a) shows that under a light load of emulated voice flows, the packet delivery delay of both DSR and improved DSR is very low, and DSR is even better than the improved version. The reason is probably that under light network load, the delay caused by the overhead of the rate limit and priority queue of improved DSR is relatively higher. However, as the traffic load increases, the delay of DSR rises quickly. The major reason is that for the DSR there are a great number of ACKs in the MAC queue, which makes the queues at each node very long and the channel competition caused by ACKs also cumbers data packet from sending as analyzed above. Both of the negative impacts will be more remarkable as the load becomes heavier. The above figure also shows that under the load of 600kbps, the delay of DSR is higher than 4 times of that of improved DSR. As the load increases further, although the delay of improved DSR rises to 211ms quickly, there is obvious superiority compared to 264ms of DSR. We have to point out that according to QoS standard prescribed by the latest VoIP technique, the delay of voice transmission should be less than 400ms. Here our improved DSR test bed is able to provide less than 16ms delay for the high quality voice flows of 64Kbps in the two hops situation. Thus, we believe that, even if several hops are added to the transmission path, the delay cannot exceed 400ms. Under the medium load, improved DSR is able to provide less than 38.3ms delay for the video flows. Since the requirement for delay of video flows is not as high as that of voice flows, the new DSR is able to guarantee the QoS requirement for delay of the real time voice and video traffic.

Delay jitter directly affects the playing delay of voice and video and the buffer size. Figure 2 (b) compares the delay jitter of the two DSR protocol under different multimedia traffic bandwidth. It shows that the delay jitter of DSR is smaller than that of improved DSR under light load of simulated voice flows. The major reason is that in the latter case packets are delayed by the rate limit according to the MAC queue status of that time before they are sent to the MAC queue, which causes some delay jitter. When the load is light, the jitter caused by rate limit is evident. As the load increase gradually, the delay jitter of the delivered packets increases in both DSR and improved DSR. However, the increasing rate of DSR is more rapid. Because when the load is relative heavy, DSR will begin to have link failure and packet loss more frequently. So except for the light load situation, the delay jitter of DSR is higher than improved DSR. Moreover, it is obvious that under medium load (between 300 to 600kbps), the delay jitter of improved DSR has the most superiority. Its average value is less than $1/4$ of that of DSR. Even under the relatively heavy load of 600Kbps, the delay jitter caused by improved DSR is only around 10ms.

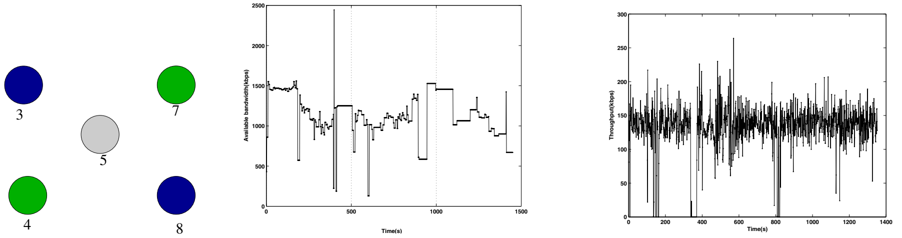
VoIP technique requires that when voice data packets are transmitted in the network, the loss probability should be less than 8%. Figure 2 (c) shows that both old DSR or improved DSR has no packet loss when transmitting the simulated voice flows. As the traffic load increases, the performance of improved DSR is quite good. Until the UDP traffic load rises to 600kbps, slight packet loss appears and the loss probability is only 0.11%. However, old DSR is hypersensitive to the increase of traffic load and the packet loss probability rises rapidly. Under 900Kbps traffic load, its packet loss probability is already as high as 67.7%, which is as high as two times of that of improved DSR. New DSR's lower packet loss probability can be attributed to following improvements: (1) the reduction of the number of ACKs (thus, with the same physical layer bandwidth, the maximal available bandwidth provided is much higher than old DSR); (2) the reduction of link failure judgments and then route change times is less than old DSR. Notice one route change will cause a great number of packet loss for the higher layer (3) the introduction of traffic class priority mechanism to improved DSR. During the process of finding a new route after link failure in old DSR, a number of data packets will be lost. While in improved DSR, though link failures could happen, there is no packet loss or only a few packets lost during the link failures. We also found that the link failures of new DSR are much less than those of DSR under the same traffic load. This can be mainly attributed to the new link probing algorithm of new DSR, which reduces the link-failure judgment times. caused by the protocol Observe that one of the major reasons for the high packet loss of old DSR protocol is that it misjudged too many link failures and the route thus changes frequently.

Although the reported results here are mainly based on DSR, we did implement the improvements over some other routing protocols (*e.g.*, AODV) also and the approaches generally apply to them.

2.2 Using BSR to Support Multimedia

When the primary path breaks, one way to improve the performance is to establish and maintain backup routes. A key advantage of using a backup path is the reduction of the frequency of route discovery flooding, which is recognized as a major overhead in on-demand protocols. We will study the performance improvement using backup source routing compared with routing using primary path only. The BSR algorithm, which extends DSR by selecting a backup path piggybacked with the primary path in the header of data packets to achieve the most reliable routes between each communicating mobile node. BSR is an extension of DSR, which uses the conception of backup route to improve the route reliability. Backup routes consist of the primary path, the shortest delay path (or the shortest hop-count path), and a backup path. BSR consists of two phases: (a) Route Discovery and (b) Route Maintenance. Route Discovery is only invoked when needed, and Route Maintenance operates only when the route is used actively to send individual packets.

Experiment Evaluation: Figure 3 reports our experimental study of BSR in a testbed. Here node 3 is receiving video sent from node 7 with rate 150kbps, and node 8 is receiving video from node node 4 with rate 300kbps. These 5 nodes are moving at



(a) test-bed setup (b) available bandwidth for 7 → 3 (c) Node 7's available bandwidth

Fig. 3. Performance of BSR

a walking speed as follows: nodes 3, 4, 7, 8 move east-west and node 5 moves south-north. We recorded the changes of routes and quality of the videos during the movement. For nodes pair $(7, 3)$, in our experiments, we found that the prime route $(7, 5, 3)$ is quite stable. We found route $(7, 4, 3)$ and route $(7, 3)$ as backup routes. We observed that there are 8 occasions the prime route fails, 28 occasions the backup routes fail, and 8 occasions both routes fail simultaneously. The data delivery ratio improves accordingly. Node 7 sent 22,192 packets, among which 21,590 packets (*i.e.*, 97.29%) have a valid route to node 3 at the time. Node 3 received 19750 packets, implying a delivery ratio 91.48%. See Figure 3 (b) for illustration. Although we set the network card in the mode of 5.5Mbps fixed rate, the achieved rate fluctuates at times. Sometimes the rate even dropped to 0, *i.e.*, there is no route from source to target. Figure 3 (c) shows the change of bandwidth in node 7. We can see the throughput of node 7 is always about 150kbps. But the throughput is 0 during the period between 15:58:25 and 15:59:11. The reason is that mpeg4ip can not send packets during that time and is not caused by route broken. Because of only a few route brokens, the transmission is quite stable.

3 Rate Adaptive Video Transmission over Ad Hoc Networks

Given a target bit rate, many video encoders can control their output stream's bitrate as close as to the target as possible using a rate control technique. This type of encoders is called adaptive video encoder. Of course, the higher the bit rate, the better the reconstruction video quality. However, over bandwidth-constrained ad hoc network, high source bit rate may lead to network congestion, hence more packets are dropped when they arrive after the deadline. Therefore, it is expected to allow the video encoder's output bitrate to be in accordance with the available network bandwidth. Ban [13] presented an adaptive rate control approach which would control the source bit rate with respect to the hop-counts. However, the source bit rate would be constant, if the hop-counts keep unchanged. In fact, the available bandwidth would vary with time even under the same hop-counts. We thus propose a cross-layer adaptive rate control mechanism, called CLARC. The basic idea behind CLARC is to adapt the video encoder's output bit-rate to the available bandwidth (network congestion and the routing information from the underlying network layer) to improve the quality of the video.

3.1 The Cross-Layer Adaptive Rate Control

The video will be transmitted using the RTP/UDP/IP protocol stack. We adopted the public domain H.263+ standard TMN8.0 as the video encoder. Its rate control model is modified to be capable of matching the output bitrate to the target bit rate for each frame. Before the encoder encodes a frame, it would call CLARC to get its target bit rate. There are two kinds of feedback information input to CLARC. One is the frame loss information from RTCP at the application layer; the other is the routing information from the routing protocol. According to the feedback information, CLARC would determine the target bit rate using the algorithm described next.

Frame loss information: We considered frame loss rather than packet loss for the following reasons. In wireless networks, the channel error is burst, which will cause one or several consecutive packets lost. This kind of packet loss should not be considered as network congestion indication. But it is difficult to differ it from packet loss caused by network congestion. In fact, one frame, especially Intra-frame, is always much longer than one MTU. In addition, the decoder can recover the error caused by some packets loss in one frame using error concealment technique. Therefore, we are allowed to feed back frame loss information. It makes CLARC robust to random error to some extent.

Routing information: We also use the information of hop-count and route breakage to facilitate the rate control [4]. Many ad hoc network routing protocols can provide the above information, such as DSR and AODV [5]. In this paper, DSR is adopted. A route breakage can also be detected by looking at the routing table information and/or control messages of the routing protocol.

3.2 The CLARC Algorithm

To detect the available network bandwidth, CLARC utilizes a probing method. Beginning from a small rate, CLARC would increase the rate gradually til the network is nearly congested. It responds quickly to the topology change by using cross-layer information. A key component in CLARC is network congestion decision. If the destination detects a frame loss, it will send NAK to the source. Otherwise, it will send ACK for each frame correctly received. If the source node receives n ACKs consecutively, the network is considered as being in good condition. If it receives n NAKs consecutively, and the route was not broken recently, the network is considered congested. Here n is a controllable parameter. We also designed a window-based structure to avoid network congestion. The window is an FIFO buffer. When a frame is sent, its timestamp is put into the window sequentially at the same time. When it is acknowledged, this frame and the ones before it will be cleared from the window. That is to say, the frames in the window are the ones that have been sent out but not acknowledged yet. The window length can indicate the degree of network congestion to some extent. A threshold CONG_AVOID is set. When the window length is longer than the CONG_AVOID, it warns the network congestion incoming. Algorithm 1 describes our CLARC method.

Here, the variables used in the above algorithm are

- **MIN_BIT_RATE** $R_{\min}$: It is the minimal bitrate of a video to achieve acceptable reconstruction quality. For CIF video (352×288 pixels), $R_{\min}$ is set to 100kbps.
- **MAX_BIT_RATE** $R_{\max}$: The maximum bitrate allowed under current network condition.

Algorithm 1. CLARC: Cross-Layer Adaptive Rate Control

```
Set $R_{\max}$ according to the hop count and set $R_{\min}$. Set $R_{cur} = R_{init}$.
if received $n$ ACKs then
    $R_{cur} = \min(R_{\max}, R_{cur} + n \cdot \delta)$. Here $\delta$ is the minimum rate increment, which
    we set to be $10kbps$.
else if received $n$ NACKs and no route brokens then
    $R_{cur} = \min(R_{\max}, R_{cur} - n \cdot \delta)$.
else if  window_length > CONG_AVOID then
    $R_{cur} = \min(R_{\max}, R_{cur} - \delta)$.
if hop count changes then
    $R_{cur} = \min(R_{\max}, R_{cur})$.
```

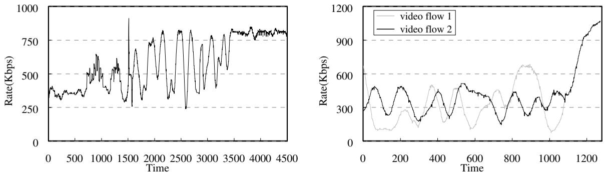
- **INITIAL_BIT_RATE** R_{init}: The initial bitrate. We set $R_{init} = (R_{\max} + R_{\min})/2$.
- **TARGET_BIT_RATE** R_{cur}: The bitrate to encode video based on network condition.

3.3 Test-Bed Implementations

To validate the performance of CLARC and its feasibility, we implemented an ad hoc video transport testbed with notebook computers and 802.11b WLAN cards. In the experiments, the maximum bandwidth was set to 11Mbps. The server has a camera and it captures live video (encoded using H.263+ TMN8.0 with 15 frame/s at CIF format) and sends streaming video to clients. We test CLARC using two different scenarios.

Scenario 1: In this scenario, we aimed to show how CLARC adapts the rate to the varying bandwidth. The network is formed by 4 laptops (as server, router$_1$, router$_2$, and client) placed along a line. The server communicates with the client though router$_1$ and router$_2$ with 3 hops. We let the client laptop move toward router$_1$ and finally stop there. As a result, the distance between the server and the client varied from 3 hops to 1 hop, and the available bandwidth also increases accordingly. In this experiment, we start one live video flow from the server to the client. Figure 4 (a) shows the results of the actual bit rate achieved. At the beginning, the bit rate is small at about 350Kbps. With the decreasing of hops, the available bandwidth increases and the bit rate also increases to 800Kbps at last. In the client, the received video remains fluent during the experiment (with reduced video quality at smaller bit rate). Without CLARC, the received video at clients paused many times and have mosaic many times also at a smaller bit rate.

Scenario 2: In this scenario, the network is composed of 5 nodes: two video servers placed in two left corners of a square, two video clients placed in the right corners of



(a) The bit rate at scenario 1 (c) Bit rates of 2 video flows

Fig. 4. The experimental results for CLARC

the square, and a router in the middle of the square. The server and the client cannot hear each other directly, but all of them can hear the router. Video flow 1 was transmitted from server 1 to the client 1 through the router, and video flow 2 is from server 2 to the client 2. Flow 1 starts early and also terminates early. Part of the experimental results are shown in Figure 4 (b). Using CLARC, the two video flows can share the wireless channel fairly well. When one flow stops, the other flow can increase its rate as much as possible to occupy the bandwidth, hence the received video also becomes better. As in scenario 1, CLARC makes the received video remain fluent during the experiment with a dynamically changed quality adapting to the bit rate.

4 Related Work

Supporting multimedia applications over wireless links has been one of the main fields of attention in the networking and video coding communities in the last decade. As the number of nodes of a wireless network grows, interference increases, reducing the achievable data rates. The capacity of a static wireless ad hoc network is shown to asymptotically vanish as the number of nodes increases in a landmark paper [16]. The art of video streaming over ad hoc wireless networks is still in its infancy, especially when addressed via a multi(or cross)-layer network design. In [12] path diversity in an 802.11 network combined with multistream coding of video is proposed and analyzed. Several cross-layer approaches have been suggested, as in [2, 3]. In [3], source, channel coding, packetization, and MAC layer retransmissions are performed together to reach optimized usage of the wireless channel. [4] proposed a cross layer feedback control mechanism that can allow the application layer to adapt itself to a dynamically changing network topology. Power and flow may also be allocated jointly through convex optimization to minimize network congestion [17]. [15] evaluates the ability of a wireless mesh architecture to provide high performance Internet access.

It is widely known that multiple paths routing will improve the throughput in wireless networks. Lin *et al.* [18] and Mao *et al.* [12] proposed to do video transport over ad hoc networks using multiple paths. Setton *et al.* [6] studied congestion-optimized scheduling for video over wireless ad hoc networks, and how to minimize distortion for multipath video streaming. Wu and Chuang [7] proposed a dynamic QoS allocation for multimedia in ad hoc wireless networks. Rate adaptive routing scheme for supporting multimedia with QoS was also studied in [10]. Fu *et al.* [8] proposed a transport protocol for supporting multimedia streaming in mobile ad hoc networks. To our best knowledge, none of the methods studied adaptive rate control for source video encoding.

5 Conclusion

In this paper, we investigated in detail some improvements on a number of layers to enable the multimedia transmission over wireless networks with QoS support. We implemented all our protocols in some test-beds to study their real-time performances. We first studied a number of improvement of some existing routing protocols to support multimedia transmission. Some new admission control and rate control mechanisms are studied and their performance gains are verified in our experiments. We also implemented mobile gateway protocol to enable Internet access of ad hoc nodes. To ease the burden of network management, we implemented several protocols to improve the network performance via SNMP. In our current implementation, the routing protocol is mainly based on DSR. We extend the function of original DSR by introducing the concept of mobile gateway. We implemented all functions that are necessary for Internet access in our test-bed, specifically, the support of DSR and NAT at gateway nodes, a gateway discovery protocol by non-gateway nodes in MANET. In our testbed, the gateway node can link to an AP or a base station of a cellular network (*i.e.*, Internet by GPRS). We conducted extensive experiments to study the performance and the experimental results are not included here due to space limit.

References

1. JOHNSON, D. B., AND MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, Imielinski and Korth, Eds., vol. 353. Kluwer Academic Publishers, 1996.
2. SETTON, E., YOO, T., ZHU, X., GOLDSMITH, A., GIROD, B.: Cross-layer design of ad hoc networks for realtime vdeo streaming. In: Magzine of Wireless Communications. (2005)
3. van der Schaar et al., M.: Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 wlans. IEEE JSAC **21** (2003) 1752–1763
4. Gharavi, H., Ban, K.: Dynamic adjustment packet control for video communications over ad-hoc networks. In: Proc. IEEE ICC. (2004) 3086–3090
5. PERKINS, C., AND E.M.ROYER. Ad-hoc on demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Appl.,* (1999), pp. 90–100.
6. Setton, E., Zhu, X., Girod, B.: Congestion-optimized scheduling of video over wireless ad hoc networks. In: IEEE Int. Symp. Circuits and Sys. (2005)
7. Wu, H., Chuang, P.: Dynamic qos allocation for multimedia ad hoc wireless networks. ACM Mobile Networks and Applications (2001) 377–384
8. Fu, Z., Meng, X., Lu, S.: A transport protocol for supporting multimedia streaming in mobile ad hoc networks. IEEE JSAC **21** (2003)
9. Wang, L., Zhang, L.F., Shu, Y.T., Dong, M., W.Yang, O.W.: Adaptive multipath source routing in wireless ad hoc networks. (In: IEEE ICC'01)
10. Kim, Y., Ryu, J., Cho, D.: A novel adaptive routing scheme for the qos-based multimedia services in mobile ad-hoc networks. In: IEEE VTC. (2001)
11. Srinivasan, M., Chellappa, R., Burlina, P.: Adaptive source-channel subband video coding for wireless channels. In: First IEEE Workshop on Multimedia Signal Processing. (1997) 407–412
12. Mao, S., Lin, S., Wang, Y., Panwar, S.S., Li, Y.: Video transport over ad hoc networks: Multistream coding with multipath transport. IEEE JSAC **21** (2003) 1721–1737
13. H.Gharavi, K.Ban: rate adaptive video transmission over ad-hoc networks. (IEEE electronics letters, vol. 40, no. 19, 16th Sept. 2004)
14. Biswas, S., Morris, R.: Opportunistic routing in multi-hop wireless networks. In: Proceedings of the ACM SIGCOMM '05 Conference. (2005)
15. Bicket, J., Aguayo, D., Biswas, S., Morris, R.: Architecture and evaluation of an unplanned 802.11b mesh network. In: ACM MobiCom. (2005)
16. Gupta, P., Kumar, P.: Capacity of wireless networks. Technical report, University of Illinois, Urbana-Champaign (1999)
17. Xiao, L., Johansson, M., Boyd, S.: Simultaneous routing and resource allocation via dual decomposition. IEEE Trans. Commun. **52** (2004) 1136–1144
18. Lin, S., Wang, Y., Mao, S., Panwar, S.: Video transport over ad-hoc networks using multiple paths. In: IEEE International Symposium on Circuits and Systems. (2002) 57–60

Trust Extended Dynamic Security Model and Its Application in Network

Xiaofei Zhang[1,5], Fang Xu[2], Yi Liu[3,4], Xing Zhang[3,5], and Changxiang Shen[4,5]

[1] State Key Laboratory of Information Security,
Graduate University of Chinese Academy of Sciences, Beijing, 100049, China
[2] Institute of Commanding and Automatization,
Academe of Navy Equipment, Beijing 100036, China
[3] Institute of Electronic Technology,
Information Engineering University, Zhengzhou 450004, China
[4] Computer Technology Institute of Navy, Beijing 100841, China
[5] Trusted Computing Lab, Institute of Computer Science and Technology,
Beijing University of Technology, Beijing 100022, China
zhxf04b@mails.gucas.ac.cn

Abstract. With the development of ad hoc networks, it is one of the most important problems to protect the security of information flows between the nodes. The paper applies the technique of trusted computing to designing the Trust Extended Dynamic (TED) security model. TED model protects the security of information flows between the nodes built on trusted platforms. The security model is based on BLP and Biba models. Due to introducing the concept of reliability and the functions for the trustworthy state measurements, the trustworthiness of subjects and some objects is monitored, and the accessing ranges of subjects are adjusted at runtime. Through these means, the ability of TED security model to resist running attacks is enhanced. The formal description of TED model is specified in detail, and its security properties are described. The paper also introduces the application of the model in the Trusted Web Server (TWS).

Keywords: Trusted computing, Access control, Security model, Reliability, Measurement.

1 Introduction

Ad hoc networks are dynamically formed anywhere without any hardware infrastructure. With the development of wireless communication and terminal techniques, ad hoc networks have been widely used in commerce. As the networks constantly change their topologies, the security problems of them are more than the normal networks. Unprotected ad hoc networks face several threats, such as wiretapping, playback attacks and so on. Because the considerations on the security of routing protocols are limited, some malicious or unsafe nodes may be contained in the networks. The confidentiality and integrity of the information

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 404–415, 2006.

in ad hoc networks may be damaged by them. Therefore, controlling information flows between the nodes in ad hoc networks improves the security of the networks.

Access control prevents unauthorized information flows, and it enforces security polices based on security models. But current security models have some shortages as follows. Above all, process (subject) in conventional security models inherits the security properties of the user who runs it, but the trustworthiness of the running process itself and input data is ignored. So these security models couldn't resist running malicious codes. Moreover, most of these models don't satisfy the requirements of some high-assurance environments, in which confidentiality and integrity are needed at the same time.

With the increasing demands for information protection, the technique of trusted computing is promoted by Trusted Computing Group (TCG), which creates a foundation of trust for software processes based on a trusted component named Trust Platform Module (TPM), in the form of build-in hardware. Trusted Platform (TP) enhances the security of computing terminal by improving platform authentication, platform integrity, protected storage, and authenticated booting. Now, the technique has been supported by above 200 companies all over the world. Every node founded on a TP should enhance the security of ad hoc networks. Although trusted computing has improved the security of current computing terminals, it only detects the integrity of system configurations and doesn't control the confidentiality and integrity of information flows in a computing system.

On one hand, security models protect legitimate information flows and prevent unauthorized accesses, which remedy the disadvantage of trusted computing which cannot confine the unsafe information flows. On the other hand, trusted computing assures that the access control module is loaded correctly, and it can supervise the running environments of a TP. Thus designing a security model with trusted computing can reflect the current trustworthiness of the system environments, which makes the model more reasonable and secure than the traditional models.

The paper puts forward TED security model, which defends the security of information flows between the nodes built on TPs in ad hoc networks. The model is founded on BLP [1] and Biba [2] models. Besides protecting the confidentiality and integrity of system objects, the concept of reliability and the functions for trustworthy state measurements are promoted, which monitor the trustworthiness of subjects and objects. And it also adjusts the accessing ranges of subjects according to the security rules. Then the ability of TED security model to stand against running attacks is enhanced. The design notion, components and security rules of TED model are specified in detail. And its security properties are proved. The transformation of the trustworthy states of requests is also introduced in the paper. The usage of the model in TWS is described, including the structure and processing course of TWS.

The paper organizes as follows. Section 2 analyzes the attacks on system processes and important files, and it shows the necessity of checking the trustworthiness of running environments. Section 3 illuminates the notion, components and security rules of TED security model, and its security properties are proved. The transformation of the trustworthy states of requests is also described. Section 4 provides the application of TED model in TWS, including its structure and processing course. The related work is specified in section 5. Section 6 concludes the paper and presents the future work.

2 Analysis of Attacks

Various types of runtime attacks, such as stack and heap overflow, Trojan horse, malicious scripts, are often encountered. As the frangibility and limited protection in ad hoc networks, these attacks damage the nodes in the networks more easily than normal network nodes. Because the trustworthiness of running processes is not detected, processes can run as normal even if aggressive codes have been embedded in them. On the other hand, the trustworthiness of accessed objects is not validated. So some important system files and configurations may be modified without any notifications. These make attackers to enter and control the system easily. In this section, we analyze two types of attacks from the view of trustworthiness checking: Rootkit and system files attack. These are representative examples of such attacks involving processes and objects respectively.

2.1 Rootkit Attack

Rootkit is one of the most popular activities of serious intrusions. Once intruders obtain root privileges of a terminal running SunOS, UNIX or Linux, they will replace some system files to concealed their trails, and steal sensitive information in the systems.

The most simple and classic example of Rootkit may implement as following steps. To begin with, an intruder obtains a copy of the source code to /bin/login for the version of Linux the target host is running. And he edits its source code to include a password that will let him login as root. Then he replaces the original /bin/login with the new /bin/login. After login in the host, he will replace some system binaries including netstat, ifconfig, ps, ls, inetd, i.e.. So his activities can be concealed. If the intruder is far more skilled in his attacks, he will modify the target kernel, and kernel Rootkits can hide files, directories, and processes.

It is the most useful method to avoid Rootkit that the trustworthiness of processes is inspected regularly. In some cases, even OS kernel should be detected before it starts. If the integrity measurements are different from the normal values, it may be confirmed that the systems have been destroyed, and the destructive processes should be terminated.

2.2 Attack on System Files

Init is a user mode process in Linux. When operating system starts up, init starts all the other processes and services. It is one of the most important processes in a system. Init reads the /etc/inittab file, which plays a crucial role in the boot sequence. It supplies the script to the init command's role as a general process dispatcher. If the runlevel in inittab was altered to 0 or 6 (system halt or system reboot), the system should not start as normal, and it can be viewed as a kind of Denial of Service (DOS).

To prevent it, the trustworthiness of inittab should be examined to judge whether the important contents of inittab were modified during booting. When some differences were found, the current booting shouldn't continue unless the destroyed inittab was covered with the backup file.

3 Formal Description of TED Security Model

As stated in the above section, without checking the trustworthiness of running processes and accessed objects, quite a number of malicious programs can intrude upon a system. Then the sensitive information in it may be modified or accessed with on permission. In order to protect the information flows between the nodes in ad hoc networks, we propose a new security model in this section, which imports the concept of reliability to reflect the trustworthiness of subjects and objects. The ability of the model to resist runtime intrudes is enhanced.

3.1 Notion of Design

BLP (Bell-La Padula) model [1] proposed in 1973 is based on finite state machine. The model is strictly proved and has been widely adapted to various secure applications. However, it only concerns confidentiality of system information. In 1975, another model named Biba [2] is promoted to protect the integrity of information flows, which is a mathematical dual of BLP model. Due to the importance of integrity to many applications, especially network applications, TED security model is founded on the integration of BLP and Biba model.

According to the definitions of BLP model, the security properties of a process inherit the properties of the user who invokes it. Actually, a user in a high security level may start a process that isn't trusted. If the integrity of the process was damaged, and it contained some malice codes, the security of the resources in the system shouldn't be ensured. Whether the processes can perform a user's instructions dependably affects the safety of a system directly. Moreover, if a trusted process accessed an object including a Trojan horse or the other kinds of intrusive programs, the safety of the information in the system should also be damaged. With the increasing complexities of application environments, the descriptions of the reliability of subjects and objects cannot be ignored in a security model.

Therefore, the concept of reliability introduced in TED describes the trustworthy degrees of subjects and objects. Reliability comprises the reliability value of a subject (or an object) and its current state of trustworthiness. The functions of trustworthy state measurements monitor the trustworthiness of current subjects and some objects which are important system files and have fixed contents.

If their integrity was destroyed, they should not be trusted any more, and any requests concerning them should be rejected. If a subject requested to access an undecided object, the accessing range of the subject should be reduced. After the operation, the current state of the subject turns to be undecided while its original state is trusty. Similarly, if a trusted object was requested by an undecided subject, the accessing range of the undecided subject should be dwindled. After the access, the current state of the object changes to be undecided while its original state is trusted. As a result, the ability of TED model to prevent runtime attacks is improved.

3.2 Definitions

Suppose S is the set of subjects, $S^T$of trusted subjects. Then $S' = S - S^T$ is the set of subjects that aren't trusted. $TS = \{trusty, unchecked, untrusty\}$ is the set of the trustworthy states. O represents the set of objects.

Definition 1. ***Trustworthy State Measurement to Subject:*** $im_s : S \rightarrow TS$ *is a mapping of the set of subjects to* TS*.* $\forall s \in S$*, if* $im_s(s) = trusty$*, the subject* s *is trusted; if* $im_s(s) = unchecked$*,* s *is undecided; if* $im_s(s) = untrusty$*,* s *isn't trusted.*

Definition 2. ***Value of Reliability:*** $D = \{d_1, d_2, \ldots, d_n\}$ *is the set of the values of reliability.* $\forall d \in D$ *is not a negative integer, which denotes the trusted degree of a subject or an object.*

Definition 3. ***Reliability of Subject:*** $R_s = \{r_{s_1}, r_{s_2}, \ldots, r_{s_n}\}$ *is the set of reliability of subjects.* $\forall s_i \in S$*,* $r_{s_i} = (d_{s_i}, ts_{s_i})$ *specifies the reliability of* s_i *is* r_{s_i}*. And* $d_{s_i} \in D$ *is the reliable value of the process* s_i *itself. If* $s_i \in S^T$*,* d_{s_i} *is the biggest reliable value in the system.* $ts_{s_i} \in TS$ *is the current trustworthy state of* s_i*.*

It must be pointed out that if and only if $s \in S_T$ and $im_s(s) = trusty$, s is a trusted subject at runtime.

Definition 4. ***Trustworthy State Measurement to Object:*** $im_o : O \rightarrow TS$ *is a mapping of the set of objects to* TS*.* $\forall o \in O$*, if* $im_o(o) = trusty$*, the object* o *is trusted; if* $im_o(o) = unchecked$*,* o *is undecided; if* $im_o(o) = untrusty$*,* o *isn't trusted.*

Definition 5. ***Reliability of Object:*** $R_o = \{r_{o_1}, r_{o_2}, \ldots, r_{o_n}\}$ *is the set of reliability of objects.* $\forall s_i \in S$*,* $r_{o_i} = (d_{o_i}, ts_{o_i})$ *specifies the reliability of* o_i *is* r_{o_i}*. And* $d_{o_i} \in D$ *inherits the reliable value of the subject who creates* o_i*.* $ts_{o_i} \in TS$ *is the current trustworthy state of* o_i*. If an object doesn't have the fixed contents, its trustworthy state is denoted as "unchecked".*

s is a process which is running. The security properties of s comprise its confidentiality level, integrity level, categories, and reliability. Its security properties except the reliability inherit the user's, who invokes the process.

o is a passive computer system repository which is used to store information. Data, files, and I/O devices can all be viewed as objects. The security properties of an object comprise its confidentiality level, integrity level, categories, and reliability. Its security properties are determined by the security label rules in system.

$A = \{r, w, a, e\}$ is the set of access modes. It explains the operations which a subject can request. It has the same meaning as [1].

Definition 6. *L_f is the set of confidentiality levels, which is expressed by the set of positive integrities. If $\forall m, n \in L_f$, and $m > n$, the confidentiality level m is higher than the confidentiality level n. Suppose $SystemHigh_f$ is the highest confidentiality level in L_f, and $SystemLow_f$ is the lowest in L_f.*

Definition 7. *L_i is the set of integrity levels, which is expressed by the set of positive integrities. If $\forall m, n \in L_i$, and $m > n$, the integrity level m is higher than the integrity level n. Suppose $SystemHigh_i$ is the highest integrity level in L_i, and $SystemLow_i$ is the lowest in L_i.*

Definition 8. ***System State:*** *$\forall v \in V, v = (B \times M \times F \times I \times K \times T \times H)$ is a system state.*

*$B = P(S \times O \times A)$**:** $\forall b \in B$ records current access of subjects to objects in various modes.*

Access Matrix *M**:** describes the access modes that a subject can operate on an object, which embodies discretionary security.*

Confidentiality Level Vector *$F = (f_s, f_c, f_o)$**:** $f_s(s)$ is the maximal confidentiality level of the subject s. $f_c(s)$ is the current confidentiality level of the subject s, and $f_s(s) \geq f_c(s)$. $f_o(o)$ is the confidentiality level of the object o.*

Category Function *K**:** $k(s)$ is the categories of a subject s. $k(o)$ is the categories of an object o.*

Reliability Vector *$T = (t_s, t_o)$**:** $t_s(s)$ is the reliability of a subject s. $t_o(o)$ is the reliability of an object o.*

Hierarchy *H**:** H represents objects hierarchy, and it can be a collection of rooted, directed trees and isolated points.*

Definition 9. ***Function of Adjusting Confidentiality Level*** *$\delta : D \to L_f$**:** $\forall s \in S, o \in O, \delta(d_s, d_o)$ represents the current adjusting confidentiality level l_f while the reliable value of s is d_s and the reliable value of o is d_o. l_f satisfies $f_c(s) - \delta(d_s, d_o) \geq SystemLow_f$ and $f_c(s) + \delta(d_s, d_o) \leq SystemHigh_f$.*

Definition 10. ***Function of Adjusting Integrity Level*** *$\lambda : D \to L_i$**:** $\forall s \in S, o \in O, \lambda(d_s, d_o)$ represents the current adjusting integrity level l_i while the reliable value of s is d_s and the reliable value of o is d_o. l_i satisfies $i_c(s) - \lambda(d_s, d_o) \geq SystemLow_i$ and $i_c(s) + \lambda(d_s, d_o) \leq SystemHigh_i$.*

3.3 Security Rules

Rule1 Trust Extended Discretionary Security: A state $v = (b \times M \times f \times i \times k \times t \times H)$ satisfies the Trust Extended Discretionary Security, iff $(s_i, o_j, x) \in b \Rightarrow x \in M_{ij}$, and $ts_{s_i} \neq untrusty$, and $ts_{o_j} \neq untrusty$.

Rule2 Trust Extended Simple Security: A state $v = (b \times M \times f \times i \times k \times t \times H)$ satisfies the Trust Extended Simple Security, iff $(s, o, x) \in b \Rightarrow$

a) $x = e$, and $ts_s \neq untrusty$, and $ts_o \neq untrusty$;

or b) $x = r$, and $f_s(s) \geq f_o(o)$, and $k(o) \subseteq K(s)$, and $ts_s \neq untrusty$, and $ts_o \neq untrusty$;

or c) $x = a$, and $i_s(s) \geq i_o(o)$, and $k(o) \subseteq K(s)$, and $ts_s \neq untrusty$, and $ts_o \neq untrusty$;

or d) $x = w$, and $f_s(s) \geq f_o(o)$, and $i_s(s) \geq i_o(o)$, and $k(o) \subseteq K(s)$, and $ts_s \neq untrusty$, and $ts_o \neq untrusty$.

Rule3 Trust Extended Reading Security: A state $v = (b \times M \times f \times i \times k \times t \times H)$ satisfies the Trust Extended Reading Security to $S' \subset S$, iff $(s, o, r) \in b \Rightarrow$

e) $f_c(s) \geq f_o(o)$, and $i_c(s) \leq i_o(o)$, and $k(o) \subseteq k(s)$, and $ts_s = trusty$, and $ts_o = trusty$;

or f)$ts_s = unchecked$ or $ts_o = unchecked$, and satisfies the following conditions:

(1) $f_c(s) - f_o(o) \geq \delta(d_s, d_o)$, and $i_o(o) - i_c(s) \leq \lambda(d_s, d_o)$, and $k(o) \subseteq k(s)$;

(2) $b' = b \cup (s, o, r)$, and $v' = (b', M, f, i, k, t', H)$, and $t' = (t'_s, t'_o)$,

if $ts_s = trusty$, then $ts'_s = unchecked$, i.e. $t'_s(s) = (d_s, unchecked)$, else $t'_s = t_s$;

if $ts_o = trusty$, then $ts'_o = unchecked$, i.e. $t'_o(o) = (d_o, unchecked)$, else $t'_o = t_o$.

Rule4 Trust Extended Appending Security: A state $v = (b \times M \times f \times i \times k \times t \times H)$ satisfies the Trust Extended Appending Security to $S' \subset S$, iff $(s, o, a) \in b \Rightarrow$

g) $f_c(s) \leq f_o(o)$, and $i_c(s) \geq i_o(o)$, and $k(s) \subseteq k(o)$, and $ts_s = trusty$, and $ts_o = trusty$;

or h)$ts_s = unchecked$ or $ts_o = unchecked$, and satisfies the following conditions:

(1) $f_o(o) - f_c(s) \leq \delta(d_s, d_o)$, and $i_c(s) - i_o(o) \geq \lambda(d_s, d_o)$, and $k(s) \subseteq k(o)$;

(2) $b' = b \cup (s, o, a)$, and $v' = (b', M, f, i, k, t', H)$, and $t' = (t'_s, t'_o)$,

if $ts_s = trusty$, then $ts'_s = unchecked$, i.e. $t'_s(s) = (d_s, unchecked)$, else $t'_s = t_s$;

if $ts_o = trusty$, then $ts'_o = unchecked$, i.e. $t'_o(o) = (d_o, unchecked)$, else $t'_o = t_o$.

Rule5 Trust Extended Writing Security: A state $v = (b \times M \times f \times i \times k \times t \times H)$ satisfies the Trust Extended Writing Security to $S' \subset S$, iff $(s, o, w) \in b \Rightarrow$

i) $f_c(s) = f_o(o)$, and $i_c(s) = i_o(o)$, and $k(o) = k(s)$, and $ts_s = trusty$, and $ts_o = trusty$;

or j)$ts_s = unchecked$ or $ts_o = unchecked$, and satisfies the following conditions:

(1) $f_c(s) = f_o(o)$, and $i_c(s) = i_o(o)$, and $k(o) = k(s)$;

(2) $b' = b \cup (s, o, w)$, and $v' = (b', M, f, i, k, t', H)$, and $t' = (t'_s, t'_o)$,

if $ts_s = trusty$, then $ts'_s = unchecked$, i.e. $t'_s(s) = (d_s, unchecked)$, else $t'_s = t_s$;

if $ts_o = trusty$, then $ts'_o = unchecked$, i.e. $t'_o(o) = (d_o, unchecked)$, else $t'_o = t_o$.

3.4 Theorems

Theorem 1. *The reading range of a subject in the undecided state is the subset of its reading range in the trusty state.*

Proof. Argue by contradiction. Contradiction yields the proposition " the reading range of a subject in the undecided state is not the subset of its reading range in the trusty state.".

Suppose $Range_r$ is the reading range of s in the trusty state, $Range'_r$ of s in the undecided state. Then $Range_r \subset Range'_r$ can be concluded.

Suppose $s \in S^T$. Then $Range_r \subset Range'_r$ contradicts the Trust Extended Simple Security, which concludes $Range_r = Range'_r$.

Suppose $s \in S'$. According to the rule of Trust Extended Reading Security, the categories of s in the undecided state are equal to its categories in the trusty state. So the range changes of confidentiality and integrity levels need to concern.

While s is trusty, suppose $Range_r|_f$ is the reading range of confidentiality level, and $Range_r|_i$ is the reading range of integrity level. According to the rule e, for $\forall o \in O'$, $Range_r|_f = \{SystemLow_f, \ldots, f_c(s)\}$ and $Range_r|_i = \{i_c(s), \ldots, SystemHigh_i\}$ can be educed.

While s is undecided, suppose $Range'_r|_f$ is the reading range of confidentiality level, and $Range'_r|_i$ is the reading range of integrity level. According to the rule f, for $\forall o \in O'$, $Range'_r|_f = \{SystemLow_f, \ldots, f_c(s) - \delta(d_s, d_o)\}$ and $Range'_r|_i = \{i_c(s) + \lambda(d_s, d_o), \ldots, SystemHigh_i\}$ can be concluded.

Suppose $Range_r|_f \subset Range'_r|_f$, then it leads to $f_c(s) < f_c(s) - \delta(d_s, d_o)$, i.e. $\delta(d_s, d_o) < 0$.

On account of $\delta(d_s, d_o) \in L_f$, $\delta(d_s, d_o) < 0$ contradicts definition 9 in which L_f is a set of positive integrities.

The argument is complete. □

Theorem 2. *The appending range of a subject in the undecided state is the subset of its appending range in the trusty state.*

The proof is similar to theorem 1. So it isn't specified in detail.

The above theorems show that TED security model adjusts the accessing ranges of a subject. While a subject is trusty, its accessing range is bigger than that in the undecided state. If it isn't trusty, it can access nothing. So the damage of runtime attacks is depressed.

3.5 Transformation of Trustworthy States of Requests

Every request in an operating system contains a subject, a requested object and a type of access mode. The current trustworthy states of the subject and the object in a request form the trustworthy state of the request.

Suppose s_{trusty} is a trusted subject, and $s_{unchecked}$ is an undetermined subject, and $s_{untrusty}$ is a subject that isn't trusted. Similarly, o_{trusty} is a trusted object, and $o_{unchecked}$ is an undetermined object, and $o_{untrusty}$ is an object that isn't trusted. Then every pair of a subject and an object has five trustworthy states of a request: (s_{trusty}, o_{trusty}), $(s_{unchecked}, o_{unchecked})$, $(s_{trusty}, o_{unchecked})$, $(s_{unchecked}, o_{trusty})$,$(s_{untrusty}, *)/(*, o_{untrusty})$. Here, $*$ denotes every possible trustworthy state of a subject or an object. Let a, b, c, d, e denote the above trustworthy states of a request respectively.

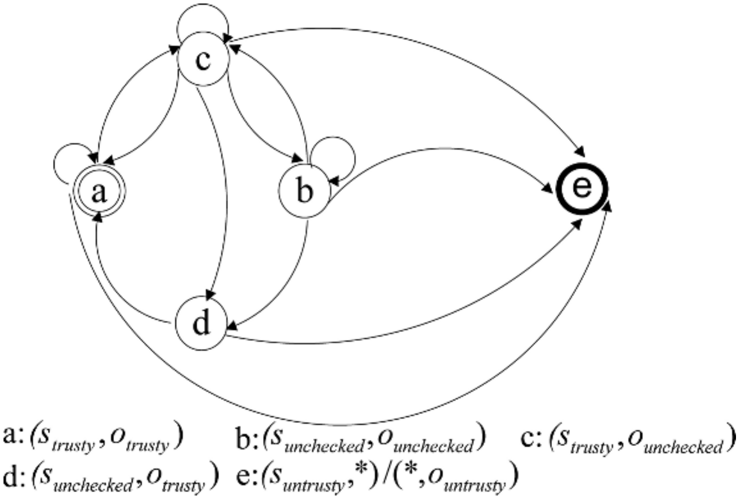

c
a
b
e
d
a: (s_{trusty}, o_{trusty})
b: $(s_{unchecked}, o_{unchecked})$
c: $(s_{trusty}, o_{unchecked})$
d: $(s_{unchecked}, o_{trusty})$
e: $(s_{untrusty}, *)/(*, o_{untrusty})$

Fig. 1. Transformation of trustworthy states of requests

In accordance with the security rules in section 3.3, the transformation of trustworthy states of requests is shown in figure 1. In TED model, every subject starts from a trusted state. Suppose a subject firstly requests to access a trusted object, i.e. the trustworthy state of the request starts from the state a. If the subject still accesses a trusted object, the subject's trustworthy state doesn't change, and the trustworthy state of the request doesn't transfer, too. While the subject tries to access an unchecked object, the state enters c. After operating the requested object, the subject is undetermined. If the subject requests to access a trusted object, the trustworthy state of the request changes to d. If an unchecked object is requested, the state enters b. The trustworthiness of the subject is measured after it sends a request. If the subject is trusted, the trustworthy state of the request may change to state a or c. If not, the state enters e, and the subject stop running. If the trustworthiness of an object having fixed contents is measured after it is requested by the subject, the trustworthy state of the request may change to a or d when the object is trusted. If the object isn't trusted, the state enters e, and it can't be accessed any more.

4 Application of TED Security Model in Trusted Web Server

In the above section, TED model is specified in detail. This section will introduce how to use the model to protect the security of information flows between the nodes in ad hoc networks. Http is the network protocol of web, which is simple and powerful. But it doesn't maintain any connections and user information between transactions. Consequently, it is difficult to implement fine-grained access control to sensitive resources in server.

TWS has been developed on Linux in a TP. It is supported by Integrity Measurement Architecture (IMA), which is a loadable kernel module and has been realized by IBM [3]. It measures all executable contents when they are loaded. As it is the open source software, we have extended it to sustain the integrity measurement of some important system files. So it can support TED model to detect the integrity of subjects and some objects at runtime.

During the trusted booting, TPM assures system hardware, OS kernel, IMA and TWS are loaded correctly. At running, user authentication is achieved by Digest Access Authentication (DAA) offered by HTTP and SSL. SSL also affords protection to transmitted data. When a browser asks to access some resources, TWS firstly attests the integrity of hardware and software in the remote platform after authenticating the requester. Then the security polices of TED model are adopted, which protect sensitive objects against unauthorized access. Attackers cannot enter the system through the security leakages of the web server.

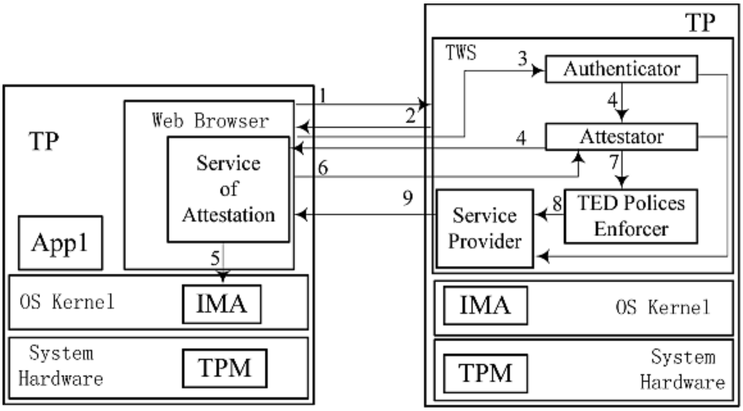


Fig. 2. The structures of TWS and the corresponding browser

The structures of TWS and the corresponding browser are shown in figure 2. The main components of TWS include authenticator, attestator, TED polices enforcer and service provider. The corresponding browser contains the module of service of attestation. The steps of process are described as follows.

Step 1: A browser requests to access some resources in TWS.

Step 2: Above all, TWS decides whether the requested object needs to authenticate after receiving the request of the browser. If not, TWS provides the requested service. If authentication is needed, TWS asks to authenticate the requester's identity.

Step 3: The browser sends the authenticated information of the requester to authenticator in TWS.

Step 4: TWS determines whether the requester is authorized. If not, TWS rejects the request, and returns an error message. If the user is authorized, attestator in TWS asks the browser to send the integrity values of its platform.

Step 5: The browser reads the current integrity measurements of the platform from the PCRs in the TP's TPM through IMA.

Step 6: The browser sends the values after signing them.

Step 7: TWS verifies the sign, and attestator attests the integrity values of the remote platform. If the values aren't right, the platform may be damaged, and it isn't trustworthy. Then the request isn't approved. If the remote platform is trusted, attestator sends the security properties of the subject and the object in the request to TED polices enforcer.

Step 8: TED polices enforcer decides whether the access should be accepted by the requester's security properties, the security properties of the demanded object, and the rules of TED model.

Step 9: If the request is accepted, TWS supplies for the requester. If it disobeys the security rules, TWS refuses the request and returns an error.

5 Related Work

The related researches on protecting the security of a system on a TP are described in this section. Using LSM and TPM, David Safford [4], et al, in Watson research center of IBM, have realized Trusted Linux Client (TLC). TLC implements three complementary loadable kernel modules for protecting client integrity. In TLC, all files are labeled with the security levels and measured before they are used. The method limits it only to verify the files in local platform and cannot solve the problem of remote attestation.

Reiner Sailer [3], et al, have designed and implemented a secure Integrity Measurement Architecture (IMA) for Linux 2.6. All executable contents loaded into the Linux system are measured before execution, and the values of these measurements are protected by TPM. IMA extends the trust concept of TCG to dynamic executable contents from BIOS to the application layer. It doesn't prevent systems from running malicious codes because it only measures the integrity of executable programs when they are loaded. Moreover, the confidentiality and integrity of the information flows in a system haven't been considered.

A new measurement approach called the Policy-Reduced Integrity Measurement Architecture (PRIMA) is proposed by Trent Jaeger [5], et al, which is based on information flow integrity. It settles the problem that the load-time measurements of code alone do not accurately reflect runtime behaviors. The

PRIMA prototype has been built on IMA using SELinux policies to protect the information flows. It only checks the integrity of executable programs and ignores the integrity of accessed objects.

6 Conclusion

The paper analyzes the attacks on processes and important system files, and it shows the necessity of checking running environments. Especially in ad hoc networks, these attacks are more harmful than in normal networks. A new approach to design security model is provided by trusted computing. TED security model is promoted in the paper, which is based on BLP and Biba model. The model is used in a TP. Adopting the concept of reliability and the functions for trustworthy state measurements, the trustworthiness of processes and input data is monitored and the accessing ranges of subjects are adjusted dynamically. So the ability of TED security model to defend running attacks is enhanced. The formal description of TED model is specified in detail, including its components and security rules. Its security properties are also proved. And the transformation of the trustworthy states of requests is described in the paper. The model is applied to TWS which protects the authorized information flows in ad hoc networks.

In the future, we shall try to improve the flexibility of TED model, so it will be much fine-grained and feasible for various applications.

References

1. Bell D. E., LaPadula L. J.: Secure computer system: unified exposition and multics interpretation. Mitre Report, MTR-2997 Rev.1, 1976.
2. Biba K. J.: Integrity considerations for secure computer systems. ESD-TR-76-372, Hanscom ATB, MASS.: Air force electronic systems division, 1977.
3. Reiner S., Xiaolan Z., et al: Design and implementation of a TCG-Based integrity measurement architecture. *13th USENIX Security Symposium* San Diego, CA, USA, pp. 223-238, 2004.
4. David S., Mimi Z.: A trusted Linux Client. http://www.acsaadmin.org/2004/workshop/David-Safford.pdf. 2004
5. Trent J., Reiner S., et al: PRIMA: policy-reduced integrity measurement architecture. *11th ACM Symposium on Access Control Models and Technologies-SACMAT 2006*, 2006.
6. Hyung C. K., Wook S., et al: Design and implementation of an extended reference monitor for trusted operating systems. *Second International Conference on Information Security Practice and Experience-ISPEC 2006* Springer-verlag, pp. 235-247, 2006.

An Improvement of Remote User Authentication Scheme Using Smart Cards

Jun-Cheol Jeon, Byung-Heon Kang, Se-Min Kim, Wan-Soo Lee,
and Kee-Young Yoo*

Department of Computer Engineeing, Kyungpook National University,
Daegu, 702-701 Korea
{jcjeon33, bhkang, resemin, complete2}@infosec.knu.ac.kr,
yook@knu.ac.kr

Abstract. In 2006, Manik et al. proposed a novel remote user authentication scheme using bilinear pairings. Chou et al. identified a weakness in Manik et al.'s scheme and improved the scheme. And besides, Thulasi et al. pointed that both Manik et al.'s and Chou et al.'s schemes are vulnerable to forgery attacks and replay attacks. In this paper, we analyze the previous schemes based on a timestamp and provide further comments and an improved scheme using a nonce. Our scheme also provides mutual authentication between a user and a remote server while the previous schemes only provided unilateral authentication.

Keywords: Mutual Authentication, Bilinear Pairing, Forgery Attack, Replay Attack, Off-line Dictionary Attack.

1 Introduction

With the rapid change of computing environment, electrical commerce, business transaction and government services have been conduct and offered in a public channel. Remote user authentication scheme is one of the important factors to use the above services. Remote user authentication scheme allows the authenticated user to login the remote system for accessing the services. For this reason, various kinds of authentication scheme have been announced [1, 2]. In 1981, Lamport proposed a password authentication scheme for insecure communication [3]. The scheme requires the remote server to maintain a password table for purpose of verification. In 2000, Hwang and Li proposed a new scheme using smart cards [4]. The advantage of the Hwang-Li's scheme is that it does not need any password table.

Recently, Manik et al. proposed a remote user authentication scheme using bilinear pairings [5]. In the scheme, they use timestamps to avoid replay attacks while sending the authentication request over a public channel. But this is completely insecure as an adversary can use this information for illegal login later.

Chou et al. identified that the verification of Manik et al.'s scheme involves subtraction of two components, which are passed over the public channel leading to replay attack [6]. One can do replay by adding same information to those two

[*] Corresponding author.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 416 – 423, 2006.

components, as it results in valid verification. To overcome replay attack, they suggested a modification in verification part of Manik et al.'s scheme.

Thulasi et al. point out that Chou et al.'s modified scheme still suffers from the replay attack [7]. Namely, the modified verification by Chou et al. as $\hat{e}(DID, P) = \hat{e}(TsH(ID) + V, P)$ also holds valid for $DID' = DID + a'$ and $V' = V' + a'$, where $a' \in G_1$. And Thulasi et al. identified that Manik et al.'s scheme is also vulnerable to forgery attack and has the weakness in password change phase.

The above mentioned protocols used a timestamp for generating parameters though the inverse of a timestamp is easily computed by well-known algorithm such as extended Euclidian algorithm. In this paper, we analyze the substantial problems on the mentioned schemes and propose an efficient protocol to avoid the attacks that identified Chou et al.'s and Thulasi et al.'s scheme. Our scheme also provides mutual authentication between a user and a remote server while the previous schemes only provided unilateral authentication.

The rest of this paper is organized as follows: Section 2 illustrates background of a bilinear paring and mathematical problems associated with the current work, Section 3 briefly reviews the previous schemes and comments of Manik et al.' scheme. Section 4 provides the further comments, our improved scheme and security and performance analysis. Finally, Section 5 gives concluding remarks.

2 Mathematical Background

Suppose G_1 is an additive cyclic group generated by P, whose order is a prime q, and G_2 is a multiplicative cyclic group of the same order. A map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ is called a bilinear mapping if it satisfies the following properties:

- Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.
- Non-degenerate: there exists $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.
- Computable: there exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all P, $Q \in G_1$.

We note that G_1 is the group of points on an elliptic curve and G_2 is a multiplicative subgroup of a finite field. Typically, the mapping will be derived from either the Weil or the Tate paring on an elliptic curve over a finite field. Other mathematical problems associated with the current work are as followings:

- Discrete Logarithm Problem (DLP): Given two elements $P, Q \in G_1$, find an integer $a \in Z_q^*$, such that $Q = aP$ whenever such an integer exists.
- Computational Diffie-Hellman Problem (CDHP): Given (P, aP, bP) for any $a, b \in Z_q^*$, compute abP.
- Decisional Diffie-Hellman Problem (DDHP): Given (P, aP, bP, cP) for any $a, b, c \in Z_q^*$, decide whether $c = ab \bmod q$.
- Gap Diffie-Hellman (GDH) group: G_1 is a GDH group if there exists an efficient polynomial time algorithm which solves the DDHP in G_1 and there is no probabilistic polynomial time algorithm which solves the CDHP in G_1 with non negligible probability of success.
- Bilinear Diffie-Hellman Problem (BDHP): Given (P, aP, bP, cP) for any a, $b, c \in Z_q^*$, compute $\hat{e}(P, P)^{abc}$.

3 Review of Previous Schemes

In this section, we briefly review the previous schemes and cryptanalysis of Manik et al., Chou et al., and Thulasi et al.

3.1 Manik et al.'s Scheme

[**Setup phase**] The remote server (*RS*) selects a secret key s and compute $Pub_{RS} = sP$. Then the *RS* publishes the system parameter $< G_1, G_2, \hat{e}, q, P, Pub_{RS}, H>$.

[Registration phase] This phase is invoked whenever U_i initially registers or re-registers to the *RS*.

(1) U_i submits his identity ID_i and PW_i to the *RS*.
(2) On receiving the registration request, the *RS* computes $Reg_{IDi} = sH(ID_i) + H(PW_i)$.
(3) The *RS* personalizes a smart card with the parameters ID_i, Reg_{IDi}, H and sends it to U_i over a secure channel.

[Authentication phase] This phase is invoked whenever U_i wants to login the *RS*. This phase is further divided into login and verification phase.

Login Phase: The user U_i inserts smart card in a terminal and keys ID_i and PW_i. If ID_i is identical to the one that is stored in the smart card, the smart card performs the following operations:

(1) Computes $DID_i = TReg_{IDi}$, $V_i = TH(PW_i)$, where T is the user system's timestamp.
(2) Sends the login request $< ID_i, DID_i, V_i, T >$ to the *RS* over a public channel.

Verification phase: This phase is invoked whenever the *RS* receives U_i's login request.

(1) Verifies the expected valid time interval $\Delta T \geq (T^* - T)$.
(2) Checks whether $\hat{e}(DID_i - V_i, P) = \hat{e}(H(ID_i), Pub_{RS})^T$. If it holds, the *RS* accepts the login request; otherwise, rejects it.

[Password change phase] This phase is invoked whenever U_i wants to change his password.

(1) U_i attaches the smart card to a terminal and keys ID_i and PW_i. If ID_i is identical to the one that is stored in the smart card, proceeds to the step (2); otherwise, terminates the operation.
(2) U_i submits a new password PW_i^*.
(3) The smart card computes $Reg_{IDi}^* - H(PW_i) + H(PW_i^*) = sH(ID_i) + H(PW_i^*)$.
(4) The password has been changed now with the new password PW_i^* and the smart card replaced the previously stored Reg_{IDi} value by Reg_{IDi}^* value.

3.2 Chou et al.'s Improvement on Manik et al.'s Scheme

Chou et al. [6] pointed that the verification in [5], $\hat{e}(DID_i - V_i, P) = \hat{e}(H(ID_i), Pub_{RS})$ holds valid even when $DID_i' = DID_i + a$ and $V_i' = V_i + a$ where $a \in G_1$, as shown below.

$$\begin{aligned}\hat{e}(DID_i' - V_i', P) &= \hat{e}(DID_i + a - V_i - a, P)\\ &= \hat{e}(DID_i - V_i, P)\\ &= \hat{e}(H(ID_i), Pub_{RS})\end{aligned}$$

To avoid this, Chou et al. proposed different verification technique as $\hat{e}(DID_i, P) = \hat{e}(TsH(ID_i) + V_i, P)$ to avoid the subtraction effect of [5].

3.3 Thulasi et al.'s Cryptanalysis

The verification in [5] is modified by Chou et al. [6] as $\hat{e}(DID_i, P) = \hat{e}(TsH(ID_i) + V_i, P)$. However, Thulasi et al. pointed that this verification also holds valid for $DID_i' = DID_i + a'$and $V_i' = V_i + a'$where $a' \in G_1$, as shown below.

$$\begin{aligned}\hat{e}(DID_i', P) &= \hat{e}(DID_i + a', P)\\ &= \hat{e}(DID_i, P)\hat{e}(a', P)\\ &= \hat{e}(TsH(ID_i) + V_i, P) + \hat{e}(a', P)\\ &= \hat{e}(TsH(ID_i) + V_i + a', P)\\ &= \hat{e}(TsH(ID_i) + V_i', P)\end{aligned}$$

Thus the approach of Chou et al., by adding V_i on the right side instead of the left side, cannot solve the problem as shown above.

Forgery attack. Given P and $Pub_{RS} = sP$, finding s is DLP but given x and xQ, it is feasible to compute Q. In the login phase, the tuple $< ID_i, DID_i, V_i, T >$ is being sent to the RS over a public channel. Any adversary tapping this message can compute a valid $< ID_i, DID_i', V_i', T'>$.

As $DID_i = TReg_{IDi}$ where $T \in Z_q^*$, $V_i = TH(PW_i)$, attacker can compute T^{-1}, Reg_{IDi} and $H(PW_i)$ as below.

$$\begin{aligned}Reg_{IDi} &= T^{-1}DID_i = T^{-1}TReg_{IDi}\\ H(PW_i) &= T^{-1}V = T^{-1}TH(PW_i)\end{aligned}$$

Attacker, who knows Reg_{IDi} and $H(PW_i)$**,** can form the valid tuple $<ID_i, DID_i', V_i', T'>$ for time stamp T^* by computing, $DID_i = TReg_{IDi}$, $V_i' = T'H(PW_i)$**.** The attacker can use the information ID_i, Reg_{IDi}, $H(PW_i)$ for accessing remote system whenever he wants. So the scheme is completely insecure against replay attacks and forgery attacks. Anyone can forge the login request, so it is also possible for an insider, leading to the insider attack.

Weakness in password change phase. In the password change phase, a user submits ID_i, old password PW_i and new password PW_i^*. But there is no verification in which the old password is valid one. Namely, in the equation, $Reg_{IDi}^* = Reg_{IDi} - H(PW_i) + H(PW_i^*)$ of the password change phase, if the value of $H(PW_i)$ is not the original value, the password change process is done. So anyone knowing the ID_i and having the smart card can change the secret value Reg_{IDi} in the smart card.

4 Further Comments and Improved Scheme

In this section, we illustrate further flaws on Manik et al.'s scheme and provide an improved scheme. We also provide security and performance analysis at the end of the section.

4.1 Further Weakness of Manik et al.'s Scheme

Manik et al.'s scheme has fundamental flaw about generation of the parameters such as DID_i, V_i. In the login phase, the parameters are computed as $DID_i = TReg_{IDi}$ and $V_i = TH(PW_i)$. The first of all, the designer overlooked the inverse of a timestamp, T^1. In finite group operation, the inverse is easily computed by well-known algorithm such as extended Euclidian algorithm. Therefore, In the above equations, the secret value is only one parameter, Reg_{IDi} and $H(PW_i)$ respectively. Thus we easily identify the values which are kept in smart card, and the following off-line guessing attack is possible.

Off-line guessing attack. Besides the replay attack and forgery attack on Manik et al,'s scheme, their scheme is still insecure against a guessing attack. When U_i sends the login request $<ID_i, DID_i, V_i, T>$ to the *RS* over a public channel. An active adversary can guess the password of a user from the equation, $V_i = TH(PW_i)$.

Comment on password change phase. Thulasi et al. worried that there is no verification in which the old password is valid one. However if the password change phase is operated after checking the validity of the inputted password, the phase does not have any problem.

Unilateral Authentication. In the above mentioned schemes, only a user is authenticated to the *RS*. In order to protect a server spoofing attack and achieve a reliable protocol, it is better to provide mutual authentication.

4.2 Our Improved Scheme

As mentioned above in Section 4.1, the previous scheme is insecure against not only forgery attack, replay attack but guessing attack. One of the key reason of those weaknesses is that Reg_{IDi}, $H(PW_i)$ can be revealed based on computation of the inverse of T. To remedy of the weakness, we generate a nonce at login phase and two cryptographic hash operations: one is a general hash operations such as MD-5 or SHA and the other is map-to-point hash operation. The following procedure illustrates our improved scheme.

[**Setup phase**] The remote server (*RS*) selects a secret key s and compute $Pub_{RS} = sP$ and two cryptographic hash functions, H_1: $\{0, 1\}^* \rightarrow G_1$ and H_2: $\{0, 1\}^* \rightarrow Z_q^*$ are also chosen by *RS*. Then the *RS* publishes the system parameter $< G_1, G_2, \hat{e}, q, P, Pub_{RS}, H_1, H_2>$.

[Registration phase]

R1 U_i submits his identity ID_i and PW_i to the *RS*.

R2 On receiving the registration request, the *RS* computes $Reg_{IDi} = sH_1(ID_i) + H_1(PW_i)$.

R3 The *RS* personalizes a smart card with the parameters ID_i, Reg_{IDi}, H_1, H_2 and sends it to U_i over a secure channel.

[Authentication phase]

Login phase: The user U_i inserts smart card in a terminal and keys ID_i and PW_i. The smart card performs the following operations.

L1 Checks if ID_i is identical to the one that is stored in the smart card.

L2 Generates a nonce $n_i \in Z_q^*$.

L3 Computes $V_i = n_iP$ and $t = H_2(T \parallel V_i^x \parallel V_i^y)$ where T is user system's time stamp, and V_i^x and V_i^y are the x and y coordinates of the point V_i.

L4 Computes $DID_i = n_i^{-1}(Reg_{IDi} - H_1(PW_i) + tP)$.

L5 Sends the login request $< ID_i, DID_i, V_i, T >$ to the RS over a public channel.

User authentication phase: On receiving the login request $< ID_i, DID_i, V_i, T >$ from the user, the RS performs the following operations:

U1 Verifies the expected valid time interval $\Delta T \geq T^* - T$.

U2 Checks whether $\hat{e}(DID_i, V_i) = \hat{e}(H_1(ID_i), Pub_{RS})\ \hat{e}(P, P)^t$. If it holds, the RS accepts the login request; otherwise, rejects it.

U3 Computes $V_R = T_R\ sH_1(ID_i)$ and sends $< V_R, T_R >$ to the user where T_R is the RS's time stamp.

Server authentication phase: On receiving the message $< V_R, T_R >$ from the server, the user performs the following operations:

S1 Verifies the expected valid time interval $\Delta T \geq T^* - T_R$.

S2 Checks whether $T_R(Reg_{IDi} - H_1(PW_i)) = V_R$. If it holds, the user authenticates the RS.

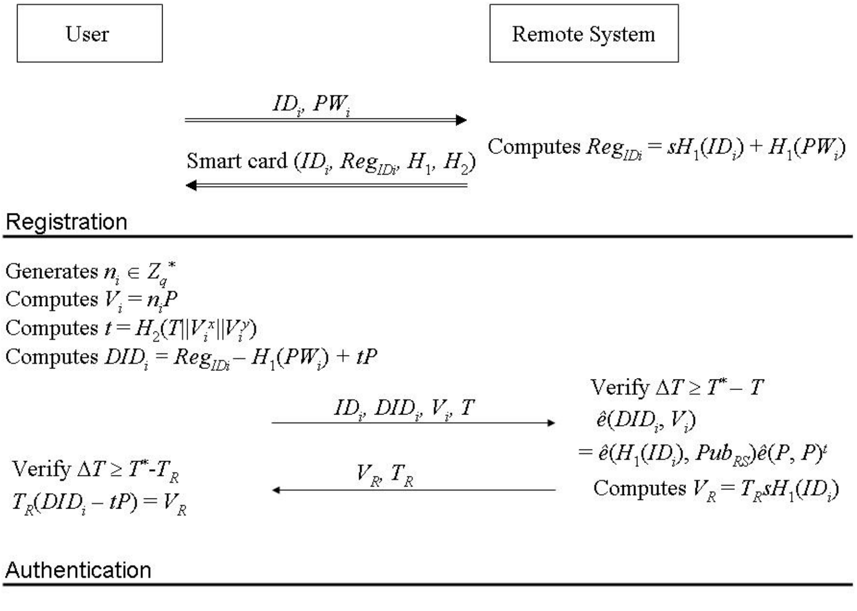

User
Remote System
ID_i, PW_i
Smart card (ID_i, Reg_{IDi}, H_1, H_2)
Computes $Reg_{IDi} = sH_1(ID_i) + H_1(PW_i)$
Registration
Generates $n_i \in Z_q^*$
Computes $V_i = n_iP$
Computes $t = H_2(T \| V_i^x \| V_i^y)$
Computes $DID_i = Reg_{IDi} - H_1(PW_i) + tP$
ID_i, DID_i, V_i, T
Verify $\Delta T \geq T^* - T$
$\hat{e}(DID_i, V_i)$
$= \hat{e}(H_1(ID_i), Pub_{RS})\hat{e}(P, P)^t$
Computes $V_R = T_RsH_1(ID_i)$
V_R, T_R
Verify $\Delta T \geq T^*$-T_R
$T_R(DID_i - tP) = V_R$
Authentication

Fig. 1. Proposed remote user authentication scheme using bilinear pairings

4.3 Security Analysis and Performance

Suppose any adversary eavesdrop and record the tuples ID_i, DID_i, V_i, and T that are being sent to the RS over a public channel in login phase. In Manik et al.'s scheme, an attacker can obtain Reg_{IDi} and $H_1(PW_i)$ using T^1 where T is a user system's time stamp, and they can forge the RS. However, in our scheme based on a nonce, Reg_{IDi} and $H_1(PW_i)$ cannot be computed by any attacker.

Correctness. The verification in V2 of login request is verified as followings.

$$\begin{aligned} \hat{e}(DID_i, V_i) &= \hat{e}(n_i^{-1}(Reg_{IDi} - H_1(PW_i) + tP), n_iP) \\ &= \hat{e}(s\, H_1(ID_i) + H_1(PW_i) - H_1(PW_i) + tP, P) \\ &= \hat{e}(s\, H_1(ID_i), P)\ \hat{e}(tP, P) \\ &= \hat{e}(H_1(ID_i), Pub_{RS})\ \hat{e}(P, P)^t \end{aligned}$$

Replay attack. Though an adversary replays an intercepted valid login request and the RS receives the request at time T'. The attack does not work because the time interval exceeds the expected time delay ΔT.

Forgery attack. In the login phase, an adversary can get the tuple $< ID_i, DID_i, V_i, T >$, however, he cannot extract Reg_{IDi} or $H_1(PW_i)$ from the equation, $DID_i = n_i^{-1}(Reg_{IDi} - H_1(PW_i) + tP)$ since an adversary cannot find the nonce n_i. Thus he cannot compute the inverse of the nonce. An invalid password input cannot generate a correct DID_i value at L4 step so that the request will be failed. Only a valid time stamp can be passed through the verification phase since an invalid T' cannot make a legitimate value, t.

Server spoofing attack. The attacker cannot impersonate the RS since he/she cannot compute a legitimate value, V_R. Though he/she can obtain the values, T_R, ID_i and H_1, the server's secret value, s cannot be found based on DLP problem.

Insider attack. If the user login request is password-based and the RS maintains password or verifier table for login request verification, this attack can be achieved by an insider of RS. In our scheme, however, the user login request is based on the user's password as well as RS's secret and a nonce which is not maintained in verifier table, thus our scheme can withstand this attack.

Table 1. Computational cost in proposed authentication scheme

	User's smart card	Remote server
Registration		1 scalar multiplication 2 map-to-point hash operations 1 point addition in G_1
User authentication	2 scalar multiplication 1 hash operation (H_2) 1 map-to-point hash operations 2 point addition in G_1	1 scalar addition 1 map-to-point hash operations 3 pairing operations 1 group exponent in G_2
Server authentication	1 scalar addition 1 scalar multiplication 1 point addition in G_1	2 scalar multiplications 1 map-to-point hash operation

Moreover, our scheme can simply prevent the scenario of many logged in users with the same login-ID as the other person cannot login to the RS without the smart card. In our scheme, RS does not posses any password or verifier table for user verification so that our scheme also withstands a stolen verifier attack. Table 1 shows the computational cost of proposed paring-based authentication scheme. In our scheme, a user's smart card has only four scalar computations, two hash operations and three point additions.

5 Conclusion

In remote authentication, the most important factor is the computational cost of a user. Several researchers indicated that Manik et al.'s scheme has security flaws but it is quite hard to improve their scheme in given environment because of their fundamental problem. Thus we analyzed that Manik et al.'s scheme has a fundamental flaws in construction of parameters and constructed an efficient scheme based on employing a nonce and two kinds of hash operations. Moreover we have shown that our mutual authentication provides not only security satisfaction but also low computational cost so that it is well-suited and practical for applications in remote user systems.

Acknowledgment

The authors would like to thank the anonymous referees for their valuable suggestions on how to improve the quality of the manuscript. This research was supported by the MIC of Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0026).

References

[1] G. Ateniese, M. Steiner and G. Tsudik: New multiparty authentication services and key agreement protocols. IEEE Journal on Selected Areas in Communication 18 (2000) 628-639

[2] H. M. Sun and L. H. Li: An efficient remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 46 (2000) 958-961

[3] L. Lamport: Password authentication with insecure communication. Communication of ACM 24 (1981) 770-772

[4] M. S. Hwang and L. H. Li: A new remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 46 (2000) 28-30

[5] Manik. L. Das, Ashutosh Saxena, V. P. Gulati, D. B. Phatak: A novel remote user authentication scheme using bilinear pairings. Computers & Security 25(2006) 184-189

[6] J. S. Chou, Y. Chen, J. Y. Lin. Improvement of Manik et al.'s remote user authentication scheme. http://eprint.iacr.org/2005/450.pdf

[7] Thulasi Goriparthi, Manik. L. Das, Atul Negi and Ashutosh Saxena: Cryptanalysis of recently proposed Remote User Authentication Schemes. http://eprint.iacr.org/2006/28.pdf

A Secure Architecture for Mobile Ad Hoc Networks

Abderrezak Rachedi[1] and Abderrahim Benslimane[2]

[1] LIA/CERI, University of Avignon, Agroparc
BP 1228, 84911 Avignon, France
abderrezak.rachedi@univ-avignon.fr
[2] LARIM/Computer Engineering Department,
Ecole Polytechnique of Montreal P.O. Box 6079
Station Centre-ville Montreal H3C 3A7 Canada
abderrahim.benslimane@polymtl.ca

Abstract. In this paper, we propose a new architecture based on an efficient trust model and clustering algorithm in order to distribute a certification authority (CA) for ensuring the distribution of certificates in each cluster. We use the combination of fully self-organized security for trust model like PGP adapted to ad-hoc technology and the clustering algorithm which is based on the use of trust and mobility metric, in order to select the clusterhead and to establish PKI in each cluster for authentication and exchange of data. Furthermore, we present new approach Dynamic Demilitarized Zone (DDMZ) to protect CA in each cluster. The principle idea of DDMZ consists to select the dispensable nodes, also called registration authorities; these nodes must be confident and located at one-hope from the CA. Their roles are to receive, filter and treat the requests from any unknown node to CA. With this approach, we can avoid the single point of failure in each cluster. This architecture can be easily extended to other hierarchical routing protocols. Simulation results confirm that our architecture is scalable and secure.

Keywords: Wireless ad hoc networks, security, clustering algorithm.

1 Introduction

In recent years, much interest has been involved in the design of Mobile Ad-hoc Network (MANET) technologies. Mobile ad-hoc networks are characterized by their self-configuration, open peer-to-peer network architecture, shared wireless medium, stringent resource constraints, and highly dynamic network topology. These characteristics make them vulnerable to security attacks. Existing security solutions for wired or wireless networks with infrastructure cannot be directly applied to MANETs. Designing security solutions for MANET is the nontrivial challenges. The goal of security solutions is to provide security services, such as authentication, confidentiality, integrity, and availability to mobile users. In order to achieve this goal, we must develop some key management systems adapted to the characteristics of MANET.

In this article, we propose a new architecture based on an efficient trust model and distributed clustering algorithm for designing the specific public key

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 424–435, 2006.

management systems. Our trust model is based on PGP (Pretty Good Privacy) approach [1] adapted to MANET characteristics, like fully self-organized security proposed by Hubaux et al. in [2]. Our distributed clustering algorithm uses the trust level and mobility metric for the selection of the cluster head (CH) which becomes certification authority (CA) in the cluster. In order to secure the cluster formation, we propose a new scheme which uses dispensable confident nodes, called registration authorities (RA). It consists to provide dynamic demilitarized zone (DDMZ) at one-hope from the CA in each cluster. The role of RAs is to protect CA, by receiving requests of certification, filtering and treating these demands before forwarding them to the CA.

The rest of the paper is organized as follows. In section 2, we discuss the related work on current key management systems developed for MANET. In section 3, we describe our global architecture, our trust model and the distributed clustering algorithm. In section 4, we present the simulation results of our distributed hierarchical architecture. In section 5, we study and analyse the security of our system. The last section consists of the conclusion.

2 Related Work

Several works have been proposed in the literature to deal with the security problems in ad hoc networks. Specially, we investigate the distributed CA approach using threshold cryptography scheme and a clustering concept.

2.1 Distributed CA Approach Using Threshold Cryptography

Having a single central authority to distribute the public key certificate to all nodes is not suitable for MANET, because this scheme is vulnerable to single point of failure like ARAN (Authenticated Routing in Ad-hoc Networks protocol) [13]. If this node is compromised, the entire network becomes compromised.

Zhou and Haas's idea consists on distributing the CA role among n nodes of the network using $(n, k+1)$ threshold cryptography scheme [11]. In this scheme the secret key is divided into n partial shares $(S1, S2, \cdots, Sn)$ where at least $k+1$ of n are partial shares which are needed to generate a secret S. The advantage is its increased availability, since any $k+1$ among n nodes in the local neighborhood of the requesting node can issue or renew a certificate. Another advantage is that any node, which does not have a secret share yet, can obtain a share from any group of at least $k+1$ nodes which has already a share. Unfortunately, this scheme has some drawbacks: First, the $k+1$ nodes must be initialized by a trusted authority. Second, the number k must be a trade-off between availability and robustness. Third, the system overloads the network because instead of sending only one request for obtaining certificate or revocation the node must send at least $k+1$ request (k traffic add in network).

2.2 A Clustering Concept

Mobile ad-hoc network may be represented as a set of clusters. Each cluster is represented by a cluster-head (CH) and gateway nodes which manage the communication with adjacent clusters. Among several secure solutions based on clustering ad hoc networks which we studied, one is a cluster based security architecture proposed by Becheler et al. [12]. This architecture use the threshold cryptography scheme (n, k) to distribute CA. The idea is to distribute the private key of CA over CHs where every CH holds a fragment of the whole key. In order to be certified, any guest node must possess a certain number (W) of warranty certificates from warrantor nodes. After that, it must request at least (k) certificates from different CHs, whose association of these k certificates gives the network certificate. The drawbacks of the log-on procedure are as follow. First, this approach is not realistic, because the warrantors do not have any information about the new node to be guaranteed (the warrantors must have minimal information about nodes, so that they can decide to guarantee or not). Second, even the guest has already W certificates from guarantors, it cannot succeed to have K certificates from CHs and it will not be certified. Third, the network traffic generated by each new node in these procedure is at least $2 * (W + K)$ packets. The Becheler's architecture has some other drawbacks in merging networks process, it assembles several networks in one network. As the two network keys cannot be mixed, one of them must be dropped and the other must be distributed over the whole network. The criterion to choose the dominate key among these different network's keys depends on the number of CHs of each network. The network which has maximum CHs will become dominant network and its network key remains private key of CA. These processes present a point of failure, because in this architecture any node can construct its own cluster. Thus, a set of malicious nodes can form their network with the maximum of CHs, and then attack the network in order to merger in the network and take the CA control.

This architecture does not contemplate the presence of two CHs in only one cluster due to the mobility of the nodes. Furthermore, the selection criteria of CHs are not mentioned in the paper. Also, to renew the network key, the intervention of a trust third party is needed so that it can subdivide the new key and distribute the fragment of the key over CHs. In the light of above factors, we believe that this architecture is not well adapted for Ad hoc environments.

3 Architecture

Firstly, we define a new trust model on which is based our architecture. Secondly, we present a clustering algorithm based trust and mobility metric to ensure a selection of trust and relatively stable confident node as CH which will become CA node in the cluster. Finally, we discuss how to evaluate certificate chain between clusters.

3.1 Primitives

The basic idea of our architecture consists of establishing dynamic public key infrastructure with CA as clusterhead that will change according to topology changes. We propose a clustering algorithm based on this trust model. A new concept is proposed to protect CA node in each cluster based on dispensable nodes.

Definition 1. *DDMZ is defined as the zone of 1-hop or more from CA. It is formed by at least one or more confident nodes (RA). Their role is to not authorize unknown nodes to communicate directly with CA node. All guest nodes must be passed by DDMZ to request certificate from CA.*

We assume that there are spare social relationships among nodes in order to establish trust relationship of any to-be-trust node with confident nodes. Also, every node has its own private/public key pair. Furthermore, the initial trust nodes (or confident nodes) are honest and do not issue false certificates. Moreover, each node manages a trust table. Initially, each trust node knows the identity and public key of other trust nodes $(ID, K+)$. It means, if we have initially k trust nodes these nodes have k-1 entries (known mutually) in their trust table.

3.2 Trust Model

In our trust model, we define trust metric (Tm) as continuous value on the [0..1] interval. A node i has a high trust value $(Tm(i) = 1)$, if it is known and exchanged keys over secure side channel (physical encounters and friends) with one or more of confident nodes [8][2]. Another manner to obtain a high trust value is that a node must prove its good faith by adapting a good behavior and a cooperation. If a new node is added in the trust table by one or more confident nodes all others confident nodes will be aware. This is because confident nodes update and exchange their trust tables. Each new unknown node starts with $Tm = 0.1$ its lower trust level.

Five roles of nodes are defined in each cluster and each role has particular trust level:

- CA_k : Certification authority of cluster k which certificate public key of nodes belonging in the same cluster. CA_k has high trust level, Tm value must be equal one.
- $RA_{i,k}$: Registration Authority of cluster k assured by trust node i. The mean goal of RA is to protect CA against attackers that by DDMZ formation in order to prevent direct communication between unknown nodes and CA, for example, they treat and filter the requests of certification toward CA. Also, RAs must be confident nodes with $Tm(i) = 1$.
- $GW_{i,j}$: It is a gateway node ensuring a connection between two different clusters i and j. These nodes must be certified by two different CA. GW nodes must haves good trust level with $Tm(g) \in [0.7 - 1.0]$.
- $MN_{i,k}$: it represents a member node i belonging to the cluster k which success to pass from visitor to member status by well behaviour and good cooperation. This status can be recommended by CA_k to another CA node. Node i can communicate inside and outside its cluster. It has an average trust level $Tm(i) \in [0.5 - 0.7]$.

- $VN_{i,k}$: It is a visitor node i that belongs to cluster k, it has low trust certificate, because CA_k and $RA_{j,k}$ nodes need to have more information about node i behavior. Node i cannot communicate outside its cluster. It has a minimal trust level $Tm(i) \in [0.1 - 0.5]$.

Figure 1 shows the state transition diagram where each state represents the node's role in each cluster.

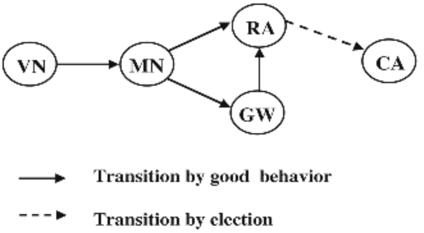


Fig. 1. State transition diagram

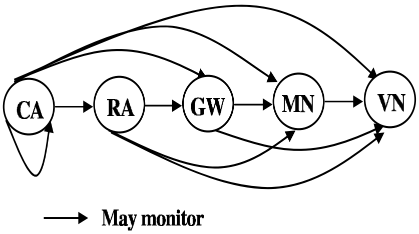


Fig. 2. Monitoring scheme

The hierarchical monitoring process consists to supervise behaviors of nodes. Each node with high trust value monitors its neighbors nodes with low trust value. Figure 2 shows the possibility of a node with certain status to monitor other status nodes. CA can monitor other CAs and all other status. RA can monitor {GW,MN,VN} status, also GW can supervise {MN,VN} status. Finally MN node can supervise only VN status but VN cannot supervise any node.

In our trust model, the trust relationship is ensured by CAs between clusters. A CA can recommend node, with certain trust level, belonging in its cluster to another CA. It is also ensured by RA in order to recommend nodes which belong in the same cluster to the CA.

The trust value of a path depends on its trust chain which is represented by its certificate chain. The inter-cluster communication is based on the evaluation of the certificate chain. The trust evaluation between two nodes consists to take the small trust value of nodes (eg. Trust between RA and GW is $min(1, w)$ where $w \in [0.7 - 0.9]$). Figure 3 shows two examples of certificate chain. The best trust chain (TC) is given in the case of b: $TC(b) > TC(a)$.

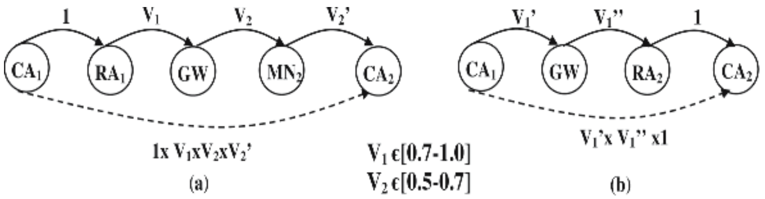


Fig. 3. Certificate Chain

3.3 Distributed Clustering Algorithm

A clustering network in our architecture is ensured by a Secure Distributed Clustering Algorithm (SDCA). The main rules of this algorithm are as follow:

1. Only confident nodes ($Tm(i) = 1$) can be candidate to become CA.
2. Each cluster-head is CA of only one cluster.
3. All confident neighbors of CA, can become RA in the cluster.
4. Other nodes are at distance of maximum d-hop from the CA according the predefined size of cluster.

Our algorithm selects CA of the cluster according to trade-off between security and stability. It is based on sending periodic beacons by each confident node to its neighbors at predefined interval time. Based on information available in the received beacons and after authentication and verification of beacon's integrity, the receivers update their information and decide about their cluster status.

The security parameter depends on trust metric, only nodes with $Tm = 1$ and at least one trust neighbor (to establish DDMZ) can be candidate to become CA in the cluster. This constitutes the cluster formation condition. Moreover, to reinforce the security of the cluster, the algorithm selects the candidate with maximum trust degree; its means the trust node which has a maximum trust neighbors.

The stability parameter is very important on clustering algorithm, this parameter is defined as cluster-head duration. Several clustering strategies have been proposed in order to increase system stability, such as: Lowest-ID cluster-head selection based on ID [10], max-connectivity algorithm [9]. In our algorithm, we adopt a mobility metric [3][7] because this strategy gives good result 33% of reduction in the number of cluster-head changes compared with last approach [10][9].

Each trust node puts the following information in the beacon before transmission:

- CA (Cluster-head): ID of the CA to which the node is attached.
- HopCount: hop count number to CA.
- Degree of Trust neighbors (DTN): each transmitter puts the number of its trust neighbors and their identities.
- Relative mobility (RM): it indicates the relative stability of the trust node with respect to its trust neighbors as presented in [4].
- ID number of the beacon (ID-num): it is the sequence number of the beacon which is incremented by one at each beacon transmission by the node.
- Message Authenticated Code (MAC): this field is reserved to authenticate the beacon information signed with private key of the sender.

$$(MAC_{K-}[CA, Hopcount, DTN, RM, ID - num])$$

This information permits to any trust receiver to authenticate the sender of the beacon and verify the integrity of information.

At first, each trust node sends successive hello packet in order to calculate the relative mobility RM, after that, it announces itself as CA by assigning its own address to the CA field of the beacon and initializes the Hop Count. When a trust node receives a beacon, from one of its neighbors, it executes the clustering algorithm to change its status from clusterhead (CH that is also CA) to RA or cluster-member only. The decision to change the status from CA to cluster-member depends on two main factors: security and stability parameters. Two security parameters in this algorithm have been defined: the trust level and the numbers of trust neighbors of CA candidate. These parameters indicate the security hardiness of the future cluster and the degree of attacks resistance. Another parameter is introduced: the stability of the CA. A CA is considered more stable than another one if it has low relative mobility. Any trust node with relative mobility more than certain threshold is not considered stable and will not enter in CA competition with others candidates. When competition between two candidate CAs, the CA with lower trust neighbors and also more relative mobility, loses the competition and becomes RA or member only, it depends on the distance (i.e., hop count) from the winner CA. If the hop count is equal to 1, the candidate CA becomes RA. It means that all trust nodes, directly connected (one-hop) to CA winner, can become RA. The nodes situated between two adjacent clusters can become gateway (GW). The below algorithm 1 is executed by each node which has high trust metric $Tm = 1$; these nodes declare themselves as candidate to become CA. The extent of a node CA to manage nodes (in its cluster) at one hope or more depends on the value of d.

Algorithm 1. Clustering Algorithm executed by confident node

```
When receiving a beacon by node j from node i;
begin
    Authentication do if ($Tm(i)! = 1$) then
    RejectBeacon(); Goto(end);
    else if ($HopCount >= d$) then
        No − Competition; Goto(end);
    else if ($RM_i < RM_j$) OR (($RM_i == RM_j$) AND ($DTN_j < DTN_i$)) then
        Accept node i as CA;
        if ($HopCount == 1$) then
            $Status(j) = RA$;
            $HopCount(i) = 1$;
        else
            $HopCount(i) = HopCount + 1$;
            $Status(j) = MN$;
    else if ($RM_j < RM_i$) OR ($DTN_j > DTN_i$) then
        node j remains as CA candidate;
    else if ($RM_i == RM_j$) AND ($DTN_j == DTN_i$) then
        apply Lowest-ID;
end
```

In order to detect the topology changes, we introduce the movement detection process. Movement of CA is detected by RA nodes while not receiving any beacon from CA for predefined period of time. Also, cluster's nodes can detect movement of RA nodes by not receiving beacons from them. The movement detection of nodes CA and RA is very important for the cluster lifetime.

Algorithm 2. Algorithm executed by a node when its RA or CA is lost

```
If node i does not receive any beacon from CA after Timeout predefined;
begin
    if It can recover CA with another RA then
    Keep previous CA;
    Update RA node and $Hop\_count$;
    else if It can find another CA then
        Join the new CA node;
        if ($Tm(i) == 1$) then
            if ($HopCount == 1$) then
                $Status(i) = RA\_NODE$;
                $HopCount(newCA) = 1$;
            else
                $Status(i) = MN$;
                $HopCount(newCA) = HopCount + 1$;
        else
            Request Certificate to RA node;
end
```

Each cluster's node other than CA or RA receives the beacon from CA. It must verify the authentication and the integrity of the beacon information by using the CA's public key ($K_{CA}+$). If the verification succeeds then the node updates any change about hop-count or new RA. If CA changes, cluster nodes verify the new CA identity. The information over the nodes can be assembled by trust model.

Each member cluster's node update periodically the cluster's information (CA and RA nodes), for more detail, the reader can refer to [4].

4 Performance Evaluation

We have implemented our clustering algorithm as described previously. We use Network Simulator (NS-2) [14] with CMU wireless extensions to simulate our algorithm. Simulation scenarios were generated with parameters listed in the table 1. The movement of mobile nodes is randomly generated and continuous within the whole simulation periode.

In order to compare the algorithm proposed in the previous section with others clustering algorithms. We assume that all nodes of the network are high trust level which means that any node can become CH.

Table 1. Simulation parameters

Parameter	Value in our simulation
Number of nodes (N)	50
Network size (mxn)	$670x670m2$
Constant mobility	20 m/sec
Transmission Range	10 m - 125 m
Pause time	0.30 s
Broadcast interval (BI)	0.75-1.25 s
Discovery interval	10*BI s
Contention periode	3.0 s

In Figure 4, we note that there is difference between our algorithm, MOBIC and Lowest-ID in the transmission range 50 m, because our algorithm need at least two nodes to form cluster, only one (isolated node) cannot become CA for security raison. In this simulation, the number of conceived clusters does not exceed 25. With a transmission range between 50 and 125 m the number of clusters decreases and more of 150 m the network become more stable. However, while fixing d=2 we obtain less cluster-head compared to MOBIC and lowest-ID.

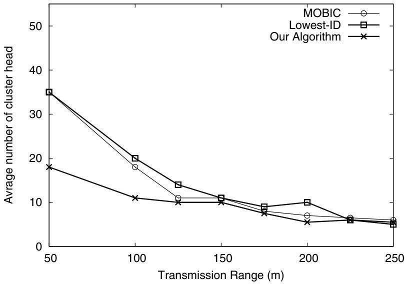


Fig. 4. Comaprison between different clustering algorithms

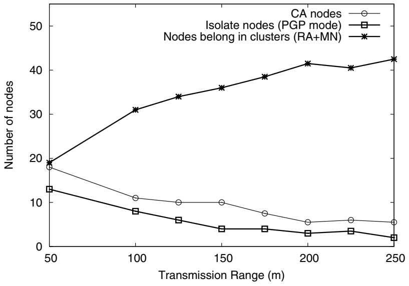


Fig. 5. Average number of different status of nodes

The cluster-head (CA) duration is the parameter which indicate the cluster stability. The longer CA duration means the system is more stable. The simulations with 100 nodes, $2500x2500$ of scenarios size and $12.5m/s$ of maximum speed gives 12.8 second of average CA duration.

Figure 5 shows the average number of different status of nodes in the network. The average number of isolated nodes (nodes cannot join any cluster) decreases when the transmission range increases. Also the number of CAs decreases with longer transmission range. The number of other nodes (member of different clusters) increases when the transmission range increases. The number of isolated nodes must be reduced to get more security communication in the network.

5 Security Analysis

The security of our architecture depends directly on the trust model. The presence of a great number of confident nodes increases the security of the network. Nodes with low trust level cannot participate in the CA election process. Only a confident node can announce itself as CA candidate. If a malicious node try to be introduced in the CA process election by announcing itself as candidate, the confident nodes can detect this in authentication phase showed in algorithm 1. If malicious nodes succeed to form their cluster and try to communicate with other clusters; the CA of cluster destination can authenticate the CA of the source cluster in inter-cluster communication. All communications from a malicious cluster are ignored.

The Denial-of-Service (DoS) attack over CA node is prevented by DDMZ where RA nodes filter all requests from unknown nodes. The robustness of DDMZ depends on the number of RAs which collaborate in order to protect CA of their cluster. If attackers try to impersonate legitimate nodes as CA or RA they will be detected by monitoring process and then isolated from the network. The malicious nodes can use the identity of legitimate nodes only if their private's keys are divulgated. If attackers try to compromise all the network, it must compromise all CAs.

The number of clusters formed by our proposed solution is related to the number and the mobility of confident nodes. The cluster size must be adapted with number of confident nodes in order to well secure CA node. The presence of two confident nodes is the minimum configuration of clustering and it must be reinforced.

We can use the thresholds cryptography scheme in each cluster after CA election. A CA divides its private key into n partial shares which are distributed over RA nodes.

Our system's architecture obliges nodes to collaborate and to adapt well behaviors to obtain more trust levels. Each unknown node must begin with a visitor status and then obtain the member status.

In order to evaluate the trust of CA authentication, we calculate the quality of authentication (QoA), so that, we apply attenuation factor to trust chain [6]. This factor is $(1-p)^{(d-1)}$ where p is the probability of the existence of compromised or a malicious node in the network and d the length of the trust chain.

$$QoA(V_1 - V_2) = TC(V_1 - V_2) * (1-p)^{(d-1)} \tag{1}$$

The more trust chain is longer the more risk to be compromised is important. In this case the cluster size must be carefully chosen.

The QoA between two clusters depends of the trust chain (TC) which attach CA nodes of clusters and also percentage of malicious nodes in the network. The communication between CAs must passed via high trust chain and it is assured by getaway nodes (GW).

The figure 6 illustrate the quality of authentication versus probability of malicious nodes. We have plot curves in the case of cluster size 1 and 2 hop with

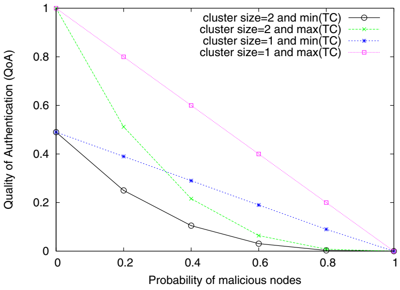


Fig. 6. QoA versus probability of malicious nodes

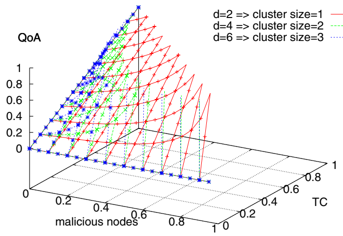


Fig. 7. QoA versus probability of malicious nodes and trust chain

maximum and minimum values of TC respectively 1 and 0.49 $(0.7 * 0.7)$. We remark the QoA linearly decrease with probability of malicious nodes increase in the case of one hop of cluster size. When we increase the cluster size at two hop we note that, QoA decrease more fast with probability of malicious nodes than the case of one hop of cluster size.

The figure 7 shows the general case of QoA with different values of TC and probability of malicious nodes. We compare three case of cluster size 1, 2, and 3 hop, we remark the best value of QoA is in the case of one hop of cluster size, low value of malicious nodes and high TC.

According to the last figures 7 and 6, we can conclude that, more of the cluster size is large, the risk to have weak QoA is hight.

6 Conclusion

In this paper, we have proposed a new architecture based on our trust model and clustering algorithm in order to distribute a certification authority (CA).

Our clustering algorithm is based on two parameters: security and stability. The security factor is related to the trust model; only confident nodes can become cluster-head and ensure CA role. The stability factor is presented by mobility metric in order to give more stable clusters. In our approach, the trust model is accomplished by monitoring process which allows any node with high trust metric to monitor and evaluate other nodes with low trust metric. In addition, we have proposed a new mechanism to protect CA, called DDMZ, which permits to increase security robustness of clusters and endures malicious nodes that try to attack CA or issue false certificates.

Our architecture ensures the security and availability of public key authentication in each cluster. This architecture is adapted to any topology changes.

Simulation results of our clustering algorithm showed the improvement of clusters stability compared to MOBIC and Lowest-ID algorithms. Furthermore, we remark that availability and robustness of DDMZ depend on the transmission range, the number and mobility of confidant nodes. We are also considering energy conservation and lifetime of the network while conceiving clusters. Our future work is to study and analyse our architecture in order to evaluate the resistance degrees of DDMZ faced to different DoS attacks.

References

1. Philip R. Zimmermann: The official PGP user's guide.MIT Press Cambridge. "MA, USA. (1995)
2. S. Capkun and L. Buttyan and J. Hubaux: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. ACM International Workshop on Wireless Security, WiSe. **2** (2002) 52–64
3. P. Basu and N. Khan and T. Little: A mobility based metric for clustering in mobile ad hoc networks. In Proceedings of Distributed Computing Systems Workshop. (2001) 43–51
4. A. Rachedi and A. Benslimane: A Hiearchical Distributed Architecture to Secure Ad-Hoc Networks. Research Technical Report LIA. (2006)
5. M. Gerla and J. T.-C. Tsai: SMulticluster, Mobile Multimedia Radio Networks. Wireless Networks. (1995) 255–256
6. S. Yi and R. Kravets: Quality of Authentication in Ad Hoc Networks. ACM, MobiCom2004. (2004)
7. I. Inn Er and Winston K.G. seah. Mobility-based d-hop Clustering Algorithm for Mobile Ad Hoc Networks. (2004)
8. S. Capkun and J. P. Hubaux and L. Buttyan: Mobility Helps Peer-to-Peer Security. IEEE Transactions on Mobile Computing. **5** (2006) 48–60
9. C. Chiang and H. Wu and W. Liu and M. Gerla: Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel. IEEE Proceedings of SICON'97. (1997) 197–211
10. M. Gerla and J. T.-C. Tsai: Multicluster, Mobile Multimedia Radio Networks. Wireless Networks. (1995) 255–256
11. Lidong Zhou and Zygmunt J. Haas: Securing Ad Hoc Networks. IEEE Network. **13** (1999) 24 –30
12. Marc Bechler and Hans-Joachim Hof and Daniel Kraft and Frank Pahlke and Lars Wolf: A Cluster-Based Security Architecture for Ad Hoc Networks. INFOCOM2004. (2004)
13. Kimaya Sanzgiri and Bridget Dahill and Daniel LaFlamme and Brian N. Levine and Clay Shields and Elizabeth M. Belding-Royer: An Authenticated Routing Protocol for Secure Ad Hoc Networks. Selected Areas in Communication (JSAC). **23** (2005) 598–610
14. UC Berkeley and USC ISI: The network simulator ns-2. Part of the VINT project. Available from http://www.isi.edu/nsnam/ns. (1998)

Interlayer Attacks in Mobile Ad Hoc Networks

Lei Guang[1], Chadi Assi[1], and Abderrahim Benslimane[2]

[1] Concordia Institute for Information System Engineering, Concordia University,
Montréal, Québec, Canada, H3G 1M8
{l_guang, assi}@ciise.concordia.ca
[2] Laboratoire Informatique d'Avignon, Université d' Avignon
84911 Avignon Cedex 9, France
Abderrahim.Benslimane@univ-avignon.fr

Abstract. In this paper, we demonstrate a new class of protocol compliant exploits that initiates at the MAC layer but targets ad hoc on-demand routing mechanisms. A misbehaved node implementing this type of attacks completely follows the specifications of IEEE802.11 standard and the existing on-demand routing protocols. However, it can cause routing shortcut attacks or detour attacks. We detail the exploits against two on-demand routing protocols: AODV and DSR. We evaluate the impact of such attacks on the network performance and propose PSD (*Prevention from Shortcut and Detour Attack*) to mitigate their impacts.

Keywords: Ad hoc networks, medium access control, security.

1 Introduction

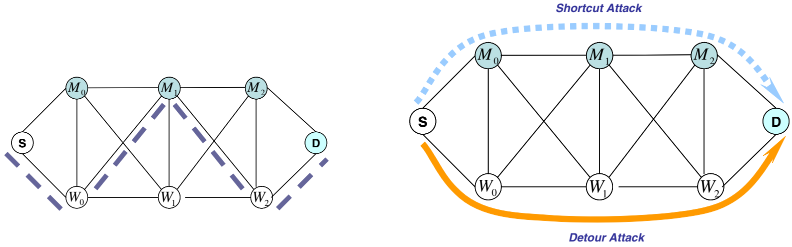
Previous attack strategies in mobile ad hoc network (MANET) target either the network layer or the MAC layer. Interlayer attack strategy, however, has not been fully addressed. In this paper, we present attacks initialized at MAC layer but aiming at ad hoc routing mechanisms and also propose the schemes to mitigate such attacks. Lately, significant research efforts have focused on improving the security of ad hoc networks. In mobile ad hoc networks (MANET), nodes are both routers and terminals and due to the lack of a routing infrastructure these nodes have to cooperate to ensure successful communication. Clearly, cooperation means ensuring correct routing establishment mechanisms, the protection of routing information and the security of packet forwarding [5]. One major challenge that was neglected previously by the research community is that of securing against MAC layer misbehaviors, particularly MAC misbehavior that can evade from the existing MAC misbehavior detection systems [3], [4], [6] and affect the cross-layer performance, such as throughput and delays.

The rest of this paper is organized as follows: Section 2 elaborates two interlayer attack strategies: ShortCut Attack (SCA) and DeTour Attack (DTA). Section 3 illustrates the impacts of SCA and DTA on the ad hoc routing protocols using case studies. Section 4 analyzes the network throughput under such attacks via modeling. Section 5 proposes a method *PSD* to mitigate SCA and DTA. Section 6 presents the simulation experiments. Finally Section 7 concludes this paper.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 436–448, 2006.

2 Interlayer Attack Strategy

In this section, we present shortcut attack (SCA) and detour attack (DTA) that initialize at MAC layer but aim to disrupt the performance of ad hoc routing mechanisms. Through simple manipulation of IEEE 802.11 backoff procedure, SCA can attract more flows and drop the packets to cause denial of service whereas DTA can help a node to reduce the chance to be selected as a forwarding node thus conserve its device energy.



(a) Normal Case

(b) Attack Case

Fig. 1. Route changes - Case (I)

Shortcut Attack: In [2], the authors introduced a new category of routing attack, namely, rushing attack. Generally, the attacker can propagate the routing messages faster than other well-behaved nodes (WN) using a wormhole. Therefore, it effectively increases the probability that any discovered new route contains the mis-behaved nodes (MN). Once selected as a relaying node, the MN can implement other attacks, such as *JellyFish* and *Dropping RTS/DATA*, to discard the data packets which leads to denial of service. In this paper, we present a simple technique to implement rushing attack. Recall that, a node with a data packet (either broadcast or unicast) to send out has to set up a backoff timer by randomly choosing a cw from $[0, CW]$. However, a MN can intentionally pick up a smaller cw upon reception of a routing packet. Note that, a MN does not need to know the exact packet type, e.g., routing packet[1]. Therefore, this MN is capable of accessing the channel faster than its neighbors to relay the RREQ. As a result, the route containing the MN will have more opportunity to be selected because the routing protocol "thinks" this route is a *short one*. In the rest of the paper, we refer to this attack as shortcut attack. Note that, unlike other backoff manipulation misbehavior [3], in this case a MN needs to manipulate cw only for a routing packet.

Detour Attack: Contrary to shortcut attack, the objective of detour attack aims at conserving the attacker's limited device energy by choosing to forward

[1] Since RREQ (RREQ is a route request control packet used by reactive routing protocol in order to discover a route between a source and a destination) is a broadcast packet, a MN can start misbehaving *only* when it receives a broadcast packet, i.e, address = 255.255.255.255.

less data packets. A MN implementing DTA forces a flow to *detour* around itself by delaying the propagation of routing messages, e.g., RREQs, which will allow RREQ forwarded by WN to arrive at the destination sooner. Hence, the attacker will reduce the possibility of being selected as a forwarding node. Here, unlike SCA, a MN chooses a larger *cw* to backoff for a longer duration than a well-behaved node. If its neighbors also receives the RREQ, they will have more chance to capture the channel and send a packet if they are WNs. Consequently, the RREQs broadcasted by WNs will arrive at the destination faster than those being forwarded by MNs. Hence, a MN could conserve its energy by evading being selected as a router.

3 Case Study

In this section, we use two simple cases to illustrate the procedure of SCA and DTA in both AODV and DSR.

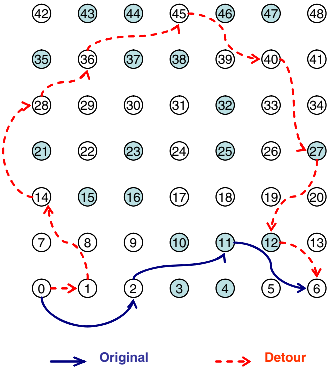


Fig. 2. Route changes - Case (II)

Case (I): In Fig. 1(a), Node S originates a data flow addressed to node D. Nodes $M_0, \cdots, M_2$ are misbehaving whereas nodes $W_0, \cdots, W_2$ are well-behaving. The solid lines in the figure represent the physical connections between each pair of nodes. The distance between node M_i (W_i) and node M_{i+1} (W_{i+1}) equals to 100m whereas the distance between node M_i and node W_i is 200m. Moreover, for the following discussions, the transmission range for each node is 250m and the carrier sense range is 550m. Initially, all nodes are well-behaved. The data flow will be routed through $S \rightarrow W_0 \rightarrow M_1 \rightarrow W_2 \rightarrow D^2$ (denoted by the dashed line). When all the MNs are activated, the route will be disrupted and changed as follows:

[2] Upon different selection of the simulation seeds, there exists multiple choice of routes from S to D. Here, we only show one specific example.

Table 1. Number of forwarded packets - Case (II)

AODV	route length	pkt (W)	pkt $M(M_{11})$
Normal	3 hops	985	981(981)
Detour	9 hops	5485	1791(0)

- *SCA:* when S broadcasts a RREQ, both M_0 and W_0 receive this message. Consider each RREQ is transmitted without any delay at the network layer, M_0 will send the RREQ to its neighbors. Here, the RREQs broadcasted by M_0 will be discarded by S and W_0, i.e., M_1, W_1, faster than W_0 because it selects a smaller *cw* than W_0. Later, when the RREQs sent by W_0 arrives at M_1 and W_1, they will be dropped because each node has already "seen" a *recent* RREQ (the one from M_0). As a consequence, both M_1 and W_1 will forward the RREQ sent from M_0. Hence, the final discovered route will contain M_0. The same scenario happens for other MNs as well. Finally, for this scenario the new discovered route will always be $S \rightarrow M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow D$ for any simulation seeds (as shown in Fig. 1(b) denoted by the dotted line). Hence, the MNs succeed in *attracting* this flow.
- *DTA:* Unlike the previous scenario, here a MN will always choose a larger *cw*, e.g., from range $[0, 3 \times CW]$ instead of $[0, CW]$. This will allow the WNs to flood the RREQs faster than MNs. As a result, the discovered route will have less probability to contain MNs. In this scenario, the MNs conserve energy while causing extra traffic load for other nodes (which might be MN or WN as discussed in Case (II)). The route change is shown in Fig. 1(b), denoted by the thick solid line.

Case (II): As shown from Fig. 2, we see that the route has successfully changed from 3 hops to 9 hops. The setup of this experiment is a grid network of 7×7 nodes. The grid unit is 100 meters. There are 49 (numbered 0 to 48) nodes that are positioned on the grid. The seed for this simulation run is 100 and the data rate is 10 packets/second with the packet size of 512 bytes. The simulation time is 100 seconds. On one side, the MN (M_{11}) "evades" from the route discovery and is successful to conserve its energy. On the other side, this route change has caused extra traffic load for other nodes. Table 1 shows that M_{11} relays 0 packet whereas WNs $(1, 14, 28, 36, 45, 40)$ and MNs $(27, 12)$ have extra traffic load. Note that, as long as this route is maintained, the throughput of this flow will not be affected. However, when node mobility is high or the network environment is congested, a longer route is easily broken and therefore shorter routes will be preferred. In this case, when the MNs misbehave the flow will have less chance to traverse through a long route and maintain the connection.

4 Modeling and Analysis

In this section, we use simple models to explore the impacts of interlayer attacks on the network performance.

Throughput Analysis: First, we assume that there is no idle time which means that a source will always have packets to send to a destination during the whole simulations. Next, consider we have N nodes in an ad hoc network, M out of N are misbehaved nodes (MN) ($0 \leq M \leq N$). The route length for a flow is denoted as l. Let $P\{i|j\}$ denote the probability that the i_{th} forwarding node is misbehaving given that j MNs have been already selected for the same flow:

$$P\{i|j\} = \begin{cases} 0, & M = 0 \\ (M-j)/(N-i), & 0 < M < N \\ 1, & M = N \end{cases} \tag{1}$$

Assume a relaying node is randomly selected from the N nodes in the network, the probability that a route contains no misbehaved nodes P_{WN}[3] is given by:

$$\begin{aligned} P_{WN} &= (1-P_1)(1-P_2)\cdots(1-P_l) \\ &= \prod_{k=1}^{l}(1-P_k) \end{aligned} \tag{2}$$

The normalized throughput is computed by $\frac{E[T_{flow}]}{E[T_{sim}]}$, where T_{flow} is the time duration to transfer the actual data. T_{sim} is the total simulation time. Similar to [1,7], T_{sim} is given by:

$$T_{sim} = T_{flow} + T_{repair} \tag{3}$$

where T_{repair} is the time for an on-demand routing protocol to repair a route after breakage due to mobility, link failure, etc. T_{repair} consists of four parts which are given by:

$$T_{repair} = T_{diag} + T_{err} + T_{req} + T_{rep} \tag{4}$$

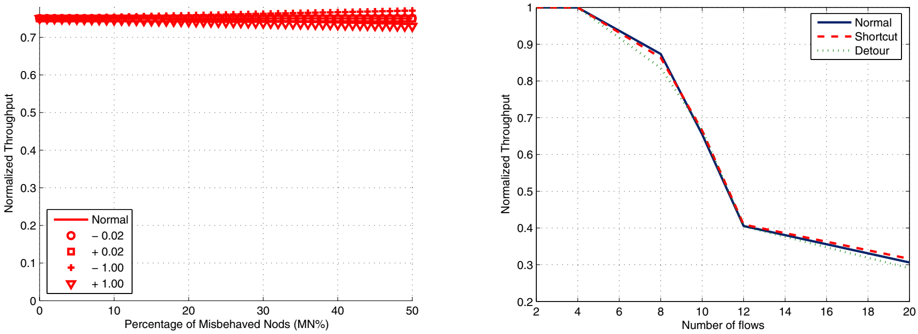
First, T_{diag} is the time to diagnose a new route breakage. Second, the time to generate a RERR message to the upstream nodes is denoted as T_{err}. Moreover, the time to initialize a new route discovery by flooding RREQs is denoted as T_{req}. Finally, the time duration for nodes to receive a RREP is denoted as T_{rep}. The life time of an existing route is denoted as T_{life}, i.e., route duration. We can see that $T^f_{flow} = T^f_{life}$ for flow f. Thus, the throughput is computed as:

$$S_{norm} = \sum_{f=1}^{F}\left(\frac{E[T^f_{life}]}{E[T^f_{life}] + E[T^f_{repair}]}\right) \tag{5}$$

where F is the total number of flows. In the presence of SCA or DTA, the T_{repair} will either decrease (SCA) or increase (DTA) a small fraction which is denoted as δ. Therefore, we have the following equation:

$$\begin{aligned} S_{SCA/DTA} &= \sum_{f=1}^{F}\left(\frac{E[T^f_{life}]^*}{E[T^f_{life}]^* + E[T^f_{repair}]^*}\right) \\ &= \sum_{f=1}^{F}\left(\frac{E[T^f_{life}] \pm \gamma\cdot\delta}{(E[T^f_{life}] \pm \gamma\cdot\delta) + (E[T^f_{repair}] \mp \gamma\cdot\delta)}\right) \end{aligned} \tag{6}$$

[3] We adopt the short notation $P_k = P\{k \mid j\}$.



(a) Modeling Throughput

(b) Simulation Throughput

Fig. 3. Throughput Analysis under SCA/DTA

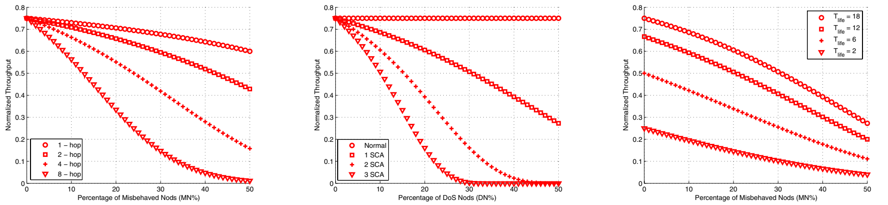
where in $\mp$, "$-$" refers to SCA, "+" refers to DTA and γ refers to the percentage of SCA/DTA nodes. From the results shown in [7], we choose $E[T^{f}_{life}]$ as 18 second corresponding to node velocity is 20 meters/second and $E[T_{repair}]$ as 5 seconds[4]. In the presence of SCA, $E[T_{repair}]$ is decreased to $E[T_{repair}]^{-}$ by $\gamma \cdot \delta$ due to accelerated propagation of routing messages. On the contrary, in DTA, $E[T_{repair}]^{+}$ represents an increased repair time duration. The delay might be introduced by both MAC layer and routing layer. In our case, we focus on delay incurred by MAC layer contention. Consider a MN that selects a $cw = 1023$ (which is the CW_{max} specified by IEEE 802.11) and $T_{slot} = 20us$, the maximum delay for one transmission attempt is 20 ms for DTA. Moreover, the minimum delay is 0 ms for SCA.

In Fig. 3(a), it is shown that the throughput comparison under SCA (denoted as "-x", where x is the time subtracted from $E[T_{repair}]$), DTA (denoted as "+x", where x is the time added to $E[T_{repair}]$) and normal case. Clearly, the throughput are all comparable to the normal case. However, in SCA the throughput slightly increases whereas in DTA the throughput decreases based on the value of the delay. We can also draw the same conclusion from the simulation results as shown in Fig. 3(b)[5]. It shows that the throughput for SCA/DTA is similar to the normal case in terms of number of flows. Note that, according to the standard, the maximum delay introduced in MAC is around 20ms. If a DTA node has the ability to completely abuse the standard, it can select a very large cw, e.g., 50000, to sharply reduce the network throughput.

Furthermore, as we explained in Section 2, SCA has to combine with DoS attacks to degrade the network performance otherwise SCA will only attract

[4] T_{req} is determined by *RREQRatelimit* (which is the maximum number of RREQs allowed to originate by a node) and *RREQRetries*. T_{rep} is determined by the *NetTraversalTime*.

[5] In this simulation, there are 49 nodes in the network with 10 flows randomly generated.



(a) Throughput Comparison under Different Route Length (b) Throughput Comparison under Different Fraction of SCA Nodes (c) Throughput Comparison under Different Life Time

Fig. 4. Impacts of SCA combined with DoS Attacks

flows. We assume M MNs can implement any type of dropping attacks and each single MN can disrupt the whole flow. Hence, P_{WN} is a constant which equals to $(1 - \frac{M}{N})^l$. As explained in [1], in the presence of misbehaved nodes, the time to repair a route that contains no misbehaved nodes is given by:

$$E[T_{repair}] = \sum_{n=1}^{\infty} n \times (E[T_{diag}] + E[T_{req}] + E[T_{rep}]) \times P_{WN} \times (1 - P_{WN})^{n-1} \tag{7}$$

where n is the number of attempt times for a node to repair a broken route. For simplicity, we suppose that all repairs have the same duration and all flows have the same length. Thus we have:

$$S^f_{SCA-DoS} = \frac{E[T^f_{life}]^-}{E[T^f_{life}]^- + E[T^f_{repair}]^- \times P^{-l}_{WN}} \tag{8}$$

As Fig. 4(a) shows, in SCA-DoS attack, the throughput of longer routes will drop to zero quickly as the $MN\%$ increases. This is due to the facts that: 1)longer routes have more probability to contain MNs; and 2) it requires longer duration to repair the broken links. For example, throughput for 2-hop is higher than 8-hop by almost 100% when $MN\%$ of both cases equals to 20%. Fig. 4(b) indicates that collisions between SCA and DoS can cause severe performance degradation if SCA nodes can successfully attract more flows than normal cases. It is clear that the more data flows attracted to the SCA nodes, the worse the network throughput is ("n-SCA" in the figure represents n flows have been attracted to the SCA nodes). Furthermore, Fig. 4(c) depicts the total network throughput when varying the route life time under 3-hop case. Different value of life time corresponds to different mobility, e.g., the lower the life time the higher the node mobility. Clearly, the throughput decreases as $MN\%$ increases. Higher mobility results shorter route duration, thus it will cause degradation of network throughput due to frequently broken routes and repairing procedure.

5 Proposed Scheme

In this section, we describe our scheme *Prevention from Shortcut Attack and Detour Attack* (PSD) to defend against interlayer attacks.

5.1 PSD - Part I: Randomized Routing Messages Selection

Upon the reception of a RREQ, a node will check to determine whether this message is a recent RREQ. For AODV, this is determined by receiving a RREQ from the same originator address with the same RREQ ID during a time period greater or equal than *PathDiscoveryTime*. In DSR, a node considers a RREQ recently seen if it still has information about that Request in its Route Request Table. Both protocols will discard the upcoming RREQ which is considered *"recent"*.

In [2], the authors proposed randomized message forwarding to mitigate rushing attacks under the assumption that each forwarding node is capable of collecting the maximum possible number of RREQs when given "perfect" information, such as network topology. However, such perfect information is usually unavailable in real ad hoc networks since there is no centralized management. Here, we propose a similar approach and specify how to determine the timeout T_{to} during which RREQs are buffered and one is randomly selected. If an intermediate node (IN) receives the first request $RREQ_1$ destined to node D at time t_{RREQ_1}, it sets T_{to}^{lb} (lower bound of T_{to}) to t_{RREQ_1} and T_{to}^{ub} (upper bound) to $T_{to}^{lb} + (2^{i_{min}} - 1) \times T_{slot}$, where $2^{i_{min}} - 1 = CW_{min}$ is the minimum contention window size. Consequently, it buffers $RREQ_1$. If the IN senses the channel is busy, the timeout timer is frozen. Otherwise it will continue to count down until it times out. During this timeout interval ($T_{to}^{lb} < t_{RREQ_i} \leq T_{to}^{ub}$), if another RREQ arrives, it is buffered. In the presence of a collision at t_{col}, the IN assigns t_{col} to T_{to}^{lb} and increases T_{to}^{ub} to $T_{to}^{lb} + (2^{i_{min}+N_{col}} - 1)T_{slot}$, where N_{col} is the number of all collisions. To avoid unlimited waiting for the retransmitted packets, the maximum $T_{to}^{ub_{max}}$ is defined as $T_{to}^{lb_1} + (2^{i_{max}+1} - 1)T_{slot}$. As we explained in Section 4, the delay incurred at MAC layer is small, the IN can also simply set its T_{to}^{ub} to the maximum value. After the expiration of the timeout timer, the IN can randomly choose a $RREQ_i$[6] to forward, which reduces (increases) the probability that a SCA (DTA) node is included (excluded) in the routing selection.

5.2 PSD - Part II: Randomized Delay of RREQs

This is a complementary scheme to *PSD - part I.* Recall that the delay introduced by the backoff manipulation is in terms of milliseconds, therefore we can ignore this fraction of delays at higher layer. For example, rather than forwarding the newly received RREQ immediately, the routing protocol can delay the transmission of this RREQ by a small delay, e.g., a delay uniformly selected from $[0, t_{jit}]$

[6] An IN can also apply certain policy to randomize the RREQ selection, e.g., randomly select RREQ from N_1 *slow* RREQ and N_2 *fast* RREQs ($N_1 > N_2$).

Algorithm 1. PSD

```
if $collisions\ happen$ then
  $N_{col}++$
end if
for $each\ recv\ RREQ_i$ do
  if $i == 1$ then
    $T_{to}^{lb_i} = t_{RREQ_1}$
    $T_{to}^{ub_i} = min(T_{to}^{ubmax}, (t_{RREQ_1} + (2^{i_{min}+N_{col}} - 1) \times T_{slot}))$
    $Buffer\ (RREQ_1)$
  else
    if $t_{RREQ_i} > T_{to}^{ub_i}$ then
      $Discard\ (RREQ_i)$
    else
      if $N_{col} > 0$ then
        $T_{to}^{lb_i} = t_{RREQ_i}$
        $T_{to}^{ub_i} = min(T_{to}^{ubmax}, (t_{col} + (2^{i_{min}+N_{col}} - 1) \times T_{slot}))$
        $Buffer\ (RREQ_i)$
      else
        $Buffer\ (RREQ_i)$
      end if
    end if
  end if
end for
if $T_{to}\ timer\ expires$ then
  $forward(RandomRREQ, jitter^8)$
  $N_{col} = 0$
end if
```

milliseconds[7]. As shown in Section 4 Fig. 3(a), it is clear that the throughput is close to that of normal case when a smaller delay is introduced. However, this operation is based on the assumption that the MN will not be able to break the routing protocol. Therefore, a secure on-demand routing protocol is required. Several protocols have been already proposed. For more details, readers please refer to [8], [5], [9].

5.3 DSD: Detection of SCA and DTA

If a MN is capable of choosing a very large *cw* beyond CW_{max}, it can defeat the PSD scheme and even cause devastating performance degradation as we mentioned in Section 4. Here we presents a simple method to detect such behavior. After the timeout timer for the RREQ buffering expires, the IN continues to record the arrival time of received RREQ for the same destination. If the RREQ is incredable *late*, e.g., in AODV the RREQ bearing the same sequence number arrives 1 second later than the first received RREQ, the IN can mark the source of the RREQ as a MN. To ensure correct diagnosis ratio, the IN can monitoring for a period to collect more information. Note that, the longer the delay of the RREQ, the higher the correct diagnosis.

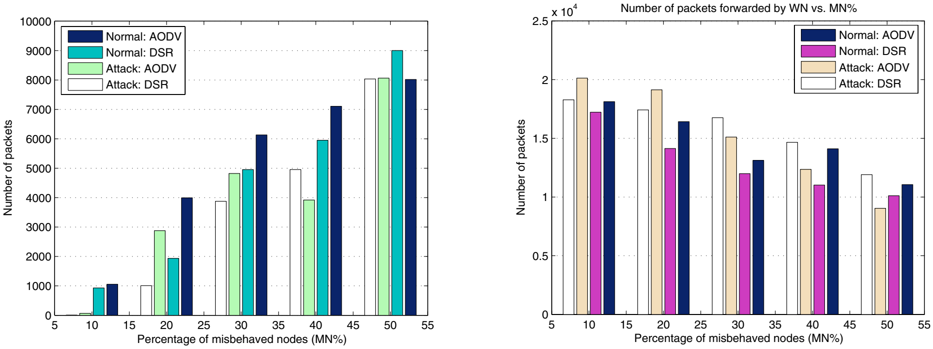
[7] AODV jitters the sending of broadcast packets by 10ms by default, however, this is not enough to counter the delay introduced by MAC. In our experiments, we jitter the broadcast messages by 100ms which is sufficient to solve the problem as shown in Table 2.

6 Simulation and Analysis

We use NS-2 to evaluate the impacts of interlayer attacks and the efficiency of our proposed scheme PSD.

6.1 Static Networks

Fig. 5 shows the number of packets forwarded by MN (WN) under DTA. The network topology in this experiment is the same as Fig. 2. There are 8 CBR flows in the network and 40% out of 49 nodes are mis-behaved (always selecting $cw = 127$ for routing messages) which are randomly positioned in the network. Fig. 5(a) indicates that the MNs in DTA always forward less packets than WNs in both AODV and DSR. However, when the fraction of the MNs is large, the MNs can not successfully detour the traffic, e.g., when $MN\% = 50$ the number of forwarded packets is comparable for attack and normal cases in AODV. This is because the neighbors of the MNs might be also misbehaving. Similar results can be seen from Fig. 5(b) as well. Table 2 shows the efficiency of PSD when applied to the Case I (see Fig. 1(a)). The results show that PSD can partially mitigate the impacts caused by DTA/SCA.



(a) Mis-behaved Nodes (MN)

(b) Well-behaved Nodes (WN)

Fig. 5. Number of packets forwarded by MN (WN) under detour attack

Table 2. Number of forwarded packets (Case - I)

AODV	normal	DTA	SCA	PSD I (DT)	PSD II (DT)
W	7992	11988	0	5994^9	7992
M	3996	0	11988	5994*	3996

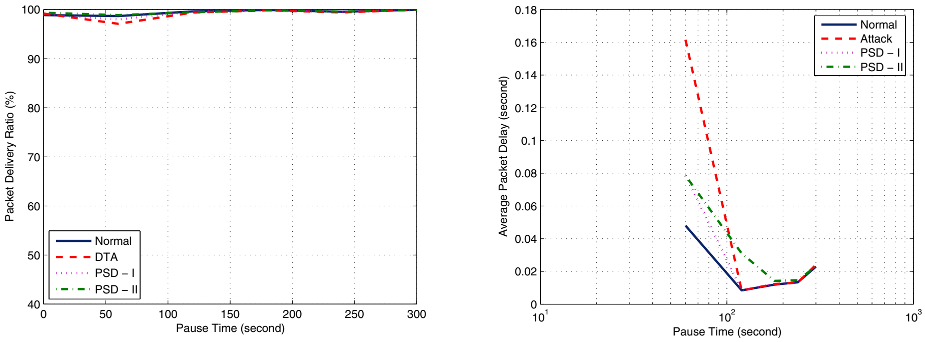
DSR	normal	DTA	SCA	PSD I (DT)	PSD II (DT)
W	7986	11979	0	5994*	7986
M	3993	0	11979	5994*	3993

6.2 Mobile Ad Hoc Networks

Our protocol evaluations are based on the simulation of 50 wireless nodes forming an ad hoc network, moving over a rectangular $1500m \times 300m$ area. Random waypoint model is used for generating the mobility pattern. The transmission range for each node is 250m and the carrier sense range is 550m. 10 flows are randomly generated between a source-destination pair. The traffic type is constant bit rate (CBR). The packet size is 512 bytes/packet and the data rate is 4 packets/second for each flow which is a relatively low load. The channel bit rate is 2Mbps. The total time for each simulation run is 300 seconds. For brevity, only DSR is discussed in the following section. Similar results can be also obtained from AODV. To model the interlayer attacks, we consider only detour attack where a misbehaved node will always select $cw = 31$ for the routing packets.

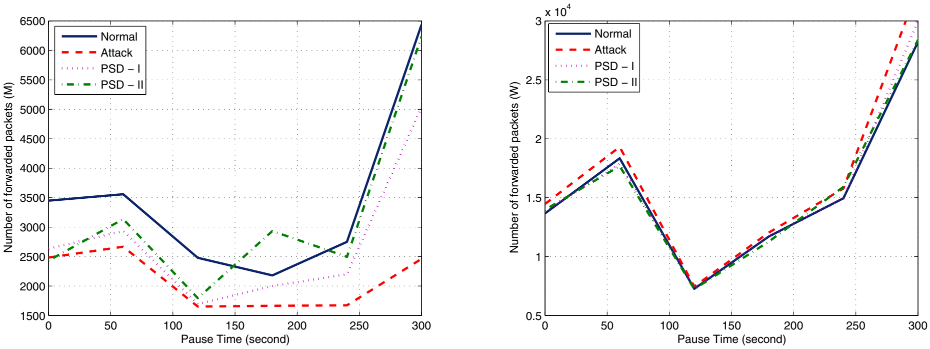
Fig. 6(a) shows the network packet delivery ratio when varying the pause time under normal case (depicted as "Normal"), attack case (depicted as "Attack") and PSD (depicted as "PSD - I" and "PSD - II"). Lower node pause time corresponds to higher mobility, e.g., 0 second pause time means that a node will continue to move around during the whole simulation. It is clear that, there is no significant difference between these cases, however, delivery ratio of DTA is smaller than the PSD and normal case when the mobility is higher. Furthermore, as shown in Fig. 6(b), the average delay in DTA is much higher normal case even in a less congested environment which reflects the fact as we discussed in Section 3, that is a flow might traverse longer routes than normal case in the presence of DTA resulting an increased delay. Clearly, both PSD - I and PSD - II can mitigate this negative impact when the node mobility is high. Also, the delay of PSD - I and PSD - II are comparable to the normal case. The largest average delay difference between the normal case and PSD is around 0.03 second. Moreover, PSD - II incurs more delay than PSD - I because it always jitters routing packets by $20ms$ where the delay caused by PSD - I is between the range $[0, 20ms]$.

Fig. 7 depicts the number of packets forwarded under normal case, attack case and PSD when the percentage of misbehaved nodes $MN\%$ is fixed at 10%



(a) Packet Delivery Ratio (b) Average Packet Delay

Fig. 6. Overall Network Performance



(a) Mis-behaved Nodes (MN)

(b) Well-behaved Nodes (WN)

Fig. 7. Number of packets forwarded by MN (WN) under detour attack

and the misbehaved node IDs are the same for each simulation run. Fig. 7(a) clearly shows that DTA nodes forward much less packets than the normal case indicating they succeed in avoiding to be selected by the routing protocol as a router. This will also cause an extra traffic load for well-behaved nodes as shown in Fig. 7(b) where in DTA the misbehaved nodes forward more packets than the normal case. Furthermore, both PSD - I and PSD - II can mitigate such behavior efficiently. PSD - II has better performance than PSD - I, however, at the cost of a higher delay compared with PSD - I and a secure routing protocol.

7 Conclusion

We presented two simple attacks implemented at MAC layer, however, affecting ad hoc on-demand routing mechanisms. A misbehaved node can use shortcut attack to increase the probability to be selected as a relaying node. After attracting flows traversing through itself, the MN can carry out DoS attacks to degrade the overall network performance. Alternatively, a node using detour attack can reduce the probability to be discovered by the routing discovery process therefore saving its limited device energy. Via simulation, we showed that SCA/DTA can affect both AODV and DSR and we proposed PSD scheme which could successfully mitigate the impacts of SCA/DTA. For future work, we plan to further study and analyze our scheme PSD - II at the network layer under the consideration of a secure protocol.

References

1. I. Aad, J. P. Hubaux, and E. W. Knightly. Denial of service resilience in ad hoc networks. In *Proc. of ACM MobiCom*, September 2004.
2. Y.-C. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proc. ACM WiSe*, 2003.

3. P. Kyasanur and N. Vaidya. Selfish MAC layer misbehavior in wireless networks. *IEEE Transactions on Mobile Computing.*, September 2005.
4. L. Guang and C. Assi. A self-adaptive detection system for MAC misbehavior in ad hoc networks. In *Proc. of IEEE ICC*, June 2006.
5. P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *Proc. of CNDS*, 2002.
6. M. Raya, J. P. Hubaux, and I. Aad. DOMINO: A system to detect greedy behavior in ieee 802.11 hotspots. In *Proc. of ACM MobiSys*, June 2004.
7. N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. PATHS: analysis of path duration statistics and their impact on reactive manet routing protocols. In *Proc. of ACM MobiHoc*, 2003.
8. M. Zapata. Secure Ad Hoc on-demand distance vector (SAODV) routing. Technical report, http://www.ietf.org/internet-drafts/draft-guerrero-manet-saodv-00.txt, Internet Draft, 2001.
9. Y.-C. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for adhoc networks. In *Proc. of MobiCom*, September 2002.

Mutual-Authentication Mechanism for RFID Systems*

Batbold Toiruul, KyungOh Lee, HyunJun Lee, YoungHan Lee,
and YoonYoung Park

Computer Science Department, Sunmoon University,
#100 Tangjeong-myun ChungNam, Asansi 336-708, Korea
tulaanaa@yahoo.com, leeko@sunmoon.ac.kr, hjlee@sunmoon.ac.kr,
hans0209@gmail.com, yypark@sunmoon.ac.kr

Abstract. The biggest challenge for current RFID technology is to provide the necessary benefits while avoiding any threats to the privacy of its users. Although many solutions to this problem have been proposed, almost as soon as they have been introduced, methods have been found to circumvent system security and make the user vulnerable. We are proposing an advanced mutual-authentication protocol between a tag and the back-end database server for a RFID system to ensure system security integrity. The three main areas of security violations in RFID systems are forgery of the tags, unwanted tracking of the tags, and unauthorized access to a tag's memory. Our proposed system protects against these three areas of security violations. Our protocol provides reader authentication to a tag, exhibits forgery resistance against a simple copy, and prevents the counterfeiting of RFID tags. Our advanced mutual-authentication protocol uses an AES algorithm as its cryptograph primitive. Since our AES algorithm has a relatively low cost, is fast, and only requires simple hardware, our proposed approach is feasible for use in RFID systems. In addition, the relatively low computational cost of our proposed algorithm compared to those currently used to implement similar levels of system security makes our proposed system especially suitable for RFID systems that have a large number of tags.

Keywords: RFID, AES, cryptograph.

1 Introduction

Radio-frequency identification (RFID) is an emerging technology. It is the next generation of an optical barcode with several major advantages over an optical barcode since a line-of-sight between the reader and the barcode is not needed, and several tags can be read simultaneously. RFID technology is rapidly finding more diversified applications in today's marketplace. For example, RFID technology is now being used for automatic tariff payment in public transport, animal identification

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment) (IITA-2006-C1090-0603-0020).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 449–460, 2006.

and tracking, automated manufacturing, and logistical control for automatic object identification since every object can be identified by a unique identification tag number. A RFID system consists of three parts: the radio-frequency (RF) tags, the RF readers, and the back-end database server. The back-end server associates records with the tag data collected by the readers. Tags are typically composed of a microchip for storage and performing logical operations and a coupling element such as an antenna coil for wireless communications. Memory chips on the tags can be read-only, write-once/read-many, or fully writable. Each memory chip holds a unique ID and other pertinent information transmitted to the tag reader using a RF. The tag readers interrogate the tags using a RF antenna and interact with the back-end database for more functionality.

However, RFID tags may pose a considerable security and privacy risk to organizations and individuals using them. Since a typical tag answers its ID to any reader and the replied ID is always the same, an attacker can easily copy the system by reading out the data of a tag and duplicating it to bogus tags. Unprotected tags may have vulnerabilities to eavesdropping, location privacy, spoofing, or denial of service (DoS). Unauthorized readers may compromise privacy by accessing tags without adequate access control. Even when the content of the tags is protected, individuals may be tracked through predictable tag responses. Even though many cryptographic primitives can be used to remove these vulnerabilities, they cannot be applied to a RFID system due to the prohibitive cost of including protection for each and every RFID tag. The RFID tag is the most costly item in a RFID system as such systems inherently use at least a minimum of several tags. Economic constraints usually dictate that the tags cost as little as possible and that as few as possible are used. Power consumption, processing time, storage, and gate count are all severely limited. For example, a practical tag costing in the order of $0.05 US may be limited to having only hundreds of bits of storage and roughly 500–5,000 gates in order to meet cost restraints.

In this paper, we propose a new mutual authentication protocol that uses AES (Advanced Encryption Standard) for the security of a RFID system. In November 2001, NIST announced that the AES algorithm, based on the Rijndael algorithm, was the new encryption standard [11], [12]. We chose the AES as our cryptographic primitive because it is standardized and considered to be secure. The AES algorithm consists of one s-box, two other kinds of transformations, and a key schedule. It supports key lengths of 128, 192, and 256 bits, and many hardware implementations of the AES algorithm exist. However, several papers have presented a low-power implementation of the AES suitable for use in RFID tags in terms of power consumption and die size [10], [13], [14].

Our proposed mutual-authentication protocol can be used to solve the inherent security problems of RFID systems. Our protocol allows high-value goods to be protected against adversarial attackers. Also, our protocols can easily meet current data rate restrictions and are compliant with existing standards as well as requirements concerning chip area and power consumption. With mutual authentication we can provide a proof for each entity of a RFID system based on an AES encryption. Therefore, our proposed protocol is sufficiently robust to withstand active attacks such as the man-in-the-middle attack, the replay attack, the eavesdropping attack, and the unwanted tracking of customers.

2 Related Works

RFID security and privacy issues have been an active and continuing area of research. We describe some of the related studies below.

Hash lock scheme, developed by MIT [6]. In this scheme, each tag verifies the reader as follows. The reader has a key (k) for each tag, and each tag holds the result metaID, where metaID = hash (k) of a hash function. A tag receives a request for ID access and sends a metaID in response. The reader sends a key that is related to the metaID received from the tag. The tag then calculates the hash function from the received key and checks whether the result of the hash function corresponds to the metaID held in the tag. Only if both data sets agree does the tag send its own ID to the reader. However, in this scheme, the adversary can track the tag via the metaID. Furthermore, both the random key and the tag ID is subject to eavesdropping by an attacker.

Randomized hash lock scheme, developed by MIT [6]. This is an extension of the hash lock scheme, and requires the tag to have a hash function and a pseudorandom generator. Each tag calculates the hash function based on the input from a pseudorandom generated r and id, i.e., c = hash (id, r). The tag then sends c and r to the reader. The reader sends the data to the back-end database. The back-end database calculates the hash function using the input as the received r and id for each ID stored in the back-end database. The back-end database then identifies the id that is related to the received c and sends the id to the reader. The tag output changes with each access, so this scheme deters tracking. However, the attacker can impersonate the tag to a legitimate reader. Also the attacker can know the r and ID_k because eavesdropping is possible.

A RFID security approach for a supply chain system, developed by the IBM China Research Lab [2]. This approach requires read-access control. When a tag receives an inquiry from a reader, the tag will first create a random number k, which it then transmits. After the random number k is received by the reader, the reader sends k back to the backend database. The backend database hashes (ReaderID || k) and sends out the hash value to the reader. The reader then sends it to the tag. In the meantime, the tag also hashes (ReaderID || k). Then the tag compares the hash value calculated by the tag to that by the reader. If they are equal, the reader passes the authentication and the tag can then provide tag ID-related information. However, in this approach, an attacker can eavesdrop on the Reader ID as no security is required for the tag to get the reader ID. Therefore, an attacker can impersonate a reader to a tag.

Cryptographic approach to "privacy-friendly" tags, developed by the NTT lab [8]. The basic idea of Ohkubo et al. is to modify the identifier of the tag each time it is queried by a reader so that the identifiers can be recognized only by authorized parties. The tag refreshes its identifier autonomously using two hash functions, G and H, as described below. Readers are (untrusted) devices that do not have cryptographic functionalities but a hash function can be embedded into the tags. Soon, this may well be a realistic assumption. Ohkubo et al.'s scheme has a complexity of mn hash computations in a closed environment (2 hash operations are carried out mn/2 times), and of 2 mn in an open environment since the database computes all of the hash chains when trying to identify a foreign tag. Thus, when the number of tags, n, or the number of read operations, m, is large, the complexity becomes unmanageable so this scheme is not scalable.

Strong authentication for RFID systems using the AES algorithm, developed by project ART [1]. The main theme of this paper is the assumption that an AES is feasible for current RFID technology without major additional costs. The ART project team selected AES as a cryptographic primitive for symmetric authentication. They analyzed several architectural possibilities for implementing AES-128 encryption functionality. The implementation of the data path of an AES-128 encryption design has a current consumption of 8.15 μA on a 0.35-μm CMOS process. It operates at a frequency of 100 kHz and needs 1,016 clock cycles for encrypting a 128-bit data block. The required hardware complexity is estimated to be 3,595 gate equivalents (GEs).

This report uses unilateral authentication, which works as follows. There are two partners, *A* and *B*. Both possess the same private key, *K*. *B* sends a random number, ***r*,** to *A*. *A* encrypts the random number $E_k(r)$ with the shared key *K* and sends it back to *B*. *B* proofs the result and can verify the identity (in other words, the possession of *K*) of *A*.

In this case, the man-in-the-middle-attack is possible. The attacker sends a random number to a tag. Then the tag replies with the encrypted value of *r* to the attacker. Therefore, it is possible for the attacker to obtain the shared *k* value from many combinations of *r* and $E_k(r)$. Then the attacker can impersonate a legitimate reader to the tag or a legitimate tag to the reader. Therefore, we need mutual authentication.

3 Our Proposed Approach to RFID Security

3.1 Notations

We use the notations summarized in Table 1 to describe our protocol throughout the remainder of this paper.

Table 1. Notations

T	RF tag, or transponder
R	RF tag reader, or transceiver
B	Back-end server, which has a database
k_1, k_2	Random secret keys, shared between T and B
K	Cryptographic key, shared between T and B
ID_k	Unique identification number of T, shared between T and B
$E_k(k_1 \oplus k_2)$	AES cipher text, using k_1, k_2, and k
$E_k(k_1 \oplus k_2 \oplus ID_k)$	AES cipher text, using k_1, k_2, k, and ID_k
$E_k(k_1, k_2)$	The notated $E_k(k_1 \oplus k_2)$
$E_k(ID)$	The notated $E_k(k_1 \oplus k_2 \oplus ID_k)$

3.2 Assumptions and Attacking Model

In our protocol, we assume that *T* has AES encryption cryptographic hardware. In [16], since an AES encryption and decryption unit with a block size of 128 bits can be implemented with only about 3.4 K-gates, our protocol only requires a small gate size. Also, we assume that *T* only has its authentication-related information, ID_k, Also, *T* has a memory for keeping values of ID_k, k_1, and k_2 to process mutual authentication. We assumed that the communication channel between *R* and *B* was secure.

To solve the security risks and privacy issues, the following attacking model must be prevented [3]–[6]. However, in our protocol, we have not considered a physical attack such as removing a RFID tag physically from a product because it is hard to carry out in public view or on a wide scale without detection. We consider the following attacks.

***Man-in-the-middle attack*:** The attacker can impersonate a legitimate reader and get the information from *T,* so he/she can then impersonate a legitimate *T* responding to *R*. Thus, a legitimate *R* can easily be fooled into authenticating an attacker before the next session.

***Replay attack*:** The attacker can eavesdrop on the response message from *T,* and retransmit the message to the legitimate *R*.

***Forgery of tags*:** A simple copy of *T's* information can be obtained through eavesdropping by an attacker.

***Unwanted tracking of customers*:** It is possible to track people's movements, social interactions, and financial transactions by correlating data from multiple tag reader locations.

3.3 Security Requirement

To protect user privacy, we consider the following requirement from a cryptographic point of view [7], [8].

Data confidentiality: T's private information must be kept secure to guarantee user privacy, and T's information must be meaningless to any unauthorized users even though it can be easily obtained through eavesdropping by an attacker.

Tag anonymity: Although T's data are encrypted, T's unique identification information can be exposed since the encrypted data are constant. An attacker can identify each T by using its permanent encrypted data. Therefore, it is important to make the information on T anonymous.

Data Integrity: If the memory of T is rewritable, forgery and data modification will occur. Thus, the linkage between the authentication information and T itself must be given in order to prevent a simple copy of T. However, data loss will result from a DoS attack, power interruption, message hijacking, etc. Thus, authentication information between T and B must be delivered without any failure, and data recovery must be provided.

In addition, we had to consider and evaluate the following security feature in the design of our RFID authentication protocol.

Mutual authentication and reader authentication: In addition to access control, the mutual authentication between T and B must be provided as a measure of trust. By authenticating mutually, the replay attack and the man-in-the-middle attack to both T and B is prevented.

3.4 Protocol Design

Our overall protocol is shown in Fig. 1. The detailed procedures for each step are described in the following:

3.4.1 Initial Setup

Each T is given two fresh random secrets, k_1 and k_2, and a unique identification, ID_k. The database (D) of B also stores them as the shared secret. In addition, D manages a record pair for each tag consisting of (ID_k, TagID). T has an AES-128 encryption circuit. If a reader requires a tag's ID, the tag must first authenticate the reader. After authentication, the reader can obtain the tag ID by the tag's response and reference to the database. In addition, both T and B have a cipher key, k, that is a 128-bit key.

3.4.2 Detailed Description

In the following, we describe our proposed protocol according to the sequence of message exchange. Also, we discuss the security goals that are achieved during the execution of each protocol message.

Step 1 (Challenging): In this step, reader *R* usually applies a collision protocol such as secure binary tree walking [4], an interleaved protocol [3], or the standard protocol of ISO 18000-3 MODE [7] to singularize *T* out of many. The Reader, *R*, receives $E_k(k_1,k_2)$ from the back-end server, *B*. Then *R* sends $E_k(k_1,k_2)$ to the queried *T*. The cipher key *k* and random numbers k_1 and k_2 are shared by *B* and *T*. Therefore, $E_k(k_1,k_2)$ is used to authenticate the validity of *R*.

Step 2 (Authentication of R): When queried, *T* generates $E^*_k(k_1,k_2)$ and verifies whether the received $E_k(k_1,k_2)$ is valid by comparing $E_k(k_1,k_2)$ with $E^*_k(k_1,k_2)$. If $E_k(k_1,k_2)==E^*_k(k_1,k_2)$, *T* authenticates *R*. Then *T* generates $E_k(k_1 \oplus k_2 \oplus ID_k)$, designated as $E_k(ID)$, which is the encryption of the AES-128 cryptographic algorithm. *T* uses this as the identification information and sends it to *R*.

Otherwise, *R* is not authenticated and *T* will keep silent. Therefore, being tracked by an attacker is not possible when no authorized readers are nearby. Cipher key *k* and random numbers k_1 and k_2 are shared only between *T* and *R*. Therefore, *T* can detect an illegal *R* and discard the message. Consequently, the man-in-the-middle attack by an illegitimate *R* and a passive eavesdropper can be prevented.

If *T* has successfully authenticated *R*, *T* updates the shared secrets keys, k_2 and k_1 by exclusive-ors with $E_k(k_1 \oplus k_2)$.

Step 3 (Authentication of T): *R* simply forwards $E_k(ID)$ to *B*. Within this step, *B* authenticates *T* with $E_k(ID)$. At first, *B* decrypts $E_k(ID)$ using cipher key *k* and random numbers k_1 and k_2 and obtains ID_k. Then *B* verifies whether ID_k is valid by comparing the obtained ID_k with ID^*_k. Random secrets, k_1 and k_2, and the cipher key, *k,* are shared only between *B* and *T*. Therefore, *B* can detect an illegal *T* and discards the message. Therefore, the man-in-the-middle attack by an illegitimate *T* and a passive eavesdropper can be prevented. If *T* is authenticated, *B* retrieves the records corresponding to ID_k and gets the real *TagID*.

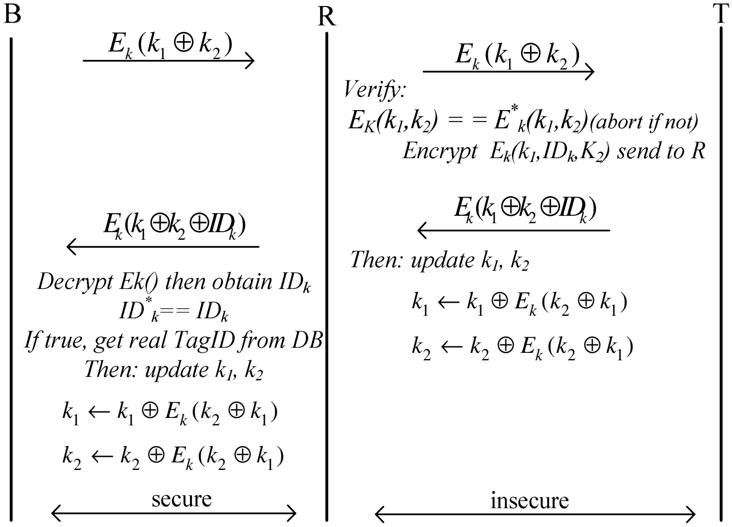

B
R
T
$E_k(k_1 \oplus k_2)$
$E_k(k_1 \oplus k_2)$
Verify:
$E_K(k_1,k_2) == E^*_k(k_1,k_2)$(abort if not)
Encrypt $E_k(k_1,ID_k,K_2)$ send to R
$E_k(k_1 \oplus k_2 \oplus ID_k)$
$E_k(k_1 \oplus k_2 \oplus ID_k)$
Then: update k_1, k_2
$k_1 \leftarrow k_1 \oplus E_k(k_2 \oplus k_1)$
$k_2 \leftarrow k_2 \oplus E_k(k_2 \oplus k_1)$
Decrypt Ek() then obtain ID_k
$ID^*_k == ID_k$
If true, get real TagID from DB
Then: update k_1, k_2
$k_1 \leftarrow k_1 \oplus E_k(k_2 \oplus k_1)$
$k_2 \leftarrow k_2 \oplus E_k(k_2 \oplus k_1)$
secure
insecure

Fig. 1. The Proposed Mutual-Authentication Protocol

Even if E_K *(ID)* is discovered through eavesdropping, the eavesdropper cannot know the ID_k value, since he/she does not know k_1, k_2, and the cipher key *k*. Since *B* initially stores the unique identification, ID_k, *B* can evaluate the linkage between $E_k(ID)$ and *T* itself in order to prevent forgery. Forgery can be detected and prevented by *B* at this time.

At the same time, *B* can detect and prevent the man-in-the-middle attack since ID_k is used as the factor of the man-in-the-middle attack detection. Similarly, the replay attack can also be detected and prevented simultaneously.

If *B* successfully finishes the authentication process, *B* generates $E_k(k_1 \oplus k_2)$ with its shared random secrets, k_1 and k_2. The database of *B* updates the shared secrets keys, k_1 and k_2, by exclusive-ors with $E_k(k_1 \oplus k_2)$. Then, mutual authentication has finally succeeded.

4 Analysis

4.1 Security Analysis

We have evaluated our protocol from a security requirement standpoint. Our protocol guarantees a secure mutual authentication only with AES-128 encryption messages, $E_k(k_1 \oplus k_2)$, $E_k(k_1 \oplus k_2 \oplus ID_k)$, and ID_k, *T* does not store user privacy information.

Thus, data confidentiality of tag owners is guaranteed and the user privacy on data is strongly protected. In every session, we use a fresh random nonce as the keys between entities. These keys are randomized and anonymous since they are updated for every read attempt. Thus, tag anonymity is guaranteed and the location privacy of a tag owner is also not compromised. Based on mutual authentication, our protocol guarantees the data integrity between *T* and *B*. The forgery-resistance feature was realized by exclusive-oring the unique authentication number, ID_k, of *T* with the authentication information. ID_k is originally stored during the initial step. Whenever *T* generates $E_k(ID)$, it refers to ID_k, so the linkage between ID_k and *T* itself can be determined. *B* keeps each tag's ID_k initially and authenticates the ownership of the authentication information for *T*. Table 2 shows the comparison of the security requirements and the possible attacks.

The man-in-the-middle attack. Through the authentication steps 1 and 2, *R* sends $E_k(k_1 \oplus k_2)$ to *T* and *T* sends $E_k(k_1 \oplus k_2 \oplus ID_k)$ to *B* for preventing the man-in-the-middle attack. *B* can verify ID_k with the decryption of the AES-128 cryptographic value of $E_k(ID)$ transmitted from *T*. The key freshness is also guaranteed for each session. The replay attack for *T* and *B* is detected and prohibited in step 3 for *B* and in step 2 for *T*.

Table 2. Comparison of the secure requirements

Protocol	HLS [6]	RHLS [6]	Ref. [2]	Ref. [8]	Ref. [1]	Our scheme
User data confidentially	x	O	O	O	O	O
Tag anonymity	x	O	O	O	O	O
Mutual authentication	Δ	Δ	Δ	Δ	Δ	O
Reader authentication	x	x	x	x	x	O
Man-in-the-middle attack prevention	O	O	x	O	x	O
Replay attack prevention	Δ	O	x	Δ	O	O
Forgery Resistance	x	x	O	O	O	O
Tracking	x	x	O	x	x	O

Notation: x – not satisfied; O – satisfied; Δ - partially satisfied

Invulnerable to eavesdropping. In the process of authentication, even when an attacker eavesdrops on the output of tag, $E_k(k_1,k_2)$, it can not pretend to be an authorized reader in the next authentication session since the random secrets, k_1 and k_2, are changed in every session. Also, the required $E_k(k_1,k_2\ )$ value is an AES algorithm cipher, and random secrets, k_1 and k_2, and cipher key *k* are shared only between *T* and *B*. Since an AES-128 encryption is extremely difficult to inverse, the tag ID_k and random secrets, k_1 and k_2, are protected even if the output is captured by an attacker. Therefore, it is invulnerable to eavesdropping. In one word, our proposed approach is secure when any communication between readers and tags are subjected to eavesdropping.

Prevent being tracked by adversary. Tags keep silent to attackers. They only respond to authenticated readers. Furthermore, as explained above, it is impossible for attackers to pretend to be an authenticated reader. Since no tag output occurs, attackers are unable to track customers by the tag value that existed as they checked out. The privacy of location and the secrecy of what objects that the customers are carrying is protected.

4.2 Performance Analysis

We analyzed the performance of our proposed scheme with respect to computation and its anti-collision mechanism.

Low computation load. When identifying a tag from N known tags, a reader performs only two AES operations, while for other approaches of randomized access control, at least N hash operations and N searches [2] are required. In addition, the AES tag's hardware has a relatively low cost and fast computation time [10].

Since the computation load remains low even with an increasing number of tags, our proposed approach is suitable for protecting RFID systems with a large number of tags. This feature is very important for a supply chain. Each part along a supply chain deploys numerous tags. In warehouses or retail stores, thousands of products need to be tagged to accelerate the supply chain process. Therefore, a secure RFID scheme that is suitable for a large number of tags is a definite prerequisite for the implementation of a RFID supply chain system.

Anti-collision mechanism. The most important command is the anti-collision sequence, which is a command every tag must implement. Therefore, a reader sends an initial inventory command. All tags in the environment make a response that is the tag's unique ID. If only one tag answers the request, the ID can be retrieved by the reader and all subsequent commands can be addressed using the ID that addresses one single tag. If two or more tags answer a request, a collision occurs. This can be detected at the reader. The reader then uses a modified inventory request in which it adds a part of the tag's ID to the request. Only tags that have this part of the ID are allowed to answer. Once the ID of one tag is identified, the reader sends a "stay quiet" command to the tag with the identified ID. This method is used as long as no more collisions occur and all tags within the environment are identified. In our proposed approach, we have suggested two anti-collision mechanisms, namely the interleaved protocol [3] and the binary-three algorithm [4].

5 RFID Tag Architecture

The RFID tag consists of the analog front-end; the controller for implementing software requirements such as data coding, implementation of the protocol commands, anti-collision mechanisms, and error detection; the EEPROM that stores k_1, k_2, ID_k, and k; the key for cryptographic algorithms; and the AES hardware module.

We selected AES as the cryptographic primitive for our proposed approach. One important criterion for selecting the AES algorithm was its structure allowing efficient implementation in hardware. In addition, several previous implementations of the AES have proven it to be low-cost and relatively fast. The tag cost can be around \$0.05 US and the die size is less than 0.25 mm^2. Power consumption is about 10 µA [3], [10], [13].

Most hardware implementations of the AES algorithm have focused on realizing a high data throughput. Recently, however, some attention has been given to hardware implementations that were designed with hardware efficiency in mind. Hardware efficiency can be increased by lower die sizes and reduced power consumption. Some recent papers have been published that focus on this issue [10], [13]–[15].

Mangard et al. [14] presented a highly regular approach. It is comparable to RFID requirements but requires a chip area of 8,500 gate equivalents while having a higher data throughput of 70 Mbps. The AES hardware of Satoh et al. [13] is a 32-bit architecture and has a hardware complexity of 5,400 gates and reaches a throughout of 311 Mbps.

Feldhofer et al. [10] presented a silicon implementation of the AES optimized for low die size that offers excellent power consumption characteristics. The AES core of the manufactured chip has an area of 0.25 mm^2 on a 0.35 mm CMOS technology, which is comparable in size to a grain of sand. In terms of circuit complexity, the size equals 3,400 gate equivalents, and the average power consumption can be lowered to <5 mW when operated at 100 kHz and 1.5 V. Feldhofer et al. [10] implemented the AES algorithm as an 8-bit architecture.

Our protocol only uses the encryption circuit of AES. Therefore, our protocol hardware requires less chip area and power consumption than previous implementations. It also has several advantages as follows.

- It is an 8-bit implementation of the AES architecture [1], [10], [17].
- We need only the encryption circuit of the MixColumns [10], [13], [15].
- The Rcon function is a constant value. It is implemented as two different constant values in the encryption and the decryption processes. Only circuitry to implement a constant value is required for the encryption process [13].
- By using RAM as detailed in [10], we do not need a ShiftRows transformation. The ShiftRows transformation can be implemented by an appropriate addressing of the RAM or we can use an 8-bit register as the ShiftRows [13].

6 Conclusions

This paper proposes an advanced mutual-authentication protocol for security and privacy protection in RFID systems using an AES algorithm as a cryptographic primitive. This protocol protects high-valued goods against attackers. With mutual authentication, we can provide a proof for each entity of a RFID system, and since this proof is based on an AES encryption, our proposed protocol is sufficiently robust to withstand active attacks such as the man-in-the-middle attack, the replay attack, the eavesdropping attack, and the unwanted tracking of customers. Also, our protocols can easily meet current data rate restrictions and are compliant with existing standards as well as requirements concerning chip area and power consumption. In addition to cipher k, our proposed protocol uses k_1 and k_2 for security. These secret random numbers, k_1 and k_2, are changed in every session, so the attacker can not obtain important data from a tag even if the tag's outputs have been eavesdropped.

All authentication messages are randomized. In addition, each tag has its own unique identification data, so user data privacy and location privacy are guaranteed.

Acknowledgments. This research was supported by the Ministry of Information and Communication (MIC), Korea, under the Information Technology Research Center (ITRC) support program supervised by the Institute of Information Technology Assessment (IITA-2006-C1090-0603-0020).

References

1. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. "Strong authentication for RFID systems using the AES algorithm." In *Conference of Cryptographic Hardware and Embedded Systems*, 2004. Proceedings, pp. 357–370. Springer 2004.
2. X. Gao, Z. Xiang, H. Wang, J. Shen, J. Huang, and S. Song. "An approach to security and privacy of RFID system for supply chain," *CEC-East*, pp. 164–168, IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004.
3. M. Aigner and M. Feldhofer. "Secure symmetric authentication for RFID tags." *Telecommunication and Mobile Computing*, March 2005.
4. R. Juels, L. Rivest, and M. Szydlo. "The blocker tag: selective blocking of RFID tags for consumer privacy." In V. Atluri, editor, *8th ACM Conference on Computer and Communications Security*, pp. 103–111. ACM Press, 2003.
5. D. Henrici and P. Muller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers." *PerSec'04 at IEEE PerCom*, pp. 149–153, March 2004.
6. S.A. Weis, S.E. Sarma, R.L. Rivest, and D.W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems." *First International Conference on Security in Pervasive Computing*, 2003.
7. S. Weis, "Security and privacy in radio-frequency identification devices." *Master's thesis*, MIT, 2003.
8. M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to "privacy-friendly" tags." In *RFID Privacy Workshop*, MIT, USA, 2003.
9. ISO/IEC JTC 1/SC 31/WG 4, "Information technology AIDC techniques—RFID for item management air interface, part 3: parameters for air interface communications at 13.56 MHz." *Version N681R*, April 2004.
10. M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES implementation on a grain of sand." *IEE Proceedings Information Security,* October 2005, Vol. 152, Issue 1, pp. 13–20.
11. J. Daemen and V. Rijmen, "The design of Rijndael." *AES—The Advanced Encryption Standard* (Springer–Verlag, Berlin, Heidelberg, New York, 2002)
12. National Institute of Standards and Technology (NIST). "FIPS-197: advanced encryption standard, November 2001." *http://www.itl.nist.gov/fipspubs/*, accessed 18 March, 2006.
13. A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization." In C. Boyd, editor, *Proc. 7th Int. Conf. on the Theory and Application of Cryptology and Information Security, Advances in Cryptology*, ASIACRYPT 2001, Gold Coast Australia, December 2001, LNCS 2248, pp. 239–254, Springer, 2001.

14. S. Mangard, M. Aigner, and S. Dominikus, “A highly regular and scalable AES hardware architecture.” *IEEE Trans. Comput.*, 2003, 52 (4), pp. 483–491.
15. J. Wolkerstorfer, “An ASIC implementation of the AESMixColumn operation.” *Proc. Austrochip 2001*, Vienna, October 2001, pp. 129–132.
16. N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer, “Efficient AES implementations on ASICs and FPGAs.” In H. Dobbertin, V. Rijmen,, and A. Sowa, editors, *Proc. Fourth Workshop on the Advanced Encryption Standard* ‘‘AES—state of the crypto analysis.’’ AES 2004, LNCS 3373, pp. 98–112, Springer, 2004.
17. J. Wolkerstorfer, E. Oswald, and M. Lamberger, “An ASIC implementation of the AES SBoxes Proc.” *The Cryptographer’s Track at the RSA Conf. Topics in Cryptology*, CT-RSA 2002 San Jose, CA, USA, February 2002, LNCS 2271, pp. 67–78, Springer, 2002.

Achieving Anonymity in Mobile Ad Hoc Networks Using Fuzzy Position Information*

Xiaoxin Wu[1], Jun Liu[2], Xiaoyan Hong[2], and Elisa Bertino[3]

[1] Intel Communication Technology Beijing Lab, Beijing, 100080, P.R. China
[2] Dept. of Computer Science, University of Alabama, Tuscaloosa, AL 35487, USA
[3] Dept. of Computer Science, Purdue University, West Lafayette, IN 47907, USA

Abstract. Traditionally the anonymity of an entity of interest can be achieved by hiding it among a group of other entities with similar characteristics, i.e., an anonymity set. In mobile ad hoc networks, generating and maintaining such an anonymity set for any ad hoc node are challenging because of the node mobility and consequently of the dynamic network topology. In this paper, we address the problem of the destination anonymity. We propose protocols that use fuzzy destination position to generate a geographic area called *anonymity zone (AZ)*. A packet for a destination is delivered to all the nodes in the AZ, which, consequently, make up the anonymity set. The size of the anonymity set may decrease because nodes are mobile, yet the corresponding management on anonymity set is simple. We design techniques to further improve node anonymity. We use extensive simulation to study the node anonymity and routing performance, and to determine the parameters that most impact the anonymity level that can be achieved by our protocol.

1 Introduction

Privacy is a major concern for today's network users. An important privacy requirement is represented by anonymity, which is becoming increasingly important in a large variety of application domains. At the same time, mobile ad hoc networks are envisioned as an effective solution for extending the last-hop network communications to any party at any time and anywhere. Therefore, communication privacy, especially anonymity for communicating parties in ad hoc networks, is highly desired. In this work we investigate the application scenario where an ad hoc node receives sensitive data from well-known servers. This receiver may not wish its identity to be revealed to the network; we refer to this requirement as *destination anonymity*.

Traditional anonymous communication protocols may not be directly applied to mobile ad hoc networks. MIX [12] and Onion routing [13] require that security associations among entities be set up and stably maintained, which is

* Work from Xiaoxin Wu was performed at Purdue University with a support from Information Infrastructure Protection (I3P) Fellowship. *Part of the work was also supported by University of Alabama RAC 2005 award.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 461–472, 2006.

very difficult in MANET because of the lack of fixed infrastructure and of its dynamic nature. Approaches based on broadcast [2] or multicast [3] are not applicable because the network has limited bandwidth. In addition, multicast in MANETs is itself a challenging research issue. The obstacles against achieving communication-end privacy, especially destination anonymity, also depend on the fact that in on-demand routing protocols, such as AODV [4] and DSR [5], a global flooding is required in the route discovery stage. The destination identity is carried in the request, therefore, it has to be revealed to the entire network. All nodes in the network may thus become aware of the communications being established.

A widely investigated class of routing protocols for ad hoc networks is based on geographic (i.e., positioning) routing algorithms [6], where node positions are used for routing. A commonly proposed positioning routing algorithm is the Greedy Perimeter Stateless Routing (GPSR) [7]. GPSR has a better potential to achieve communication privacy because of its local and stateless route discovery protocol. More importantly, the routing information required by GPSR is the node position, not the node ID. Therefore, the real identity of a node, e.g., a destination, can be hidden. Approaches [8][17] have been reported to deal with the challenges of achieving anonymity in MANET. Under the AO2P protocol [8] node positions are used as pseudonyms for node anonymity. An important pre-requisite for AO2P is a secure position service system. Designing such a system, especially in a distributed ad hoc environment, is not trivial. In [17], a proposed neighborhood authentication protocol allows neighboring nodes to authenticate each other without revealing their identities. However, the destination ID has to be revealed for on-demand route discovery. Therefore, only a conditional anonymity can be achieved for the destination, namely that a tracer knows which node is the destination, yet the tracer does not know where the destination is.

The goal of our paper is to explore the advantages of geographic assisted routing while at the same time to address the privacy problem connected with the use of the aforementioned sensitive position data. We propose an anonymous geographic routing algorithm that uses fuzzy destination positions. The notion of fuzzy position has been used in privacy-preserving location-based services [9] [10]; under such an approach, a mobile user intentionally provides inaccurate positions for services to protect its real positions. Here, we use a fuzzy position to prevent adversaries from discovering the real position of a node and a destination ID based on its position. A pseudo destination that has a position near that of the real destination is generated, toward which packets are sent. The successful delivery in such a routing algorithm relies on the broadcast nature of wireless communication, where a transmission can always be received by all the nodes within the transmission range of the sender. Therefore, if the real destination is located in a geographic area that is not far away from the pseudo destination, it will receive the packets. Such a geographic area, that we refer to as an *anonymity zone (AZ)*, is the key concept in our design. The destination anonymity is determined by the number of nodes that are located in the AZ, and the protocol is thus called *zone-based anonymous positioning routing (ZAP) protocol.*

ZAP is based on the same principle of the Crowds protocol [1], under which a receiver hides among a group of entities, referred to as anonymity set. The difference, however, is that in ZAP, the size of such anonymity set, is affected by many network conditions and varies with time. For example, the number of nodes located in an AZ depends on the size of the AZ and the node distribution. In addition, once the AZ is built, the size of the group will decrease because of the node mobility. On the other hand, if one allows a fixed anonymity set, e.g., a group consisting of some fixed ad hoc nodes, reaching every node in the anonymity set requires MANET multicast, which may result in anonymity breaches.

ZAP is a light-weight routing protocol which exploits geographical information of nodes in order to reduce overhead incurred by privacy purposes. Although other anonymous routing protocols like ANODR [14] can also achieve destination anonymity by VCI (Virtual Circuit Identifier) enabled anonymous routes, those protocols incur heavier communicational overhead when performing on-demand anonymous route discovery. Such a discovery floods larger route request messages and has initial buffering time. ZAP, instead, emphasizes the destination anonymity. Considering position information is sensitive data in many applications, ZAP achieves destination anonymity by using fuzzy positions. After all, ZAP strives to balance over tolerable losses in privacy and a simpler protocol and network management with increased efficiency.

The paper is organized to first introduce the basic zone-based anonymous positioning routing and then an enhanced scheme in Section 2. Section 2 also discusses the anonymity, the attacks, and the mitigating techniques. Section 3 reports evaluation results from simulations. And Section 4 concludes the paper.

2 Zone-Based Anonymous Positioning Routing Protocol

2.1 Assumptions

With respect to the network, we assume that nodes are uniformly distributed with a node density not too low. A node moves toward a random direction at a variable speed. The wireless channel is bi-directional. Each node knows its own position, e.g., through a GPS system. Nodes exchange their positions locally through "hello" messages.

With respect to privacy, we assume that each node has a public key that is known to all the other nodes. The public key is assigned by a certificate authority before a node joins the network. For data delivery, the identity of the destination is not revealed to the network. Each node has an equal probability to be a receiver (client).

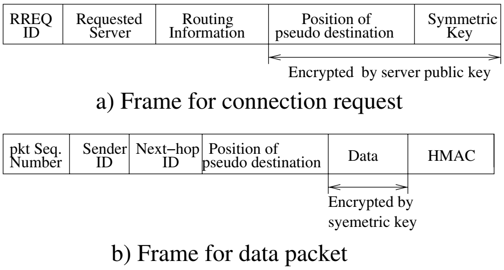
The attacker models that we consider in our work are as follows. There are internal attackers that trace or monitor the behavior of other nodes for malicious purposes. These attackers follow the protocols. They do not act aggressively (that is, do not interrupt the correct network functioning) to obtain additional information because they would like to stay in the network without being noticed. An attacker is able to eavesdrop the communication channel. It can collect position information of its neighbors by intercepting hello messages. An attacker or colluding attackers therefore can discover the *local* network topology. Finally, if a transmission lasts long enough, attackers can locate the transmitter, e.g., through directional antenna techniques, and identify it by moving to the transmitter.

2.2 ZAP with Pseudo Destination (PD-ZAP): A Basic Approach

ZAP preserves destination anonymity through anonymity zones, under which a destination is located with a number of other nodes. The protocol operates in the following steps.

A client (*destination*) sends a server (*source*) a connection request for data downloading. The request indicates parameters for setting up a private route, which includes the fuzzy position information (i.e., the pseudo destination) and, if necessary, the range of the anonymity zone. The connection request can be sent by traditional routing algorithms or flooding. To assure data confidentiality and integrity, the destination can generate a symmetric key and carry it in the connection request. Concerning the destination anonymity of this request message, our claim is that the probability of intercepting a sporadic request at its initiating location by an attacker is very small. In addition, the identity of the request originator is not carried in the message.

The message frames for connection requests and data packets can be structured as shown in Fig. 1. The routing information in a connection request is determined by the routing algorithm to be used for sending the request.


RREQ ID
Requested Server
Routing Information
Position of pseudo destination
Symmetric Key
Encrypted by server public key

a) Frame for connection request

pkt Seq. Number
Sender ID
Next–hop ID
Position of pseudo destination
Data
HMAC
Encrypted by syemetric key

b) Frame for data packet

Fig. 1. Data frame for the packet

The source retrieves the AZ information after receiving the connection request. It then initiates the greedy geographic forwarding to deliver data packets. Any forwarder (including the source) forwards the data packet to a neighboring node that is closest to the pseudo destination, which also is the geographic center of the AZ. Once a data packet reaches the AZ, a node in the AZ that first receives the packet becomes a *proxy*. The proxy then uses different local packet distribution mechanisms to deliver the packet to the destination, according to the size of AZ. The source uses the symmetric key to encrypt data, and uses HMAC [11] for data integrity. As data packets are delivered toward the AZ, not the real destination, such an approach is called ZAP with pseudo destination, or PD-ZAP.

PD-ZAP is illustrated in Fig. 2. The position of the pseudo destination is randomly selected, and is not too far from that of the real destination. This position is also the routing information carried in each data packet. Therefore, the connection request does not have to carry the real identity of the destination, as it is not required for routing. This guarantees the destination anonymity even if the source is compromised.

In PD-ZAP, a packet will finally be received by a node that is closest to the pseudo destination [1]. This node then acts as a *proxy* and broadcasts the received packet to all of its neighbors. In this paper, a broadcast is defined as the process that a node transmits a message to all of its neighboring nodes that are within its radio coverage. In Fig. 2, the solid circular represents the transmission range of the proxy, which has a radius of r. r is the maximum ad hoc channel coverage. If the real destination is within the proxy's radio coverage, it will receive the data packet. If an ACK is required, the proxy sends its neighbor list back to the source. The neighbor list has been obtained by exchanging "hello" messages.

A new session has to be started if the destination can no longer receive data packets, typically when the destination has moved away from the AZ. In this case the destination has to send a new connection request along with the updated AZ information, based on which the source initiates another private route.

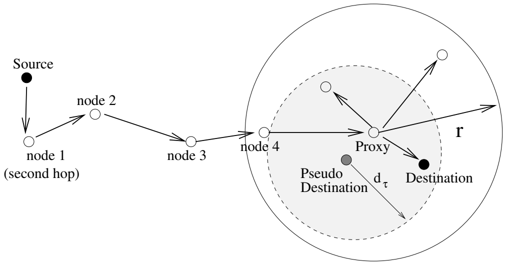


Fig. 2. The PD-ZAP approach

The generation of the pseudo destination is the key part of the algorithm. The maximum distance (or the distance threshold value) between the pseudo destination and the real destination, denoted as d_τ, determines both the node anonymity and the success of a packet delivery. The distance cannot be too long, otherwise the real destination may not receive the data packet from the proxy. It cannot be too short either, because a short distance results in a small anonymity set. As shown in Fig. 2, the destination anonymity zone (D-AZ) in PD-ZAP is the shaded circular area that is centered at the pseudo destination and has a radius of d_τ. For the attackers, only a node located in that area can be the destination. The pseudo destination selection depends on node density and node mobility.

In PD-ZAP, the position of the pseudo destination is also used as the session ID, according to which a node receiving the packet from the proxy knows whether it is the destination. Only the destination will be able to decrypt the packet using the established symmetric key. The other nodes simply drop the packet. However, since upon different packet arrivals, the node that is closest to the pseudo destination may be different, proxies can be different for the same session.

[1] It is not necessary that a node is located at that position.

2.3 Anonymity, Weaknesses, and Mitigation Techniques

In this subsection we discuss the protocol anonymity at high level. We determine possible privacy attacks, and propose mechanisms as counter measures.

Anonymity. Our anonymity goal is to hide a destination among a number of ad hoc nodes. In ZAP, the destination anonymity depends on the size of the group formed by the nodes that are located in the D-AZ. The determining factors are node distribution, size of D-AZ, and node mobility.

Intercepting a connection request may not help attackers too much in identifying the destination. As the identity of the request originator is not carried in the message, upon intercepting a request, the attacker cannot even tell whether the node from which it intercepted the message is the originator or just a forwarder. Even if an attacker knows that such node is the originator, the transmission happens too soon so that it is difficult for the attacker to locate the originator and thereafter to identify it. For the same reason, when a destination sends an ACK back using an alternative private route and the ACK is intercepted by attackers, the identity of the destination will not be discovered.

Because a node has to locally disclose its position, an attacker can stay close to its target node and monitor its behavior. Under such a target-oriented attack, communication privacy cannot be preserved by just using ZAP protocol. To mitigate such an attack, background noise is needed. A node can occasionally send out dummy packets that have the same pattern as requests and ACKs. In this case, when a real request or ACK is sent, the attacker cannot be certain.

The goal of the destination anonymity achieved by PD-ZAP can be achieved using earlier work on untraceable and anonymous routing presented in ANODR [14]. In both PD-ZAP and ANODR, the real destination hides within the radio range of the last hop towards the destination. The two protocols also share a similar way of establishing a credential between the source and the destination. The global trap door used by ANODR can be implemented the same way as in PD-ZAP given the available public keys. The difference relies on the location information. Geographic position (could be pseudo) allows a destination node (client) in PD-ZAP to send an initial request to the source (server) directly through geo-forwarding, and in return, allows the server to geo-forwarding data packets to the destination. While in ANODR, without location information, a destination has to flood its initial request and then use the signaling route discovery to establish a route. In addition, the ZAP protocols can be used in applications that favor geographic information assisted routing. However using geo-forwarding, an internal passive attacker can learn the approximate area of the destination from any hop. When that happens, destination anonymity relies on the protection within the zone. In order to increase the anonymity zone and to defend attacks against destination anonymity, we present a ZAP variant (RR-ZAP, in Section 2.4) that expands the anonymity zone towards an area other than the receiving range of the last hop.

Intersection Attack: The Impact of Node Mobility on Anonymity. Node mobility has important impact on the anonymity. To analyze such impact, the notion of *intersection attack* has to be introduced.

An intersection attack occurs when an attacker knows its Entity of Interest (EOI) is in more than one anonymity set. In this case, it concludes that the EOI must be in the intersected set among all these anonymity sets. As the intersected set is smaller than any of the original set, the anonymity level for the EOI decreases.

Node mobility helps attackers to conduct intersection attacks and therefore to degrade node anonymity. This is especially the case when the communication between the source and the destination lasts for a long time. Fig. 3 shows an example. Suppose that two packets arrive at the D-AZ at times t_1 and t_2, respectively. At time t_1, a set_1 of nodes is located in the D-AZ and at time t_2, a set_2 of nodes is located in the D-AZ. The sets set_1 and set_2 are not equal because some nodes may have moved out or into the D-AZ between the two transmissions. To an attacker, the anonymity set for the destination includes only the nodes that are in the D-AZ at both t_1 and t_2, that is, the intersection of the anonymity sets at the times t_1 and t_2. In this example, it is easy for the attacker to infer that the destination node is either e or f. The size of the anonymity set is reduced to 2, instead of 6 for set_1 or 5 for set_2.

If a session lasts long, the number of nodes remaining in the anonymity zone can be small. The destination anonymity thus can be very low. Note that the nodes that are originally out of an AZ move in the AZ during the communication do not contribute to anonymity, because the attacker knows these nodes cannot be the destination anyway.

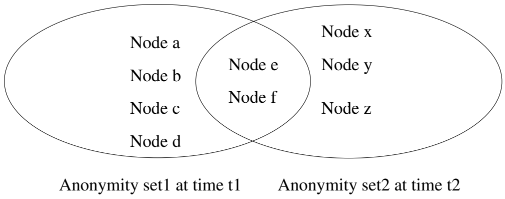


Fig. 3. Example of intersection attack

Mitigating Techniques Against Intersection Attack. Different approaches can be adopted to mitigate the impact of node mobility and to reduce the anonymity degradation. One approach is to divide a long-duration session into a number of short subsessions that use different D-AZs. For each subsession, a D-AZ and the corresponding symmetric key are generated. As a subsession does not last a long time, the destination anonymity may only decrease moderately because of mobility. The challenge is how to make these subsessions un-linkable. A straightforward solution is to increase the inter-subsession duration, which improves anonymity at the cost of the communication delay.

Tradeoff Between Privacy and Network Performance. ZAP achieves privacy at the cost of network performance. The inaccurate routing information in PD-ZAP results in a decreased data delivery ratio in MANETs. In approaches that mitigate intersection attacks resulting from node mobility, additional signaling and increased redundant transmissions are required. In general, a better performance implies a sustained communication duration that is long enough to complete a session. A longer communication, on the other hand, may decrease node anonymity because it gives a tracer more opportunities to conduct an intersection attack. In a later section, we present an analysis on the flooding overhead with respect to the initial D-AZ size. An extensive analysis on the mutual impact between network performance and privacy will be carried out as part of our future work.

2.4 ZAP with Route Redundancy: Advanced Approach

As PD-ZAP has a relatively small anonymity set, we propose to use a route with redundant hops to increase the D-AZ; we call such an approach ZAP with route redundancy (RR-ZAP) (refer to Fig. 4.). RR-ZAP can be used in a network where the position of servers (that is, sources) are well known. Like PD-ZAP, in RR-ZAP, a client (destination) creates a pseudo destination, denoted by P in the figure, for building a private route. Unlike PD-ZAP, in RR-ZAP, P is not close to the real destination, but can be a few hops away. P is selected so that the real destination is close to the direct connection between the source and the pseudo destination, which is line SP in the figure. If the network nodal density is not too low, the routing path may not deviate too far away from SP. The real destination then is close to the path, and can *intercept* the data delivered to the pseudo destination. In Fig. 4, the real destination can receive the packet, probably, from node 3.

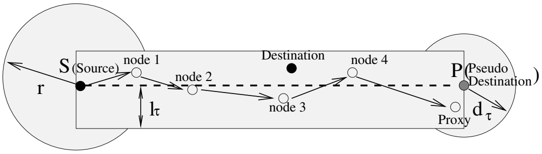


Fig. 4. The RR-ZAP approach

The distance between the real destination and SP should not be higher than a threshold value l_τ. l_τ determines the anonymity set and successful delivery ratio. It depends on node density and distribution. To an attacker, as the destination can be any node that is no more than l_τ away from the SP, the anonymity zone for the destination then includes the shaded rectangular area in the figure. Other than that, the real destination can also be located at the circular shaded areas at the two ends of the path, which are respectively the coverage of the source and the anonymity zone for PD-ZAP.

When an immediate acknowledgment from the real destination to the source is required, all the nodes in the path will collect ACKs from their neighbors and send back the lists to the source. The source then knows whether the real destination has received the packet.

3 Simulation Study

We further evaluate the destination anonymity and the network performance of the proposed protocols through simulation. The evaluation metrics include: *(i) the size of the anonymity set*: the number of nodes that remain in the anonymous zone when a session ends compared to those at the beginning of the session; *(ii) packet delivery ratio*: the ratio between the number of data packets received and those originated by the sources; *(iii) normalized packet forwarding overhead*: the number of packets transmitted by ZAPs normalized to those transmitted by GPSR under the same condition. *(vi) average end-to-end packet latency*: the average time from when the source generates the data packet to when the destination receives it.

We evaluate protocols PD-ZAP and RR-ZAP. For RR-ZAP, the simulation area limits the number of hops we can choose for redundancy. Thus in our implementation, a pseudo destination is positioned at the intersection of the boundary and it is chosen to ensure that l_τ equals to the half of the transmission range. We present GPSR for reference when appropriate.

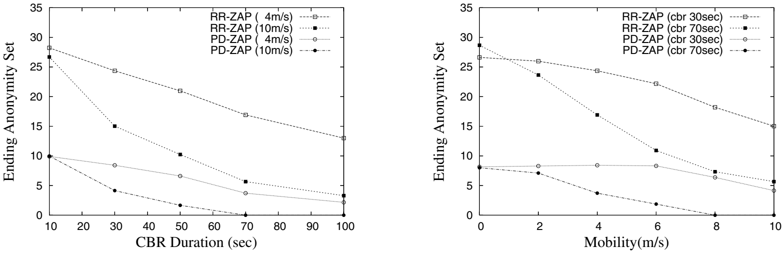
We use QualNet [15], a detailed packet-level network simulator, in investigating the impact from the protocol specific parameters and varying network conditions on the aforementioned metrics. The simulated ad hoc network has 180 nodes initially uniformly distributed in a $2000m \times 2000m$ area. The nodes move according to Random Waypoint Model [5], with a pause time of zero and the minimum and the maximum speeds set to the same (note that this configuration avoids the problem pointed out in [16]). The average density is around 20 neighbors per node. Simulations use renewal CBR application so to constantly maintain five CBR sessions. Each source generates data packets of 256 bytes at a rate of 4 packets per second. The source-destination pairs are chosen randomly from all the nodes(but we exclude the pairs that are located close to the edge of the network to be destinations). The session duration is a variable. We use IEEE 802.11b DCF at MAC layer and two-ray ground propagation model at physical layer. Network devices have link bandwidth at 2Mbps and 370m power range. The results are averaged over several simulation runs with various random seeds.

3.1 Anonymity

The destination anonymity is measured by the size of the anonymity set ($Size_{AS}$) that consists of the nodes remained in the D-AZ through out the session. We investigate how it is affected by session time, mobility, and the sizes of the anonymous zone. The default AZ sizes are 250m.

Figure 5(a) reports the change of $Size_{AS}$ as a function of the session duration. The figure illustrates several interesting facts. First, when session duration increases, all curves show a decreasing trend in anonymity set. Second, when mobility is high, the anonymity set size decreases faster because more nodes move out of the initial anonymous zone during the session. Third, in general, the anonymity set of RR-ZAP is larger than that of PD-ZAP because the entire route becomes the anonymous region, which, in most cases is larger than a destination-based D-AZ.

Figure 5(b) shows the change of $Size_{AS}$ of ZAPs as a function of mobility for long and short sessions. The trends are similar to the previous figure. RR-ZAP has larger AS size. But the set size decreases when mobility increases, especially when sessions last longer the decreasing is quicker. This is because RR-ZAP's anonymous zone is generally long and narrow. It is more sensitive to mobility. Yet PD-ZAP can tolerate higher mobility when session is short (30sec). Up to



(a) AS vs. CBR Duration

(b) AS vs. Mobility

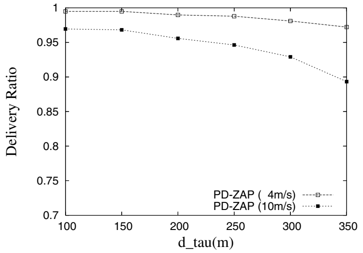
Fig. 5. Payoffs with perfect information

mobility equals to 6m/s, the sizes of the AS are mostly not affected by mobility, due to the fact that few nodes can move out of the original AS region in a short period of time. When session is long (70Sec), all the ZAPs start degradation at low mobility.

3.2 Routing Performance

We investigate how the packet delivery performance of the ZAP protocols are affected by session time, mobility, and the sizes of D-AZs. While we try to stress one condition, we keep other parameters moderate.

Figure 6 investigates how the zones affect PD-ZAP on the delivery ratio. Sessions are kept short in 30 seconds. It shows that PD-ZAP maintains high delivery ratio when mobility is low (4m/s) no matter how d_τ increases. This is because the distance a node can move in the short session time does not cause many nodes to move out of its D-AZ, which is a little smaller than a node's transmission range. But delivery ratio degrades quickly in high mobility (10m/s) as expected.

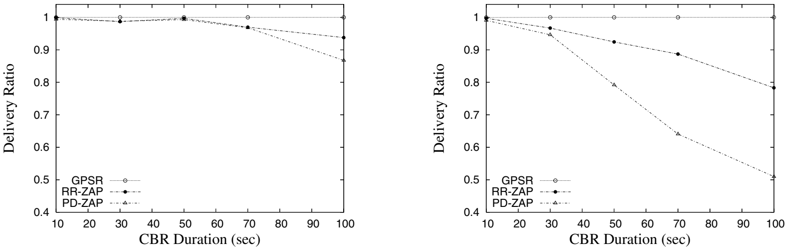


PD-ZAP

Fig. 6. D-AZ Impact on Delivery Ratio

Figure 7 reports the impact from session duration, where d_τ is 250m. The figure shows that GPSR has the nearly perfect data delivery ratio over all the session length. All ZAPs suffer from delivery ratio degradation when sessions are long. High mobility has large impact even when sessions are short. Impact from session duration and mobility is caused by the fact that destination nodes move away from the anonymous region.

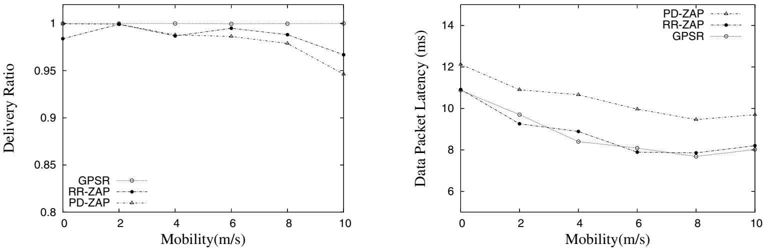
Figure 8 gives mobility impact on the performance of protocols. The configuration is: CBR sessions are 30 seconds long), d_τ is the same as in previous figure. Figure 8(a) shows the mobility impact on delivery ratio. GPSR is not affected by mobility since it can always find a closer forwarder in the current node density



(a) Low Mobility (b) High Mobility

Fig. 7. Session Duration Impact on Delivery Ratio

(nodes update location database once per second). Both RR-ZAP and PD-ZAP are not significantly affected as well because the CBR session time is relatively short. Figure 8(b) shows the latency over mobility. Again, it is expected to see that mobility has little impact on each individual protocol.



(a) Delivery Ratio (b) Packet Latency

Fig. 8. Mobility Impact on Routing Performance

In summary, our simulations show that for destination anonymity protection, RR-ZAP has successfully increased the AS size. But RR-ZAP is more sensitive to mobility and communication duration than PD-ZAP.

4 Conclusion

In this paper we proposed ZAP, an anonymous routing protocol that adopts the group-based anonymity idea in MANET. An anonymity zone is defined, and the nodes residing in the anonymity zone form the anonymity set. Because nodes are mobile, the anonymity set in our work is dynamic, which is different from that in wired networks. We use simulation to study the protocol performance such as node anonymity and packet delivery percentage. We have found that if the anonymity requirement is not high, PD-ZAP can be used because it achieves efficient node anonymity and a good routing performance (e.g., a low probability of a delivery failure). We then propose RR-ZAP, which uses redundant route to further improve anonymity. RR-ZAP is more sensitive to mobility, but it is worthy to trade-off for anonymity compared to PD-ZAP.

References

1. M. K. Reiter and A. D. Rubin, *Crowds: Anonymity ForWeb Transactions*, ACM Transactions on Information and System Security, 1(1):6–92, 1998.
2. R. Sherwood, B. Bhattacharjee, and A. Srinivasan, *p5: A Protocol for Scalable Anonymous Communication*, IEEE Symposium on Security and Privacy, pages 53–65, Oakland, CA, May 2002.
3. V. Scarlata, B. Levine, and C. Shields, *Responder Anonymity and Anonymous Peer-to-Peer File Sharing*, IEEE International Conference on Network Protocols (ICNP), Riverside, CA, 2001.
4. 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, 1999.
5. D. Johnson and D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, in proceedings of ACM SIGCOMM-Computer Communications Review, 1996.
6. I. Stojmenovic, *Position based routing in ad hoc networks*, in IEEE Commmunications Magazine, 40(7):128-134, July 2002.
7. B. Karp and H. T. Kung, *GPSR: Greedy Perimeters Stateless Routing for Wireless Network*, in proceedings of MOBICOM'00, 2000.
8. X. Wu and B. Bhargava, *AO2P: Ad Hoc On-Demand Position-Based Private Routing*, Accepted for publication in IEEE Transaction on Mobile Computing.
9. B. Gedic and L. Liu, *Location Privacy in Mobile System: A Personalized Anonymization Model* in Proceedings of ICDCS, 2005.
10. R. Cheng, D. V. Kalashnikov and S. Prabhakar, *Querying Imprecise Data in Moving Object Environments*, in IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), Vol. 16, No. 9, pp. 1112-1127, Sep 2004.
11. National Institute for Standards and Technology (NIST). *The Keyed-Hash Message Authentication Code*, FIPS 198, 2002.
12. D. L. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM, 24(2):84–88, 1981.
13. M. Reed, P. Syverson, and D. Goldschlag, *Anonymous Connections and Onion Routing*, IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection, 16(4):482–494, 1998.
14. J. Kong and X. Hong, *ANODR: Anonymous on Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks*, 4th ACM international symposium on Mobile ad hoc networking and computing, Annapolis, MD, June 2003.
15. *QualNet*, Scalable Network Technologies (SNT), http://www.qualnet.com/.
16. J. Yoon, M. Liu, and B. Noble, *Random Waypoint Considered Harmful*, in Proceedings of IEEE INFOCOM, 2003.
17. Y. Zhang, W. Liu, and W. Luo, *Anonymous Communications in Mobile Ad Hoc Networks*, in Proceedings of INFOCOM, 2005.

Proxy Signature Without Random Oracles

Xinyi Huang, Willy Susilo, Yi Mu, and Wei Wu[*]

Center for Information Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, Australia
{xh068, wsusilo, ymu}@uow.edu.au

Abstract. In mobile Ad Hoc networks, the existence and availability of trusted authorities is severely limited by intrinsic network features, and problems such as "service availability" have become a crucial issue. A proxy signature scheme allows an entity to delegate his/her signing capability to another entity in such a way that the latter can sign messages on behalf of the former when the former is not available. This is an important primitive to ensure the service availability issue. Proxy signatures have found numerous practical applications such as distributed systems, mobile agent applications, etc. However, the security of the known proxy signature schemes is proven in the random oracle which does not imply security in the real world. In this paper, we propose the *first* proxy signature schemes *without* random oracle. The unforgeability of our scheme is based on the hardness of the well known Computational Diffie Hellman (CDH) problem.

Keywords: Proxy Signature, Without Random Oracles, CDH Problem, Bilinear Pairings.

1 Introduction

In Mobile Ad hoc Networks, permanent connections between customers and servers are unnecessary and infeasible. In order to ensure service availability to the customers distributed in the whole networks, the server must delegate his rights to some other parties in the systems, such as the mobile agents. This way, replication can be achieved and there is no need to count on a single server.

A proxy signature scheme is a variation of the standard signature schemes, in which an original signer (say, Alice) can delegate her signing right to another signer, called the proxy signer (say, Bob), for signing messages. The notion of proxy signature was introduced by Mambo, Usuda and Okamoto [15]. Since then, proxy signature schemes have attracted a considerable amount of interest from the cryptographic research community. Based on the delegation type, there are

[*] The work is partially supported by Nature Science Found of Jiangsu Province (No. BK2006217) and Xidian University's Open Grant of Key Laboratory on Computer Network and Information Security of Ministry of Education of China (No. 20040105).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 473–484, 2006.

three types of proxy signatures: *full delegation*, *partial delegation*, and *delegation by warrant*. In the full delegation system, Alice's secret key is given to Bob directly so that Bob can have the same signing capability as Alice. In practice, such schemes are obviously impractical and insecure. In a partial delegation proxy signature scheme, a proxy signer possesses a key, called private proxy key, which is different from Alice's private key. Hence, proxy signatures generated by using the proxy private key are different from Alice's signatures. However, in such schemes, the messages a proxy signer can sign are *not* limited. This weakness is eliminated in delegation by a warrant that specifies what kinds of messages are delegated. Here, the original signer uses the signing algorithm of a standard signature scheme and its secret key to sign a warrant and generate a signature on the warrant which is called as delegation. The proxy signer uses the delegation and his/her secret key to create a proxy signature on behalf of the original signer. According to whether the original signer can generate a valid proxy signature, proxy signatures can be classified into *proxy-unprotected* and *proxy-protected* schemes. In a proxy-protected scheme only the proxy signer can generate proxy signatures, while in a proxy-unprotected scheme either the proxy signer or the original signer can generate proxy signatures. In many applications, proxy-protected schemes are required to avoid the potential disputes between the original signer and the proxy signer. Though there exist many proxy signature schemes, most of them are insecure [14,11,13,16,17,20].

Provable security is the basic requirement for the proxy signature schemes. Currently, all the practical secure signature schemes were proven in the random oracle model. The random oracle model was introduced by Bellare and Rogaway in [5]. The model replaces hash functions by truly random objects and provides probabilistic security proofs for the resulting schemes, showing that attacks against these can be turned into efficient solutions of well-known mathematical problems, such as the discrete logarithm problem or factorization. Although the model is efficient and useful, it has received a lot of criticism that the proofs in the random oracle model are not proofs. They are simply a design validation methodology capable of spotting defective or erroneous designs when they fail. Canetti *et al.* have shown that security in the random oracle model does not imply the security in the real world in that a scheme can be secure in the random oracle model and yet be broken without violating any particular intractability assumption, and without breaking the underlying hash functions [7]. Therefore, the search for a secure proxy signature scheme without random oracle remains an open and interesting research problem.

Our Contribution

In this paper, we propose the *first* secure proxy signature scheme whose security does *not* rely on the random oracle. We incorporate Water's signature scheme [19] to obtain a concrete secure proxy signature scheme. The new scheme is proxy-protected in the sense that even the proxy signer can not forge a valid proxy signature. The security of the proposed scheme is based on the hardness of the well-known hard problem, the Computational Diffie Hellman Problem.

Roadmap

The rest of this paper is arranged as follows. In next section, we provide the preliminaries of our scheme including bilinear pairings and security assumptions. In Section 3, we describe the formal models of our proxy signature scheme. We present our proxy signature scheme without random oracle in Section 4. In Section 5, we provide formal security analysis of the proposed scheme. Finally, we conclude our paper in Section 6.

2 Preliminaries

In this section, we will review some fundamental backgrounds used throughout this paper, namely bilinear pairings and complexity assumption.

2.1 Bilinear Pairing

Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two groups of prime order p and let g be a generator of $\mathbb{G}_1$. The map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following three conditions hold true:

- e is bilinear, i.e. $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
- e is non-degenerate, i.e. $e(g, g) \neq 1_{\mathbb{G}_T}$.
- e is efficiently computable.

We say that $(\mathbb{G}_1, \mathbb{G}_T)$ are bilinear groups if there exists a group $\mathbb{G}_T$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ as above, and e, and the group action in $\mathbb{G}_1$ and $\mathbb{G}_T$ can be computed efficiently. See [3] for more details on the construction of such pairings.

2.2 Complexity Assumption

Definition 1. Computational Diffie Hellman (CDH) Problem in $\mathbb{G}_1$ *Given $g, g^a, g^b \in \mathbb{G}_1$ for some unknown $a, b \in \mathbb{Z}_p$, compute $g^{ab} \in \mathbb{G}_1$.*

The success probability of a polynomial algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}_1$ is denoted:

$$\mathsf{Succ}^{CDH}_{\mathcal{A},\mathbb{G}_1} = Pr[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \in_R \mathbb{Z}_p,]$$

Definition 2. Computational Diffie Hellman (CDH) Assumption in $\mathbb{G}_1$ *Given $g, g^a, g^b \in_R \mathbb{G}_1$, for some unknown $a, b \in \mathbb{Z}_p$, $\mathsf{Succ}^{CDH}_{\mathcal{A},\mathbb{G}_1}$ is negligible.*

3 Formal Models of Proxy Signatures

Let Alice denote the original signer and Bob the proxy signer. Our proxy signature scheme consists of the following algorithms: ParaGen, KeyGen, StandardSign, DelegationGen, ProxySign and ProxyVerification.

1. $\mathsf{ParaGen}$: Taking as input the system security parameter ℓ, this algorithm outputs system's parameters: Para. That is: $\mathsf{Para} \leftarrow \mathsf{ParaGen}(\ell)$
2. KeyGen: Taking as input system's parameter Para, this algorithm generates a secret-public key pair (sk_i, pk_i) where $i \in \{a, b\}$ denotes Alice and Bob, respectively. That is: $(sk_i, pk_i) \leftarrow \mathsf{KeyGen}(\mathsf{Para})$
3. $\mathsf{StandardSign}$: Input system's parameter Para, the signer's secret key sk and the message M to be singed, this algorithm generates the standard signature: σ_S. That is: $\sigma_S \leftarrow \mathsf{StandardSign}\ (\mathsf{Para}, sk, M)$
4. $\mathsf{DelegationGen}$: Input system's parameter Para, the original signer's secret key sk_a and the warrant W to be singed, this algorithm uses the $\mathsf{StandardSign}$ algorithm to generate the delegation: σ_W. That is: $\sigma_W \leftarrow \mathsf{DelegationGen}(\mathsf{Para}, sk_a, W)$
5. $\mathsf{ProxySign}$: Input system's parameter Para, the warrant W, the delegation σ_w, the secret key sk_b of the proxy signer and the message M to be signed, this algorithm generates the proxy signature σ. That is: $\sigma_M \leftarrow \mathsf{ProxySign}(\mathsf{Para}, W, \sigma_W, sk_b, M)$
6. $\mathsf{ProxyVerification}$: Input system's parameter Para, original signer's public keys pk_a, proxy signer's public key pk_b, the warrant W, the signed message M and the signature σ_M, this algorithm outputs True if σ is a valid proxy signature of the message M and output $\perp$ otherwise. That is: $\{\mathsf{True}, \perp\} \leftarrow \mathsf{ProxyVerification}(\mathsf{Para}, pk_a, pk_b, W, M, \sigma_M)$

3.1 Security Models

Lee, Kim and Kim defined some properties that a strong proxy signature scheme should provide in [12]. While these informal requirements provide some intuition about the goals that a notion of security for proxy signature schemes should capture, their precise meaning is unclear. The first security model of proxy signature was proposed in [4]. In [10], the authors also proposed a security model of the proxy signature. In the model defined in[10], they divide the potential attackers into three kinds:

1. **Type I:** This type adversary $\mathcal{A}_I$ only has the public keys of Alice and Bob.
2. **Type II:** This type adversary $\mathcal{A}_{II}$ has the public keys of Alice and Bob, he additionally has the secret key of the proxy signer Bob.
3. **Type III:** This type adversary $\mathcal{A}_{III}$ has the public keys of Alice and Bob, he additionally has the secret key of the original signer Alice.

One can find that if a proxy signature scheme is secure against Type II (or Type III) adversary, the scheme is also secure against Type I adversary. We note the above classification helps to make the security model clearer, therefore, we will use this classification to redefine and improve the security model proposed in [4]. In the security model defined later, we only consider the general case of the proxy signature where the original signer and the proxy signer are distinct.

In a warrant based proxy signature, the delegation is the original signer's standard signature on the warrant which contains information regarding the particular proxy signer such as the proxy signer's public key, a period of validity, and restrictions on the class of messages for which the warrant is valid. Therefore, this kind of proxy signature can prevent the misuse of the delegation. Here after, we only focus on the unforgeability of the proxy signature.

Existential unforgeability against adaptive A_{II} Adversary
Roughly speaking, the existential unforgeability of a proxy signature scheme under a type II attacker requires that it is difficult for a user to forge a valid proxy signature under a warrant if he does not obtain the delegation of this warrant. It is defined using the following game between the challenger $\mathcal{C}$ and a type II adversary $\mathcal{A}_{II}$:

- Setup: $\mathcal{C}$ runs the ParaGen algorithm to obtain system's parameter Para, runs KeyGen to obtain the secret-public key pairs $(sk_a, pk_a), (sk_b, pk_b)$ of the original signer Alice and proxy signer Bob, respectively. $\mathcal{C}$ then sends (pk_a, pk_b, sk_b) to the adversary A_{II}.
- Delegation queries: Proceeding adaptively, A_{II} can request the delegation on the warrant W. In response, $\mathcal{C}$ runs the DelegationGen algorithm to obtain σ_W and returns σ_W to the adversary A_{II}.
- ProxySign queries: Proceeding adaptively, A_{II} can request the proxy signature on the message M under the warrant W. In response, $\mathcal{C}$ runs DelegationGen algorithm to generate the delegation on the warrant W. Then $\mathcal{C}$ runs the ProxySign algorithm to obtain the proxy signature σ_M and returns σ_M to the adversary A_{II}.
- Output: Finally, A_{II} outputs a signature σ^* with the warrant W^* and the message M^* such that
  1. W^* has not been requested as one of the Delegation queries.
  2. (M^*, W^*) has not been requested as one of the ProxySign queries.
  3. σ^* is a valid proxy signature of the message M^* under the warrant W^*.

Compared with the model defined in [4], an important refinement is that $\mathcal{A}_{II}$ can adaptively submit the ProxySign queries under warrant whose delegation is unknown to $\mathcal{A}_{II}$. The only restrictions are that when $\mathcal{A}_{II}$ outputs the forgery (M^*, W^*, σ^*), he cannot submit W^* as one of the Delegation queries or submit (M^*, W^*) as one of the ProxySign queries. However, he can even submit (M', W^*) to the ProxySign queries where $M' \neq M^*$. The success probability of an algorithm A_{II} wins the above game is defined as $Succ\ A_{II}$.

Definition 3. *We say a type II adversary $\mathcal{A}_{II}$ can $(t, q_W, q_{PS}, \varepsilon)$ break a proxy signature scheme if A_{II} runs in time at most t, $\mathcal{A}_{II}$ makes at most q_W* Delegation *queries and at most q_{PS}* ProxySign *queries and $Succ\ A_{II}$ is at least ε.*

Existential unforgeability against adaptive A_{III} adversary
The existential unforgeability of a proxy signature scheme under a type III attacker requires that it is difficult for the original signer to generate a valid proxy signature of a message M^* which has not been singed by the proxy signer. It is defined using the following game between the challenger $\mathcal{C}$ and a type III adversary $\mathcal{A}_{III}$:

- Setup: $\mathcal{C}$ runs the ParaGen algorithm to obtain system's parameter Para, runs KeyGen to obtain the secret-public key pairs $(sk_a, pk_a), (sk_b, pk_b)$ of the original signer Alice and proxy signer Bob, respectively. $\mathcal{C}$ then sends (pk_a, pk_b, sk_a) to the adversary $\mathcal{A}_{III}$.
- StandardSign: Proceeding adaptively, $\mathcal{A}_{III}$ can request proxy signer's standard signature on the message M. In response, $\mathcal{C}$ runs the StandardSign algorithm to generate the standard signature on the message M and returns to the adversary $\mathcal{A}_{III}$.
- ProxySign queries: Proceeding adaptively, $\mathcal{A}_{III}$ can request the proxy signature on the message M under the warrant W. In response, $\mathcal{C}$ runs the DelegationGen algorithm to generate the delegation on the warrant W. Then $\mathcal{C}$ runs the ProxySign algorithm to generate the proxy signature σ_M and returns σ_M to the adversary $\mathcal{A}_{III}$.
- Output: Finally, $\mathcal{A}_{III}$ outputs a signature σ^* with the warrant W^* and the message M^* such that
  1. (M^*, W^*) has not been requested as one of the ProxySign queries.
  2. σ^* is a valid proxy signature of the message M^* under the warrant W^*.

In this model, we allow the attacker $\mathcal{A}_{III}$ can submit StandardSign queries, this is to guarantee that proxy signer's standard signature on the message M^* can not help the attacker to forge a valid proxy signature on the same message. The success probability of an algorithm $\mathcal{A}_{III}$ wins the above game is defined as *Succ* $\mathcal{A}_{III}$

Definition 4. *We say a type III adversary $\mathcal{A}_{III}$ can $(t, q_S, q_{PS}, \varepsilon)$ break a proxy signature scheme if $\mathcal{A}_{III}$ runs in time at most t, $\mathcal{A}_{III}$ makes at most q_S* StandardSign*queries and q_{PS}* ProxySign *queries, and Succ $\mathcal{A}_{III}$ is at least ε.*

4 Proposed Scheme

In this section, we will describe our proxy signature scheme without random oracle. It consists of the following algorithms:

1. ParaGen: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be bilinear groups defined in Section 2.1 where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$ for some prime p, g is the generator of $\mathbb{G}_1$. e denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. The messages M to be signed in this scheme will be represented as bitstrings of length n. Furthermore, picks $2n+2$ random elements $u', v', u_1, u_2, \cdots, u_n, v_1, \cdots, v_n \in_R \mathbb{G}_1$ and set $\boldsymbol{u} = (u_1, u_2, \cdots, u_n)$, $\boldsymbol{v} = (v_1, v_2, \cdots, v_n)$. Then the common parameter $\mathsf{Para} = (\mathbb{G}_1, \mathbb{G}_T, p, g, e, n, u', v', \boldsymbol{u}, \boldsymbol{v})$.
2. Key Gen: The original Alice picks two secret values $x_a, y_a \in_R \mathbb{Z}_p^*$ and set the secret key $sk_a = (sk_{ax}, sk_{ay}) = (x_a, y_a)$. Then the signer computes the public key $pk_a = (pk_{ax}, pk_{ay}) = (g^{x_a}, g^{y_a})$. Similarly, the proxy signer's secret key is $sk_b = (sk_{bx}, sk_{by}) = (x_b, y_b)$ and the public key is $pk_b = (pk_{bx}, pk_{by}) = (g^{x_b}, g^{y_b})$

3. StandardSign: Let M be an n-bit message to be signed and M_i denote the i^{th} bit of M, and $\mathcal{M} \in \{1, \cdots, n\}$ be the set of all i for which $M_i = 1$, the standard signature is generated as follows. First, a random $r \in \mathbb{Z}_p$ is chosen. Then the standard signature is constructed as: $\sigma_S = (\sigma_{S_1}, \sigma_{S_2})$ where $\sigma_{S_1} = g^{sk_x sk_y}(u' \prod_{i \in \mathcal{M}} u_i)^r, \sigma_{S_2} = g^r$. Here sk_x, sk_y denote the secret key of the signer.
4. DelegationGen: Let W be an n-bit message to be signed by the original signer Alice and W_i denote the i^{th} bit of W, and $\mathcal{W} \in \{1, \cdots, n\}$ be the set of all i for which $W_i = 1$, the delegation is generated as follows. First, a random $r_a \in \mathbb{Z}_p$ is chosen. Then the signature is constructed as: $\sigma_W = (\sigma_{W_1}, \sigma_{W_2})$ where $\sigma_{W_1} = g^{x_a y_a}(u' \prod_{i \in \mathcal{W}} u_i)^{r_a}, \sigma_{W_2} = g^{r_a}$. Then Alice sends the delegation σ_W with the warrant W to the proxy signer Bob.
5. ProxySign: Let M be an n-bit message to be signed by the original signer Alice and M_i denote the i^{th} bit of M, and $\mathcal{M} \in \{1, \cdots, n\}$ be the set of all i for which $M_i = 1$, the proxy signature is generated as follows. First, two random values $r'_a, r_b \in \mathbb{Z}_p$ are chosen. Then the signature is constructed as:

$$\begin{aligned}\sigma_M = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3}) &= (\sigma_{W_1}(u' \prod_{i \in \mathcal{W}} u_i)^{r'_a} g^{x_b y_b}(v' \prod_{i \in \mathcal{M}} v_i)^{r_b}, \sigma_{W_2} g^{r'_a}, g^{r_b}).\\ &= (g^{x_a y_a} g^{x_b y_b}(u' \prod_{i \in \mathcal{W}} u_i)^{r_a + r'_a}(v' \prod_{i \in \mathcal{M}} v_i)^{r_b}, g^{r_a + r'_a}, g^{r_b})\end{aligned}$$

6. Verification: Given the public keys (pk_a, pk_b), a warrant $W \in \{0,1\}^n$, a message $M \in \{0,1\}^n$, and a signature $\sigma_M = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, verify whether

$$e(\sigma_{M_1}, g) \stackrel{?}{=} e(pk_{ax}, pk_{ay}) e(pk_{bx}, pk_{by}) e(u' \prod_{i \in \mathcal{W}} u_i, \sigma_{M_2}) e(v' \prod_{i \in \mathcal{M}} v_i, \sigma_{M_3}).$$

If the equality holds the result is True; otherwise the result is $\perp$.

Correctness:

$$\begin{aligned}e(\sigma_{M_1}, g) &= e(g^{x_a y_a} g^{x_b y_b}(u' \prod_{i \in \mathcal{W}} u_i)^{r_a + r'_a}(v' \prod_{i \in \mathcal{M}} v_i)^{r_b}, g)\\ &= e(g^{x_a y_a}, g) e(g^{x_b y_b}, g) e((u' \prod_{i \in \mathcal{W}} u_i)^{r_a + r'_a}, g) e((v' \prod_{i \in \mathcal{M}} v_i)^{r_b}, g)\\ &= e(pk_{ax}, pk_{ay}) e(pk_{bx}, pk_{by}) e(u' \prod_{i \in \mathcal{W}} u_i, g^{r_a + r'_a}) e(v' \prod_{i \in \mathcal{M}} v_i, g^{r_b})\\ &= e(pk_{ax}, pk_{ay}) e(pk_{bx}, pk_{by}) e(u' \prod_{i \in \mathcal{W}} u_i, \sigma_{M_2}) e(v' \prod_{i \in \mathcal{M}} v_i, \sigma_{M_3})\end{aligned}$$

5 Security Analysis

In this section, we will provide the formal security analysis of the proposed proxy signature scheme.

5.1 Unforgeability Against Type II Adversary

Theorem 1. *If there exists a type II adversary $\mathcal{A}_{II}$ can $(t, q_W, q_{PS}, \varepsilon)$ breaks the proposed proxy signature scheme then there exists another algorithm $\mathcal{B}$ who can use $\mathcal{A}_{II}$ to solve an instance of the CDH problem in $\mathbb{G}_1$ with the probability*

$$Succ_{\mathcal{B},\mathbb{G}_1}^{CDH} \geq \frac{\varepsilon}{27(n+1)^2} \cdot \frac{1}{(q_W + q_{PS})^2}$$

in time $t + c_1(4q_W + 7q_{PS}) + c_2((n+2)q_W + (2n+4)q_{PS})$. Here c_1, c_2 are the two constants that depend on $\mathbb{G}_1$.

Proof. Let $\mathbb{G}_1$ be a bilinear pairing group of prime order p. Algorithm $\mathcal{B}$ is given $g, g^a, g^b \in \mathbb{G}_1$ which is a random instance of the CDH problem. Its goal is to compute g^{ab}. Algorithm $\mathcal{B}$ will simulate the challenger and interact with the forger $\mathcal{A}_{II}$ as described below.

Let's recall the definition of the type II adversary $\mathcal{A}_{II}$. This type of adversary $\mathcal{A}_{II}$ has the public key of the original signer Alice and the proxy singer Bob, he also has Bob's secret key.

1. Setup: $\mathcal{B}$ chooses two integers ℓ_a, ℓ_b, and other two integers, k_a, k_b, uniformly at random between 0 and n. Then it chooses two values x'_a, x'_b and two random n-vectors, $\boldsymbol{x_a} = (x_{ai}), \boldsymbol{x_b} = (x_{bi})$ where $x'_a, x_{ai} \in_R \mathbb{Z}_{\ell_a}, x'_b, x_{bi} \in_R \mathbb{Z}_{\ell_b}$. Additionally, $\mathcal{B}$ chooses two values y'_a, y'_b and two random n-vectors $\boldsymbol{y_a} = (y_{ai}), \boldsymbol{y_b} = (y_{bi})$ where $y'_a, y'_b, y_{ai}, y_{bi} \in_R \mathbb{Z}_p$. $\mathcal{B}$ keeps all the values secret.

   For an n-bit X, we let $\mathcal{X} \subseteq \{1, 2, \cdots, n\}$ be the set of all i for which $X_i = 1$. Then, for a warrant W, $\mathcal{W}$ be the set of all i for which $W_i = 1$. Similarly, for a message M, $\mathcal{M}$ be the set of all i for which $M_i = 1$. To make the notation easy to follow, we define six functions $F_a(X), F_b(X), J_a(X), J_b(X)$ and $K_a(X), K_b(X)$ as [19]:

   (a) $F_a(X) = (p - \ell_a k_a) + x'_a + \Sigma_{i \in \mathcal{X}} x_{ai}$ and $F_b(X) = (p - \ell_b k_b) + x'_b + \Sigma_{i \in \mathcal{X}} x_{bi}$

   (b) $J_a(X) = y'_a + \Sigma_{i \in \mathcal{X}} y_{ai}$ and $J_b(M) = y'_b + \Sigma_{i \in \mathcal{X}} y_{bi}$

   (c) $K_a(X) = \begin{cases} 0, & \text{if } x'_a + \Sigma_{i \in \mathcal{X}} x_{ai} \equiv 0 \pmod{\ell_a} \\ 1, & \text{otherwise} \end{cases}$

   and $K_b(X) = \begin{cases} 0, & \text{if } x'_b + \Sigma_{i \in \mathcal{X}} x_{bi} \equiv 0 \pmod{\ell_b} \\ 1, & \text{otherwise} \end{cases}$

   $\mathcal{B}$ sets the public keys of the users and the common parameter as:

   (a) $\mathcal{B}$ chooses two random numbers $sk_{bx}, sk_{by} \in \mathbb{Z}_p^*$ and sets

   $$pk_{ax} = g^a, pk_{ay} = g^b, pk_{bx} = g^{sk_{bx}}, pk_{by} = g^{sk_{by}}.$$

   Where g^a, g^b are the input of the CDH problem.

   (b) $\mathcal{B}$ assigns $u' = pk_{ay}^{p - k_a \ell_a + x'_a} g^{y'_a}, u_i = pk_{ay}^{x_{ai}} g^{y_{ai}}, \boldsymbol{u}_a = (u_1, u_2, \cdots, u_n)$

   (c) $\mathcal{B}$ then assigns, $v' = pk_{ay}^{p - k_b \ell_b + x'_b} g^{y'_b}, v_i = pk_{by}^{x_{bi}} g^{y_{bi}}$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_n)$. Then $\mathcal{B}$ returns $(\mathbb{G}_1, \mathbb{G}_T, e, p, g, \boldsymbol{u}, u', \boldsymbol{v}, v')$ and $(pk_{ax}, pk_{ay}, pk_{bx}, pk_{by}, sk_{bx}, sk_{by})$ to the Type II adversary $\mathcal{A}_{II}$.

2. Delegation queries: Suppose $\mathcal{A}_{II}$ issues a delegation query for an n-bit warrant W. If $K_a(W) \neq 0$ (If we have $K_a(W) \neq 0$ this implies $F_a(W) \neq 0 \pmod{p}$, since we can assume $p > n\ell_a$ for any reasonable values of p, n, and ℓ_a[19]), $\mathcal{B}$ can construct the delegation of this warrant by choosing a random $r_a \in \mathbb{Z}_p$ and computing:

$$\sigma_W = (\sigma_{W_1}, \sigma_{W_2}) = \left(pk_{ax}^{\frac{-J_a(W)}{F_a(W)}} (u' \prod_{i \in \mathcal{W}} u_i)^{r_a}, pk_{ax}^{\frac{-1}{F_a(W)}} g^{r_a} \right)$$

If $K_a(W) = 0$. $\mathcal{B}$ terminates the simulation and reports failure.

3. ProxySign queries: Suppose $\mathcal{A}_{II}$ issues a delegation query for an n-bit message M under the warrant W.
   (a) If $K_a(W) = 0, K_b(M) = 0$, $\mathcal{B}$ terminates the simulation and reports failure.
   (b) Else $K_a(W) = 0, K_b(M) \neq 0$, $\mathcal{B}$ can construct the delegation of this warrant by choosing a random $r_a, r_b \in \mathbb{Z}_p$ and computing: $\sigma_M = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$. where

$$\sigma_{M_1} = \left(pk_{ax}^{\frac{-J_b(M)}{F_b(M)}} (u' \prod_{i \in \mathcal{W}} u_i)^{r_a} \cdot g^{sk_{bx} sk_{by}} (v' \prod_{i \in \mathcal{M}} v_i)^{r_b} \right),$$

$$\sigma_{M_2} = g^{r_a}, \sigma_{M_3} = pk_{ax}^{\frac{-1}{F_b(M)}} \cdot g^{r_b}$$

   (c) Otherwise $K_a(W) \neq 0$. In this case, $\mathcal{B}$ can compute the delegation of the warrant W as he does in response to the delegation queries. Since $\mathcal{B}$ knows the secret key sk_{bx}, sk_{by} of proxy signer, $\mathcal{B}$ can run the ProxySign algorithm as defined in Section 4 to compute the proxy signature and return the signature to $\mathcal{A}_{III}$.

Finally, the adversary $\mathcal{A}_{II}$ outputs a proxy signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ of the message M^* under the warrant W^* such that
(a) W^* has not been submitted as one of the Delegation queries.
(b) (M^*, W^*) has not been submitted as one of the ProxySign queries.
(c) $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ is a valid signature, that is:

$$\sigma_1^* = g^{sk_{ax} sk_{ay}} g^{sk_{bx} sk_{by}} (u' \prod_{i \in \mathcal{W}^*} u_i)^{r_a^*} (v' \prod_{i \in \mathcal{M}^*} v_i)^{r_b^*}, \sigma_2^* = g^{r_a^*}, \sigma_3^* = g^{r_b^*}$$

If $F_a(W^*) \neq 0$ or $F_b(M^*) \neq 0$, $\mathcal{B}$ will abort. Otherwise, $F_a(W^*) = 0$, $F_b(M^*) = 0$. In this case,

$$\begin{aligned} \sigma_1^* &= g^{sk_{ax} sk_{ay}} g^{sk_{bx} sk_{by}} (u' \prod_{i \in \mathcal{W}^*} u_i)^{r_a^*} (v' \prod_{i \in \mathcal{M}^*} v_i)^{r_b^*} \\ &= g^{ab} g^{sk_{bx} sk_{by}} (g^{J_a(W^*)})^{r_a^*} (g^{J_b(M^*)})^{r_b^*} \\ &= g^{ab} g^{sk_{bx} sk_{by}} (g^{r_a^*})^{J_a(W^*)} (g^{r_b^*})^{J_b(M^*)} \\ &= g^{ab} g^{sk_{bx} sk_{by}} (\sigma_2^*)^{J_a(W^*)} (\sigma_3^*)^{J_b(M^*)} \end{aligned}$$

Therefore, $\mathcal{B}$ can compute $g^{ab} = \frac{\sigma_1^*}{g^{sk_{bx} sk_{by}} (\sigma_2^*)^{J_a(W^*)} (\sigma_3^*)^{J_b(M^*)}}$

This completes the description of the simulation. It remains to analyze the probability of $\mathcal{B}$ not aborting. $\mathcal{B}$ will not abort if all the following cases happen:

$$\begin{aligned}
&\mathsf{A}: K_a(W) \neq 0 \pmod{\ell_a} \text{ during } \mathsf{Delegation} \text{ queries}\\
&\mathsf{B}: K_a(W) \neq 0 \pmod{\ell_a} \text{ or } K_b(M) \neq 0 \pmod{\ell_b} \text{ during } \mathsf{ProxySign} \text{ queries}\\
&\mathsf{C}: F_a(W^*) = 0 \pmod{p} \text{ and } F_b(M^*) = 0 \pmod{p}
\end{aligned}$$

The success probability is $Succ_{\mathcal{B}}^{CDH}$=$\Pr[A \wedge B \wedge C]\varepsilon$.

$$\begin{aligned}
\Pr[A \wedge B \wedge C] &= \Pr[\bigwedge_{i=1}^{q_W} K_a(W_i) \neq 0 \bigwedge_{i=1}^{q_{PS}} \Big(K_a(W_i) \neq 0 \bigvee K_b(M_i) \neq 0\Big)\\
&\bigwedge F_a(W^*) = 0 \pmod{p} \bigwedge F_b(M^*) = 0 \pmod{p}]\\
&\geq \frac{1}{(n+1)^2 \ell_a \ell_b}(1 - \frac{2(q_W + q_{PS})}{\ell_a})
\end{aligned}$$

Therefore, $Succ_{\mathcal{B},\mathbb{G}_1}^{CDH} \geq \frac{1}{(n+1)^2\ell_a\ell_b}(1-\frac{2(q_W+q_{PS})}{\ell_a})\varepsilon$. We can optimize it by setting $\ell_a = \ell_b = 3(q_W + q_{PS})$, then

$$Succ_{\mathcal{B},\mathbb{G}_1}^{CDH} \geq \frac{\varepsilon}{27(n+1)^2} \cdot \frac{1}{(q_W + q_{PS})^2}$$

Algorithm $\mathcal{B}$'s running time is the same as $\mathcal{A}'_{II}s$ running time plus the time it takes to respond to q_W Delegation queries and q_{PS} ProxySign queries. Each Delegation query requires 4 exponentiation operations and $n+2$ multiplication operations in $\mathbb{G}_1$. Each ProxySign query requires at most 7 exponentiation operations and $2n+4$ multiplication operations in $\mathbb{G}_1$. If we assume each exponentiation takes time c_1 and each multiplication takes time c_2, the total running time is at most $t + c_1(4q_W + 7q_{PS}) + c_2((n+2)q_W + (2n+4)q_{PS})$. This completes the proof. □

5.2 Unforgeability Against Type III Adversary

Theorem 2. *If there exists a type III adversary $\mathcal{A}_{III}$ can $(t, q_S, q_{PS}, \varepsilon)$ breaks the proposed proxy signature scheme then there exists another algorithm $\mathcal{B}$ who can use $\mathcal{A}_{III}$ to solve an instance of the CDH problem in $\mathbb{G}_1$ with the probability*

$$Succ_{\mathcal{B},\mathbb{G}_1}^{CDH} \geq \frac{\varepsilon}{27(n+1)^2} \cdot \frac{1}{(q_S + q_{PS})^2}$$

in time $t + c_1(4q_W + 7q_{PS}) + c_2((n+2)q_W + (2n+4)q_{PS})$. Here c_1, c_2 are the two constants that depend on $\mathbb{G}_1$.

Proof. It is similar to the proof of Theorem 1.

6 Conclusion

In this paper, we proposed the first proxy signature scheme without random oracle based on Water's signature scheme [19]. We showed that our scheme is unforgeable against an adaptively chosen message attacker. Even the original signer can not forge a valid proxy signature of our scheme. The security of our scheme is based on the Computational Diffie Hellman problem.

Acknowledgement

The authors would like to thank the anonymous referees of the Second International Conference on Mobile Ad Hoc and Sensor Networks (MSN 2006) for the suggestions to improve this paper.

References

1. D. Boneh and X. Boyen. Short signatures without random oracles. In Advances in Cryptology, Proc. EUROCRYPT 2004, Lecture Notes in Computer Science 3027, pages. 56–73. Springer–Verlag, 2004.
2. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In Advances in Cryptology, Proc. CRYPTO 2001, Lecture Notes in Computer Science 2139, pages. 213–229. Springer–Verlag, 2001.
3. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In Advances in Cryptology–ASIACRYPT 2001, Lecture Notes in Computer Science 2248, pages. 514–532. Springer–Verlag, 2001.
4. A. Boldyreva, A. Palacio and B. Warinschi. Secure proxy signature scheme for delegation of signing rights. IACR ePrint Archive, available at `http://eprint.iacr.org/2003/096/`, 2003.
5. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. Advances in Cryptology - Eurocrypt'96, Lecture Notes in Computer Science 950, pages 399–416, Springer-Verlag, Berlin, 1996.
6. J. H. Cheon. Security analysis of the strong diffie-hellman problem. EUROCRYPT 2006, to appear.
7. R. Canetti, O. Goldreich and S. Halevi. The Random Oracle Methodology, revisited. In Proceedings of the 30th Annual Symposium on the Theory of Computing (STOC'98), pages 209–218, 1998.
8. S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptively chosen message attacks. SIAM Journal on Computing, 17(2):281-308, 1988
9. S.D. Galbraith and K.G. Paterson and N.P. Smart. Pairings for cryptographers. IACR ePrint Archive, available at http://eprint.iacr.org/2006/165, 2006.
10. X. Huang, Y. Mu, W. Susilo, F. Zhang and X. Chen. A short proxy signature scheme: efficient authentication in the ubiquitous world. The Second International Symposium on Ubiquitous Intelligence and Smart Worlds (UISW2005), Lecture Notes in Computer Science 3823, pages. 480–489, Springer-Verlag, 2005.
11. J.-Y. Lee, J. H. Cheon and S. Kim. An analysis of proxy signatures: Is a secure channel necessary? In Topics in Cryptology - CT-RSA 2003, Lecture Notes in Computer Science 2612, pages. 68–79. Springer–Verlag, 2003.

12. B. Lee, H. Kim and K. Kim. Strong proxy signature and its applications. In Proc of SCIS'01, pages. 603–08. 2001.
13. B. Lee, H. Kim, and K. Kim. Secure mobile agent using strong nondesignated proxy signature. In Information Security and Privacy (ACISP01), Lecture Notes in Computer Science 2119, pages. 474–486. Springer–Verlag, 2001.
14. S. Kim, S. Park and D. Won. Proxy signatures, revisited. In Information and Communications Security (ICICS97), Lecture Notes in Computer Science 1334, pages. 223–232. Springer–Verlag, 1997.
15. M. Mambo, K. Usuda and E. Okamoto. Proxy signature: delegation of the power to sign messages. IEICE Trans. Fundamentals, Vol. E79-A, No. 9, Sep., pages. 1338–1353, 1996.
16. T. Okamoto, A. Inomata, and E. Okamoto. A proposal of short proxy signature using pairing. In International Conference on Information Technology (ITCC 2005), pages. 631–635. IEEE Computer Society, 2005.
17. T. Okamoto, M. Tada, and E. Okamoto. Extended proxy signatures for smart cards. In ISW 99, Lecture Notes in Computer Science 1729, pages. 247–258, Springer-Verlag, 1999.
18. H.-U. Park and I.-Y. Lee. A digital nominative proxy signature scheme for mobile communications. In Information and Communications Security (ICICS 2001), Lecture Notes in Computer Science 2229, pages. 451–455, Springer–Verlag, 2001.
19. R. Waters. Efficient identity-based encryption without random oracles. In EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pages. 114–127. Springer–Verlag, 2005.
20. G. Wang, F. Bao, J. Zhou and Robert H. Deng. Security analysis of some proxy signatures. In ICICS 2003, Lecture Notes in Computer Science 2971, pages. 305–319. Springer–Verlag, 2003.
21. R. Zhang, J. Furukawa and H. Imai. Short signature and universal designated verifier signature without random oracles. In Applied Cryptography and Network Security (ACNS 2005), Lecture Notes in Computer Science 3531, pages. 483–498. Springer-Verlag, 2005.
22. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In Public Key Cryptography (PKC'04), Lecture Notes in Computer Science 2947, pages 277–290. Springer–Verlag, 2004.

Building Hierarchical Public Key Infrastructures in Mobile Ad-Hoc Networks

Cristina Satizábal[1,2], Jordi Forné[1], Juan Hernández-Serrano[1], and Josep Pegueroles[1]

[1] Department of Telematics Engineering, Technical University of Catalonia,
Jordi Girona 1-3 C3, 08034 Barcelona, Spain
{isabelcs, jforne, jserrano, josep.pegueroles}@entel.upc.edu
[2] Department of Engineering and Architecture, Pamplona University,
Km 1 via Bucaramanga, Pamplona, Colombia

Abstract. Dynamism of mobile ad-hoc networks implies changing trust relationships among their nodes that can be established using peer-to-peer PKIs. Here, certification paths can be built although part of the infrastructure is temporarily unreachable because there can be multiple paths between two entities but certification path discovery is difficult since all the options do not lead to the target entity. On the contrary, in hierarchical PKIs, there is only one path between two entities and certification paths are easy to find. For that reason, we propose a protocol that establishes a virtual hierarchy in a peer-to-peer PKI. The results show that this protocol can be executed in a short time. In addition, our protocol does not require to issue new certificates among PKI entities, facilitates the certification path discovery process and the maximum path length can be adapted to the characteristics of users with limited processing and storage capacity.

Keywords: Public Key Infrastructure (PKI), hierarchical trust model, peer-to-peer trust model, Mobile Ad-hoc Networks (MANETs).

1 Introduction

The development of wireless networks has allowed that mobile users with compatible wireless devices can establish short duration networks that permit them to satisfy their communication necessities in a certain moment called ad hoc networks. Mobile Ad-hoc Networks (MANETs) are open peer-to-peer networks where nodes can move randomly and organize themselves arbitrarily. The network topology is highly dynamic as nodes frequently join or leave the network, and roam in the network on their own will.

Trust establishment is a difficult task in MANETs that requires fast and flexible protocols, independent of a fixed networking infrastructure. Peer-to-peer Public Key Infrastructures (PKIs) can provide such trust since they are quite dynamic and certification paths can be built although part of the infrastructure is temporarily unreachable. However, certification path discovery is difficult because they can have multiple paths between two entities and all paths do not lead to the target entity.

The purpose of our proposal is to take advantage of the efficiency in the path discovery process offered by hierarchical PKIs, where trust relationships are unidirectional and paths are easy to find. For that reason, our protocol establishes a virtual

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 485 – 496, 2006.

hierarchy among the Certification Authorities (CAs) of a peer-to-peer PKI, contributing to simplify the path discovery process. In a hierarchical model, there is only one path between two entities so the verifier must carry out less search operations to discover a path than in a peer-to-peer architecture. In addition, our protocol is adaptable to the characteristics of the mobile users with limited processing and storage capacities, since it establishes a maximum path length that can be set in accordance with the features of the users' terminals.

Section 2 defines certification path and describes the certification path validation process. Also, it compares the characteristics of hierarchical and peer-to-peer PKIs. In section 3, we explain how our protocol creates a virtual hierarchy in a peer-to-peer PKI. Section 4 contains a practical example of our protocol and determines the time required to execute the protocol in this example. Finally, section 5 concludes.

2 PKI Trust Models. Processing Certification Paths

2.1 Certification Path Validation Process

PKI [1] uses Trust Third Parties (TTPs), known as certification authorities, to digitally sign Public Key Certificates (PKCs), ensuring that a particular public key belongs to a certain user. Therefore, certificates, and the keys they contain, give the communicating parties information about the owner of the certificate and the authority that issued it.

A CA's *certification domain* defines the organizational or geographical boundaries within which the CA is considered trustworthy. Thus, all the PKI users in a CA's certification domain consider this authority like their trust anchor.

A *trust anchor* is a certification authority that a PKI user explicitly trusts under all circumstances and is used by the client application as the starting point for all certificate validation. Each user receives the public key of its trust anchor when it is registered in the PKI.

When two users belong to the same certification domain and they want to communicate each other, one can obtain easily the other's public key, since they know the public key of their trust anchor. But when the users belong to different certification domains, their communication is only possible if there is an uninterrupted chain of trust points between them, which supposes the intervention of several CAs and therefore the necessity of an agreement among their policies. Cross certification allows CAs to build these trust chains from an entity to another.

Cross certification is the establishment of a trust relationship between two certification authorities through a certificate signed by a CA and that contains the public key of another CA, referred to as *cross certificate*.

Therefore, a *certification path* [2] is a chain of public key certificates through which an entity can obtain the public key of another entity.

The primary goal of a path validation process is to verify the binding between a subject and a public key. Then, the verifier must check the signature and validity of each certificate in the path in order to trust the public key of the target entity. Thus, the path is traced from the verifier's trust anchor to the CA public key required to validate the target entity's certificate and the certification path length is equal to the number of CAs in the path plus one: a certificate for each CA and the target entity's certificate.

In general, a path validation process involves the following steps:

- *Discovering a Certification Path:* It is to build a trusted path between the verifier's trust anchor and the target entity based on the trust relationship among the CAs of the PKI. A certification path can be built in the *forward* direction, that is, from the target entity to a trust anchor, or in the *reverse* direction, that is, from a trust anchor to the target entity[3].
- *Retrieving the Certificates:* It is to retrieve each certificate in the path from the place(s) where they are stored. In today's enterprise, it is common practice to post (or publish) certificates and certificate revocation information (particularly revocation information based on CRLs) to a repository. A *repository* is a generic term used to denote any logically centralized database capable of storing information and disseminating that information when requested to do so [4].
- *Verifying the Digital Signatures*: It is to verify the validity of the digital signature of each certificate in the path. It involves:
  1. Decrypting the signed part of the certificate with its issuer's public key.
  2. Calculating a hash of the certificate's content.
  3. Comparing the results of 1 and 2. If they are the same then the signature is valid.
- *Verifying the Validity of the Certificates:* It is to determine if the certificates have expired or have been revoked. The certificate validity period is used to verify the expiration, while the revocation status depends on the revocation mechanism used. Certificate revocation is the mechanism under which an issuer can revoke the binding of an identity with a public-key before the expiration of the corresponding certificate. It is possible to use periodic publication mechanisms such as Certificate Revocation Lists (CRLs)[1], or on-line query mechanisms such as the Online Certificate Status Protocol (OCSP) [5].

2.2 Hierarchical and Peer-to-Peer Trust Models

Certification architectures or trust models provide a technological framework for creating and managing trust relationships among the different entities of a PKI. Thus, certification architectures describe how the trust relationships and the necessary rules to find and to cross the certification paths are built. The most popular PKI trust models are hierarchical and peer-to-peer (see [6], [7])

2.2.1 Hierarchical Model

This is the most common model. In this configuration, all users trust the same root CA (RCA). That is, all the users of a hierarchical PKI begin their certification paths with the RCA's public key. In general, the root CA does not issue certificates to users but only issues certificates to subordinate CAs. Each subordinate CA may issue certificates to users or another level of subordinate CAs, if it is permitted by the certification policies. In a hierarchical PKI, the trust relationships are unidirectional, that is, subordinate CAs do not issue certificates to their superior CAs (Fig. 1).

Hierarchical PKIs are scalable. Certification paths are easy to develop because trust relationships are unidirectional and the longest path is equal to the depth of the tree less one, because RCA's certificate is not part of the path since it is known by all entities in the architecture.

The drawbacks of the hierarchical model result from the reliance on a single trust point. The compromise of the RCA's private key results in a compromise of the entire PKI. In addition, transition from a set of isolated CAs to a hierarchical PKI may be logistically impractical because all users must adjust their trust points.

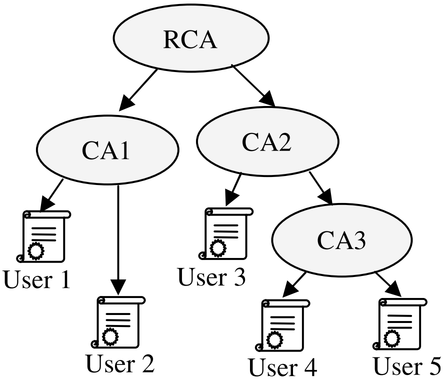


Fig. 1. Hierarchical model

2.2.2 Peer-to-Peer Model

It is also known as cross-certificate architecture or mesh model. Here, the user's trust anchor is its local CA and all the CAs can be trust points because they are autonomous. Autonomy refers to the fact that the CA does not rely on a superior CA in a hierarchy. An autonomous CA can perform peer-to-peer cross-certification with other autonomous CAs. Thus, a pair of certificates describes their bidirectional trust relationship (Fig. 2). However, the trust relationship may not be unconditional. If a CA wants to limit the trust, it must specify these limitations in the certificates issued to its peers. All certificate validation, by clients within an autonomous CA, starts with the local CA's self-signed certificate.

Peer-to-peer PKI can easily incorporate a new community of users and although the management cost is high, there is not a single point of failure since it counts on different trust points and they can have multiple paths between two users. In addition, a peer-to-peer PKI can easily be constructed from a set of isolated CAs because the users do not need to change their trust points. This model serves to represent the dynamic changes of the organizational structures or environments where communicating entities are not related hierarchically.

The drawback of this model is that the number of trust relationships is directly proportional to the number of CAs (n), that is, the number of trust relationships is equal to n*(n – 1), what causes scalability problems. The maximum length of a certification path in a mesh PKI is the number of CAs in the infrastructure.

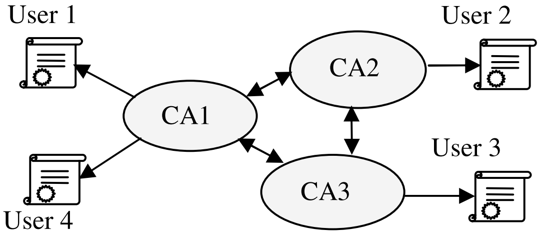


Fig. 2. Peer-to-peer model

Table 1 compares the characteristics of hierarchical and peer-to-peer trust models.

Table 1. Hierarchical model vs. peer-to-peer model

Characteristic	Hierarchical	Peer-to-Peer
Trust Anchor	Root CA	Local CA
Trust Relationships	Unidirectional	Bidirectional
Scalability	Yes	No
Number of paths between two entities	One	Multiple
Path Discovery	Easy	Difficult
Longest Path	Depth of the tree less one	Number of CAs in the infrastructure

3 Protocol Description

In this section, we describe how our protocol establishes a virtual hierarchy in a peer-to-peer PKI, based on the trustworthiness level of the participant CAs. The hierarchy is built from the leaves to the root (upwards). This protocol facilitates the certification path discovery process and can be adapted to users with limited capacities.

Some aspects of our protocol are inspired on the algorithm proposed by J. Hernandez-Serrano et al in [8], although the application area is different. Table 2 shows the notation used in this paper.

We divide our protocol in two phases to understand it better.

- *Trustworthiness order among CAs:* In this phase, the neighboring CAs are arranged from the less trustworthy to the most trustworthy.
- *Construction of the hierarchy:* In this phase, it is established a hierarchical trust relationship among the CAs of the peer-to-peer PKI.

3.1 Trustworthiness Order Among CAs

The protocol begins when an authority CA_0 declares to its neighbors (CAs that issued a certificate to CA_0 and CAs that CA_0 issued a certificate) that it wants to establish a hierarchical trust relationship with them. In addition, CA_0 propose a maximum certification path length (L_{MAX}) based on the processing and storage capacity of its users.

Thus, CA_0 sends a request message to its neighbors containing the value of L_{MAX}. These messages and all the messages sent among CAs along the protocol must be authenticated by the receiver.

Table 2. Notation

Notation	Meaning
L_{MAX}	Maximum path length allowed
CA_i	Certification authority i
L_i	Number of certificates from the leaves to the authority i
IN_i	Number of CAs which CA_i trusts (received certificates)
OUT_i	Number of CAs that trust CA_i (issued certificates)
CA_0	Current authority

Each neighbor can accept or refuse to collaborate in the establishment of that hierarchy, sending to the demanding authority an acceptance or rejection message.

Once authority CA_0 receives the responses from all its neighbors, it determines the number of CAs that want to be part of the hierarchy and issued a certificate to CA_0 (IN_0), and the number of CAs that want to participate in the hierarchy and received a certificate from CA_0 (OUT_0). Then, CA_0 sends these values to its participant neighbors in an information message and these neighbors send to CA_0 their own parameters IN_i and OUT_i. We assume that there is a secure system by which each authority always sends truthful information to its neighbors.

Later, CA_0 compares OUT_0 with the received OUT_i values and puts them in order from the lowest to the highest. The authority with the lowest OUT_i is the less trustworthy, that is, the neighbor that less the other participants trust. If there are two or more CAs with the same OUT_i, they are arranged in accordance with the IN_i value from the lowest to the highest too. For the sake of simplicity, we have not considered other parameters to put in order the CAs such as existing policy mapping or distance between authorities, but these can be considered if parameters OUT_i and IN_i are the same for two or more CAs. Thus, each authority put in order its neighbors, from the less trustworthy to the most trustworthy, determining which of its neighbors are less trustworthy and more trustworthy than itself. At the beginning of the protocol, L_i=0 for all the CAs.

3.2 Construction of the Hierarchy

In this phase of the protocol, the CAs act from the less trustworthy to the most trustworthy in accordance with the order established at the first phase. Therefore, the less trustworthy authority in the neighborhood acts first and the other CAs must wait for the intervention of their less trustworthy neighbors.

The objective of the second phase is that each authority chooses a superior CA among the participant neighbors that issued it a certificate (trusted neighbors). Thus, when an authority CA_0 acts, it looks for the most trustworthy authority of its trusted neighbors, based on the trustworthiness order established at the first phase of the protocol, and chooses this neighbor like superior CA. If L_0 is higher than L_i of superior CA and (L_0 + 1) is less than or equal to (L_{MAX} – 1), L_i of superior CA takes the value of (L_0 +1). In case that (L_0 +1) is higher than (L_{MAX} – 1), the chosen superior CA is not appropriate and CA_0 must choose the next trusted neighbor like superior CA provided that this neighbor is more trustworthy than CA_0. CA_0 checks again if L_0 is higher than L_i of the new superior CA and so on until CA_0 finds a suitable superior CA. Nevertheless, it can be possible that none of the trusted neighbors that are more trustworthy than CA_0 can be used like superior CA. Thus, when CA_0 concludes this procedure, it sends an association message to its neighbors informing the identity of its superior CA or a failure message if it was not possible to choose a superior CA.

Later, the following less trustworthy authority in the neighborhood, according to the order established in the first phase, repeats the procedure and so on until all CAs act, except for the most trustworthy authority that must not carry out this procedure because there is not a neighbor more trustworthy than it. Instead of that, this authority sends a root_CA message to its neighbors.

The authorities that did not choose a superior CA in this phase of the protocol, including the most trustworthy authority, are considered root CAs. If there are more than one root CA at the end of the second phase, the protocol must be repeated with the resulting root CAs, considering only the certificates issued among them to determine the new value of *OUTi* and IN_i parameters. L_i maintain the value that they obtained during protocol execution. In addition, when the protocol is repeated, the value of L_i can be less than or equal to L_{MAX} in the second phase, instead of (L_{MAX} – 1).

Even so, hierarchy can have more than one root CA after the repetition of the protocol. In this case, root CAs must find the shortest path among them using an alternative method.

Root CAs send their public key to all the authorities below them in a root_CERT message, at the end of the protocol execution.

4 Practical Example

Fig. 3 shows a mobile ad-hoc network with 10 nodes. Arrows represent the certificates issued from one node to another.

At the first round, node 1 wants to carry out the protocol, so it sends a request message to its neighbors (2, 3, 4, 5 and 7) and proposes a maximum path length L_{MAX}=3.

We define a *round* as the set of messages that are sent or received at the same time slot. Round time will depend on processing time, network speed, latency, etc.

In the second round, if node 2 wants to collaborate with node 1, it sends an acceptance message to 1 and a request message containing the value of L_{MAX} to its other neighbors (3, 4, 5, 6, 9 and 10).

For the sake of simplicity, we suppose that all nodes want to be part of the hierarchy. Thus, at the same round, node 3 sends an acceptance message to node 1 and a request message to nodes 2, 4, 6, 7, 9 and 10. Node 4 sends an acceptance message to node 1, and a request message to nodes 2, 3, 7, 8, 9 and 10. Node 5 sends an acceptance message to node 1, and a request message to nodes 2, 6, 7, 9 and 10. And node 7 sends an acceptance message to node 1, and a request message to nodes 3, 4, 5, 6, 8 and 9.

In the third round, node 6 sends an acceptance message to nodes 2, 3, 5 and 7, and a request message to nodes 9 and 10. Node 8 sends an acceptance message to nodes 4 and 7, and a request message to node 9. Node 9 sends an acceptance message to nodes 2, 3, 4, 5 and 7, and a request message to nodes 6, 8 and 10. Node 10 sends an acceptance message to nodes 2, 3, 4 and 5, and a request message to nodes 6 and 9. On the other hand, node 2 sends an acceptance message to nodes 3, 4 and 5; node 3 sends an acceptance message to nodes 2, 4 and 7; node 4 sends an acceptance message to nodes 2, 3 and 7; node 5 sends an acceptance message to nodes 2 and 7; and node 7 sends an acceptance message to nodes 3, 4 and 5.

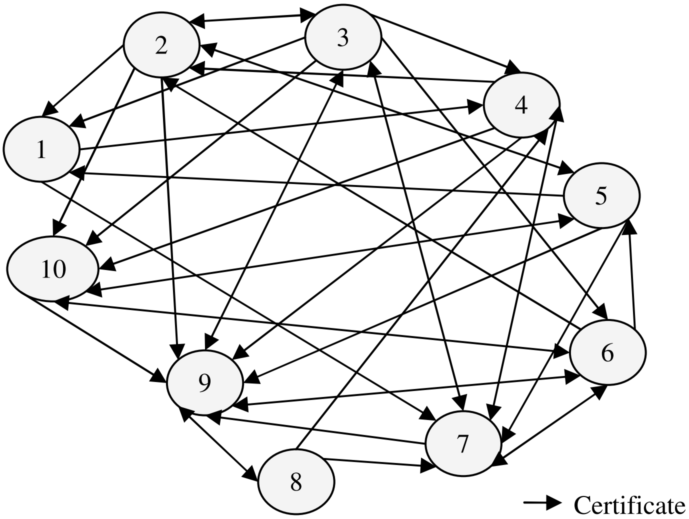


Fig. 3. Mobile ad-hoc network

In the fourth round, node 6 sends an acceptance message to nodes 9 and 10; node 8 sends an acceptance message to node 9; node 9 sends an acceptance message to nodes 6, 8 and 10; and node 10 sends an acceptance message to nodes 6 and 9.

Until now, 84 messages have been sent: 42 request messages and 42 acceptance messages.

When nodes receive response to its request messages, they must determine their OUT_i and IN_i values and send them in an information message to their neighbors in the fifth round. Altogether, they are 62 information messages. Table 3 shows the parameters of each node.

Table 3. Parameters of the nodes

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\boldsymbol{OUT_i}$	2	5	7	4	5	5	3	3	4	3
$\boldsymbol{IN_i}$	3	4	3	4	3	4	7	1	7	5

Once each node obtains the parameters of its neighbors, puts them in order from the less trustworthy to the most trustworthy. Thus, for node 1 the trustworthiness order is: **1**, 7, 4, 5, 2, 3; for node 2 is: 1, 10, 4, 9, 5, **2**, 6, 3; for node 3 is: 1, 10, 7, 4, 9, 2, 6, **3**; for node 4 is: 1, 8, 10, 7, **4**, 9, 2, 3; for node 5 is: 1, 10, 7, 9, **5**, 2, 6; for node 6 is: 10, 7, 9, 5, 2, **6**, 3; for node 7 is: 1, 8, **7**, 4, 9, 5, 6, 3; for node 8 is: **8**, 7, 4, 9; for node 9 is: 8, 10, 7, 4, **9**, 5, 2, 6, 3; for node 10 is: **10**, 4, 9, 5, 2, 6, 3.

According to this order, nodes 1, 8 and 10 act first in the second phase, then node 7, next node 4, later node 9, after that node 5, then node 2, next node 6 and finally node 3 is the most trustworthy.

Node 1, among its trusted neighbors (2, 3 and 5), chooses 3 like superior CA, because it has the highest OUT_i. Since, L_1=L_3 and (L_1+1) < (L_{MAX}-1), L_3=L_1+1=1. Thus, node 3 is a suitable superior CA for 1 and it sends an association message to its neighbors (2, 3, 4, 5 and 7) in the sixth round. At the same round, node 8 chooses 9 like superior CA, so L_9=L_8+1=1, and node 10 chooses 3. These nodes also send association messages to their neighbors.

In the seventh round, node 7 chooses 3 like superior CA. Since L_7<L_3 and L_3< (L_{MAX}-1), the choice of node 7 is appropriate.

Node 4 trusts nodes 1, 3, 7 and 8, and node 3 is the most trustworthy. Therefore, this is chosen like superior CA by node 4 that sends an association message to its neighbors (1, 2, 3, 7, 8, 9 and 10) in the eighth round.

Now, node 9 chooses a superior CA. Among its trusted neighbors (2, 3, 4, 5, 6 and 10), node 3 is the most trustworthy, L_9=L_3 and (L_9+1) = (L_{MAX}-1), so this node is a suitable superior CA for 9 and L_3=L_9+1=2. Thus, in the ninth round, node 9 informs its neighbors the identity of its superior CA through an association message.

In the tenth round, node 5 chooses 6 like superior CA and L_6=L_5+1=1. Next, in the eleventh round, node 2 chooses 3 like superior CA. And, in the twelfth round, node 6 chooses 3 like superior CA. Finally, in the thirteenth and fourteenth rounds, node 3 sends a root_CA message to its neighbors and a root_CERT message to the subordinate CAs (1, 2, 4, 5, 6, 7, 8, 9 and 10). Altogether, they are 71 messages: 55 association messages, 7 root_CA messages and 9 root_CERT messages. Fig 4 shows the established hierarchy, where node 3 is the root CA.

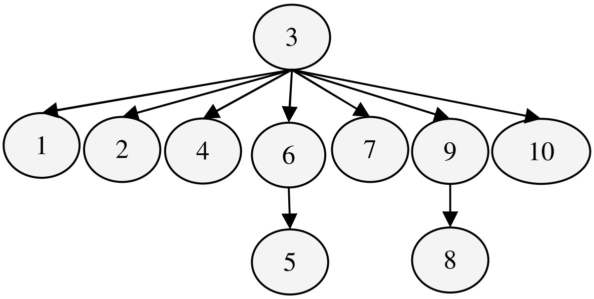


Fig. 4. Established hierarchy

4.1 Run Time of the Protocol

To calculate the time of a successful transmission in a WLAN 802.11b (11Mbps) using RTS/CTS scheme, we can use the values in Table 4 and equations (1), (2), (3), (4) and (5), that have been taken from [9]. In addition, the average real rate (C) for this type of network, when there is a high concurrence level is 2,739Mbps, according to tables in [10]. Therefore, if we assume that the maximum length of a message is 700 bytes (l=5600 bits) and a propagation delay δ=1μs, the time of a successful transmission T_S^{RTS} is 2,8ms. Thus, the time needed to carry out fourteen (14) rounds is 39,2ms.

On the other hand, we can calculate the time needed to sign the messages sent by each node and to verify the signature of the received messages. If we assume that each node is a laptop with Pentium 4 2,1GHz processor under Windows XP SP1, according to [11], the run time of a signature operation using a RSA-1024 algorithm is 4,75ms/operation, whereas the run time of a signature verification operation is 0,18ms/operation . Therefore, the time needed to sign and to verify the signature of the 14 messages is 69,02ms.

Finally, the total time needed to execute our protocol in this ad-hoc network is approximately 108,22ms. Nevertheless, this must be considered like a minimum time since it depends also on the response time of the nodes and the congestion of the network. On the other hand, the run time of signature and verification operations of limited devices such as PDAs and mobile phones is higher, so this will increase the run time of our protocol.

$$T_S^{RTS} = \mathrm{DIFS}+T_{RTS}+\mathrm{SIFS}+T_{CTS}+\mathrm{SIFS}+T_{header}+(l/C)+\mathrm{SIFS}+T_{ACK}+4\delta \quad (1)$$

$$T_{header} = (MAC_{hdr}/C) + (PHY_{hdr}/C_{control}) \quad (2)$$

$$T_{ACK}=l_{ACK}/Cc_{ontrol}; \quad (3)$$

$$T_{RTS}=l_{RTS}/Cc_{ontrol} \quad (4)$$

$$T_{CTS}=l_{CTS}/Cc_{ontrol} \quad (5)$$

Table 4. DSSS system parameters used in 802.11b standard

Parameter	**Value**
MAC header, MAC_{hdr}	272 bits
PHY header (long), PHY_{hdr}	192μs
RTS packet, l_{RTS}	160 bits + PHY_{hdr}
CTS packet, l_{CTS}	112 bits + PHY_{hdr}
ACK packet, l_{ACK}	112 bits + PHY_{hdr}
DIFS	50μs
SIFS	10μs
Control rate, $C_{control}$	2Mbit/s

5 Conclusions

Dynamism of mobile ad-hoc networks implies changing trust relationships among their nodes. Although peer-to-peer PKIs are quite dynamic and certification paths can be built where part of the infrastructure is temporarily unreachable, the discovery of certification paths is not an easy task since there can be multiple paths between two entities and all the options do not lead to the target entity. This is not the case of hierarchical PKIs, where there is only a path between two entities.

In this paper, we describe a protocol that establishes a virtual hierarchy in a peer-to-peer PKI, based on the trustworthiness of the participant CAs. The level of trustworthiness of each authority is determined in accordance with two parameters: the number of issued certificates (OUT_i) and the number of received certificates (IN_i).

An advantage of our protocol is that it does not establish new trust relationships among the CAs but it takes the existing relationships to establish the hierarchy. Thus, it is not necessary to issue new certificates or adjust the trust points.

The practical example of section 4 shows that our protocol can be carried out in a short time, what is a token of its efficiency. In addition, thanks to unidirectional trust relationships of the hierarchy, the verifier can discover easier and more rapidly the paths than in a peer-to-peer model.

Also, our protocol is adaptable to users with limited processing and storage capacities, since hierarchy is established considering a maximum certification path length (L_{MAX})

Our protocol not always finds a single root CA, what not implies that there is not a path among the authorities. For that reason, in those cases, we advise to use alternative methods to find the shortest path among the resulting root CAs.

Acknowledgements

This work has been supported by the Spanish Research Council under the projects ARPA (TIC2003-08184-C02-02) and SECONNET (TSI2005-07293-C02-01).

References

[1] ITU-T, "Recommendation X.509: Information Processing Systems - Open Systems Interconnection - The Directory: Authentication Framework (Technical Corrigendum)", International Telecommunication Union, 2000.

[2] R. Housley, W. Polk, W. Ford and D. Solo, "RFC3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", 2002.

[3] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman and S. Proctor, "Building Certification Paths: Forward vs. Reverse", *Network and Distributed System Security Symposium (NDSS 2001)*, 2001.

[4] C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*, Addison-Wesley, 2003.

[5] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, "RFC2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", 1999.

[6] W. T. Polk and N. E. Hastings, "Bridge Certification Authorities: Connecting B2B Public Key Infrastructures", NIST, 2000.

[7] R. Perlman, "An Overview of PKI Trust Models", *IEEE Network*, 1999, vol. 13, pp. 38-43.

[8] J. Hernandez-Serrano, J. Pegueroles and M. Soriano, "GKM over large MANET", *IEEE International Workshop on Self Assembling Wireless Networks (SAWN2005)*, 2005, pp. 484-490.

[9] P. Chatzimisios, A. C. Boucouvalas and V. Vitsas, "Optimisation of RTS/CTS Handshake in IEEE 802.11Wireless LANs for Maximum Performance", *IEEE Global Telecommunications Conference Workshops, 2004 (GlobeCom Workshops 2004)*, 2004, pp. 270-275.

[10] G. Anastasi, M. Conti and E. Gregori, "Chapter 3: IEEE 802.11 AD HOC Networks: Protocols, Performance, and Open Issues", in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, Eds.: Wiley-Interscience, 2004, pp. 94.

[11] W. Dai, "Crypto ++ 5.2.1 Benchmarks", Accessed at: 4/10/2004, http://www.eskimo.com/~weidai/benchmarks.

Key Exchange in 802.15.4 Networks and Its Performance Implications

Moazzam Khan, Fereshteh Amini, and Jelena Mišić

Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada
{umkhanm, amini, jmisic}@cs.umanitoba.ca

Abstract. The IEEE 802.15.4 specification is a recent low data rate wireless personal area network standard. While basic security services are provided for, there is a lack of more advanced techniques which are indispensable in modern personal area network applications. In addition, performance implications of those services are not known. In this paper, we describe a secure data exchange protocol based on the Zigbee specification and built on top of 802.15.4 link layer. This protocol includes a key exchange mechanism. Then, we evaluate the overhead of this scheme under different application scenarios. Initial results show the range of network and traffic parameters wherein the proposed scheme is feasible to use.

1 Introduction

The IEEE 802.15.4 specification outlines a class of wireless radios and protocols targeted at low power devices, personal area networks, and sensor devices. IEEE 802.15.4 specification employs a number of well-known security services that can be implemented but at the cost of memory and communication overhead. Currently, not many wireless sensor network overhead statistics are available when security is employed in such networks. Sensor network application developers and network administrators always need these overhead statistics in choosing the security option that best suites the security for a particular threat environment. For evaluating these security overheads on wireless sensor networks, we will simulate IEEE 802.15.4 media access control layer and try to implement secure data exchange once the devices exchange link keys with the PAN coordinator. We will try to measure the costs that are incurred after employing these security features under different inputs to wireless sensor network model.

For the remaining of this paper, we will give an overview of IEEE 802.15.4 specification in section 2 and later in section 3 we will introduce the security features addressed in IEEE 802.15.4. As IEEE 802.15.4 does not address any keying model, we are relying on keying model from Zigbee specification and will discuss about this in section 3.2. In section 4 we will explain the simulation model, how it is implemented and in the same section will present our results. Finally we will conclude our work in section 5.

2 IEEE 802.15.4

The need for low-cost, low-power and short-range communication is the main reason of introducing IEEE 802.15.4 Low Rate Wireless Personal Area Network (LR-WPAN)

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 497–508, 2006.

standard [1]. According to this specification, such WPAN consists of devices which are the basic components of these networks. Two or more devices communicating in a common physical channel create a WPAN.

Star topology is one option for communication in LR-WPAN. In this topology devices communicate via a single central controller called PAN coordinator. After deciding on a PAN identifier, PAN coordinator may decide whether a device can join the PAN.

In the current work we concentrate on beacon-enabled based communication. In this form of communication, devices first listen for the network beacon. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate point, the device transmits its data packet, using slotted CSMA-CA, to the coordinator (uplink). The coordinator acknowledges the successful reception of the data by transmitting an acknowledgment frame.

On the other hand, when the PAN coordinator has something to send to a device (downlink), it informs the device by including in the network beacon that a data message is pending. The device periodically listens to network beacon and, if a message is pending, transmits a request frame to the coordinator using slotted CSMA-CA. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. The pending data frame is then sent using slotted CSMA-CA. The device acknowledges the successful reception of the data by transmitting an acknowledgment frame.

3 Security in IEEE 802.15.4

IEEE 802.15.4 standard provides physical and link layer solutions for wireless personal area networks. It also provides well-known and well-understood cryptographic techniques [2,3] by supporting Authentication, Message integrity, Confidentiality and Freshness check for preventing replay attacks. Application of such security mechanisms comes at a cost that include processing overhead, memory overhead, power consumption and resulting low bandwidth [4]. In this paper, we will mainly focus on measuring the processing and communication overhead of secure IEEE 802.15.4 networks.

An application implemented using IEEE 802.15.4 has choice of different security suites that control the type of security protection by setting appropriate control parameters in the link layer security suite stack. A long Message Authentication Code (MAC) size improves the security feature of authentication and it is very difficult for an adversary to break or guess a MAC of longer size [5]. But this improved security is achieved at the cost of longer packet size. In IEEE 802.15.4 compliant wireless sensor networks, packet size is very crucial to the overall throughput that is required by the application. Applications that support continuous data flow would be affected more than the applications in which data flow is periodic. Applications used for real time monitoring of some critical environments rely on continuous flow of data and hence by implementing security will affect the overall throughput and lifetime of such network by increasing the packet size. For the current work we will employee the security suite specified in IEEE 802.15.4 that supports both encryption and data integrity with MAC size of 128 bits. The security suite uses Counter with CBC-MAC (CCM) [6] mode of AES (Advanced

Encryption Standard) for encryption and authentication. This cryptographic technique uses counter by first applying integrity protection both on message header and data payload and later it encrypts the data payload and MAC using AES. At the receiver end the receiver gets the packet; applies decryption using parameters based on sender's address from its Access Control List.

3.1 Security Building Blocks

The IEEE 802.15.4 specification provides basic security mechanisms but these security features can not work at their own. The level of security in any network revolves around the keys that are shared among devices. Different approaches have been suggested to distribute and manage these keys. While IEEE 802.15.4 does not suggest any keying mechanism hence in this paper we will follow the keying mechanism from Zigbee alliance specifications [3]. In this section we will first introduce the keying mechanisms and later explain how this is handled in Zigbee specification by taking advantage of the inherent security mechanisms already provided by IEEE 802.15.4.

Keying Model. As explained above, the IEEE 802.15.4 addresses good security mechanisms but it still does not address what type of keying mechanism will be used to employ above techniques.

Zigbee alliance [3] is an association of companies working together to enable wireless networked monitoring and control products based on IEEE 802.15.4 standard. After the acceptance of 802.15.4 as IEEE standard, Zigbee alliance is mainly focused on developing network and Application layer issues. Zigbee alliance is also working on Application Programming Interfaces (API) at network and link layer of IEEE 802.15.4. Alliance also introduces secure data transmission in wireless sensor network that are based on IEEE 802.15.4 specification but most of this work is in general theoretical descriptions of security protocol at network layer. There is no specific study or results published or mentioned by Zigbee alliance in regards to which security suite perform better in different application overheads. Zigbee alliance has also recommended both symmetric and asymmetric key exchange protocols for different networking layers. Asymmetric key exchange protocols that mainly rely on public key cryptography are computationally intensive and their feasibility in wireless sensor networks is only possible with devices that are resource rich both in computation and power.

Application support sub-layer of ZigBee specification provides the mechanism by which a Zigbee device may derive a shared secret key (Link Key) with another ZigBee device. Key establishment involves two entities, an initiator device and a responder device and is prefaced by a trust provisioning step. Trust information (e.g. MASTER key) provides a starting point for establishing a link key and can be provisioned in-band or out-band.

Zigbee alliance uses Symmetric-Key Key Establishment (SKKE) protocol for link key establishment. In SKKE an initiator device establishes a link key with a responder device using a master key. This master key, for example, may be pre-installed during manufacturing, may be installed by a trust center, or may be based on user-entered data (PIN, password). In current study we assume that all the devices and Pan coordinator have pre-installed Master keys and we will focus mainly on Link key establishment.

Keyed Hash Function for Message Authentication. A hash function is a way of creating a small digital fingerprint of any data. Cryptographic hash function is a one-way operation and there is no practical way to calculate a particular data input that will result in a desired hash value thus is difficult to forge. A practical motivation for constructing hash functions from block ciphers is that if an efficient implementation of block cipher is already available within a system (either in hardware or in software), then using it as the central component for a hash function may provide latter functionality at little additional cost. IEEE 802.15.4 protocol supports a well known block cipher AES and hence Zigbee Alliance specification also relied on AES. Zigbee alliance suggested the use of Matyas-Meyer-Oseas [7] as the cryptographic hash function that will be based on AES with a block size of 128 bits.

Mechanisms that provide integrity checks based on a secret key are usually called Message Authentication Codes (MACs). Typically, message authentication codes are used between two parties that share a secret key in order to authenticate information transmitted between these parties. Zigbee alliance specification suggest the keyed hash message authentication code (HMAC) as specified in the FIPS Pub 198 [8]. A Message Authentication code or MAC takes a message and a secret key and generates a $MACtag$, such that it is difficult for an attacker to generate a valid (message, tag) pair and are used to prevent attackers forging messages. In this paper, the calculation of $MacTag$ (i.e HMAC) of data $MacData$ under key $MacKey$ will be shown as follows

$$MacTag = MAC_{MacKey}MacData$$

3.2 Symmetric-Key Key Establishment Protocol (SKKE)

Key establishment involves two entities, an initiator device and a responder device, and is prefaced by a trust-provisioning step. Trust information (e.g., a master key) provides a starting point for establishing a link key and can be provisioned in-band or out-band. In the following explanation of the protocol we assume unique identifiers for initiator device's as U and for Responder Device (PAN Coordinator) as V. The master key shared among both devices is represented as $Mkey$.

We will divide Symmetric-Key Key Establishment Protocol (SKKE) between initiator and responder in following major steps.

Exchange of Ephemeral Data. Figure 1 illustrates the exchange of the ephemeral data where the initiator device U will generate the Challenge QEU. QEU is a statistically unique and unpredictable bit string of length $challengelen$ by either using a random or pseudorandom string for a challenge $Domain\ D$. The challenge domain D defines the minimum and maximum length of the Challenge.

$$D = (minchallengeLen, maxchallengeLen)$$

Initiator device U will send the Challenge QEU to responder device which upon receipt will validate the Challenge QEU by computing the bit-length of bit string Challenge QEU as $Challengelen$ and verify that

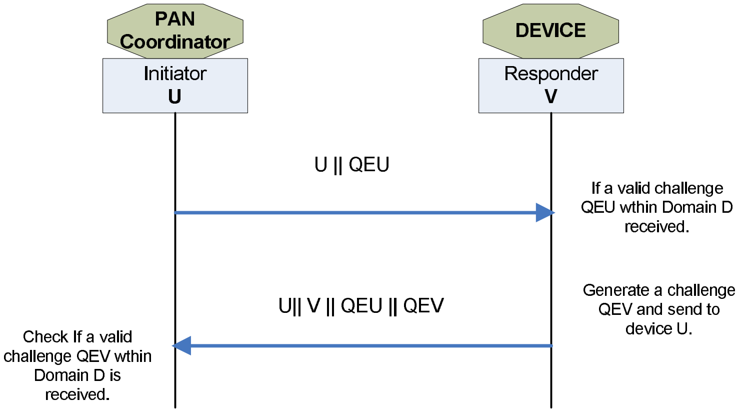

PAN Coordinator
DEVICE
Initiator
U
Responder
V
U || QEU
If a valid challenge QEU wthin Domain D received.
U|| V || QEU || QEV
Generate a challenge QEV and send to device U.
Check If a valid challenge QEV wthin Domain D is received.

Fig. 1. Exchange of ephemeral data

$$Challengelen \in [minchallengelen, maxchallengelen]$$

Once the validation is successful the Responder device will also generate a Challenge QEV and send it to initiator device U. The initiator will also validate the Challenge QEV as described above.

Generation of Shared Secret. Both parties involved in the protocol will generate a shared secret based on unique identifiers (i.e. distinguished names for each parties involved), symmetric master keys and Challenges received and owned by each party (Figure 2).

1. Each party will generate a $MACData$ by appending their identifiers and respective valid $Challenges$ together as follows

   $MACData = U||V||QEU||QEV$

2. Each party will calculate the $MACTag$ (i.e Keyed hash) for $MACData$ using $Mkey$ (Master Key for the device) as the key for keyed hash function as follows.

   $MACTag = MAC_{Mkey}MACData$

3. Now both parties involved have derived same secret Z
   (note: This is just a shared secret not the Link key. This Shared secret will be involved in deriving the link key but is not the link key itself.)

   $Z = MACTag$

Derivation of Link Key. Each party involved will generate two cryptographic hashes (this is not keyed hash) of the shared secret as described in ANSI X9.63-2001 [9].

$$Hash_1 = H(Z||01)$$
$$Hash_2 = H(Z||02)$$

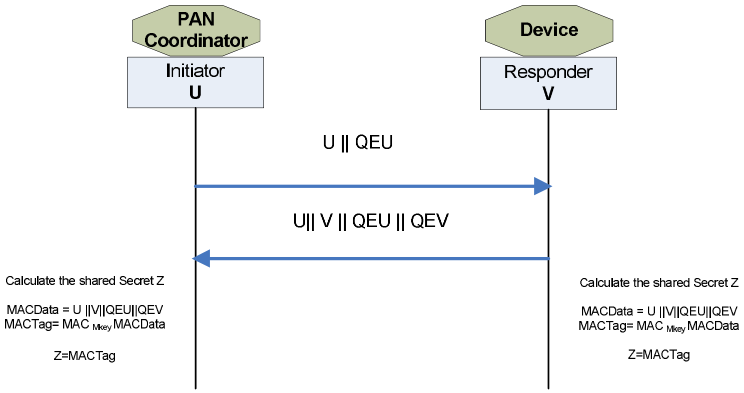


Fig. 2. Generation of shared Secret

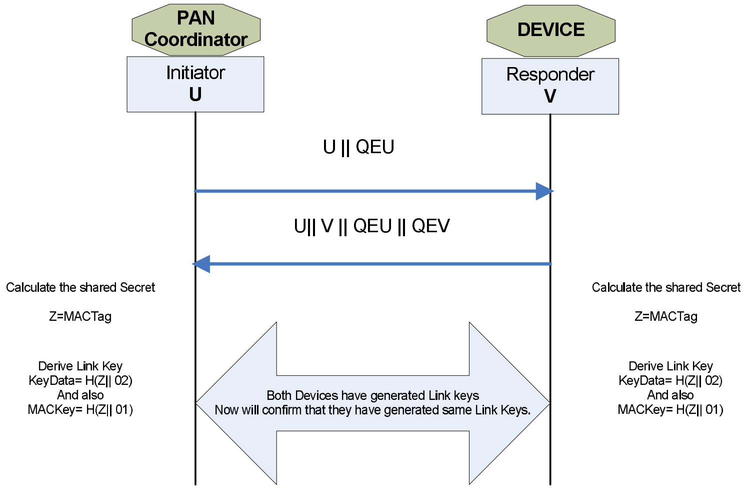


Fig. 3. Generation of Link Key

The hash value $Hash_2$ will be Link key among two devices (Figure 3). Now for confirming that both parties have reached on same Link key $(KeyData = Hash_2)$ we will use value $Hash_1$, as key for generating Keyed hash values for confirming stage of the protocol.

$$MACKey = Hash_1 \tag{1}$$

$$KeyData = Hash_2 \tag{2}$$

$$KKeyData = Hash_1 || Hash_2 \tag{3}$$

Confirming Link Key. Till this stage of protocol both parties are generating the same values and now they want to make sure that they reached on same Link key values but they do not want to exchange the actual key at all. For this they will once again rely on keyed hash functions and now both devices will generate different $MACTags$ based on different Data values but will use same key (i.e. $MACKey$) for generating the keyed hashes ($MACTags$).

1. Generation of MACTags
   Initiator and responder devices will first generate $MACData$ values and based on these values will generate $MACTags$. Initiator device D will receive the $MACTag_1$ from the responder device V and generate $MACTag_2$ and send to device V.

   We explain the generation of both $MACData$ values and $MACTags$ as follows

   First both devices will calculate $MACData$ values

   $$MACData_1 = 02_{16}||V||U||QEU||QEV$$
   $$MACData_2 = 03_{16}||V||U||QEU||QEV$$

   From the above $MACData$ values both devices will generate the $MACTags$ using the key $MACkey$ (Equation 1) as follows

   $$MacTag_1 = MAC_{MacKey}MacData_1$$
   $$MacTag_2 = MAC_{MacKey}MacData_2$$

2. Confirmation of MACTags
   Now the initiator device D will receive $MacTag_1$ from responder and Responder device V will receive $MACTag_2$ from device D and both will verify that the received $MACTags$ are equal to corresponding calculated $MACTags$ by each device. Now if this verification is successful each device knows that the other device has computed the correct link key (Figure 4).

3.3 Use of SKKE in Our Simulation Model

We have implemented SKKE in four major communication steps as are described in ZibBee specification [3] (Figure 5).

SKKE-1
Initiator U will send the Challenge QEU and wait for the Challenge QEV from responder V.

SKKE-2
Responder V will receive the Challenge QEU from initiator U, calculates its QEV and in the same data packet will send the $MacTag_1$.

SKKE-3
Initiator will verify the $MacTag_1$ and if it is verified successfully, will send its $MacTag_2$. Now the initiator has a Link key but will wait for an acknowledgment that its $MacTag_2$ has been validated by the Responder V.

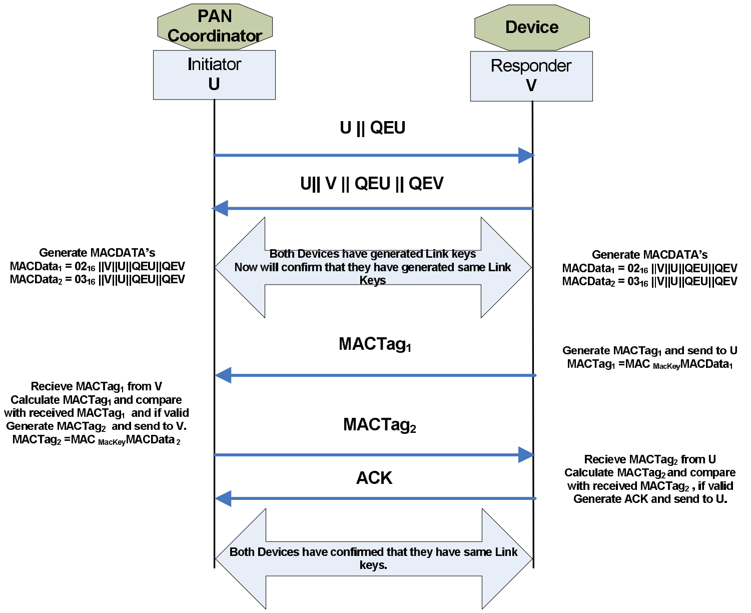

PAN Coordinator
Device
Initiator U
Responder V
U || QEU
U|| V || QEU || QEV
Generate MACDATA's
MACData$_1$ = 02$_{16}$ ||V||U||QEU||QEV
MACData$_2$ = 03$_{16}$ ||V||U||QEU||QEV
Both Devices have generated Link keys
Now will confirm that they have generated same Link Keys
Generate MACDATA's
MACData$_1$ = 02$_{16}$ ||V||U||QEU||QEV
MACData$_2$ = 03$_{16}$ ||V||U||QEU||QEV
MACTag$_1$
Generate MACTag$_1$ and send to U
MACTag$_1$ =MAC $_{MacKey}$MACData$_1$
Recieve MACTag$_1$ from V
Calculate MACTag$_1$ and compare
with received MACTag$_1$ and if valid
Generate MACTag$_2$ and send to V.
MACTag$_2$ =MAC $_{MacKey}$MACData $_2$
MACTag$_2$
ACK
Recieve MACTag$_2$ from U
Calculate MACTag$_2$ and compare
with received MACTag$_2$, if valid
Generate ACK and send to U.
Both Devices have confirmed that they have same Link keys.

Fig. 4. Confirmation of Link Keys

SKKE-4

Responder will receive and validate the $MacTag_2$ from the Initiator. If $MacTag_2$ validated successfully, the responder will send an acknowledgment and now both Initiator and Responder have Link keys. Once initiator receives this SKKE-4 message, keys establishment is complete and now regular secure communication can proceed using Link key among the initiator and the responder.

4 Simulation Model

We have simulated the key exchange mechanism using the IEEE 802.15.4 network using Artifex [10] a general development platform for discrete event simulations. For the remaining of this section we first give a quick introduction of beacon-enabled simulation model of 802.15.4 [11,12] and later explain the simulated key exchange process simulated in our current work.

4.1 Beacon-Enabled IEEE 802.15.4 Simulation Model

The network communication model of this simulation is based on star topology. The model is built on three primary objects : PAN coordinator, Device and Medium. The device and PAN coordinator objects are inter-connected via medium object in our simulation model.

Two different Token types are defined that play the role of packet and backoff. Packets can be any of beacon, MAC request, data and acknowledgment (ack) types. The communication is initiated when PAN coordinator first sends beacon to medium (beacons are sent after every 48t where t is duration of one backoff period). After receiving the beacon the medium starts a clock and sends pulse to all devices every t time.

Data packets are generated by device object following exponential distribution and are destined to a randomly chosen device. The packet is then sent to the medium and a copy of it is kept for retransmission if needed. Data packets are then received by the medium. If the number of received packets in medium is greater than 1, collision occurs. If there are no collisions, data packets are sent successfully to the PAN coordinator and the medium status is set to busy.

PAN coordinator is the next stop for data packets and is responsible for sending ack type packets to corresponding device after a specified delay. As of every packet, ack will be received first by the medium and then sent to corresponding device. When PAN coordinator is sending data to a device it keeps finite buffer for each device in the PAN. If the buffer of the device which the data packet is destined for is full, the packet will be discarded. In the case that there is still room in that device's buffer, the coordinator adds the destination ID of packet to the pending devices list and advertises the ID in the beacon. The device will notice that there is packet waiting for it and will initiate a MAC request packet to be sent to the coordinator. The PAN coordinator after receiving the request will perform round robin scheduling algorithm and choose the device to send the packet from its corresponding downlink buffer.

4.2 Adding Key Exchange Mechanism to the Simulation Model of IEEE 802.15.4 Network

In this section we describe the communication between the ordinary nodes and PAN coordinator which occurs as result from the link key exchange. We assume that devices are attached to the cluster and the formation of the piconet is finalized. Also, we assume the master keys are established, so that there is no threat of eavesdropping during exchange of master keys. The next step is generating link keys between each device

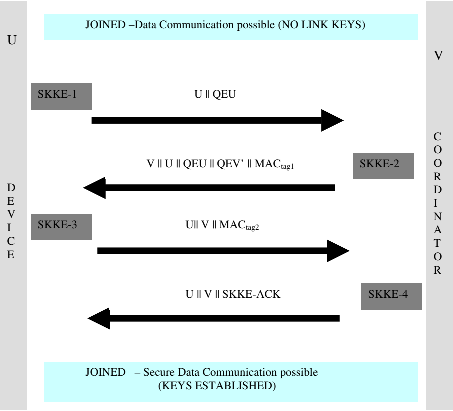

U
DEVICE
V
COORDINATOR
JOINED –Data Communication possible (NO LINK KEYS)
SKKE-1
U || QEU
SKKE-2
V || U || QEU || QEV' || MAC$_{tag1}$
SKKE-3
U|| V || MAC$_{tag2}$
SKKE-4
U || V || SKKE-ACK
JOINED – Secure Data Communication possible
(KEYS ESTABLISHED)

Fig. 5. SKKE protocol

and PAN coordinator. For the exchange of link keys, we will follow SKKE protocol as describe in Section 3.3.

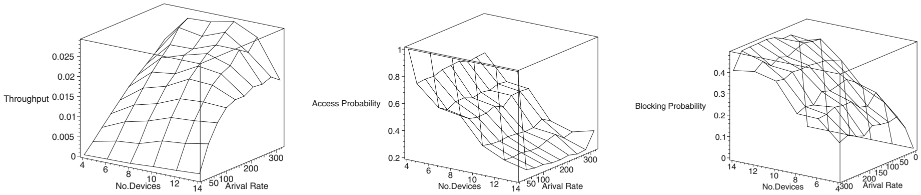
The process of key generation starts by PAN coordinator's advertisement for the first phase of key generation packets. Depending on which stage of generation we are in, the corresponding SKKE type of data packet (ranging from 1 to 4) will be processed (e.g the first data packet has the type of SKKE-1 and so on). According to the standard specification at most 7 devices can be advertised in each beacon. Therefore the PAN coordinator will advertise 7 devices in each beacon. According to the standard, each device listens to each beacon and if its ID has being advertised the device will send a request packet. Request packet is transmitted in CSMA-CA mode and can collide with other packets. If it is received successfully by the PAN coordinator it will be acknowledged and downlink packet transmission carrying the SKKE protocol data will follow in the downlink transmission.

In our model, key exchange packets have non-preemptive priority over data packets. If the node has started backoff process for data packet and it hears its ID in the beacon it will finish the current packet transmission before sending the request packet. However, if data packet arrives to the device's buffer while the key exchange is going on, its transmission will be postponed until device receives the new link key. PAN-Coordinator will first check key for the destination device from its access control list and no packet will be sent to the specific destination until the corresponding link key is already exchanged between PAN coordinator and the node. From this point on regular secure data packets will be immediately send to the destination.

4.3 Simulation Run and Analysis

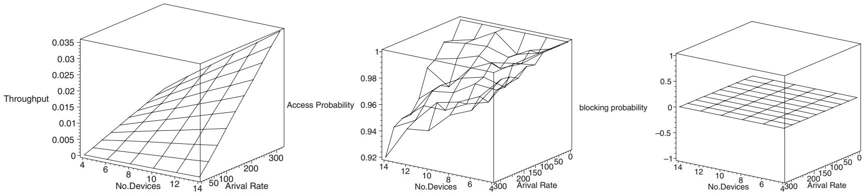
We have implemented the physical, data link and security layer of an IEEE 802.15.4 cluster operating in beacon enabled, slotted CSMA-CA mode. The packet size without security overheads includes all physical layer and Medium Access control layer headers, and it is set to 30 bytes i.e. to three backoff periods. When packet signature (message authentication code) of 16 bytes is added to the total packet size had to be rounded to 5 backoff periods (the largest packet size could be set to 13 backoff periods).

The cluster under consideration contains 14 devices, each having buffer capacity for three packets. Packet arrival per device followed the Poisson process with average rate of 90.5 packets per minute. When the coordinator announces key exchange in the beacon, all nodes had to temporarily stop uplink data transmissions until they receive new key initialization values from the coordinator in the downlink packets. Due to complex downlink data-link transmission algorithm we expected that key exchanges will adversely affect the regular sensing traffic. we considered the impact of the increase of packet size due to addition of Message Authentication Code, increased processing time needed for encryption in AES with CBC-MAC, and key exchange between the nodes over various packet arrival rates and cluster sizes. Figure 6 presents throughput, access probability (probability of no packet collision) and blocking probability at the node's buffer when all security overhead is included. Results were taken for varying number of nodes and varying packet arrival rate per node. Figure 7 presents the same parameters (except the key exchange cost since it does not exist) when no security measures are deployed in the network. We observe that without security measures, blocking probability is equal to zero i.e. that network works without losses.



(a) Throughput (b) Access probability (c) Blocking Probability

Fig. 6. Throughput, Access Probability and Blocking Probability as the function of simulation time (backoffs) for the case when security is employed and all devices stop their communications to update their keys



(a) Throughput (b) Access probability (c) Blocking probability

Fig. 7. Throughput, Access Probability and Blocking Probability as the function of simulation time(backoffs) when no security technique is employeed

5 Conclusion and Future Work

We have studied and simulated the key exchange process in IEEE 802.15.4 on top of simulation model of this network and the results confirm our expectations. Data encryption is provided by exchanging link keys between each device and PAN coordinator. The signature payload plays a big role on performance of the network. Also we have observed that the total access delay is higher when encryption and decryption is provided.

For the future works we will measure more realistically the performance of secure IEEE 802.15.4 personal area network. Calculation of timeout for SKKE packets should be less than usual packets because device needs to throw error if it does not receive response quickly. The SKKE packets should be processed and selected in a priority queue at PAN Coordinator. Also different key exchange protocols need to be studied and if they are suitable for this kind of networks, their performance will be compared to the SKKE protocol.

References

1. Standard for part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPAN). IEEE Std 802.15.4, IEEE (2003)
2. Stallings, W.: Cryptography and Network Security: Principles and Practice. Prentice Hall, Upper Saddle River (2003)

3. ZigBee specification (ZigBee document 053474r06, version 1.0). ZigBee Alliance (2004)
4. Sastry, N., Wagner, D.: Security considerations for IEEE 802.15.4 networks. In: WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security. (2004) 32–42
5. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Computer and System Sciences **61** (2000) 362–399
6. Whiting, D., Housley, R., Ferguson, N.: Counter with cbc-mac (CCM). http://www.rfc-archive.org/getrfc.php?rfc=3610 (2003)
7. Menezes, A., Oorschot, P.V., Vanstone, S.: Handbook of Applied Cryptography. CRC Press (1997)
8. FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication 198,US Department of Commerce/N.I.S.T. (2002)
9. ANSI X9.63-2001,Public Key Cryptography for the Financial Services Industry- Key Agreement and Key Transport Using Elliptic Curve Cryptography. American Bankers Association (2001)
10. Inc., R.D.: Artifex v.4.4.2 (2003)
11. Shafi, S.: Performance of a beacon enabled IEEE 802.15.4-compliant network. Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada (2005)
12. Mišić, J., Shafi, S., Mišić, V.B.: Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic. IEEE Transactions on Parallel and Distributed Systems **17** (2006) 1–16

An Authenticated Key Agreement Protocol for Mobile Ad Hoc Networks*

Xukai Zou[1], Amandeep Thukral[1], and Byrav Ramamurthy[2]

[1] Indiana University, Purdue University, Indianapolis, IN 46202, USA
{xkzou, athukral}@cs.iupui.edu
http://www.cs.iupui.edu/~xkzou/
[2] University of Nebraska-Lincoln, Lincoln, NE 68588, USA
byrav@cse.unl.edu
http://csce.unl.edu/~byrav/

Abstract. The growing popularity of wireless ad hoc networks has brought increasing attention to many security issues for such networks. A lot of research has been carried out in the areas of authentication and key management for such networks. However, due to lack of existing standards for such networks, most of the proposed schemes are based on different assumptions and are applicable only in specific environments. Recently Balachandran et al. proposed CRTDH [1], a novel key agreement scheme for group communications in wireless ad hoc networks. The protocol has many desirable properties such as efficient computation of group key and support for high dynamics. However, the protocol does not discuss mutual authentication among the nodes and hence, suffers from two kinds of attacks: man-in-the-middle attack and Least Common Multiple (LCM) attack. This paper identifies the problems with the current CRTDH scheme and discusses these attacks. AUTH-CRTDH, a modified key agreement protocol with authentication capability, is also presented. Results from extensive experiments that were run on the proposed protocol and some other key agreement protocols including CRTDH are also discussed. It can be observed from the experiments that the new scheme is comparable with the CRTDH scheme and better than many other non-authenticated schemes in terms of performance.

Keywords: Network security, Chinese remainder theorem, Key management, Mobile ad hoc networks, Secure group communication.

1 Introduction

The recent developments in wireless networks, in particular IEEE 802.11 networks, have revolutionized the way people use computers and networks. Wireless networks offer convenience and ease of use and hence find applications in numerous fields such as business, home and military. Mobile ad hoc networks (MANETs) are a special type of wireless networks consisting of a set of autonomous mobile nodes that form a temporary infrastructure-free network to

* This work was partially supported by the U.S. NSF grant CCR-0311577.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 509–520, 2006.

carry out basic networking functions. Ad hoc networks offer a convenient mode of communication over a shared wireless medium.

However, the advantages attributed to such wireless networks come with an associated cost of complicating endpoint management and security. Wired networks usually employ a trusted central authority, which provides the security services such as authentication, key management and authorization. Ad hoc networks operate in an infrastructureless setting and the communication links in such networks use an open shared medium. Hence they are vulnerable to various kinds of active and passive attacks [2]. In such a setting, it is untenable to assume the presence of an online server to provide the above mentioned security services. As a result, the members provide for these services themselves.

The security requirements of ad hoc networks, when considered at a high level are identical to that of wired networks [3]. These include availability, confidentiality, integrity, authentication and non-repudiation. Achieving these security requirements in wireless ad hoc networks becomes more challenging due to many reasons, which include limited computation capabilities, inherent mobility, and shared access medium [4].

Security for ad hoc networks has been a field of active research in recent years. The primary focus of the previous research had been on securing ad hoc routing protocols. Recently much interest has been shown in authentication and key management issues for such networks [5,3,6,7,8,9,10]. However, due to the lack of existing standards for general ad hoc networks, all the proposed protocols are based on different assumptions and security requirements. As a result, most of them are applicable in certain specific environments only. Moreover, most of the solutions proposed in literature are not comprehensive in nature i.e. the key management schemes avoid the issue of authentication and most authentication protocols do not discuss key management.

Recently Balachandran et al. [1] proposed CRTDH, a novel and efficient key agreement scheme for wireless ad hoc networks. The protocol uses Chinese Remainder Theorem and Diffie-Hellman key exchange to obtain a contributory key agreement scheme for secure group communication. The proposed scheme achieves key agreement in an efficient manner and has additional advantage of not having the requirement for member serialization. However, the protocol does not consider mutual authentication among nodes and hence, is vulnerable to two kinds of attacks, namely, the man-in-the-middle attack and the least common multiple (LCM) attack. In this paper the security of the CRTDH scheme is analyzed along with a discussion on these attacks. AUTH-CRTDH, a new key agreement scheme for secure group communication (SGC) with authentication capability is also proposed. The scheme possesses the desirable characteristics of the original scheme including good support for user dynamics and also provides authentication capability in an efficient way. Extensive experiments were performed on the proposed scheme, original CRTDH scheme and some other typical key agreement protocols. The proposed protocol is comparable to CRTDH in terms of performance and better than many other non-authenticated protocols.

The rest of the paper is organized as follows. The related work along with a description of CRTDH and the corresponding attacks are presented in Section 2. Section 3 describes the proposed protocol and the experimental results along with a brief discussion are presented in Section 4. Section 5 presents our conclusions.

2 Related Work

2.1 Existing Key Agreement Protocols

A lot of research has been done in the area of key management schemes for secure group communication and several contributory key agreement schemes have been proposed in literature. The initial attempt to extend the two party Diffie Hellman (DH) key exchange to group communication was done by Ingemarsson et al. [11], and the proposed scheme is known as the ING protocol. The protocol executes in $n-1$ rounds and requires that all the members be arranged in a logical ring. The protocol has the advantage that it does not have a group controller (GC), but it suffers from high communication overhead.

Steiner et al. proposed an elegant extension to the two party Diffie Hellman (DH) exchange for dynamic peer groups called Group Diffie Hellman (GDH) [12,13,14]. Three closely related protocols (GDH.1,2 and 3) were discussed in the study. These protocols achieve contributory key agreement even though the computation load is not equally distributed among the different members. The protocols also require the members to be serialized or structured in a particular order and the information is sent from one node to another in a serial fashion.

Another extension to the DH protocol was proposed by Steer et al. in [15]. This scheme also requires members to be serialized as in the GDH schemes. None of the above mentioned schemes discuss mutual authentication among users in the respective protocols.

2.2 Description of CRTDH

In 2005, Balachandran et al. proposed CRTDH [1], an efficient key agreement scheme for wireless ad hoc networks. The scheme uses the Chinese Remainder Theorem and Diffie-Hellman key exchange to achieve totally distributed key agreement for secure group communication. The key agreement procedure in the CRTDH protocol is discussed below.

Assume n users $\{U_1, U_2, ..., U_n\}$ wish to form a group and compute the shared group key. Each user U_i $(i = 1, ..., n)$ selects the Diffie-Hellman (DH) private share x_i and broadcasts the public share $y_i = g^{x_i} \bmod p$, where p and g are the prime modulo and generator used in the DH computation. On receiving the public shares (y_j) from all other members in the group, each U_i computes the DH key shared with each of them as

$$m_{ij} = y_j^{x_i} \bmod p$$

where $j = 1, ..., i-1, i+1, ..., n$. Each user then finds the Least Common Multiple (LCM) of the DH keys computed in the previous step. Let the LCM for U_i be lcm_i. U_i then randomly selects k_i (such that $k_i < min\{m_{ij}, j \neq i\}$) , its share for the group key. It also selects two arbitrary numbers, D and D_p such that $D \neq k_i$ and $gcd(D_p, lcm_i) = 1$. Each member then solves the CRT

$$\begin{aligned} crt_i &\equiv k_i \bmod lcm_i \\ crt_i &\equiv D \bmod D_p \end{aligned}$$

and broadcasts it to the group. Once each user U_i receives the CRT values from all the other members in the group, it obtains their corresponding secret shares by computing

$$k_j = crt_j \bmod m_{ij}, \text{ where } j \neq i.$$

It then computes the group key as $GK = k_1 \oplus k_2 \oplus ... \oplus k_n$.

2.3 Attacks on CRTDH

The CRTDH protocol, in its current form, lacks mutual authentication among members and hence is vulnerable to the man-in-the-middle attack. The paper does not address this issue since the major focus is on key management. The Diffie-Hellman (DH) key exchange in its basic form is susceptible to impersonation attacks and the existing CRTDH scheme uses DH in its basic form, hence, suffering from such kind of attack. CRTDH also suffers from another kind of attack, which we call the *Least Common Multiple (LCM)* attack. The attack is possible due to the fact that the LCM for any given set of numbers is not unique to the given set. In other words, there could be many more numbers that could possibly be added to the set and still result in the same LCM value. This could cause problems in the member join and member leave operations.

Problem with Member Join. Assume that there exists a group of four members, $\{U_1, U_2, U_3, U_4\}$ who share the group key GK and a user U_5 wishes to join the group. The member join operation in the current scheme requires one of the members (closest to the newly added member) to provide the new member with the hash value of the current group key, along with the public DH shares of the current group members i.e. $h(GK)$ and y_1, y_2, y_3, y_4. User U_5 executes the steps in the key agreement procedure (as described previously) and broadcasts the CRT value crt_5 along with its DH public share y_5. Existing members obtain the secret share selected by U_5 (k_5) after computing the DH key they share with it. The new group key is obtained by XORing the hash of the current group key and the key share of the newly joining member U_5.

The problem arises in the step where an existing user computes the DH key that it shares with the newly joined member. There could be a case where the addition of the shared DH key does not affect the LCM and hence the LCM value remains the same as before. This could lead to breaching of backward secrecy where the newly added member would be able to obtain the secret share of the existing member. To better explain the problem, assume that user U_4 computes the following after receiving the public share of U_5.

$\{U_4\} \rightarrow m_{41} = 6, m_{42} = 4, m_{43} = 8, m_{45} = 12.$

As can be observed, lcm_4 (=24) remains unchanged upon the addition of m_{45}. Hence user U_5 could obtain the shared secret k_4, if it could capture previous messages sent by user U_4. Similarly, it is possible that the values for all other lcms remain unchanged after U_5 joins, thus making it possible for U_5 to obtain all the previous key shares. This way U_5 can compute the previous group key.

Problem with Member Leave. In the current scheme, when an existing member of the group decides to leave, say U_i, the rekeying operation is performed in order to maintain forward secrecy. Any of the remaining members, say U_j, repeats the key agreement steps, wherein it selects a new k'_j and computes lcm_j again, but leaves the DH key it shares with U_i out of the LCM computation. It then solves the CRT and broadcasts it to the group. The idea here is that since the DH key shared between U_i and U_j i.e. m_{ij} is not included in the LCM and CRT computations, U_i would not be able to obtain the new key share k'_j.

The problem arises once again due to the fact that there may be cases where the new LCM value for a user may still cover the DH key value that it shared with the departing member. In such a case, the departing member would still be able to decrypt new messages.

3 AUTH-CRTDH: A Key Management Scheme for Ad Hoc Networks with Authentication Capability

In this section, we describe AUTH-CRTDH, a key management scheme for ad hoc networks with authentication capability. The users utilize the services of a central key generation center (KGC) for obtaining a secret corresponding to the ID. This central entity is different than a Group Controller (GC) (present in many schemes) as the services of the KGC are required only at system setup and it does not participate in the key agreement procedure. The operations performed at the KGC can be thought of as *offline* operations that need to be performed prior to the formation of any ad hoc environment. However, these operations ensure that each node in the ad hoc network can authenticate itself to any other node in the network. We also propose changes in the join and leave algorithms in the existing scheme to make the scheme resistant to LCM attacks. The proposed scheme is described below.

A. Offline System Setup

The users in the system are identified with a unique identity (ID). The scheme is analogous to RSA public key cryptosystem, with a value of e=3. The system setup procedure carried out at the KGC is described below.

- Step 1: Generate two large prime numbers p_1 and p_2, and let $n = p_1 \cdot p_2$.
- Step 2: Select the center's secret key d from the computation:

$$3 \times d \equiv 1 \bmod (p_1 - 1)(p_2 - 1).$$

- Step 3: Select an integer g that is a primitive element in Z_n^*.
- Step 4: Select a secure hash function h (i.e. one-way and collision-resistant) which is used to compute the extended identity (EID_i) of user U_i as:

  $$EID_i = h(ID_i)$$

  The hash function is made public.
- Step 5: Generate the user secret key S_i as

  $$S_i = EID_i^d (\bmod n)$$

  As a result of the above relations, the following equation holds.

  $$EID_i = S_i^3 (\bmod n)$$

When a user U_i registers with the system, he sends his ID_i to the KGC, which performs the steps 4 and 5 mentioned above. The KGC sends (n, g, h, S_i) to U_i. U_i keeps S_i secret and stores the public information (n, g, h). In addition, the ID of each user is publicly known.

B. Key Agreement

In order to establish the group key for a group with m members, each member U_i should execute the following steps, where $i = 1, 2, ..., m$.

- Step 1: Select the Diffie-Hellman (DH) private share x_i, and compute the following values for the first broadcast.

  $$\begin{aligned} A_i &= S_i \cdot g^{2x_i} \pmod n \\ B_i &= g^{3x_i} \pmod n \end{aligned}$$

- Step 2: Broadcast A_i and B_i to all members of the group.
- Step 3: Receive the public shares A_j and B_j from other members in the group and authenticate the users. Each member U_i calculates $EID_j = h(ID_j)$ and checks the validity of the member's broadcast message through the following:

  $$EID_j = A_j^3 / B_j^2$$

  If the equation holds true, then the user computes the DH shared secret with each of the members as follows:

  $$m_{ij} = B_j^{x_i} \bmod n, \text{ where } j \neq i.$$

  Otherwise, U_j's authentication fails and action would be initiated to remove user U_j from the group. Note: Step 6 and Step 7 conduct a second time verification on A_i.
- Step 4: Find the least common multiple (LCM) of all the DH keys calculated in Step 3 as lcm_i.
- Step 5: Select a random share for the group key k_i, such that $k_i <$min$\{m_{ij}, \forall j \neq i\}$. Also select an arbitrary number D such that $D \neq k_i$ and another number D_p such that $gcd(D_p, lcm_i) = 1$ (similar to original CRTDH scheme).

- Step 6: Solve the CRT: (as in original CRTDH scheme)

  $$crt_i \equiv k_i \bmod lcm_i$$
  $$crt_i \equiv D \bmod D_p$$

  For authentication purposes, the user also computes the following

  $$\begin{aligned} X_i &= h(k_i) \cdot g^{2D} \cdot S_i \pmod{n} \\ Y_i &= g^{3D} \pmod{n} \\ Z_i &= \{A_i || X_i\}_{k_i} \end{aligned}$$

  and broadcasts $\{X_i, Y_i, Z_i, crt_i\}$ to the group.
- Step 7: Receive the CRT values from all the other members in the group and calculate the following: (similar to CRTDH scheme)

  $$k_j = crt_j \bmod m_{ij}$$

  for all $j \neq i$. To validate the authenticity of the key k_j, the user also computes EID_j and verifies the following equation

  $$(h(k_j))^3 = X_j^3/(Y_j^2 \cdot EID_j)$$

  In addition, $\{A_j || X_j\}_{k_j}$ will be computed and verified against Z_j. This aims at authenticating A_i (Step 1) and X_i.

  After both of the above verifications succeed, the user then computes the group key (as in CRTDH scheme)

  $$GK = k_1 \oplus k_2 \oplus ... \oplus k_n$$

Thus the Chinese Remainder Theorem is used to send the secret key share of each user (disguised) to all the other members in the group. The mutual authentication is provided by using an ID based scheme.

To understand the details of the protocol, let us consider a group of 4 members $\{U_1, U_2, U_3, U_4\}$. In the first two steps, the users generate and distribute their DH public share in a way that their identity could be verified. U_1 selects a DH private share x_1 and computes $A_1 = S_1 \cdot g^{2x_1} \pmod{n}$ and $B_1 = g^{3x_1} \pmod{n}$. U_1 then broadcasts A_1 and B_1 to the other members in the group.

Step 3 of the protocol involves mutual authentication of users and generation of m_{ij} values, which are the DH keys shared between U_1 and the other members. U_1 calculates EID_2, EID_3, EID_4 and authenticates U_2, U_3, U_4. On successful authentication, U_1 calculates three m values m_{12}, m_{13}, m_{14}, which are equal to $B_2^{x_1}, B_3^{x_1}, B_4^{x_1}$ respectively. The three DH keys (m_{12}, m_{13}, m_{14}) generated by U_1 are equal to m_{21}, m_{31}, m_{41} generated by U_2, U_3, U_4 respectively. U_1 then calculates the LCM of the three DH keys m_{12}, m_{13}, m_{14}.

In step 5 of the protocol, U_1 generates its random key share k_1. The random key share must be less than all DH keys m_{12}, m_{13}, m_{14}. U_1 selects two arbitrary numbers, D and D_p, which are used in solving the CRT. Care should be taken while selecting D such that it is not equal to k_1. Otherwise, the solution of the CRT would be equal to k_1.

In step 6 of the protocol U_1 solves the CRT and also computes $X_1 = h(k_1) \cdot g^{2D} \cdot S_1 \bmod n$ and $Y_1 = g^{3D} \bmod n$ and broadcasts $\{X_1, Y_1, crt_1\}$ to the group. After receiving the CRT values crt_2, crt_3, crt_4 from other members of the group, U_1 can obtain k_2, k_3, k_4 by performing the following operations.

$$k_2 = crt_2(\text{mod } m_{12})$$
$$k_3 = crt_3(\text{mod } m_{13})$$
$$k_4 = crt_4(\text{mod } m_{14})$$

The shares k_i are then XOR-ed to obtain the group key $GK = k_1 \oplus k_2 \oplus k_3 \oplus k_4$.

C. Member Join

The operations when a new member joins the group are described below.

- Step 1: The newly added member U_i executes Steps 1 and 2 of the Key Agreement procedure and broadcast A_i and B_i to the group.
- Step 2: Every other member U_j (for all $j \neq i$) authenticates the incoming message from U_i and on successful authentication computes its shared DH key with U_i, (m_{ji}). It then recomputes the value for lcm_j. Let us call the new value lcm'_j
- Step 3: Now the user U_j checks for the following condition:

  $$lcm'_j \text{ mod } m_{ji} = 0,$$

  If the above condition holds, then the new member U_i is asked to re-select its secret x_i and broadcast the public share B_i again.
- Step 4: Once this has been done, and there is no member with the above condition satisfied, the closest member sends the hash of the current group key and the public shares of the existing members of the group to the newly joined member. The new member performs the rest of the steps of the Key Agreement procedure and the new group key is obtained in a similar fashion as in the original scheme, by XORing the hash of the current key and the key share of the newly joining member U_5 as follows:

  $$GK_{new} = h(GK) \oplus k_5$$

D. Member Leave

The leave operation in AUTH-CRTDH is discussed below.

- Step 1: When a member, say U_i leaves the group, any one of the remaining members, say U_j recomputes its LCM (lcm'_j).
- Step 2: It then evaluates the following condition:

  $$lcm'_j \text{ mod } m_{ji} = 0,$$

  If the above condition holds, then the new LCM covers the shared DH key m_{ji} and hence U_j needs to reselect its secret x_j and broadcast the public DH share B_j. The other members update their corresponding shared DH keys with U_j and the leave operation from step 1 is repeated.
- Step 3: If the above condition is not valid anymore, the new group key is obtained in the same way as in the current scheme, by XOR-ing the existing group key with the secret share of U_j.

4 Experimental Results

This section discusses the results from experiments conducted on several protocols, including the AUTH-CRTDH scheme. Different key agreement protocols were implemented and compared on the basis of computation times for the users. The implementation was done in C++ using the Crypto++ library [16], which is a free C++ library of cryptographic schemes. All tests were carried out on a Dell PowerEdge server with Dual 3.2 GHz Pentium CPU and 4GB RAM running Linux OS. The schemes that were implemented and compared were AUTH-CRTDH, CRTDH, ING, BD and Steer (STR) protocol. Graphs representing the computation time for a single member for key establishment are discussed. Additionally computation times for member join and leave and overheads on existing members when a member joins or leaves the group are also discussed.

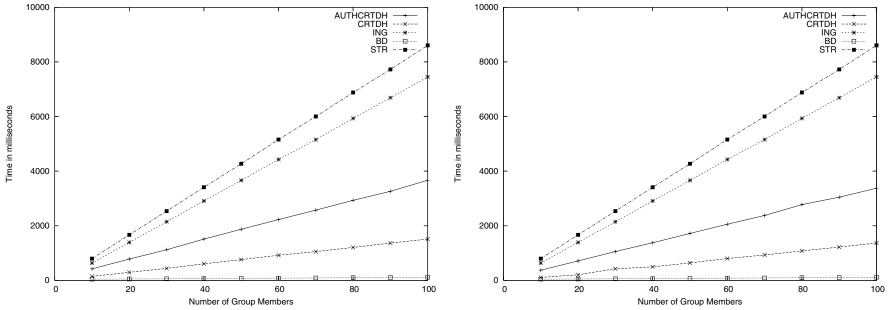


Fig. 1. Computation time of a single member for group key establishment (left) and Join operation time of joining member (right)

The first graph (See Figure 1 (left)) shows the computation time for a single member for key establishment in different protocols. Group sizes with the number of members ranging from 10 to 100 members were considered for the tests. The time was reported in milliseconds and the group key of size 128 bits was used in each case. As can be observed from Figure 1 (left), the AUTH-CRTDH computation time increases linearly and depends on the size of the group as well as the group key size. The computation times for ING and STR protocol are close. The AUTH-CRTDH scheme performs better than these protocols and is only a little more expensive than the original CRTDH scheme. However, this is a small cost paid to achieve mutual authentication among the members, which no other scheme discussed achieves. The BD scheme is most computationally efficient among all the schemes for key establishment. However, the BD protocol requires member serialization that is not required in AUTH-CRTDH.

The computation time (for the joining member) for the join operation is shown in Figure 1 (right). As can be observed, the join operation times are close to the key establishment times. This is due to the fact that the newly joining member essentially performs all operations of the key establishment procedure. The

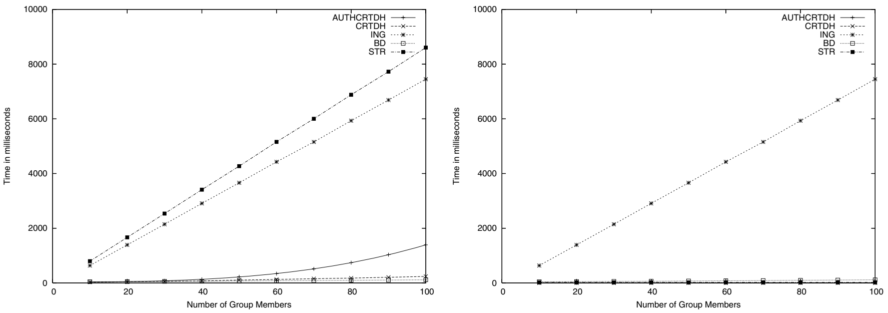


Fig. 2. Computation time: Leave operation time for leaving member (left) and Computation overhead for existing member when another member joins/leaves (right)

AUTH-CRTDH and CRTDH computation times for the newly joining member is slightly less than the key establishment time since the newly joining member does not have to perform the modular operation (for all other members) to obtain their secret shares. Instead the newly joining member computes the group key by XORing the hash of the old key and the key share that it selected.

A point to note here is that even though the BD protocol performs well computationally during the join operation, it involves two rounds of communication. On the other hand, both AUTH-CRTDH and CRTDH involve only one round of communication. Additionally in the BD protocol, existing members other than the joining members need to perform considerable amount of computation, which is not the case in AUTH-CRTDH. The computation load on existing members is also discussed later in this section.

Figure 2 (left) discusses the computation times for a member leave operation. The ING, BD and STR protocols have similar computation times as the key establishment and join operations since all these three operations are quite similar. As mentioned above for the join operation, the AUTH-CRTDH protocol involves less overhead for existing members when a member leaves the group, unlike the BD protocol.

The overhead on an existing member when another member joins or leaves is an important factor. This specifies the amount of computation that a user has to perform due to a membership change. This is an important factor in ad hoc networks as there may be frequent group changes due to high mobility. Hence, the overhead of the system should be minimal to save battery power.

Figure 2 (right) shows the computational overhead for an existing member for member join/leave. As can be observed from the figure, the overhead is highest in the ING protocol. In order to make the graph more clear, the ING protocol was removed and another graph was obtained as shown in Figure 3.

It can be observed from Figure 3 that the overhead is constant for the STR protocol since it involves one DH public key exponentiation. The computational overhead for the BD protocol increases as the group size increases. The AUTH-CRTDH and CRTDH schemes have the least overhead for an existing member

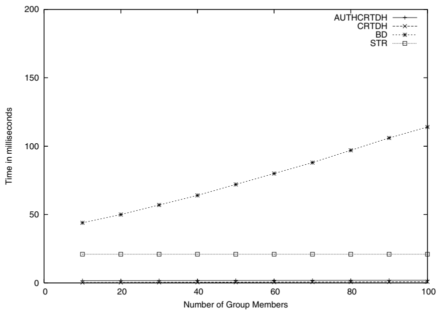


Fig. 3. Computation overhead for existing member when another member joins/leaves without ING computation times

compared to all the other protocols when a member joins or leaves the group. This is a desired property for ad hoc networks.

Thus it can be inferred from the above experiments that the proposed AUTH-CRTDH protocol retains the desirable characteristics from the original CRTDH scheme while adding authentication to the protocol. The protocol in fact, performs better than some of the non-authenticated protocols. The proposed protocol does not require any pre-shared secrets between the nodes and supports high user dynamics. The protocol uses Chinese Remainder Theorem, which is not computationally intensive.

The scheme is very efficient communication wise as it requires only two rounds for initial key agreement and member join operation and one round for leave operation. The scheme does not require member serialization, a requirement in some protocols which is not feasible in ad hoc networks. Every node is treated equally and the workload is distributed among all nodes equally. As the proposed scheme is efficient in terms of amount of computation time also, it could be employed for secure conference applications.

5 Conclusion

In this paper, we identify the lack of key management protocols for ad hoc networks with authentication capabilities. We study the security of the CRTDH protocol and describe several practical attacks on it. We also proposed AUTH-CRTDH, a modified key agreement scheme with authentication capability. The new scheme maintains the efficiency of the original CRTDH scheme while defeating the attacks discussed in the paper.

References

1. Balachandran, R., Ramamurthy, B., Zou, X., Vinodchandran, N.: CRTDH: An efficient key agreement scheme for secure group communications in wireless ad hoc networks. Proceedings of IEEE ICC 2005 (2005)

2. Luo, W., Fang, Y.: A survey of wireless security in mobile ad hoc networks: Challenges and Solutions. Ad Hoc Wireless Networking, Kluwer Academic Publishers (2003) 319–364
3. Weimerskirch, A., Thonet, G.: A distributed light-weight authentication model for ad-hoc networks. Proceedings of the 4th International Conference Seoul on Information Security and Cryptology (2001) 341–354
4. Zhou, L., Haas, Z.J.: Securing ad hoc networks. IEEE Network **13**(6) (1999) 24–30
5. Wu, B., Wu, J., Fernandez, E., Magliveras, S.: Secure and efficient key management in mobile ad hoc networks. Proceedings of the 1st Int'l Workshop on Systems and Network Security (SNS2005) (2005)
6. Khalili, A., Katz, J., Arbaugh, W.: Toward secure key distribution in truly ad-hoc networks. Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03) (2003)
7. Perrig, A., Canetti, R., Tygar, D., Song, D.: The TESLA broadcast authentication protocol. Cryptobytes, Volume 5, No. 2 (RSA Laboratories, Summer/Fall 2002) (2002) 2–13
8. Zhu, S., Xu, S., Setia, S., Jajodia, S.: LHAP: A lightweight hop-by-hop authentication protocol for ad-hoc networks. IEEE International Conference on Distributed Computing Systems (2003)
9. Lu, B., Pooch, U.: A lightweight authentication protocol for ad-hoc networks. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) (2005)
10. Stajano, F., Anderson, R.: The resurrecting duckling: Security issues for ad-hoc wireless networks. Proceedings of the 7th International Workshop on Security Protocols (1999) 172–194
11. Ingemarsson, I., Tang, D., Wong, C.: A conference key distribution system. IEEE Transactions on Information Theory **28**(5) (1982) 714–720
12. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. ACM Conference on Computer and Communications Security (ACM CCS 1996) (1996) 31–37
13. Steiner, M., Tsudik, G., Waidne, M.: Key agreement in dynamic peer groups. IEEE Transactions on Parallel and Distributed Systems **11**(8) (2000) 769–780
14. Steiner, M., Tsudik, G., Waidner, M.: CLIQUES: A new approach to group key agreement. IEEE International Conference on Distributed Computing Systems (ICDCS 1997) (1997) 380–387
15. Steer, D., Strawczynski, L., Diffie, W., Wiener, M.: A secure audio teleconference system. Advances in Cryptology-CRYPTO'88, LNCS, Springer-Verlag **403** (1990) 520–528
16. Dai, W.: Crypto++ library. At http://www.escimo.com/weidai/cryptlib.html (2005)

Efficient ID-Based Authenticated Group Key Agreement from Bilinear Pairings

Lan Zhou, Willy Susilo, and Yi Mu

Center for Information Security Research
School of IT and Computer Science
University of Wollongong, Wollongong, NSW 2522
Australia
{lz815, wsusilo, ymu}@uow.edu.au

Abstract. The nature of mobile ad-hoc networks does not permit a member of the group or a central authority to determine a single key to be used among the group members. Group key agreement offers a solution to this problem by allowing the group members to collaboratively determine the common key for the group. Additionally, authenticated group key agreement (AGKA) is an important issue in many modern collaborative and distributed applications. During the last few years, a number of authenticated group key agreement protocols have been proposed in the literature. In this paper, we present a secure ID-based AGKA protocol which only requires *one round* by using pairing-based cryptography. We prove that the scheme is secure against an active adversary under the decisional bilinear Diffie-Hellman assumption in the Random Oracle Model. We then extend our scheme to a two-round AGKA protocol which is more efficient in communication costs, and this scheme outperforms any existing AGKA protocols in the literature.

Keywords: authenticated group key agreement, bilinear pairings, ID-based, cryptography.

1 Introduction

Mobile ad-hoc networks, comprised of constrained devices, offer convenient communication over the shared wireless channels in the (partial) absence of any fixed infrastructure. Securing such networks becomes a very important issue. In ad-hoc networks, key distribution techniques are less useful than key agreement techniques since the trust in the network to allow a member of the group of a central authority to determine the group key is lacking. Group key agreement protocols, which enable a set of participants to agree on a common secret value based on each participant's public contribution, provide a good solution to secure mobile ad-hoc networks.

Since the publication of the well-known Diffie-Hellman (DH) key exchange [9], many solutions have been proposed to extend it to the multi-party setting. Notable solutions, which can be viewed as the first group key agreement schemes, have been proposed by Ingemarsson *et al.* [10] in 1982.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 521–532, 2006.

Group key agreement is a protocol that allows a group of users communicating over an insecure, open network to come up with a common session key. This session key, which is only known to the users who are the valid members of the group, may later be used to facilitate the communication among these users. By using group key agreement protocols, the presence of a central authority is no longer required. Moreover, when the group composition changes, one can employ supplementary key agreement protocols to obtain a new group key. Thus, a transient secure channel can be constructed during the lifetime of one session of a group.

Authenticated group key agreement is a group key agreement protocol ensured with an authentication mechanism, which is used to guarantee that *no* other users aside from the valid members of the group can learn any information about the session key. Authenticated group key agreement can be classified into two categories: Certificate-based and ID-based.

The certificate-based protocols work by assuming that each user has a (long-term) public/private key pair, and each user knows the public key of each other user. Thus, the problem of authenticating the session key is replaced by the problem of authenticating the long-term public keys. Hence, in a certificate-based system, the participants *must* firstly verify the certificate of the user *before* using the user's public key. Consequently, the system requires a large amount of computing time and storage.

The ID-based protocols allow each user using their identities (IDs) of other users as their public keys. Many ID-based AGKA protocols have been proposed in recent years. Nevertheless, some efficient results in [3,4,11,17] require two rounds to construct a session key and some of these protocols are found to be flawed [4,17]. In [12,13,16], some single round tripartite authenticated key agreement protocols were proposed but these methods cannot be extended to large groups consisting of more than three parties since these methods rely on the bilinearity property of bilinear pairing. Very recently, a single round ID-based AGKA protocol was proposed in [15]. However, we note that the scheme requires the user to keep a *public key* issued by the group administrator (GA) and verify each other's public key before using it, which is the idea of a certificate-based group key agreement protocol. This means, that their scheme is *not* an ID-based scheme, rather than a public key based scheme. Moreover, the scheme in the paper [15] is flawed actually. We will show it in Appendix A.

Our Contribution

In this paper, *for the first time in the literature*, we present a provably secure one-round ID-based AGKA protocol. The protocol is a *contributory key agreement* in which each group member takes responsibility for contributing to the generation of group session key. We also present an efficient AGKA protocol, which is a variant of our one-round ID-based AGKA. The scheme itself requires two rounds. However, as we shall show in this paper, this scheme is very efficient in communication costs than other previously known ID-based AGKA protocols, and hence, this scheme outperforms any other existing schemes in the literature. We provide security proofs for our schemes, and show that they are secure against active adversary under the decisional bilinear Diffie-Hellman assumption in the Random Oracle Model. Our protocols provide *forward secrecy* in the sense that any exposure of any user's long-term private keys *does not* compromise the security of previous session keys.

Organization of The Paper
In Section 2, we first review some security assumptions used throughout the paper, and define our security model and some notations. Then, we propose our one round authenticated group key agreement protocol (O-AGKA) in Section 3. In Section 4, we provide the security proof of our O-AGKA protocol. In Section 5, we present an efficient two-round authenticated group key agreement protocol (T-AGKA), which uses the same technique as our O-AGKA protocol. We compare the efficiency between our two efficient AGKA protocols and some other efficient AGKA protocols proposed in the literature in Section 6. Finally, Section 7 concludes the paper.

2 Preliminaries

In this section, we first review some cryptographic assumptions that will be used throughout the paper. Then, we describe the security model in which we prove the security of our group key agreement protocol.

2.1 The Bilinear Maps and Assumption

Let $\mathbb{G}_1$ be a cyclic additive group of prime order q. Let $\mathbb{G}_2$ be a cyclic multiplicative group of same order q. We assume that the discrete logarithm problems (DLP) in both $\mathbb{G}_1$ and $\mathbb{G}_2$ are hard to solve.

BDH Parameter Generator
Let Bilinear Diffie-Hellman (BDH) parameter generator $\mathcal{IG}_{DBH}$ be a probabilistic polynomial time (PPT) algorithm. When running in polynomial time, $\mathcal{IG}_{DBH}$ outputs two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same order q and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ which satisfies the following properties:

- *Bilinear:* for all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$ we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
- *Non-degenerate:* if for $P \in \mathbb{G}_1$ we have $\hat{e}(P, Q) = 1$ for all $Q \in \mathbb{G}_1$, then $P = \mathcal{O}$.
- *Computable:* for all $P, Q \in \mathbb{G}_1$, the pairing $\hat{e}(P, Q)$ is computable in polynomial time.

Decisional Bilinear Diffie – Hellman (DBDH) problem
The decisional BDH problem is to distinguish between tuples of the form $(P, aP, bP, cP, \hat{e}(P, P)^{abc})$ and $(P, aP, bP, cP, \hat{e}(P, P)^d)$ for random $P \in \mathbb{G}_1$, and $a, b, c, d \in \mathbb{Z}_q^*$. An algorithm $\mathcal{A}$ is said to solve the BDH problem with an advantage of ϵ if

$$|Pr[\mathcal{A}(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - Pr[\mathcal{A}((P, aP, bP, cP, \hat{e}(P, P)^d)) = 1]| \leq \epsilon$$

DBDH Assumption: We assume that the probability of a polynomial time algorithm to solve DBDH problem is negligible.

2.2 Security Model

The model described below follows Bresson *et al.*'s [6] formal security model. We restrict to recalling some details of their cryptographic proof model used for the security proof below. A more detailed discussion of this model can be found in [5,6].

Participants. A finite set $\mathcal{U}$ of PPT Turing machines U_i models the users that constitute the (potential) protocol participants. In this model we allow each user $U_i \in \mathcal{U}$ to execute a protocol many times with different users. A user may execute a polynomial number of protocol instances in parallel. We denote the instance $t \in \mathbb{N}$ of principal $U_i \in \mathcal{U}$ by Π_i^t.

Initialization. During this phase, which is conducted before the first execution of the key establishment protocol, the master secret key s and global parameters Params are generated by algorithm Setup. Each user $U_i \in \mathcal{U}$ gets public and private keys from a group administrator GA by using algorithm Setup, while the long-term private key S_{U_i} is only revealed to U_i, the corresponding public key is given to all users.

Adversarial model. Normally, the security of a protocol is related to the adversary's ability. The abilities are formally modeled by queries issued by adversaries. We assume that a probabilistic polynomial time adversary $\mathcal{A}$ controls the communications completely and can make queries to any instance. The list of queries that $\mathcal{A}$ can make is summarized below:

- $\mathsf{Execute}(\{U_1, U_2, \ldots, U_r\})$: This query executes a protocol run between the users $\{U_1, U_2, \ldots, U_r\}$, and the adversary $\mathcal{A}$ gets the complete transcripts of all the messages sent during the protocol execution.
- $\mathsf{Send}(\Pi_i^t, M)$: This query allows the adversary $\mathcal{A}$ to send a message M to instance Π_i^t, and $\mathcal{A}$ gets back the reply generated by this instance.
- $\mathsf{Reveal}(\Pi_i^t)$: This query returns the session key. $\mathcal{A}$ is allowed to use this query only if the oracle Π_i^t has accepted, then $\mathcal{A}$ gets the session key.
- $\mathsf{Corrupt}(U_i)$: This query allows the adversary $\mathcal{A}$ to get the long-term private key corresponding to U_i. But the adversary $\mathcal{A}$ does not get the internal data of any instance of U_i executing the protocol.
- $\mathsf{Test}(U_i, t)$: The adversary $\mathcal{A}$ can use this query only once. This query models the semantic security of a session key. $\mathcal{A}$ can ask any of the above queries, and once, asks a Test query. Then, a random bit b is drawn and the session key is returned if $b = 1$, otherwise a random value is returned.

In the model, we consider two types of adversaries according to their attack types. The attack types are simulated by the queries issued by adversaries.

A *passive adversary* is not allowed to use Send and Corrupt queries, while an *active adversary* can issue all the above queries.

2.3 Security Notions

We define session IDS (SIDS) for oracle Π_i^t in a execution protocol as $SIDS(\Pi_i^t) = \{SID_{ij}, j \in \mathcal{U}\}$ where SID_{ij} is the concatenation of all messages exchanged by oracle Π_i^t with Π_j^w. The partner ID for an oracle Π_i^t, denoted by $PIDS(\Pi_i^t)$, is a set of the users with whom Π_i^t intends to establish a session key.

Definition 1. Partnering: *Now we define instances Π_i^t and Π_j^w are partnered if and only if $PIDS(\Pi_i^t) = PIDS(\Pi_j^w)$ and $SIDS(\Pi_i^t) = SIDS(\Pi_j^w)$.*

Definition 2. Freshness: *We define a user instance Π_i^t that has accepted fresh if:*

- *For a $U_j \in \mathcal{U}$, a* **Corrupt**(U_j) *query was never executed before a query of the form* **Send**$(\Pi_i^t, *)$ *or* **Send**$(\Pi_j^w, *)$*, where Π_i^t and Π_j^w are partnered, has taken place.*
- *Π_i^t has accepted a session key $K \neq NULL$ and neither Π_i^t nor one of its partners has been asked for a* **Reveal** *query.*

Before we look into the security of the protocol, we first define the following game between the adversary $\mathcal{A}$ and a set of oracles Π_i^t for $U_i \in \mathcal{U}$.

1. Each user is given a long-term private key during the initialization phase.
2. Adversary $\mathcal{A}$ interacts with some queries and gets back the reply generated by the corresponding oracles.
3. $\mathcal{A}$ executes a **Test**(Π_i^t) query for a fresh oracle Π_i^t, and finally outputs a guess bit b'.

In above game, we denote by Succ the probability that the bit b' outputs by $\mathcal{A}$ satisfies $b = b'$.

Definition 3. Protocol Security: *We denote the advantage of the adversary $\mathcal{A}$ attacking the protocol as $\mathsf{Adv}_{\mathcal{A}}(k) = |2 \cdot \mathsf{Succ} - 1|$. We say the group key establishment protocol secure if for all PPT adversary $\mathcal{A}$ $\mathsf{Adv}_{\mathcal{A}}(k)$ is negligible.*

Definition 4. Authentication: *A key agreement protocol is said to provide authentication if for a user U_i, no other users except partners can learn the value of a session key.*

Definition 5. Forward Secrecy: *Forward secrecy means that an adversary gets negligible advantage in knowing information about previously established session keys when making a* Corrupt *query.*

3 A One Round Group Key Agreement Scheme

In this section, we propose our one-round ID-based AGKA scheme, which we called O-AGKA. The protocol involves a group administrator GA. In the following description $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ and $H_3 : \{0,1\}^n \rightarrow \{0,1\}^n$ are cryptographic hash functions. H_1, H_2 and H_3 are considered as random oracles in the security analysis.

Setup. GA runs BDH parameter generator to generate a prime q, two groups $\mathbb{G}_1, \mathbb{G}_2$ of order q, and an bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Choose a random generator $P \in \mathbb{G}_1$. Then GA picks a random $s \in \mathbb{Z}_q^*$ and sets $P_{pub} = sP$. GA keeps s secret as the *master secret key* and publishes system parameters $\mathsf{params} = \{\hat{e}, \mathbb{G}_1, \mathbb{G}_2, q, P, P_{pub}, H_1, H_2, H_3\}$.

Extract. When a user U_i with identity ID_i wishes to obtain a key pair, GA computes $Q_i = H_1(ID_i)$ and the long-term private key $S_i = sQ_i$, and returns S_i to the user U_i.

Let $U_1, \ldots, U_n$ be the n users who want to establish a session key. The protocol is as follows:

Interacting. Each user U_i picks $\delta_i \overset{R}{\leftarrow} \mathbb{G}_2$ and $r_i, k_i \overset{R}{\leftarrow} \{0,1\}^n$. Then U_i computes $P_i^j = H_2(\hat{e}(S_i, Q_j) \cdot \delta_i) \oplus r_i$, where $1 \leq j \leq n$ and $j \neq i$. U_i then computes

$$D_i = \langle \delta_i,\ P_i^1,\ \ldots,\ P_i^{i-1},\ P_i^{i+1},\ \ldots,\ P_i^n,\ H_3(r_i) \oplus k_i,\ \mathcal{L} \rangle,$$

where $\mathcal{L}$ is a label that contains information about how "P_i^j" is associated with each receiver. Then U_i broadcasts D_i to all others.

Key Computation. Let $D_j = \langle R_j, P_j^1, \ldots, P_j^n, V_j, \mathcal{L} \rangle$. Upon receiving D_j, each responder U_i, using $\mathcal{L}$, finds appropriate P_j^i and computes

$$k_j' = H_3(H_2(\hat{e}(Q_j, S_i) \cdot R_j) \oplus P_j^i) \oplus V_j$$

Each U_i can now compute the common session key as follows:

$$K = K_i = k_1' \oplus \cdots \oplus k_{i-1}' \oplus k_i \oplus k_{i+1}' \oplus \cdots \oplus k_n'$$

4 Security Proof

In this section, we show that the protocol O-AGKA is secure against an active adversary under the *DBDH* assumption. In other words, if there exists an active adversary who has non-negligible probability of breaking the protocol O-AGKA, then he also has non-negligible probability of breaking the *DBDH* assumption.

Theorem 1. *The above O-AGKA protocol is secure against an active adversary under the DBDH assumption in the Random Oracle Model. Concretely,*

$$\mathsf{Adv}_{\mathcal{A}} \leq 2n \cdot q_{ex} \cdot \mathsf{Adv}_{G}^{DBDH}$$

Proof. Let $\mathcal{A}$ be an active adversary that can get an advantage in breaking O-AGKA. We first consider the case that an adversary $\mathcal{A}$ makes only one $\mathsf{Execute}$ query and then extend this to the case that $\mathcal{A}$ makes multiple $\mathsf{Execute}$ queries. Let n be the number of users chosen by the adversary $\mathcal{A}$. The distribution of the transcript $\mathcal{T}$ and the resulting group session key K is given by:

$$\mathsf{params} = \begin{bmatrix} (\mathbb{G}_1, \mathbb{G}_2, \hat{e}) \leftarrow \mathcal{IG}_{BDH}; P \leftarrow \mathbb{G}_1; s \leftarrow \mathbb{Z}_q^*; P_{pub} = sP \\ Q_1, \ldots, Q_n \leftarrow \mathbb{G}_1; S_1 = sQ_1, \ldots, S_n = sQ_n : \\ (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}) \end{bmatrix}$$

$$Real = \begin{bmatrix} \delta_1, \ldots, \delta_n \leftarrow \mathbb{G}_2; r_1, \ldots, r_n, k_1, \ldots, k_n \leftarrow \{0,1\}^n; \\ R_1 = \delta_1, \ldots, R_n = \delta_n \\ P_1^i = H_2(\hat{e}(S_1, Q_i) \cdot \delta_1) \oplus r_1, \ldots, P_n^i = H_2(\hat{e}(S_n, Q_i) \cdot \delta_n) \oplus r_n \\ V_1 = H_3(r_1) \oplus k_1, \ldots, V_n = H_3(r_n) \oplus k_n; \\ k_j' = H_3(H_2(\hat{e}(Q_j, S_i) \cdot R_j) \oplus P_j^i) \oplus V_j \\ \mathcal{T} = \langle R_1, \ldots, R_n, P_1^i, \ldots, P_n^i, V_1, \ldots, V_n \rangle; \\ K = k_1' \oplus \cdots \oplus k_{i-1}' \oplus k_i \oplus k_{i+1}' \oplus \cdots \oplus k_n' \end{bmatrix}$$

Consider the distributions $Fake_i$ defined as follows:

$$Fake_1 = \begin{bmatrix} \delta_1, \ldots, \delta_n \leftarrow \mathbb{G}_2; r_1, \ldots, r_n, k_1, \ldots, k_n \leftarrow \{0,1\}^n; b_1, \ldots, b_n \leftarrow \mathbb{Z}_q^* \\ R_1 = \delta_1, \ldots, R_n = \delta_n \\ P_1^i = H_2(\hat{e}(b_1 P_{pub}, b_i P) \cdot \delta_1) \oplus r_1, \ldots, P_n^i = H_2(\hat{e}(S_n, Q_i) \cdot \delta_n) \oplus r_n \\ V_1 = H_3(r_1) \oplus k_1, \ldots, V_n = H_3(r_n) \oplus k_n; \\ k_1' = H_3(H_2(\hat{e}(b_1 P, b_i P_{pub}) \cdot R_1) \oplus P_1^i) \oplus V_1 \\ k_j' = H_3(H_2(\hat{e}(Q_j, S_i) \cdot R_j) \oplus P_j^i) \oplus V_j | 2 \leq j \leq n, j \neq i \\ \mathcal{T} = \langle R_1, \ldots, R_n, P_1^i, \ldots, P_n^i, V_1, \ldots, V_n \rangle; \\ K = k_1' \oplus \cdots \oplus k_{i-1}' \oplus k_i \oplus k_{i+1}' \oplus \cdots \oplus k_n' \end{bmatrix} \cdots$$

Continuing in this way, we obtain the distribution:

$$Fake_n = \begin{bmatrix} \delta_1, \ldots, \delta_n \leftarrow \mathbb{G}_2; r_1, \ldots, r_n, k_1, \ldots, k_n \leftarrow \{0,1\}^n; b_1, \ldots, b_n \leftarrow \mathbb{Z}_q^* \\ R_1 = \delta_1, \ldots, R_n = \delta_n \\ P_1^i = H_2(\hat{e}(b_1 P_{pub}, b_i P) \cdot \delta_1) \oplus r_1, \ldots, P_n^i = H_2(\hat{e}(b_n P_{pub}, b_i P) \cdot \delta_n) \oplus r_n \\ V_1 = H_3(r_1) \oplus k_1, \ldots, V_n = H_3(r_n) \oplus k_n; \\ k_j' = H_3(H_2(\hat{e}(b_j P, b_i P_{pub}) \cdot R_j) \oplus P_j^i) \oplus V_j \\ \mathcal{T} = \langle R_1, \ldots, R_n, P_1^i, \ldots, P_n^i, V_1, \ldots, V_n \rangle; \\ K = k_1' \oplus \cdots \oplus k_{i-1}' \oplus k_i \oplus k_{i+1}' \oplus \cdots \oplus k_n' \end{bmatrix}$$

Let $\epsilon = \mathsf{Adv}_G^{DBDH}$. Assume that $\mathcal{A}$ made q_{se} times Send queries and q_{ex} times $\mathsf{Execute}$ queries. Then $\mathcal{A}$ randomly chooses $(\mathcal{T}, K)$ pairs to make a Test query and outputs b'. Since $\mathcal{A}$ can obtain $b_1 P, \ldots, b_n P$ by using multiple H_1 queries, and $P_{pub} = sP$ is public, it is obviously that $\mathcal{A}$ can distinguish $\hat{e}(S_{U_1}, Q_{U_i})$ from

$\hat{e}(b_1 sP, b_i P)$ with probability ϵ', where $\epsilon' \leq \epsilon$. Hence $\mathcal{A}$ can correctly guesses $b = b'$ with probability ϵ'. The remaining steps continue in the same way and we obtain:

$$\begin{aligned}
&|Pr[\mathcal{T} \leftarrow Real; K \leftarrow Real; \mathcal{A}(\mathcal{T}, K) = 1] - \\
&\qquad Pr[\mathcal{T} \leftarrow Fake_1; K \leftarrow Fake_1; \mathcal{A}(\mathcal{T}, K) = 1]| \leq \epsilon \\
&|Pr[\mathcal{T} \leftarrow Fake_1; K \leftarrow Fake_1; \mathcal{A}(\mathcal{T}, K) = 1] - \\
&\qquad Pr[\mathcal{T} \leftarrow Fake_2; K \leftarrow Fake_2; \mathcal{A}(\mathcal{T}, K) = 1]| \leq \epsilon \\
&\qquad\qquad \vdots \\
&|Pr[\mathcal{T} \leftarrow Fake_{n-1}; K \leftarrow Fake_{n-1}; \mathcal{A}(\mathcal{T}, K) = 1] - \\
&\qquad Pr[\mathcal{T} \leftarrow Fake_n; K \leftarrow Fake_n; \mathcal{A}(\mathcal{T}, K) = 1]| \leq \epsilon
\end{aligned}$$

Combining the above equations, we obtain the following.

$$\begin{aligned}
\epsilon'' = &|Pr[\mathcal{T} \leftarrow Real; K \leftarrow Real; \mathcal{A}(\mathcal{T}, K) = 1] - \\
&\qquad Pr[\mathcal{T} \leftarrow Fake_n; K \leftarrow Fake_n; \mathcal{A}(\mathcal{T}, K) = 1]| \leq n \cdot \epsilon
\end{aligned}$$

Hence ϵ'' is the probability that the session key can be correctly guessed when $\mathcal{A}$ make the Test query. Assume that $\mathcal{A}$ has made q_h times H_1 queries during the breaking process, then there will be a $\mathcal{H}$-list which contains all the messages that $\mathcal{A}$ has queried before. Let Ask be the event that what $\mathcal{A}$ make to the $Hash$ query is on the $\mathcal{H}$-list when $\mathcal{A}$ make the Test query. The advantage $\mathcal{A}$ in breaking the protocol conditioned by the fact that the session key is correctly guessed, is:

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}} &= 2 \cdot \mathsf{Succ} - 1 = 2Pr[b = b'] - 1 \\
&= 2Pr[b = b' | \neg\mathsf{Ask}] Pr[\neg\mathsf{Ask}] + 2Pr[b = b' | \mathsf{Ask}] Pr[\mathsf{Ask}] - 1 \\
&= 2Pr[b = b' | \neg\mathsf{Ask}] + 2Pr[b = b' | \mathsf{Ask}] - 1 \\
&= 2Pr[b = b' | \mathsf{Ask}] = 2\epsilon''
\end{aligned}$$

In the random oracle model, $2Pr[b = b' | \neg\mathsf{Ask}] - 1 = 0$, since $\mathcal{A}$ cannot gain any advantage on a random oracle without asking for it. Then we can have the probability that $\mathcal{A}$ breaks the O-AGKA, which is less than $2n \cdot \mathsf{Adv}_G^{DBDH}$. By adapting a standard hybrid argument, we obtain the probability that an active adversary $\mathcal{A}$ breaks the protocol O-AGKA as follows:

$$\mathsf{Adv}_{\mathcal{A}} \leq 2n \cdot q_{ex} \cdot \mathsf{Adv}_G^{DBDH}$$

5 An Efficient Group Key Agreement Scheme

In this section, we present a two-round authenticated group key agreement protocol called T-AGKA, which is more efficient in communiacation costs than other previously known AGKA protocols. The T-AGKA protocol is a variant of O-AGKA protocol presented above and described as follows:

Setup. As in the O-AGKA scheme. In addition, we pick three new hash functions $H_4 : \mathbb{G}_2 \rightarrow \{0,1\}^n$, $H_5 : \{0,1\}^n \rightarrow Z_q^*$ and $H_6 : \mathbb{G}_1 \rightarrow \{0,1\}^n$.

Extract. As in the O-AGKA scheme.

Round 1. The initiator U_1 picks $\delta \xleftarrow{R} \mathbb{G}_2$, $r \xleftarrow{R} \{0,1\}^n$ and $k_1 \xleftarrow{R} \mathbb{Z}_p^*$. Then U_1 computes $D_1 = \langle R, P_2, \ldots, P_n, V, W, \mathcal{L}\rangle$ such that

$$D_1 = \langle \delta, r \oplus H_4(\hat{e}(S_1, Q_2)\cdot\delta),\ \ldots,\ r \oplus H_4(\hat{e}(S_1, Q_n)\cdot\delta),\ H_5(r)\cdot k_1 P,\ k_1 P_{pub},\ \mathcal{L}\rangle,$$

where $\mathcal{L}$ is a label that contains information about how "P_i" is associated with each receiver, and broadcasts D_1 to all others.

Round 2. Upon receiving D_1, each responder U_i, $2 \le i \le n$, using $\mathcal{L}$, finds appropriate P_i, and computes $r' = H_4(\hat{e}(Q_1, S_i)\cdot R) \oplus P_i$. If D_1 is the valid message, it is obvious that $r' = r$. Then U_i picks $k_i \xleftarrow{R} \mathbb{Z}_p^*$, computes

$$D_i = \langle H_5(r)\cdot k_i P,\ k_i P_{pub}\rangle$$

and broadcasts D_i to all others.

Key Computation. Let $D_j = \langle X_j, Y_j\rangle$. When received D_j, each U_i, including the initiator U_1, computes $z_1 = H_5(r)^{-1}\cdot V$ and $z_j = H_5(r)^{-1}\cdot X_j$, $2 \le j \le n$. Then each U_i can hold a list of z_j, and U_i can verify all the z_j:

$$\hat{e}(P, \sum_{j=1}^{n} Y_j) \stackrel{?}{=} \hat{e}(P_{pub}, \sum_{j=1}^{n} z_j)$$

If the above equation holds, U_i can assume that all the parties involved are valid members of the group with the corresponding long-term private keys.

Each U_i now can compute the common session key as follows:

$$K = K_i = H_6(z_1) \oplus \cdots \oplus H_6(z_n)$$

Theorem 2. *The two-round group key agreement protocol T-AGKA is secure against an active adversary under the DBDH assumption in the Random Oracle Model. Concretely,*

$$\mathsf{Adv}_{\mathcal{A}} \le 2q_{ex}\cdot \mathsf{Adv}_G^{DBDH}$$

Due to lack of spaces, we omit the security proof of this theorem since it is similar to the proof of Theorem 1. We refer the reader to the full version of this paper [18] for a more complex account.

6 Comparison

In this section, we compare our protocols O-AGKA, T-AGKA with some previously known AGKA protocols, the ID-GKA by Choi *et al.* [4], the two round multi-party key agreement protocol MP-KA of Du *et al.* [17] and the AGKA by Shi *et al.* [15]. Since ID-GKA is already found to be flawed, we use an improved ID-GKA scheme in [14] instead in the following table. We use the following notations:

n	total number of the users in the group
Round	total number of rounds
Pairing	total number of pairing computations for all users
Ucasts	total number of unicast of all members
Bcasts	total number of broadcast of all members
Msize	total number of the messages of all members

Then, we obtain the following comparison.

Table 1. Comparison of AGKA protocols

Protocol	Round	Pairing	Ucasts	Bcasts	Msize	Type
Choi's AGKA	2	4n	0	2n	3n	ID-based
Du's AGKA	2	4n	0	2n	3n	ID-based
Shi's AGKA	1	n	$(n-1)^2$	0	n^2	Non ID-based & flawed
Our O-AGKA	1	n^2	0	n	n(n+2)	ID-based
Our T-AGKA	2	4n	0	n	3n	ID-based

As shown in Table 1, our two-round T-AGKA protocol is the most efficient one as compared to other protocols. Our O-AGKA protocol requires to involve more messages compared to other protocols, but it *only* requires one round. We note that Shi's AGKA actually is flawed and is *not* an ID-based protocol (refer to our justification in Appendix A), and hence, our O-AGKA protocol is *the only* provably secure protocol that requires one round that is known to date.

7 Conclusion

We proposed a single round ID-based authenticated group key agreement protocol. We also provided an alternative way of achieving an efficient two rounds ID-based AGKA protocol. We proved that our ID-based AGKA protocols are secure against active adversary under the assumption of $DBDH$ in the Random Oracle Model(ROM).

References

1. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *Lecture Notes in Computer Science 2139*, pages 213 – 229, 2001.
2. D. Boneh, B. Lynn, and H. Shacham. Short signature from the Weil Pairing. *in Proc. of Asiacrypt 2001, LNCS 2248*, pages 514–532, 2001.
3. J. Bohli, B. Glas, R. Steinwandt. Towards Provably Secure Group Key Agreement Building on Group Theory. *Cryptology ePrint Archive, Report 2006/079*, 2006.
4. K. Y. Choi, J. Y. Hwang and D. H. Lee. Effcient ID-based Group Key Agreement with Bilinear Maps. *PKC 2004, Lecture Notes in Computer Science 2947*, pages 130 – 144, Springer-Verlag, 2003.

5. E. Bresson, O. Chevassut, D. Pointcheval and J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. *in Proc. 8th ACM Conference on Computer and Communication Secuirty (CCS)*, 2001.
6. E. Bresson, O. Chevassut, and D. Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange - The Dynamic Case. *Asiacrypt 2001, LNCS 2248*, pages 290 – 309, Springer-Verlag, 2001.
7. J. Katz and M. Yung. Scalable Protocols for Authenticated Group Key Exchange. *Proc. of Crypto 2003, LNCS 2729*, pages 110 – 125, Springer-Verlag, 2003.
8. J. Katz and M. Yung. Scalable Protocols for Authenticated Group Key Exchange. Full version.
9. W. Diffie and M. Hellman. New Directions In Cryptography. *IEEE Transactions on Information Theory*, vol. IT-22(6), pages 644 – 654, November, 1976.
10. I. Ingemarsson, D. T. Tang, and C. K. Wong. A Conference Key Distribution System. *In IEEE Transactions on Information Theory 28(5)*, pages. 714 – 720, 1982.
11. R. Dutta and R. Barua. Constant Round Dynamic Group Key Agreement . *Cryptology ePrint Archive, Report 2005/221*, 2005.
12. Z. Cheng, L. Vasiu, and R. Comley. Pairing-based one-round tripartite key agreement protocols . *Cryptology ePrint Archive, Report 2004/079*, 2004.
13. F. Zhang, S. Liu and K. Kim. ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings . *Cryptology ePrint Archive, Report 2002/122*, 2002.
14. X. Du, Y. Wang, J. Ge, Y. Wang. An Improved ID-based Authenticated Group Key Agreement Scheme . *Cryptology ePrint Archive, Report 2003/260*, 2003.
15. Y. Shi, G. Chen and J. Li. ID-Based One Round Authenticated Group Key Agreement Protocol with Bilinear Pairings. *International Conference on Information Technology: Coding and Computing (ITCC'05)*, - Volume I pages 757 – 761, 2005.
16. A. Joux. An one round protocol for tripartite Diffie-Hellman . *Proc. ANTS4, LNCS 1838*, pages 385 – 394, 2000.
17. X. Du, Y. Wang, J. Ge and Y. Wang. ID-Based Authenticated Two Round Multi-Party Key Agreement . *Cryptology ePrint Archive, Report 2003/247*, 2003.
18. L. Zhou, W. Susilo and Y. Mu. Efficient ID-based Authenticated Group Key Agreement from Bilinear Pairings (full version) . *Manuscript*, 2006.

A Observation of One-Round ID-Based Authenticated Group Key Agreement Protocol in [15]

Firstly, we review the One-round ID-based AGKA in [15].

Setup. GA generates a prime q, two groups $\mathbb{G}_1, \mathbb{G}_2$ of order q, and an bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Choose a random generator $P \in \mathbb{G}_1$. Then GA randomly picks $s_1, s_2 \in \mathbb{Z}_q^*$ and sets $P_{pub} = s_1 P, P'_{pub} = s_2 P$. GA then publishes system parameters $\mathsf{params} = \{\hat{e}, \mathbb{G}_1, \mathbb{G}_2, q, P, P_{pub}, P'_{pub}, H\}$.

Extract. When a user U_i with identity ID_{U_i} wishes to obtain a key pair, GA computes $I_i = H(ID_i)$, $Q_i = (I_i s_1 + s_2)P$ and the secret $S_i = (I_i s_1 + s_2)^{-1} P$, and returns S_i to the user U_i. Q_i is the user's public key.

Let $U_1, \ldots, U_n$ be the n users who want to establish a session key. The protocol is as follows:

Interacting. Each user U_i picks $a_i \overset{R}{\leftarrow} \mathbb{Z}_p^*$. Then U_i computes $T_i^j = a_i Q_j$, where $1 \leq j \leq n$ and $j \neq i$. Now U_i can check public key of each user:

$$Q_j \overset{?}{=} I_j P_{pub} + P'_{pub}$$

If the above equation holds, U_i can assume that U_j is a valid member of the group. Then U_i sends T_i^j to U_j.

Key Computation. Upon receipt of T_j^i, each U_i now can compute the common session key as follows:

$$K = K_i = \hat{e}(T_1^i + \cdots + T_{i-1}^i + a_i Q_i + T_{i+1}^i + \cdots + T_n^i, S_i) = \hat{e}(P, P)^{(a_1 + \ldots + a_n)}$$

Now we show how to attack the protocol above. An adversary $\mathcal{A}$ asks the GA for a key pair, and thus he gets $I_{\mathcal{A}} = H(ID_{\mathcal{A}})$, $Q_{\mathcal{A}} = (I_{\mathcal{A}} s_1 + s_2)P$ and the secret $S_{\mathcal{A}} = (I_{\mathcal{A}} s_1 + s_2)^{-1} P$, where $I_{\mathcal{A}} \neq I_i, 1 \leq i \leq n$. Then $\mathcal{A}$ chooses T_i^1 and T_j^1 which are two messages sent to U_i and U_j by U_1, and computes:

$$\begin{aligned} a_1 s_1 P &= (I_i - I_j)^{-1}(T_i^1 - T_j^1) \\ &= (I_i - I_j)^{-1}(a_1(I_i s_1 + s_2)P - a_1(I_j s_1 + s_2)P) \\ &= (I_i - I_j)^{-1}(a_1(I_i - I_j)s_1 P) \end{aligned}$$

$$\begin{aligned} a_1 s_2 P &= (I_i^{-1} - I_j^{-1})^{-1}(I_i^{-1} T_i^1 - I_j^{-1} T_j^1) \\ &= (I_i^{-1} - I_j^{-1})^{-1}(I_i^{-1} a_1(I_i s_1 + s_2)P - I_j^{-1} a_1(I_j s_1 + s_2)P) \\ &= (I_i^{-1} - I_j^{-1})^{-1}(a_1(I_i^{-1} - I_j^{-1})s_2 P) \end{aligned}$$

In the same way, $\mathcal{A}$ can get $a_2 s_1 P, \ldots, a_n s_1 P$ and $a_2 s_2 P, \ldots, a_n s_2 P$. Then $\mathcal{A}$ continues to compute:

$$T_i^{\mathcal{A}} = a_i s_1 P \cdot I_{\mathcal{A}} + a_i s_2 P = a_i (I_{\mathcal{A}} s_1 + s_2) P$$

$\mathcal{A}$ now can compute the group session key as follows:

$$K = K_i = \hat{e}(T_1^{\mathcal{A}} + \cdots + T_n^{\mathcal{A}}, S_{\mathcal{A}}) = \hat{e}(P, P)^{(a_1 + \ldots + a_n)}$$

From the description above, we can observe that if there is a user who is belong to this group and has the valid group key pair, he can know the group session key of any execution of the group key agreement protocol even if he is not involved in it. □

Efficient Augmented Password-Based Encrypted Key Exchange Protocol

Shuhua Wu and Yuefei Zhu

Department of Networks Engineering,
Zhengzhou Information Engineering Institute,
Zhengzhou 450002, China
wushuhua726@sina.com.cn

Abstract. In this paper, we propose an efficient augmented password-based encrypted key exchange protocol based on that of Bellovin and Merritt. The protocol is more efficient than any of the existing augmented encrypted key exchange protocols in the literature we can document and thus is popular in low resource environments. Furthermore, we have proved its security under the assumptions that the hash function closely behaves like a random oracle and that the computational Diffie-Hellman problem is difficult.

Keywords: password, encrypted key exchange, Diffie-Hellman assumptions.

1 Introduction

Password-based encrypted key exchange are protocols that are designed to provide pair of users communicating over an unreliable channel with a secure session key even when the secret key or password shared between two users is drawn from a small set of values. Humans directly benefit from this approach since they only need to remember a low-quality string chosen from a relatively small dictionary (e.g. 4 decimal digits). The vast majority of protocols found in practice do not account, however, for such scenario and are often subject to so-called dictionary attacks.

To address this problem, several protocols have been designed to be secure even when the pre-shared password is short. The seminal work in this area is the Encrypted Key Exchange (EKE) protocol proposed by Bellovin and Merritt in [1]. EKE is a classical Diffie-Hellman key exchange wherein the two flows are encrypted using the password as a common symmetric key. Then an attaker making a password guess could decrypt the symmetric encryption. Following EKE, many password authenticated key exchange protocols were proposed [2,3,4,5,6,7,8,9,10,11,12,13,14,15]. Some of these protocols were, in addition, designed to protect against server compromise, so that an attacker that was able to steal data from a server could not later masquerade as a user without having performed a dictionary attack. However, they contained only informal arguments for security and some of them were subsequently shown to be insecure, e.g. [16].

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 533–542, 2006.

Therefore, the importance of formal proofs of security should be emphasized to design protocols. In fact, some recent ones, e.g. [10,11,14], have been formally proven secure but they are not in the augmented mode with the exception of [15].

In this paper, we propose an efficient augmented password-only encrypted key exchange and provide a rigorous proof of security for our protocol based on the computaitonal Diffie-Hellman assumption. Our protocol is also a variation of their EKE protocol. In contrast to some previous work (e.g. SPEKE proposed in [14] recently), one of the primary advantages in our case is that users need remember only a short password, and no cryptographic key(s) of any kind: this is so since the public parameters can be "hard-coded" into any implementation of the protocol. However, some privious protocols such as SPEKE suffer from the disadvantage that the client must store server's public keys (and if the client will need to authenticate to multiple servers, the client must store multiple public keys); in some sense, this obviates the reason for considering password-based protocols in the first place: namely, that human users cannot remember or securely store long, high-entropy keys. This drawback has partially motivated our work.

Another primary advantage in our case is that our protocol is an efficient augmented password-based encrypted key exchange protocol. The protocol is remarkably efficient, requiring computation only a little more than "classical" Diffie-Hellman key exchange which provides no authentication at all. So far as we know, it is more efficient than any of the existing augmented EKE protocols we can document, even than some not in the augmented model. In the augmented password-based key exchange protocol, one party (commonly referred to as the client) has the password, while the other party (commonly referred to as the server) does not have the password. Instead, the server only has a password verification data derived using a function of the password. It is worthful for practical purposes because even an adversary obtains a password verification data from the server, the adversary still needs to launch offline dictionary attacks for getting the corresponding password.

To sum up, ours is the protocol for password-only authentication which is both practical and provably-secure based on the computaitonal Diffie-Hellman assumption. And the protocol is quite popular in low resource environments because of the remarkable efficiency. Finally, considering the fact that the protocol is often used in low resource environments, we implement it over elliptic curves (EC) because of the well-known advantages with regard to processing and size constraints. But the efficiency argument made in this paper does not stem from the use of elliptic curves because the performance comparison against some previous works is made under the assumption that all other related protocols are also implemented over EC.

2 Security Models for Password-Based Key Exchange

A secure password-based key exchange is a key exchange protocol where the parties use their password in order to derive a common session key sk that will be used to build secure channels. Loosely speaking, such protocols are said to be secure against *dictionary attacks* if the advantage of an attacker in distinguishing a real session key from a random key is less than $O(n/|\mathcal{D}|) + \epsilon(k)$ where $|\mathcal{D}|$ is the size of the dictionary $\mathcal{D}$, n is the number of active sessions and $\epsilon(k)$ is a negligible function depending on the security parameter k.

In this section, we briefly review the security model we will use in the rest of the paper to prove the security of our protocol. It is the same as the one defined by Bellare et al.[17] and is referred to as the Find-Then-Guess (FTG) model.

2.1 Communication Model

Protocol participants. Each participant in the password-based key exchange is either a client $C \in \mathcal{C}$ or a server $S \in \mathcal{S}$.

Long-lived keys. Each client $C \in \mathcal{C}$ holds a password pw_C. Each server $S \in \mathcal{S}$ holds a vector $pw_S = \langle pw_S[C] \rangle_{C \in \mathcal{C}}$ with an entry for each client, where $pw_S[C]$ is the transformed-password, as defined in [18]. In a symmetric model, $pw_S[C] = pw_C$, but they may be different in some schemes. pw_C and pw_S are also called the long-lived keys of client C and server S.

Protocol execution. The interaction between an adversary $\mathcal{A}$ and the protocol participants occurs only via oracle queries, which model the adversary capabilities in a real attack. During the execution, the adversary may create several concurrent instances of a participant. These queries are as follows, where U^i denotes the instance i of a participant U:

- $Execute(C^i, S^j)$: This query models passive attacks in which the attacker eavesdrops on honest executions between a client instance C^i and a server instance S^j. The output of this query consists of the messages that were exchanged during the honest execution of the protocol.
- $Send(U^i, m)$: This query models an active attack, in which the adversary may intercept a message and then either modify it, create a new one, or simply forward it to the intended participant. The output of this query is the message that the participant instance U^i would generate upon receipt of message m.

2.2 Security Definitions

Partnering. The definition of partnering uses the notion of session identifications (*sid*). More specifically, two instances U_1^i and U_2^j are said to be partners if the following conditions are met: (1) Both U_1^i and U_2^j accept; (2) Both U_1^i and U_2^j share the same session identifications; (3) The partner identification for U_1^i is U_2^j and vice-versa; and (4) No instance other than U_1^i and U_2^j accepts with a partner identification equal to U_1^i or U_2^j. In practice, the *sid* could be taken to be the partial transcript of the conversation between the client and the server instances before the acceptance.

Freshness. The notion of freshness is defined to avoid cases in which adversary can trivially break the security of the scheme. The goal is to only allow the adversary to ask *Test* queries to fresh oracle instances. More specifically, we say an instance U^i is fresh if it has accepted and if both U^i and its partner are unopened(The adversary has not made a *Reveal* query on them).

Semantic security in the Find-Then-Guess model. This is the definition currently being used in the literature. In order to measure the semantic security of the session key of user instance, the adversary is given access to two additional oracles: the *Reveal* oracle, which models the misuse of session keys by a user, and the *Test* oracle, which tries to capture the adversary's ability (or inability) to tell apart a real session key from a random one. Let b be a bit chosen uniformly at random at the beginning of the experiment defining the semantic security in the Find-Then-Guess (FTG) model. These oracles are defined as follows.

- $Reveal(U^i)$: If a session key is not defined for instance U^i or if a *Test* query was asked to either U^i or to its partner, then return $\perp$. Otherwise, return the session key held by the instance U^i.
- $Test(U^i)$:If no session key for instance U^i is defined, then return the undefined symbol $\perp$. Otherwise, return the session key for instance U^i if $b = 1$ or a random of key of the same size if $b = 0$.

The adversary in this case is allowed to ask multiple queries to the *Execute*, *Reveal*, and *Send* oracles in any order, but it is restricted to ask only a single query to the *Test* oracle. The goal of the adversary is to guess the value of the hidden bit b used by the *Test* oracle. The adversary is considered successful if it guesses b correctly.

Semantic security. Let Succ denote the event in which the adversary is successful. The **ftg-ake-advantage** of an adversary $\mathcal{A}$ in violating the semantic security of the protocol $\mathcal{P}$ in the FTG sense and the **advantage function** of the protocol $\mathcal{P}$, when passwords are drawn from a dictionary $\mathcal{D}$, are respectively

$$Adv_{\mathcal{P},\mathcal{D}}^{ftg-ake}(\mathcal{A}) = 2 \cdot Pr[\text{Succ}] - 1$$

and

$$Adv_{\mathcal{P},\mathcal{D}}^{ftg-ake}(t, R) = \max_{\mathcal{A}}\{\mathbf{Adv}_{\mathcal{P},\mathcal{D}}^{ftg-ake}(\mathcal{A})\},$$

where the maximum is over all $\mathcal{A}$ with time-complexity at most t and using resources at most R (such as the number of queries to its oracles). The definition of time-complexity that we use henceforth is the usual one, which includes the maximum of all execution times in the experiments defining the security plus the code size [19]. Note that the advantage of an adversary that simply guesses the bit b is 0 in the above definition due to the rescaling of the probabilities.

2.3 EC Diffie-Hellman Assumptions

We assume a finite cyclic group G of prime order q generated by the base point P in an elliptic curve $\mathcal{E}$. We also call the tuple $G = (\mathcal{E}, P, q)$ the represented group.

EC Computational Square Diffie-Hellman: ECCSDH. The EC computational Diffie-Hellman(ECCDH) assumption states that given uP and vP, where u and v are drawn at random from Z_q, it is hard to compute uvP. The EC computational square Diffie-Hellman(ECCSDH) problem is the particular case where $u = v$ and it is as hard as the basic computational Diffie-Hellman problem [11]. The ECCSDH problem can be defined more precisely by considering an experiment $\mathbf{Exp}_G^{eccsdh}(\mathcal{A})$, in which we select a value u in Z_q, compute $U = uP$ and then give U to an adversary $\mathcal{A}$. Let K be the output of $\mathcal{A}$. Then, the experiment $\mathbf{Exp}_G^{eccsdh}(\mathcal{A})$ outputs 1 if $K = u^2P$ and 0 otherwise. Then, we define advantage of $\mathcal{A}$ in violating the ECCSDH assumption as $Adv_G^{eccsdh}(\mathcal{A}) = Pr[\mathbf{Exp}_G^{eccsdh}(\mathcal{A}) = 1]$ and the advantage function of the group, $Adv_G^{eccsdh}(t)$, as the maximum value of $Adv_G^{eccsdh}(\mathcal{A})$ over all $\mathcal{A}$ with time-complexity at most t.

3 The EC Based Password Encrypted Key Exchange

In this section, we introduce the EC based password encrypted protocol in the augmented mode. The protocol is remarkably efficient, requiring computation only a little more than “classical” Diffie-Hellman key exchange which provides no authentication at all.

3.1 Description

Our protocol is an augmented password-based encrypted key exchange protocol. In the augmented password-based key exchange protocol, the client has the password, while the server does not have the password. Instead, the server only has a password verification data derived using a function of the password. It is worthful for practical purposes to reduce the risk of corruption of the server. Corruption of a server occurs when an attacker gains access to the server's local database of passwords. If client's passwords are stored directly in the database, then the attacker can immediately use any of these passwords to impersonate these clients. Fortunately, our augmented password-based key exchange protocol is designed to prevent an attacker from doing just that because even he obtains a password verification data from the server, the attacker still needs to launch offline dictionary attacks for getting the corresponding password. Although this mechanism will not prevent an adversary from mounting (off-line) dictionary attacks, it will slow him down and thus give the server's administrator time to react appropriately and to inform its clients.

Our protocol is also a variation of the password-based encrypted key exchange protocol of Bellovin and Merritt [1], in which we replace the encryption function $\varepsilon_{pw}(\cdot)$ with a simple EC-based one-time pad function. More specifically, whenever a user C wants to send the encryption of a value xP to a server S, it does so by computing $X^\star = xP - pwQ$, where the password pw is assumed to be in Z_q and held by the client while $(PW = pwQ, R = pw^{-1}P)$ held by the server instead. And whenever a server S receives the encryption $X^\star$ and he just needs to decrypt it by computing $xP = X^\star + PW$. In return, the server encrypts yP as $Y^\star = yR + PW$ and sends the latter to the client. Only when the client knows exactly pw itself can he recover yP from $Y^\star$. The session key is set to be the hash (random oracle) of the user identities, the session identification, their password and the Diffie-Hellman key, where the session identification is defined as the transcript of the conversation between C and S. The protocol is simple and efficient and thus quite popular in low resources environments. The full description of our protocol is given in Figure 1, where G is a represented group and Q is an element in the group; l is a security parameter; and $H : \mathcal{C} \times \mathcal{S} \times G^4 \rightarrow \{0,1\}^l$ is a random oracle.

$$\text{Public information: } Q \in G, |G| = q, H$$
$$\text{Secret information: } pw \in Z_q, R = pw^{-1}P, PW = pwQ$$
$$\begin{array}{rcl}
\text{Client } C & & \text{Server } S \\
x \overset{R}{\leftarrow} Z_q; & & y \overset{R}{\leftarrow} Z_q; \\
X^\star = xP - pwQ & & Y^\star = yR - PW \\
& \xrightarrow{X^\star, \quad C} & \\
& \xleftarrow{Y^\star, \quad S} & \\
K_C \leftarrow (x \cdot pw)(Y^\star + PW) & & K_S \leftarrow y(X^\star + PW) \\
SK_C \leftarrow H(C, S, X^\star, Y^\star, PW, K_C) & & SK_S \leftarrow H(C, S, X^\star, Y^\star, PW, K_S)
\end{array}$$

Fig. 1. The password-based encrypted key exchange protocol

The correctness of our protocol follows from the fact that, in an honest execution of the protocol, $K_S = K_C = xyP$.

Finally, we should point out that Q is an important parameter and should be chosen carefully in such a way that it is computationally difficult for an adversary to find the discrete logarithm of Q with P as the base. Otherwise, the protocol will be insecure.

3.2 Security

As Theorem 1 states, our EC based password key encrypted protocol is secure in the random oracle model as long as we believe that the ECCDH problem is hard in G.

Theorem 1. *Let G be a represented group and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let $\mathcal{P}$ describe the password-based encrypted key exchange protocol associated with these primitives as defined in Figure 1. Then,*

$$Adv_{\mathcal{P},\mathcal{D}}^{ftg-ake}(t, q_p, q_s, q_h) \le \frac{(q_p+q_s)^2}{q} + \frac{q_h^2}{2^l} + 2q_h(1 + 2q_sq_h)\boldsymbol{Adv}_G^{eccsdh}(t + 3\tau) + \frac{4q_s}{|\mathcal{D}|},$$

where q_s represents the number of active interactions with the parties (Send-queries); q_p represents the number of passive eavesdroppings (Execute-queries); q_h represents the number of hash queries to H; and τ represents the computational time for a point multiplication in G.

Similarly, we use the method proposed in [11] to prove Theorem 1 because the method is comprehensible and less prone to errors.

Proof. Let $\mathcal{A}$ be an adversary against the semantic security of $\mathcal{P}$. The idea is to use $\mathcal{A}$ to build adversaries for each of the underlying primitives in such a way that if $\mathcal{A}$ succeeds in breaking the semantic security of $\mathcal{P}$, then at least one of these adversaries succeeds in breaking the security of an underlying primitive. Our proof consists of a sequence of hybrid experiments, starting with the real attack and ending in an experiment in which the adversary's advantage is 0, and for which we can bound the difference in the adversary's advantage between any two consecutive experiments. In the following experiments, we study the event which occurs if the adversary correctly guesses the bit b involved in the $Test$-query.

Experiment$_0$: This is the real protocol in the random-oracle model. By defination, we have

$$Adv_{\mathcal{P},\mathcal{D}}^{ftg-ake}(\mathcal{A}) = 2Pr[S_0] - 1 \tag{1}$$

Experiment$_1$: In this experiment, we simulate the hash oracle H, but also additional private hash function $H^{'}$ that will appear in the **Experiment$_3$** as usual. The hash function $H^{'}$ is computed only with the user identities and the session identification but no password or Diffie-Hellman key any longer. We also simulate all the instances, as the real players would do, for the $Send$-queries and for the $Execute$, and $Test$-queries. From this simulation, we easily see that the game is perfectly indistinguishable from the real attack. Thus, we have

$$Pr[S_1] = Pr[S_0] \tag{2}$$

Experiment$_2$: For an easier analysis in the following, we cancel experiments in which some unlikely collisions appear: collisions on the partial transcripts and on hash values. The probability are bounded by the birthday paradox:

$$|Pr[S_2] - Pr[S_1]| \leq Pr[\text{Collision}] \leq \frac{(q_p + q_s)^2}{2q} + \frac{q_h^2}{2^{l+1}} \tag{3}$$

Experiment$_3$: We compute the session key sk using the private oracles $H^{'}$. The experiments **Experiment$_3$** and **Experiment$_2$** are indistinguishable unless some specific hash queries are asked, denoted by event AskH. The session key is computed with the random oracle $H^{'}$ that is private to the simulator, then one can remark that the bit b involved in the $Test$-query cannot be guessed by the adversary, better than at random for each attempt. Hence the advantage of the adversary in this case is 0.

$$|Pr[S_3] - Pr[S_2]| \leq Pr[\text{AskH}] \qquad Pr[S_3] = \frac{1}{2} \tag{4}$$

To bound the difference between this experiment and previous, Our goal at this point shifts to computing the probability of the event AskH.

Experiment$_4$: In order to evaluate the event AskH, we introduce a random square Diffie-Hellman instance U and let $Q = U$ in this experiment. By using a technique similar to that used in lemma 2 and 3 in [11], one can show that

$$Pr[\text{AskH}] \le q_h \cdot \mathbf{Adv}_G^{eccsdh}(t+3\tau) + 2q_s q_h^2 \mathbf{Adv}_G^{eccsdh}(t+3\tau) + \frac{2q_s}{|\mathcal{D}|} \quad (5)$$

Finally, Combining all the above equations, one gets the announced result as follows.

$$\begin{aligned} &Adv_{\mathcal{P},\mathcal{D}}^{ftg-ake}(\mathcal{A}) = 2Pr[S_0] - 1 = 2(Pr[S_0] - \tfrac{1}{2}) \\ &= 2(Pr[S_1] - \tfrac{1}{2}) \le 2(|Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\le \tfrac{(q_p+q_s)^2}{q} + \tfrac{q_h^2}{2^l} + 2q_h(1+2q_sq_h)\mathbf{Adv}_G^{eccsdh}(t+3\tau) + \tfrac{4q_s}{|\mathcal{D}|}. \end{aligned}$$

□

3.3 Remarks

Our scheme is considered much more from the practical perspective. It assumes only public parameters — i.e., a " reference string" — which can be "hard-coded" into an implementation of the protocol. The primary advantage in our case is that users need remember only a short password, and no cryptographic key(s) of any kind, when compared with some previous work, say, protocol proposed recently in [14]. As a result, human users do not have to remember or securely store long, high-entropy keys. Moreover, our protocol is in the augmented model which is contrived to resist server compromise when compared to the two very recent ones [11,14].

Furthermore, our protocol is efficient. In one run of the scheme, the server side requires to compute two scalar multiplications and the client side requires three scalar multiplications. Note that we just count the number of scalar multiplication, which entails the highest computational complexity, and neglect the computational complexity of all other operations , which can be done efficiently. The protocol is remarkably efficient, requiring computation a little more than "classical" Diffie-Hellman key exchange which provides no authentication at all and requires two scalar multiplications on each side. So far as we know, it is more efficient than any of the existing augmented EKE protocols we can document, even than some not in the augmented model, say, protocol proposed recently in [14] which require four scalar multiplications on each side. Password-based protocols designed in the augmented model are much less efficient than those are not in that model, in terms of either computation or communication costs. The protocol proposed recently in [15] is in the augmented mode and requires five scalar multiplications for each side. Our protocol is certainly more efficient than it.

Finally, we should note that the protocols proposed in [11,15] require full-domain hash functions onto the represented group G. Such hash functions are difficult to implement directly in practice, which usually contains an implicit scalar multiplication over G. In that case, the computation cost of such hashes can not be neglected. Our protocol does not need such hash functions.

4 Conclusion

We have presented the protocol for password-only authentication which is both practical and provably-secure based on the computational Diffie-Hellman assumption. So far as we know, it is more efficient than any of the existing augmented EKE protocols we can document, even than some not in the augmented model. Therefore, our protocol is quite popular in low resource environments due to the remarkable efficiency.

Acknowledgement. The authors would like to thank anonymous reviewers for their valuable suggestions and comments that highly improve the readability and completeness of the paper. This research was partially supported by the National Science Foundation of the Republic of China, under contract NSC no.60473021.

References

1. S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In 1992 IEEE Symposium on Security and Privacy, pages 72-84. IEEE Computer Society Press, May 1992.
2. S. M. Bellovin and M. Merritt. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In ACM Security (CCS'93), pages 224-250.
3. L.Gong. Optimal authentication protocols resistant to password guessing attacks. In 8th IEEE Computer Security Foundations Workshop, pages 24-29, 1995.
4. D.Jablon. Strong password-only authentication key exchange. ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996.
5. D.Jablon. Extended password key exchange protocols immune to dictionary attack. In WETICE'97 Workshop on Enterprise Security, 1997.
6. S.Lucks. Open key exchange: How to defeat dictionary attacks without encrypting public keys. In proceedings of the Workshop on Security Protocols,1997.
7. Colin Boyd, Paul Montague, Khanh Quoc Nguyen. Elliptic Curve Based Password Authenticated Key Exchange Protocols. ACISP 2001: 487-501.
8. Duncan S. Wong, Agnes Hui Chan, Feng Zhu: Password Authenticated Key Exchange for Resource-Constrained Wireless Communications (Extended Abstract). ICN (2) 2005: 827-834
9. M. Bellare and P. Rogaway. The AuthA protocol for password-based authenticated key exchange. Contributions to IEEE P1363, Mar. 2000.
10. E. Bresson, O. Chevassut, and D. Pointcheval. Security proofs for an efficient password-based key exchange. In ACM CCS 03. ACM Press, Oct. 2003.
11. E. Bresson, O. Chevassut, and D. Pointcheval. New security results on encrypted key exchange. In F. Bao, R. Deng, and J. Zhou, editors, PKC 2004, volume 2947 of LNCS, pages 145-158. Springer-Verlag, Mar. 2004.
12. K. Kobara and H. Imai. Pretty-simple password-authenticated key-exchange under standard assumptions. IEICE Transactions, E85-A(10):2229-2237, Oct. 2002. Also available at http://eprint.iacr.org/2003/ 038/.
13. P. D. MacKenzie. The PAK suite: Protocols for password-authenticated key exchange. Contributions to IEEE P1363.2, 2002.

14. Michel Abdalla and David Pointcheval. Simple Password-Based Encrypted Key Exchange Protocols. In A. J. Menezes Ed., Topics in Cryptology - CT-RSA 2005, LNCS 3376, pages 191-208, Springer-Verlag.
15. M. Abdalla, O. Chevassut, and D. Pointcheval. One-time verifier-based encrypted key exchange. In V. Serge, editor, Proceedings of the 8th International Workshop on Theory and Practice in Public Key (PKC '05), volume 3386 of Lecture Notes in Computer Science, pages 47–64. Springer-Verlag, 2005.
16. S. Patel. Number theoretic attacks on secure password schemes. In proceedings of IEEE Security and Privacy, pages 236-47,1997.
17. Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In 38th Annual Symposium on Foundations of Computer Science, pages 394-403, Miami Beach, Florida, October 19-22, 1997. IEEE Computer Society Press.
18. Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Bart Preneel, editor, Advances in Cryptology - EUROCRYPT 2000, volume 1807 of Lecture Notes in Computer Science, pages 139-155, Bruges, Belgium, May 14-18, 2000. Springer-Verlag, Berlin, Germany.
19. Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, Topics in Cryptology - CT-RSA 2001, volume 2020 of Lecture Notes in Computer Science, pages 143-158, San Francisco, CA, USA, April 8-12, 2001. Springer-Verlag, Berlin, Germany.

Secure Relative Location Determination in Vehicular Network

Lei Tang, Xiaoyan Hong, and Phillip G. Bradford

Department of Computer Science, The University of Alabama,
Box 870290, Tuscaloosa, AL 35487-0290
{ltang, hxy, pgb}@cs.ua.edu

Abstract. Relative location information is very useful in vehicular networks although it is vulnerable to various attacks. Many techniques have been proposed for relative positioning and location verification. Due to the high speed and the strict security requirements, the existing relative positioning and location verification techniques are not directly applicable to vehicular networks. Hence we present a scheme called *SRLD*, which securely determines the relative locations of a set of wirelessly connected vehicles based on the relative locations of each vehicle's surrounding vehicles. *SRLD* uses cryptographic keys to authenticate location messages and uses a vehicle's public key to identify the vehicle while protecting drivers' privacy. To defend against Sybil attacks, *SRLD* employs registration and ticket verification mechanisms. It defends Wormhole and black hole attacks by probabilistically monitoring losses of relative location messages. Analysis and simulation results show that *SRLD* is lightweight and is resilient to Sybil, Wormhole and some other attacks.

Keywords: secure vehicular relative location, security, vehicular networks.

1 Introduction

Location information is very useful in our daily life. But in many cases, we do not need detailed global location information but only relative location information. The fatality analysis report [1] by National Highway Traffic Safety Administration shows that collision with another motor vehicle is the most common harmful event for fatal and injury crashes. Aided by an accurate view of the relative locations of vehicles nearby, a driver will have better chance discovering vehicles at blind spots and avoiding accidents during lane changes and merges. Furthermore, in vehicular networks, nodes' relative location information can be used to identify the relative location of a message source so that a vehicle driver will be able to tell the relative position of the vehicle that is sending a passing/decelerating message and take corresponding maneuver to prevent the accidents.

In vehicular networks, location information is life-critical but is vulnerable to malicious attacks such as Sybil [16] and Wormhole attacks [9]. To make sure that

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 543–554, 2006.

the location information is not forged or altered by malicious nodes, we need to determine the nodes' location and verify the authenticity of their location claims in presence of malicious nodes.

The vehicles' relative locations can be computed from their accurate global locations, which can be obtained by using GPS-based techniques. But since a GPS satellite simulator is able to generate fake GPS signals that flood the real GPS signals [6], GPS-based solutions are not secure in vehicular networks without authenticating GPS signals. In addition, using vehicles' precise locations to compute their relative locations may raise privacy concern of the drivers who are unwilling to expose their precise locations.

Many non-GPS based positioning and distance estimation techniques have been proposed [12, 22, 5, 2, 3, 18, 7] to determine the relative locations of nodes. However, due to the fact that vehicles are moving at high speed, all of the above mentioned positioning techniques except [12] are not directly applicable for vehicular networks since most of them are designed for in-building environment. Furthermore, all these techniques assume that all nodes are cooperative. Hence these techniques are vulnerable to various attacks.

To authenticate nodes' location claims and determine nodes' relative locations in a hostile environment, [4, 21, 8, 6] have been proposed. They can be classified into two types. One type exploits the properties of radio and sound wave and multilateration techniques [4, 21, 6, 14]. The other uses cryptographic keys to authenticate location information [13]. The techniques using multilateration may be impractical for vehicular networks because most of time the number of nodes in the proximity is too few to perform multilateration. And the techniques using ultrasonic sound may be inaccurate since the vehicles are moving in a speed about 1/10 of the speed of sound wave.

Therefore, in this article, we propose a scheme to securely determine the nodes' relative locations in the vehicular network, named *Secure Relative Location Determination* (*SRLD*). *SRLD* is distinguished from existing relative positioning schemes in that it does not require GPS or other location information but only the relative locations of each vehicle's surrounding vehicles. Essentially, with the technique introduced in the article, every node is able to construct an image of the relative locations of a set of nearby nodes that are wirelessly connected.

SRLD uses cryptographic keys to authenticate location messages and uses a vehicle's public key for identification and privacy protection. To defend against Sybil attacks, it employs registration and ticket verification mechanism. We also design a scheme to defend against Wormhole attacks by probabilistically monitoring losses of relative location messages.

The rest of the article is organized as follows. Section 2 introduces the existing relative positioning and location verification techniques. Section 3 describes the system model and the problem statement. Section 4 presents the design of *SRLD* scheme and section 5 analyzes the resilience of *SRLD* against Sybil, Wormhole and black hole attacks. The performance of *SRLD* is simulated in section 6. Finally, we conclude in section 7.

2 Previous Work

Securely determining nodes' relative locations include determining relative locations and verifying the authenticity of relative locations in presence of malicious nodes. The first problem is referred as relative positioning problem and the second problem is referred as relative position verification problem.

2.1 Relative Positioning Techniques

The nodes' accurate positions obtained using GPS devices can be used to compute their relative locations of nodes. In [12], V. Kukshya et al. presented a technique to estimate the relative locations of neighboring vehicles based on the exchange of their individual GPS coordinates. And it uses a trilateration technique to estimate relative locations during GPS outages. The relative positioning solution in [12] requires vehicles to cooperate and does not consider security issues so it is vulnerable to various types of attacks. Moreover, GPS devices can be spoofed by GPS satellite simulators [6] , which generate fake GPS signals that overcome the real GPS signals [23].

There are a number of relative positioning techniques [5, 18, 2] that exploit radio beacons or ultrasonic pulse to infer proximity to a collection of reference points with known coordinates. However, due to the fact that vehicles are moving in a speed about 1/10 of the sound wave speed (about 331 m/s), the above mentioned positioning techniques may be inaccurate in vehicular network scenario.

Furthermore, there are some IEEE 802.11 wireless network based positioning techniques [3, 7], which learn the location of wireless devices by studying the radio signal property observed at base stations.

2.2 Position Verification Techniques

Positioning techniques introduced in 2.1 are vulnerable to malicious attacks. For instance, attackers may give false positions. Many techniques have been proposed to verify positions and to prevent malicious attacks. They can be classified into two types. One type exploits the properties of radio and sound wave and multilateration technique [4, 21, 6, 14]. The other uses cryptographic key to authenticate location information [13]. We summarize some of them below.

S.Brands and D. Chaum presented a protocol to determine an upper-bound on the distance between the verifier and the prover [4]. D. Liu et al. presented two attack-resistant location estimation techniques [14] provided that the benign beacon signals account for the majority. L. Lazos et al. [13] proposed a secure localization scheme (SeRLoc) for wireless sensor networks based on directional antennas.

However, the above-mentioned location verification techniques are not directly applicable to the vehicle networks. First, the multilateration techniques are not suitable for vehicular network because often the number of nodes in the proximity is too few to perform multilateration. And directional antennas are not efficient when used on the linear topology of vehicular networks.

3 System Model

3.1 Problem Statement and Assumptions

First of all, the notations and terminologies used in this article are defined as follows.

- N: the number of vehicles in the network
- PK_v: public key of vehicle v;
- SK_v: private key of vehicle v;
- LP_v: license plate of vehicle v;
- T_v: authentication ticket vehicle v;
- R: registration interval;

We study the problem of securely determining relative locations in vehicular networks in presence of malicious nodes. Furthermore, we explore the problem of determining the vehicles' relative locations with the following design goals: 1) resistant to fake location claims, 2) decentralized relative location determination, meaning that each node computes its own image of the network topology and the images may vary. Moreover, we focus on determining the relative location among a set of vehicles that are wirelessly connected.

We list next the assumptions in our relative location determination protocol. These assumptions follow the common practices of the contemporary public key infrastructure. In this article, when we use the term *node*, we mean a vehicle in the network.

1. Vehicles are able to verify the Certification Authority (CA) certificates of the roadside APs (*Access Points*). And the communications between the nodes and the roadside APs are encrypted using asymmetric cryptography.
2. Each vehicle has a tamper-proof electronic license plate, which can only be read by roadside APs. Every R seconds, vehicles register to the roadside APs to indicate that they are active. During registration, APs read the vehicle's electronic license plate and update its registration time.
3. A vehicle's public key [20] is uniquely linked to its license plate. And the roadside APs verify the binding between a vehicle's license plate number and its public key by accessing the interfaces provided by transportation authorities.

Assumption 1 can be implemented by adapting *Secure Socket Layer*(*SSL*) scheme [17], through which a node can both verify that the AP is not spoofed by malicious attackers and negotiate an asymmetric cryptographic key.

For assumption 3, we follow earlier work [11] that vehicles' public keys and license plates are registered and verifiable at transportation authorities.

3.2 System Architecture

The vehicular network consists of road-side APs and wireless communication enabled vehicles. The APs are connected through wired Internet and they collectively provide a wireless radio that covers the entire road. Vehicles carry public keys of Certification Authority (CA) to verify CA signatures of APs. Fig. 1 depicts such an architecture.

Relative locations of nodes are stored in a table called *Relative Location Table* (*RLT*). The structure of *RLT* is as follows.

$$\{\text{Relative Location, Public Key, Seq}\}$$

In the table, the *Public Key* field records the public key of a vehicle whose link to a license plate has been proved by the APs. The *Relative Location* field specifies the relative location of that vehicle. The *Seq* field records the sequence number of the relative location beacon message received from the corresponding vehicle.

In our scheme, we use the vehicle's public key to identify the vehicle rather than using its license plate number. Using a vehicle's license plate number as the vehicle identifier will expose the license plate numbers of the vehicles on the road, which may raise privacy concerns. Using public key as identifier has the advantage of not revealing the vehicle privacy information to other vehicles since the link between a vehicle's public key and its license plate number is only verifiable through the interface provided by transportation authority, which is not accessible for normal people.

There are two types of messages in the network. One is the relative location beacon messages. The other is the generic messages such as deceleration message and other general communication messages. In this article, when we use the term *message*, we mean generic message.

Given the *RLT*, we can determine the relative location of message source. To prevent malicious nodes from fabricating messages, every message is signed by the message source using its private key. When a vehicle receives a message, it will first locates the relative location of the message source according to the public key of the message source. Then the message receiver verifies the validity of the message source's signature using the public key of the message source stored in the *RLT*.

In next section, we introduce how to construct *RLT* through *SRLD*.

4 Design of the *SRLD* Protocol

Fig. 2 illustrates a vehicular network comprising multiple lanes. The rationale of *SRLD* scheme is to construct a graph showing the relative locations of nodes within radio range by using the relative location information of each node's surrounding nodes. Specifically, each node will use video sensor to read the license plates of nodes on front left, front, front right, rear left, rear and rear right (i.e. node 1,2,3,6,7,8 in Fig. 2). And each node uses directional RFID reader to read the electronic license plates of the nodes between front left and rear left(i.e. node 4 in Fig. 2) and the nodes between front right and rear right. If every node propagates the relative locations of its surrounding nodes to other nodes, then eventually every node will be able to build a graph showing the relative locations of all nodes in the network.

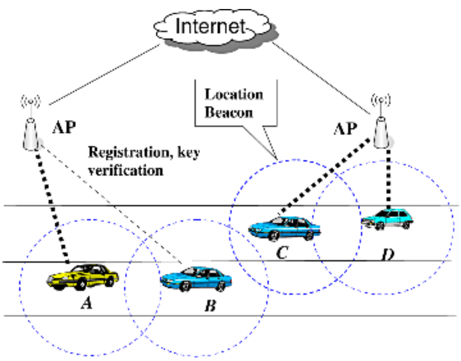


Fig. 1. System Architecture

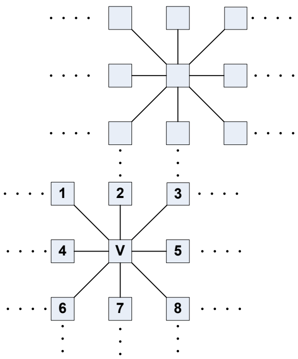


Fig. 2. Relative Locations Graph

When most of time all nodes are on a single lane, then each node only needs to propagate its predecessor and successor information instead of relative locations of all surrounding nodes. Due to the limit of space and the reason that the rationale of determining vehicle relative locations in multi-lane scenario is essentially the same as in single-lane scenario, we describe our scheme in a single-lane scenario.

4.1 SRLD Protocol

SRLD protocol is showed in Fig. 3. *SRLD* may work in two modes: *AP mode* and *distributed mode*, which are called *SRLD-AP* and *SRLD-D*, respectively. Their difference is as follows. In *SRLD-AP*, APs compute the *RLT* and propagate it to vehicles. In *SRLD-D*, vehicles distributedly construct *RLT* by exchanging the relative locations of each vehicle's surrounding vehicles.

In the first phase of *SRLD-AP* and *SRLD-D*, a vehicle v in the vehicular network obtains the license plate number LP_p of its immediate predecessor p using the video cameras mounted on the front of v. Then v requests an authentication tickets T_p from p and send T_p to a AP to authenticate it. The format of T_p is as follows: $T_p = \{PK_p,\ TS,\ E_{SK_p}(LP_p||TS)\}$.

When verifying the validity of T_p, AP first check the binding between LP_p and PK_p by accessing the interface provided by transportation authorities. Furthermore, AP verifies that the timestamp TS in T_p does not exceed 10 seconds to prevent stale tickets. Similarly, v requests a ticket T_s from its immediate successor and checks the validity of T_s. If working in *SRLD-AP* mode, AP will also record the predecessor and successor information of the node v when checking the tickets T_p and T_s.

The second phase of *SRLD-AP* is already showed in the Fig. 3 so we focus on introducing the second phase of *SRLD-D*, during which every vehicle instead of AP disseminates its immediate predecessor and successor information to other nodes using relative location beacon message B. Suppose B is generated by vehicle v, whose predecessor is p and successor is s. The format of B is as follows.

SRLD **protocol**

Verification Phase:

1. Node v observes the license plate number of its immediate predecessor p and successor s: LP_p and LP_s. Then node v sends LP_p to p and requests for an authentication ticket T_p from p. Similarly node v requests an authentication ticket T_s from s.
2. Node v sends $\{LP_p,\ T_p\}$ and $\{LP_s,\ T_s\}$ to AP to authenticate p and s.

Dissemination Phase of SRLD-D:

1. After verifying p and s, v broadcasts a relative location beacon B to neighbors.
2. When a node receives B, it updates its RLT using B and broadcasts B after verifying the validity of B.

Dissemination Phase of SRLD-AP:

1. APs compute the RLT from the predecessor and successor information collected during the verification phase and disseminate $\{RLT, sig_{AP}\}$ to all nodes. And nodes verify the validity of RLT by checking the signature sig_{AP} using public key of APs.

Fig. 3. *SRLD* Protocol

PMLD **protocol**

1. When an AP receives a predecessor/successor authentication request, it probabilistically determines if it monitors the beacon message B_{select} corresponding to this authentication request. If it determines to monitor B_{select}, it conducts the following steps to monitor which nodes faithfully forward the B_{select}.

2. Notify all APs to monitor who is forwarding B_{select} within $T_{monitor}$. Here we set $T_{monitor}$ as 30 seconds.

3. When an AP receives a beacon B, it checks whether B is the same as B_{select}. If B is the same as B_{select}, then AP records the identity of the forwarder of B.

4. After $T_{monitor}$, APs know that who have forwarded B_{select} and who have not. APs then increment the *malicious value* of those nodes that did not forward B_{select}. After the *malicious value* of a node x reaches a threshold value, AP informs all nodes that x is a possible malicious node.

Fig. 4. *PMLD* Protocol

$$B = \{PK_p,\ PK_v,\ PK_s,\ T_p,\ T_s,\ seq,\ sig'\}$$
$$sig' = E_{SK_v}(\ H(\ \{PK_p,\ PK_v,\ PK_s,\ T_p,\ T_s,\ seq\}))$$

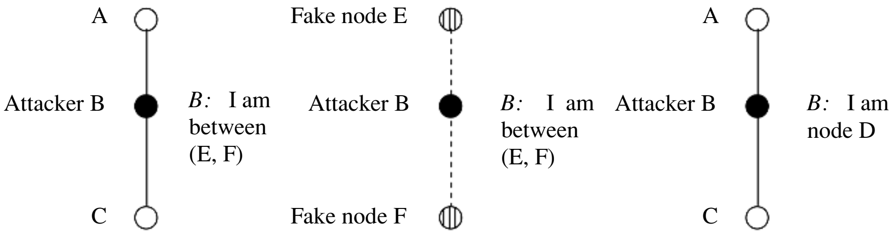
When other nodes receive B from v, they will verify the validity of B. First of all, receivers will check that the signature sig' is valid. Then they communicate with APs to verify the validity of the T_p and T_s and verify that the registrations of p, v and s are within R. If any of the above verifications fails, the receivers will ignore B.

We assume there are N vehicles in a linear topology network and T_h is the time needed for propagating B to a one-hop neighbor and processing it. We now compute how long it takes for a location change to reach all vehicles. The largest number of hops traveled by B ranges from $N-1$ to $\lceil \frac{N}{2} \rceil$. So on average the time for a location change to reach all vehicles is as follows:

$$T_h \times \frac{1}{N} \times (N-1+N-2+\ldots+\frac{N}{2}+(\frac{N}{2}+1)+\ldots+N-1) = 0.75 \times N \times T_h.$$

4.2 RLT Construction Algorithm

Now we present an algorithm to construct RLT using the relative location beacons received. RLT construction algorithm is executed by APs when working in



(a) Inject wrong location info (b) Fake non-existent nodes (c) Impersonate legitimate node

Fig. 5. Attacks against relative location determination schemes

SRLD-AP mode and is executed by individual vehicles when working in *SRLD-D* mode. The algorithm is as follows and its time complexity is $O(N^2)$.

1. From beacon messages received, search for a node h which has no predecessor. Add h to list L as list head. Make pointer P point to h.
2. Find the node s which is the successor of the node pointed by P. Then add s to list L and make pointer P point to s.
3. Continue step 2 until all nodes in the beacon messages are added to the list L. The relative location of a node in the list L is its relative location in the RLT.

5 Security Analysis

In this section, we analyze the resilience of *SRLD* to Sybil attacks, Wormhole attacks, denial-of-service attacks and black hole attacks. The goal of the *SRLD* protocol is to verify that the relative location information is not forged or altered by malicious nodes.

5.1 Sybil Attack

A Sybil attack occurs when a malicious node illegitimately takes on multiple identities as Sybil nodes [16]. First, malicious nodes may spoof roadside APs. Second, adversaries may lie about its predecessor and successor (Figure 5(a)). Third, adversaries may inject relative location information of non-existent nodes (Figure 5(b)). Last, attackers may impersonate legitimate nodes (Figure 5(c)).

For the first type attacks, since nodes will first verify the CA certificate of roadside APs and the communication between roadside APs and nodes are encrypted using asymmetric cryptography, it is hard for the attackers to impersonate APs, alter message content, or fabricate messages between APs and vehicles.

For the second type of attacks, in *SRLD*, it is difficult for an attacker to lie about its predecessor and successor since forging the tickets of predecessor or successor is challenging. And the attacks by reusing the stale tickets will be defended by checking whether the timestamps of the tickets have been expired.

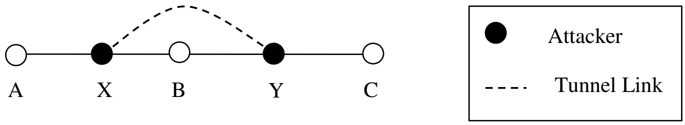


Fig. 6. An example of Wormhole attack

For the the third type of attacks, malicious nodes are unable to insert non-existent nodes because the receivers will verify the registrations of the nodes included in the relative location beacons by consulting roadside APs. Moreover, a node's registration will expire after R, which makes it difficult for adversaries to re-use stale tickets overheard previously.

For the fourth type of attacks, since the receiver verifies the authenticity of the public key of the beacon source and the signature of the message, it is difficult for malicious nodes to impersonate legitimate nodes and alter the beacon messages sent by legitimate nodes unless they know the private key of the source.

5.2 Wormhole Attack

Another significant attack is a Wormhole attack, where malicious nodes collude to selectively discard relative location messages of legitimate nodes. Figure 6 illustrates a basic Wormhole attack. The attackers control node X and Y, which are connected by a tunnel link. Regular messages and relative location messages received by X are tunneled to Y and retransmitted at Y, and vice versa. By selectively discarding messages, colluding attackers may launch DoS attacks and prevent some nodes from being known to others. For instance, X and Y may only transmit relative location messages initiated by A and C while discarding all relative location messages initiated by B. Thus, other nodes will not know the presence of B.

Some countermeasures have been presented to defend the Wormhole attacks [24, 9]. Y. Hu et al. proposed a MAC layer protocol named TIK [24] to restrict the packet's maximum allowed transmission distance, which prevents Wormhole attacks by detecting if the packet traveled further than that is allowed. In [9], the authors presented an approach to detect Wormhole attack, which depends on nodes maintaining accurate sets of their neighbors.

However, there is no solution designed specifically for defending Wormhole attacks in vehicular network. Hence, we propose *Probabilistic Message Loss Detection (PMLD)* protocol, which defend Wormhole attacks by probabilistically monitoring the losses of relative location messages. When working in *SRLD-AP* mode, APs instead of the vehicles propagate relative locations so the attackers are unable to launch Wormhole attacks. But *PMLD* protocol can be used to defend Wormhole attacks when the system works in *SRLD-D* mode.

PMLD protocol is showed in Fig.4. In *PMLD*, we assume legitimate nodes account for majority and if a node A can hear node B then B can hear A. APs probabilistically select a beacon message B_{select} and check if there are attackers discarding the selected beacon message. Since each beacon message will be transmitted by every node in the network, malicious nodes expose themselves when they discard B_{select}. In *PMLD* protocol, monitoring is performed probabilistically so that malicious nodes will not know which messages are going to be monitored.

5.3 Black Hole Attack

An attack similar to Wormhole attack is Black hole attack [10], in which a malicious node behaves like a black hole and discards all or a fraction of the relative location beacons passing it. Black hole attacks may create network partition so that a vehicle is unable to know the relative location of interested vehicles due to the network partition.

Black hole attackers can be detected by neighboring nodes, which identify and put the attackers on blacklist. However, as Y. Hu et al. pointed out in [10], the above watchdog-like method [15] may enable attackers to add legitimate nodes to blacklists and interfere the normal function of legitimate nodes.

In our system, we employ *PMLD* protocol as the countermeasure of Black hole attacks. The APs identify the black hole attackers by probabilistically monitoring message transmissions and inform legitimate nodes about the attackers. Compared with watchdog-like method, our approach exploits the authority of APs and will not cause legitimate nodes to be blackmailed by attackers.

5.4 Replay Attack and Denial-of-Sevice Attack

During Replay attacks, attackers retransmit stale messages recorded previously. In *SRLD-AP*, since *RLT* is transmitted by APs with asymmetric cryptography, it is difficult to launch Replay attacks. In *SRLD-D*, since each location message has a sequence number and a signature, it is hard for attackers to inject stale location messages since they are unable to forge the message signature.

Moreover, malicious nodes may initiate denial-of-service attacks such as constantly retransmitting stale messages or garbage messages, the normal wireless transmission around the malicious nodes will be severely affected due to the heavy radio collisions. Denial-of-service attacks are difficult to prevent due to the sharing nature of wireless medium. One way to resume communication in face of the denial-of-service attacks is to switch channels. Another way is to stop attacking nodes physically. Roadside APs record the electronic license plates of the attacking nodes and report the positions of attacking nodes to law enforcement department to stop the attacking nodes.

6 Evaluation

In this section, we evaluate *SRLD-D* regarding to the following two metrics.

- *location beacon latency:* This metric measures the maximal time it takes for a location beacon message to reach all nodes.

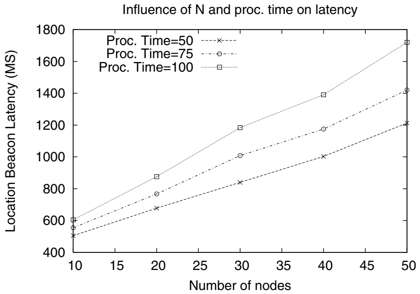


Fig. 7. Latency vs. N and processing time

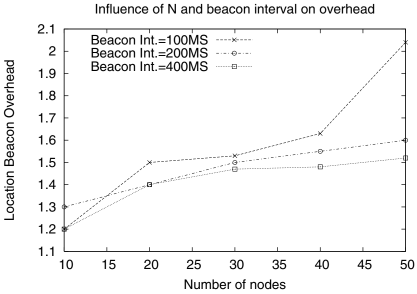


Fig. 8. Overhead vs. N and beacon int

- *location beacon overhead:* This metric measures on average how many messages are sent on each node to propagate a location beacon to all nodes.

We conduct the simulations using Qualnet Network Simulator [19]. The nodes in the network move according to the mobility pattern of the vehicles on the free way. The nodes use IEEE 802.11b radio to communicate. When measuring *location beacon latency*, we vary N and the processing time of a location beacon on each node. Fig. 7 demonstrates that the larger the processing time the larger the location beacon latency and the location beacon latency is linearly proportional to N. Moreover, we measure the influences of N and location beacon interval on *location beacon overhead*. Fig. 8 shows that when N becomes larger or when beacon interval shinks, location beacon overhead increases due to the increase of radio collisions.

7 Conclusions and Future Plan

In this paper, we have presented *SRLD*, a novel scheme for securely determining the relative locations of vehicles in vehicular networks. *SRLD* does not require any GPS or accurate position information but only the relative locations of each vehicle's surrounding vehicles. *SRLD* uses cryptographic keys to authenticate relative location messages and uses a vehicle's public key to identify the vehicle for protecting the driver's privacy. The scheme is designed to defend against Sybil attacks, Wormhole attacks, black hole attacks, and replay attacks. In the future, we will evaluate the scheme under the aforementioned attacks. Moreover, we plan to evaluate *SRLD* in the multi-lane scenario.

References

[1] Fatality analysis reporting system (FARS) web-based encyclopedia. http://www-fars.nhtsa.dot.gov/.

[2] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications Magazine*, 4(5):42–47, October 1997.

[3] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE Infocom 2000*, volume 2, pages 775–784, 2000.

[4] S. Brands and D. Chaum. Distance-bounding protocols. In *EUROCRYPT '93*, pages 344–359, 1994.

[5] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[6] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *IEEE INFOCOM*, 2005.

[7] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. A probabilistic room location service for wireless networked environments. In *UbiComp '01*, pages 18–34, 2001.

[8] D. Singelee, and B. Preneel. Location verification using secure distance bounding protocols. *International workshop on wireless and sensor networks security*, 2005.

[9] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *Network and Distributed System Security Symposium (NDSS)*, February,2004.

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. In *MobiCom '02*, pages 12–23, 2002.

[11] J.-P. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security and Privacy*, 2(3):49–55, 2004.

[12] Kukshya, V.; Krishnan, H.; Kellum, C. Design of a system solution for relative positioning of vehicles using vehicle-to-vehicle radio communications during gps outages. *Vehicular Technology Conference 2005*, 2:1313–1317, October 2005.

[13] L. Lazos and R. Poovendran. SeRLoc: Robust localization for wireless sensor networks. *ACM Trans. Sen. Netw.*, 1(1):73–100, 2005.

[14] D. Liu, P. Ning, and W. Du. Attack-resistant location estimation in wireless sensor networks. In *IPSN '05*, pages 99–106, Los Angeles, California, USA, 2005.

[15] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00*, pages 255–265, 2000.

[16] J. Newsome, R. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: Analysis and defenses. In *IPSN '04)*, Apr. 2004.

[17] P. Persiano and I. Visconti. A secure and private system for subscription-based remote services. *ACM Trans. Inf. Syst. Secur.*, 6(4):472–500, 2003.

[18] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00*, pages 32–43, 2000.

[19] Qualnet Network Simulator. `http://www.qualnet.com/`.

[20] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26(1):96–99, 1983.

[21] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *WiSe '03*, pages 1–10, New York, NY, USA, 2003.

[22] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *MobiHoc '03*, pages 201–212, 2003.

[23] J. S. Warner and R. G. Johnston. Think GPS cargo tracking = high security? Think again. Technical report, Los Alamos National Laboratory,2003.

[24] Y. Hu, A. Perrig, and D. Johnson. A defense against wormhole attacks in wireless ad hoc networks. In *Proc. of INFOCOM 2003*, San Francisco, CA, USA, 2003.

A Local-Control Algorithm to Prolong the Lifetime of Wireless Ad Hoc Networks

Jacques M. Bahi, Ahmed Mostefaoui, and Michel Salomon

Computer Science Laboratory (LIFC), University of Franche-Comte
FRE CNRS 2661, IUT de Belfort-Montbeliard, BP 527
90016 Belfort Cedex, France
{Jacques.Bahi, Ahmed.Mostefaoui, Michel.Salomon}@univ-fcomte.fr

Abstract. Energy efficiency is a major design issue in wireless ad hoc networks since nodes are battery constrained. In such a network each node has to support, in addition to the application workload, the one resulting from the lack of any infrastructure or centralized administration in the network. In particular far-off communications between nodes are done using a multi-hop route via other nodes. Thus they may decide to relay or not packets, but using an inappropriate routing approach can sunk the performances. In this paper we propose a distributed local-control algorithm that guarantees a fair workload distribution across the network. This approach ensures that each node will contribute proportionally to its available energy. Furthermore, the algorithm is able to deal with dynamic networks like mobile ad hoc networks (MANETs). We study the relevance of our approach to prolong the lifetime in dynamic heterogeneous networks through simulations using OMNeT++.

1 Introduction

Wireless ad hoc networks have received a lot of attention over the last years, becoming a mature and viable alternative to infrastructure-based wireless networks. Each node of such a network communicates directly only with nodes in its physical neighborhood and uses multi-hop routes to reach the other nodes. Moreover, nodes mobility and their decentralized operating manner give rise to many issues that have no counterpart in the wired networks. Indeed, since there is no network infrastructure each node has to perform, in addition to its own workload, some fundamental networking tasks (namely relaying and routing). Hence, each node of a wireless ad hoc network has to find the best balance between performing its own application tasks and the networking ones.

Maximizing the nodes lifetime supposes to reduce the number of tasks a node is charged with performing, especially networking tasks such as relaying. Unfortunately this context usually raises a node behavior that is unsuitable: selfish nodes are not willing to relay packets, since it implies energy consumption due to radio communications. Nodes must maximize their lifetime, but also ensure that the connectivity of the network is preserved. This point is highlighted by the motivating example given in figure 1.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 555–566, 2006.

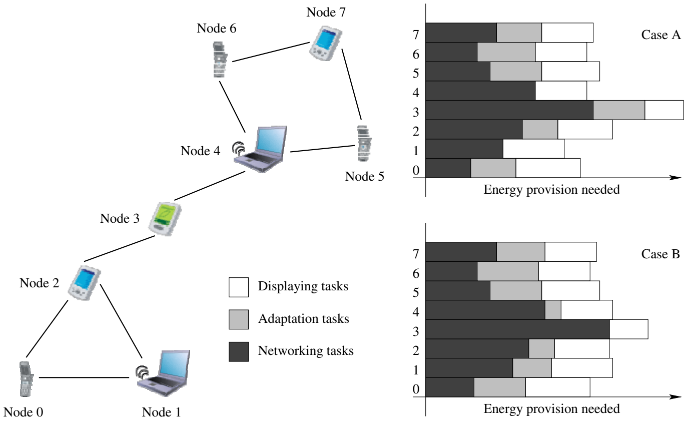


Fig. 1. Motivating example

In this example, each node is supposed to take pictures or videos from a sporting competition and exchange them with neighbors. Clearly the node denoted 3 is in a key position, since it acts as a bridge node. Therefore if this node dies the network will become partitioned. Let us assume now that node 3 has limited capabilities, then, to display the pictures or videos the node has to perform adaptation tasks (resizing pictures, etc.). In addition to that, it must also relay packets to the rest of the nodes. In this context, as shown by case A in figure 1, contrary to the other nodes, node 3 will rapidly consume all its energy, leading to a significant reduction of the network lifetime. To address this problem, we propose that the nodes make a deal between themselves, in order to reduce node 3 workload. More precisely, as shown in case B, the idea is to share its adaptation tasks among nodes that have a higher battery energy level.

In this paper, we design a lifetime-aware task assignment algorithm that can be applied to a static or a dynamic ad hoc network (like MANETs) where nodes may be heterogeneous. The algorithm, of iterative nature, induces a local-controlled migration of tasks between nodes, based on the residual energy of each node. The relevance to prolong lifetime via task migration is assessed through OMNeT++ simulations considering a real multimedia scenario.

2 Related Work

Minimizing energy consumption in wireless ad hoc networks has been investigated by many researchers. The primary goal is, of course, to prolong network lifetime. A way to improve lifetime is to design a suitable routing protocol, therefore several power-aware routing protocols and topology control algorithms have been developed [1,2]. Most of those focus on minimizing the energy consumed per packet in order to deliver it to the destination. A centralized algorithm to determine the maximum lifetime, based on the Garg-Koenemann algorithm for multicommodity flow, has been proposed in [3]. A flow-based approach is also used in [4] (maximum concurrent flow). Several other papers have considered the topic of energy-aware broadcasting in wireless networks [5,6]. In that case the problem can be stated as follows: maintain a transmission graph connecting a given source node to all the other nodes. The aim of many of those works is to minimize the global power consumption of the whole network.

This paper is an extension of a previous work [7] in which we have proved theoretically the validity of our method. Several other differences can be noticed between the papers. In particular, the convergence detection has been explicitly integrated in the proposed algorithm, and the simulation results reported in this paper have been done in the context of dynamic networks.

3 Problem Formulation

The system is supposed to be time-slotted, s denoting the time-slot length. Hence, each time-step t satisfies $t = k \times s$ where $k \in \mathbb{N}$, but for the sake of simplicity we use k instead of t. The network is assumed to be a set of N nodes $i \in \{1, \ldots, N\}$. We consider M subsets of tasks and the energy needed for task execution differs from a set to another. Moreover, as the nodes can be heterogeneous the energy cost of tasks belonging to the same subset (or class) also depends on the node characteristics. Further notations are given in table 1.

The energy spent by node i during time slot k is given by $\xi_i^{(k)} = \sum_{j=1}^{M} N_{i,j}^{(k)} e_{i,j}$, hence the energy provision satisfies

$$E_i^{(k)} = E_i^{(0)} - \sum_{l=1}^{k} \xi_i^{(l)} = E_i^{(0)} - \sum_{l=1}^{k} \sum_{j=1}^{M} N_{i,j}^{(l)} e_{i,j} \quad . \tag{1}$$

As we are trying to optimize the energy, the only way to fulfill this objective is, according to equation (2), to adapt the number of tasks that node i executes $\left(\sum_{l=1}^{k} \sum_{j=1}^{M} N_{i,j}^{(l)}\right)$. Obviously, decreasing this number for each node is not suitable, since this scenario would lead to a Quality of Service drop. Thus, our objective is to balance tasks among nodes considering their available energy. More precisely, we search for a time slot S such that $\forall k > S$ we have

$$\frac{\sum_{j=1}^{M} T_{1,j}^{(k)} e_{1,j}}{E_1^{(k)}} = \ldots = \frac{\sum_{j=1}^{M} T_{N,j}^{(k)} e_{N,j}}{E_N^{(k)}} \tag{2}$$

Table 1. Notations used

Symbol	Description
$T_{i,j}^{(k)}$	: number of class j tasks held by node i at time step k.
$e_{i,j}$	: energy spent by node i to execute one task from class j.
$N_{i,j}^{(k)}$	: number of class j tasks executed by node i during time slot k.

where $T_{i,j}^{(k)}$ stands for the remaining workload $T_{i,j}^{(k)} = T_{i,j}^{(0)} - \sum_{l=1}^{k} N_{i,j}^{(l)}$.

Equation (2) expresses that the ratio between the energy needed to execute the remaining workload and the energy available is the same for all nodes. Thus, we guarantee that each node participates based on its energy resource, and consequently the fairness among all nodes of our approach. In the following the ratio of node i will be denoted $x_i^{(k)} = \frac{\sum_{j=1}^{M} T_{i,j}^{(k)} e_{i,j}}{E_i^{(k)}}$.

Clearly our approach defines a fixed-point system. Indeed, let us denote by $X^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, \ldots, x_N^{(k)}\right)$ the ratio vector then we have $\forall k > S$

$$X^{(k+1)} = X^{(k)} = X^{(S)} = \left(x_1^{(S)}, x_2^{(S)}, \ldots, x_N^{(S)}\right) = (x^*, \ldots, x^*) = X^* \ . \qquad (3)$$

4 Lifetime-Aware Task Migration

Each node executes an algorithm that is driven by its remaining energy provision and workload. These informations together with the ratio of the node's one-hop neighbors are used to control for each of them whether the node has to send tasks or not to it. The algorithm runs iteratively until it detects that no task exchange has to take place: ratio difference between the node and its single-hop neighbors is below a fixed threshold. Let us also emphasize on the robustness and insensitivity of our approach to broken links between nodes or changing network topologies.

Let $V_i^{(k)}$ denote the set of one-hop neighboring nodes of node i at time step k. Each node performs algorithm 4.1 where the symbols $ea_i^{(l-1)}$ and $D_{i,j}^{(l-1)}$ denote respectively:

- the algorithm's energy consumption. It means the energy expenditure due to computations and communications.
- the number of class j tasks generated during time slot $l-1$. This task generation process is assumed to follow a Poisson process of parameter λ_j.

The principle of our algorithm is to exchange the ratio value of the node with its neighbors, and the algorithm uses these values to deduce to which one tasks must be send and how many. The aim is for each node to send tasks to neighboring nodes having a larger ratio value. Task migration may be done until ratio convergence is detected: the ratio difference (line 6) is below a threshold ϵ. The $A_{iv}^{(k)}$ weight factor is a component of the diffusion matrix $A_{iv}^{(k)}$ which is constructed at each time step, in order to tackle changing topologies. The energy provision decreases due to task and algorithm execution (line 15), whereas the number of tasks may increase thanks to the generation process (line 16).

4.1 Task Migration

Due to its decentralized nature, algorithm 4.1 requires that each node communicates with its one-hop neighbors. Now we will provide the amount of tasks that must be sent during an iteration to a neighbor node having a larger ratio value.

Algorithm 4.1. Lifetime-aware task migration algorithm

```
1: $l = k$
2: Convergence = false
3: repeat
4:   if $V_i^{(l)} \neq \emptyset$ then
5:     Exchange $x_i^{(l)}$ with each $v \in V_i^{(l)}$ (i.e. send $x_i^{(l)}$ and receive its $x_v^{(l)}$)
6:     $x_i^{(l+1)} = x_i^{(l)} + \sum_{v \in V_i^{(l)}} A_{iv}^{(l)} \cdot \left(x_v^{(l)} - x_i^{(l)}\right)$
7:     if $x_i^{(l+1)} - x_i^{(l)} > \epsilon$ then
8:       Send tasks to neighboring nodes $\left(v \in V_i^{(l)}\right)$ according to algorithm 4.2
9:     else
10:      Convergence = true
11:    end if
12:  else
13:    Convergence = true
14:  end if
15:  $l = l + 1$
16:  $E_i^{(l)} = E_i^{(l-1)} - \sum_{j=1}^{M} N_{i,j}^{(l-1)} e_{i,j} - ea_i^{(l-1)}$
17:  $T_{i,j}^{(l)} = T_{i,j}^{l-1} - N_{i,j}^{(l-1)} + D_{i,j}^{(l-1)}$
18:  $x_i^{(l)} = \frac{\sum_{j=1}^{M} T_{i,j}^{(l)} e_{i,j}}{E_i^{(l)}}$
19: until Convergence is true
```

From algorithm 4.1, line 6, we have for node i

$$x_i^{(k+1)} = x_i^{(k)} + \sum_{v \in V_i^{(k)}} A_{iv}^{(k)} \cdot \left(x_v^{(k)} - x_i^{(k)}\right) \quad . \tag{4}$$

Let us consider any neighbor v. Several observations can be made:

1. $x_v^{(k)} < x_i^{(k)}$ leads to a decrease in the ratio value by $A_{iv}^{(k)} \cdot \left(x_v^{(k)} - x_i^{(k)}\right)$. Therefore node i must send tasks to node v so that it results in the previous ratio decrease factor.
2. if $x_v^{(k)} = x_i^{(k)}$ no task migration will happen
3. $x_v^{(k)} > x_i^{(k)}$ is the dual case of case one.

We denote by $\alpha_{i,v,j}^{(k)}$ the number of class j tasks to be sent from node i to node v at instant k. Then equation (4) can be rewritten as

$$x_i^{(k+1)} = x_i^{(k)} + \sum_{v \in V_i^{(k)}} \operatorname{sgn}(x_v - x_i) \cdot \frac{\sum_{j=1}^{M} \alpha_{\phi_1(i,v),\phi_2(i,v),j}^{(k)} e_{i,j}}{E_i^{(k)}} \tag{5}$$

where

$$\phi_1(i,v) = \frac{(1 + \operatorname{sgn}(x_v - x_i)) \cdot i + (1 - \operatorname{sgn}(x_v - x_i)) \cdot v}{2} \quad , \tag{6}$$

$$\phi_2(i,v) = \frac{(1 - \operatorname{sgn}(x_v - x_i)) \cdot i + (1 + \operatorname{sgn}(x_v - x_i)) \cdot v}{2} \quad .$$

Naturally, we focus on case 1. From equations (4) and (5) we can deduce the number of tasks to be sent from node i to node v at instant k

$$\frac{\sum_{j=1}^{M} \alpha_{i,v,j}^{(k)} e_{i,j}}{E_i^{(k)}} = A_{iv}^{(k)} \cdot \left(x_i^{(k)} - x_v^{(k)} \right) \quad . \tag{7}$$

Consequently, we obtain

$$\sum_{j=1}^{M} \alpha_{i,v,j}^{(k)} e_{i,j} = A_{iv}^{(k)} \cdot \left(\sum_{j=1}^{M} T_{i,j}^{(k)} e_{i,j} - E_i^{(k)} x_v^{(k)} \right) = A_{iv}^{(k)} \cdot \left(\sum_{j=1}^{M} \gamma_{i,v,j}^{(k)} e_{i,j} \right) \quad . \tag{8}$$

Let us now discuss the problem of estimating the coefficients $\gamma_{i,v,j}^{(k)}$. We decide to define them according to the node workload, since these coefficients are directly related to it. The value of coefficient with index j is proportionally defined from both class j and global workload. Hence we have

$$\sum_{j=1}^{M} \gamma_{i,v,j}^{(k)} e_{i,j} = \sum_{j=1}^{M} \frac{T_{i,j}^{(k)}}{\sum_{l=1}^{M} T_{i,l}^{(k)}} \cdot \left(\sum_{j=1}^{M} T_{i,j}^{(k)} e_{i,j} - E_i^{(k)} x_v^{(k)} \right) \tag{9}$$

and coefficient $\gamma_{i,v,j}^{(k)}$ is given as:

$$\gamma_{i,v,j}^{(k)} = \frac{T_{i,j}^{(k)}}{e_{i,j} \cdot \sum_{l=1}^{M} T_{i,l}^{(k)}} \cdot \left(\sum_{j=1}^{M} T_{i,j}^{(k)} e_{i,j} - E_i^{(k)} x_v^{(k)} \right) \quad . \tag{10}$$

However, using this expression exhibits a major problem, namely, that in some cases it results in sending more tasks than the node has. To solve this problem we add constraints to ensure that the number of tasks a node has to execute, whatever the class, is always a positive number or zero. The constraints we add can be stated as follows:

$$0 \leq \left(\alpha_{i,v,j}^{(k)} = A_{iv}^{(k)} \cdot \gamma_{i,v,j}^{(k)} \right) \leq \frac{T_{i,j}^{(k)}}{|V_i^{(k)}|} \quad \Leftrightarrow \quad 0 \leq \gamma_{i,v,j}^{(k)} \leq \frac{T_{i,j}^{(k)}}{A_{iv}^{(k)} \cdot |V_i^{(k)}|} \quad . \tag{11}$$

Finally, considering equation (10) and inequation (11), each coefficient $\gamma_{i,v,\sigma}$, $\sigma \in \{1, \ldots, M\}$ is given by:

$$\gamma_{i,v,\sigma}^{(k)} = \min \left(\begin{array}{l} \max \left(0, \frac{T_{i,j}^{(k)}}{e_{i,j} \cdot \sum_{l=1}^{M} T_{i,l}^{(k)}} \cdot \left(\sum_{j=1}^{M} T_{i,j}^{(k)} e_{i,j} - E_i^{(k)} x_v^{(k)} \right) \right), \\ \min \left(\frac{\sum_{j=1}^{M} \gamma_{i,v,j}^{(k)} e_{i,j} - \sum_{m=1}^{\sigma-1} \gamma_{i,v,m}^{(k)} e_{i,m}}{e_{i,\sigma}}, \frac{T_{i,\sigma}^{(k)}}{A_{iv}^{(k)} \cdot |V_i^{(k)}|} \right) \end{array} \right) \quad . \tag{12}$$

There is just a little drawback: the coefficients $\alpha_{i,v,j}^{(k)}$ are real-valued. Since all task number must be a discrete value, we have to introduce a suitable discretization step. We will discuss this point in the section devoted to implementation issues.

Taking into account all the previous remarks, the algorithm used to send tasks between neighboring nodes is given thereafter.

Algorithm 4.2. Sending tasks

```
for all node $v \in V_i^{(k)}$ do
    if $\left(x_v^{(k)} - x_i^{(k)}\right) < 0$ then
        Compute $\gamma_{i,v,j}^{(k)}, j \in \{1, \ldots, M\}$
        Discretize all coefficients $\alpha_{i,v,j}^{(k)} = A_{iv}^{(k)} \cdot \gamma_{i,v,j}^{(k)}$
        for all $j$ such that $1 \leq j \leq M$ do
            Send $\beta_{i,v,j}^{(k)}$ tasks to node $v$ ($\beta_{i,v,j}^{(k)}$ denote the discrete value of $\alpha_{i,v,j}^{(k)}$)
        end for
    end if
end for
```

4.2 Diffusion Matrix

Diffusion algorithms are well known for load balancing. They were first introduced for networks with fixed topology and more recently for dynamic networks. As can be seen from equation (4), we have used a general ("anisotropic") diffusion law: $x_i^{(k+1)} = \left(1 - \sum_{v \neq i} A_{iv}^{(k)}\right) \cdot x_i^{(k)} + \sum_{v \neq i} A_{iv}^{(k)} x_v^{(k)}$, which in turn may be written in a matrix form as $X^{(k+1)} = A^{(k)} \cdot X^{(k)}$.

The fraction of the ratio difference which is transferred between nodes i and v can vary for different links and in time, in order to be able to respond to the changing nature of the network's topology. We have chosen a weight-conserving diffusion rule: $A_{iv}^{(k)} = A_{vi}^{(k)}$. On the one hand the off-diagonal elements $A_{iv}^{(k)}$ are set to zero when there is no link between nodes i and v, otherwise $A_{iv}^{(k)} \geq 0$. On the other hand the diagonal elements, that represent the fraction of ratio difference that will not induce task migration, satisfy $A_{ii}^{(k)} = 1 - \sum_v A_{iv}^{(k)}$, $A_{ii}^{(k)} \geq 0$. All these constraints make $A^{(k)}$ a symmetric doubly stochastic matrix.

The behavior of our approach is conditioned on the form of the diffusion matrix. The algorithm used to construct this matrix can be found in [7].

5 Implementation Issues

5.1 Discretizing Task Migration

The coefficients $\alpha_{i,v,j}^{(k)}$ are discretized using a two-step method. In the first one, each coefficient is rounded to the nearest integer value, the deviation between the real value and the corresponding integer one is also calculated. In the second step, the conversion accuracy is estimated through the accuracy of the sum. By accuracy of the sum we mean the difference between the rounded exact sum of exact values compared to the sum of the rounded values. If the difference is not equal to zero, one rounded coefficient is modified (increased by 1 if the difference is positive, decreased by 1 otherwise), the coefficient is chosen according to the deviation value. This process is repeated until the difference is zero. We also add constraints in order to ensure that no more tasks than available will be sent and also that it remains positive.

5.2 Energy Consumption Model

To evaluate the ability of our approach to achieve energy-efficiency by migrating tasks between nodes, and assess its effectiveness, an accurate modeling of the energy consumed is essential. Obviously, there are two sources of energy consumption in our algorithm: computation and communication. We decide to set a fixed energy cost for the computation part, based on the energy consumption characterization of the network nodes chosen in our simulation setup.

Several communication energy models have been proposed. A simple model uses the voltage and current characteristics of the radio used to estimate the power dissipated. In [8] the authors measured the energy consumption of a 802.11 wireless NIC operating in ad hoc mode. The obtained measures allow them to model energy consumption (E_S) through linear equations. These equations model data communication activities (reception and transmission) considering different types of communication (broadcast, unicast or packet discarding):

$$E_S = m_S \times \text{PacketSize in bytes} + b_S \tag{13}$$

where m_S and b_S are communication-specific. m_S models the impact of the data packet size on energy consumption, whereas b_S takes into account the energy cost resulting from the channel access and the MAC layer control packets. We used the equations obtained for the Lucent WaveLAN Silver card (11 Mbps).

6 Experimental Results

We ran our algorithm over a set of simulated networks having different topologies and sizes, and compared our results with those obtained without using our method. The simulator OMNeT++ [9] is used to perform the simulations.

6.1 Simulation Setup

To study the sensitivity of our approach to network characteristics, we investigate the algorithm performances under different types of topologies and network sizes. The three topologies we have studied are: fully connected, linear and a mixed topology. The latter, in which subsets of nodes communicate with each other via intermediate nodes, has been randomly generated using a realistic link-level topology generator [10]. We have investigated the first two topologies for networks of 5 and 10 nodes, while the size of the mixed one was set to 20 nodes.

A very important property of the lifetime-aware task migration algorithm is its ability to deal with heterogeneous networks. Therefore, the simulation scenario involves wireless sensing nodes monitoring a perimeter through webcams and mobile computing platforms (laptops) to view the images. The energy consumption characterizations conducted by Margi *et al.* for both wireless DELL Latitude C600 laptop [11] and Crossbow Stargate platform [12] allow us to simulate this real life case network. Thus, we have considered two types of nodes. The first type (termed A) has limited resources (computing and storage), corresponding to a Stargate node. The second one (termed B) denotes nodes having higher battery power and processing facilities, which means DELL laptops. At the beginning of each simulation the type of each node is randomly chosen. In addition to heterogeneous networks our approach can support dynamic changes, whether due to node mobility or to node failures.

Simulations are performed according to the node parameter settings, depending on its type, given in table 2. All energy values, battery capacity and consumption induced by task execution, are based on real specifications and measures. The initial workload per node is defined with respect to its type: an A node is assigned 250 tasks, while a B node receives 500 tasks. These tasks are randomly distributed among three classes, and the energy cost for executing a unitary task is fixed for each class using the energy characterization case studies mentioned above. We can note that we set arbitrarily the fixed energy cost of our algorithm. Initially, a task of each class is executed at every node, when a class has no more task to execute, the free execution slot is allocated to another class. As soon as the remaining workload of a node reaches zero, baseline tasks are executed until the battery is depleted. Data flows between pairs of nodes, generated by task migration, are of the order of 460 kB per task for the first class and 46 kB for the two other ones.

The initial battery charge values are workload-based. In this way, the starting ratio value of each node is randomly drawn from a uniform distribution and then, using the initial workload, we calculate the corresponding energy value. Two uniform distributions are considered. The first one has a range of $[0.25, 1)$, while the range of the second one is $[1, 2)$. The underlying idea is to have some nodes that will die before a complete execution of all their tasks.

6.2 Simulation Results

The first thing we have studied is the fairness of our approach. In other words, can we ensure that all the nodes of a network reach nearly the same ratio value? The different network configurations have been evaluated considering 20 randomly generated initial settings. The same random seeds are used for each network. Fig. 2(a), (b) and (c) show representative ratio evolutions for each type of topology. As one can see, the ratios always converge toward the same value and the convergence speed is clearly related to the network topology and size. Obviously, the fastest convergence is obtained with the fully connected topology

Table 2. Node settings (in particular energy consumption) used in the experiments

Node		Crossbow Stargate		Dell Laptop	
Battery capacity / load		7.4 Wh / 50%		56.25 Wh / 50%	
Task class		$e_{i,j}$	$N_{i,j}$	$e_{i,j}$	$N_{i,j}$
	1	5 W	1	28.65 W	1
j	2	4.75 W	1	24.75 W	1
	3	4.25 W	1	21 W	1
Baseline (system idle)		1.53 W		10.59 W	
Algorithm (fixed cost)		1.74 W		11 W	

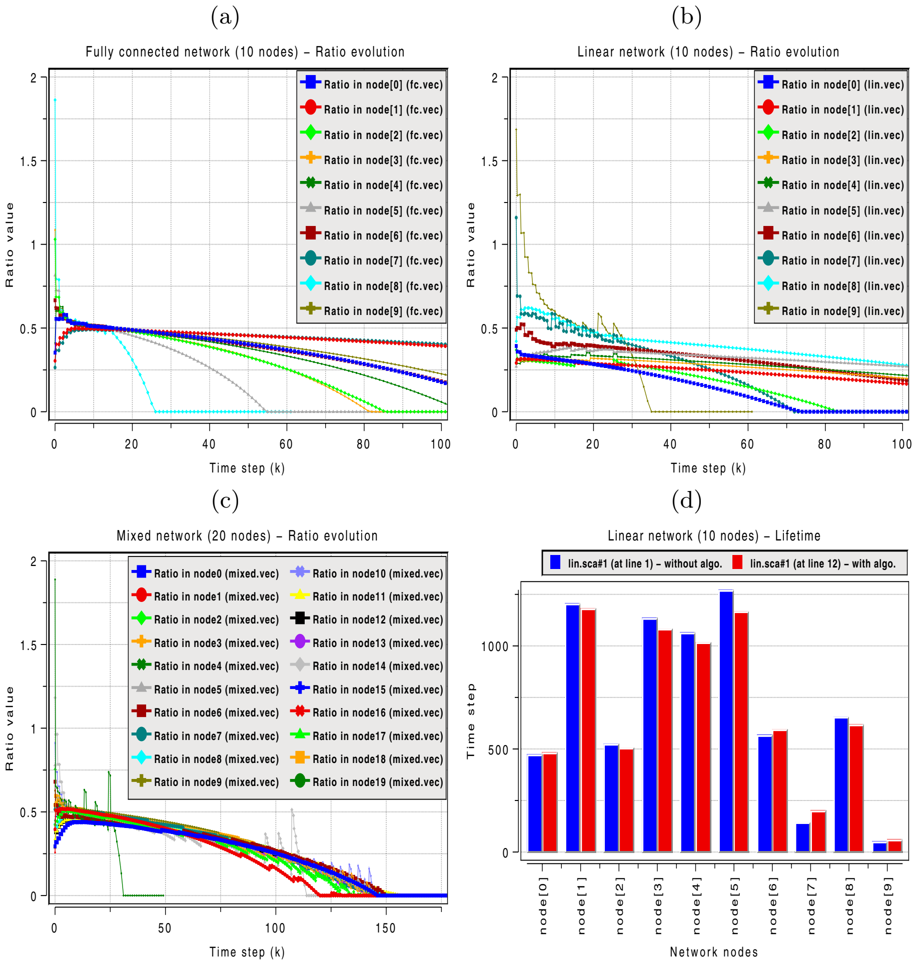


Fig. 2. Ratio evolution for different network configurations and lifetime improvement

which requires in average about 15 time steps to reach convergence. The number of iterations needed by the linear topology is of the order 26 iterations and far more for the mixed one. Furthermore, as shown by Fig. 2(c) our algorithm is robust to node failure, since the first failing node dies before time step 50 (the curve is stopped), while the algorithm is still running.

Another interesting point to observe is that the ratio values get closer and closer to zero, and finally reach this last value for all the nodes. It means that, thanks to the proposed algorithm, no node dies before having executed all its tasks. This is because the generation process has been only taken into account during the first ten iterations. Thus, the proposed algorithm provides a certain level of Quality of Service, ensuring that no task will remain unexecuted. Finally, we can remark the oscillations of some ratio curves, in particular in Fig 2(c).

Table 3. Average lifetime improvement for different topologies

Topology	Fully connected		Linear		Mixed
Size	5	10	5	10	20
$\epsilon = 10^{-2}$	53.50%	57.80%	46.40%	23.50%	12.75%

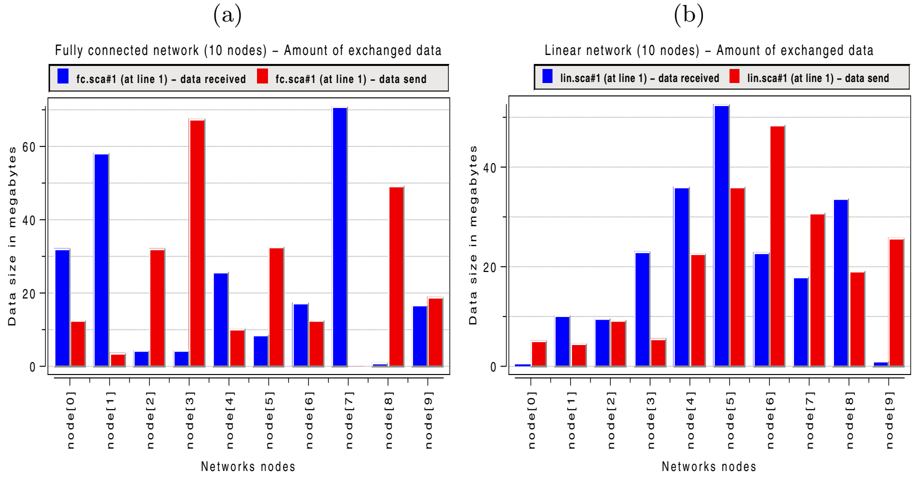


Fig. 3. Amount of exchanged data: (a) fully connected network and (b) linear network

Those oscillations appear when a node is coming closer to its end of life. Let us remember that tasks migrate between nodes based on their ratio. Therefore, when a node has one of the lowest ratio value and a low energy provision, the tasks sent by neighboring nodes to it result in an important increase of the ratio value. This problem is overcome by introducing an energy level under which a node will no more execute the algorithm, except if its ratio is above one.

Table 3 presents the average network lifetime improvement obtained. The best performances are induced by the fully connected topology, whatever the network size, with a lifetime increase above 50%. This fact can be explained by the fastest speed of convergence. Clearly, the linear topology is very sensitive to the network size. The main reason for this observation is the highest average node distance. Although the 20-node-mixed network presents the worst performances, our algorithm still remains interesting, allowing a satisfactoy increase. So, the energy overhead due to our method does not lead to a lifetime decrease. Fig 2(d) shows, for the same linear network as in figure 2(b), the lifetime of each node with and without our algorithm. From this bar chart, we can see that four nodes have a longer lifetime, in particular the first two failing ones (nodes 7 and 9), whereas the lifetime is reduced for some other nodes.

Fig. 3(a) and (b) present the global amount of exchanged data, respectively for a fully connected and a linear network. We think that these values are realistic and representative. In addition to the communications due to our algorithm, we have included a normal traffic in the networks. In the fully connected context we can see that node 8 sends almost only data, since it has the highest starting ratio value (see Fig. 2(a)). On the other hand, node 7 which has the lowest ratio value at the beginning, receives nearly only data. A look at the linear case shows, as expected, that the data communications increase from the ends toward the inner nodes.

7 Conclusions

Energy-efficiency is crucial in power-limited wireless ad hoc networks, since nodes have significant power constraints (battery life). In this paper, we have investigated the problem of prolonging the lifetime of dynamic heterogeneous wireless ad hoc networks. To solve this problem, we have proposed a local-control algorithm that migrates tasks between neighbouring nodes, based on their remaining energy provision. Extensive simulations results of a real-life scenario were provided to evaluate the performance of our algorithm. Those results clearly demonstrate the effectiveness and relevance of our approach.

References

1. Stojmenovic, I., Lin, X.: Power-aware localized routing in wireless networks. In: IPDPS, IEEE Computer Society (2000) 371–376
2. Wattenhofer, R., Li, E.L., Bahl, P., Wang, Y.M.: Distributed topology control for wireless multihop ad-hoc networks. In: INFOCOM. (2001) 1388–1397
3. Chang, J.H., Tassiulas, L.: Fast approximate algorithms for maximum lifetime routing in wireless ad-hoc networks. In Pujolle, G., al., eds.: NETWORKING. Volume 1815 of Lecture Notes in Computer Science., Springer (2000) 702–713
4. Sankar, A., Liu, Z.: Maximum lifetime routing in wireless ad-hoc networks. In: INFOCOM. (2004)
5. Cagalj, M., Hubaux, J.P., Enz, C.: Minimum-energy broadcast in all-wireless networks: : Np-completeness and distribution issues. In Akyildiz, I.F., Lin, J.Y.B., Jain, R., Bharghavan, V., Campbell, A.T., eds.: MOBICOM, ACM (2002) 172–182
6. Papadimitriou, I., Georgiadis, L.: Energy aware broadcasting in wireless networks. Mobile Networks and Applications **9** (2004) 567–583
7. Bahi, J., Mostefaoui, A., Salomon, M.: Increasing lifetime of wireless ad hoc networks using a decentralized algorithmic approach. In: ICON 2006, 14th IEEE Int. Conf. on Networks, Singapore (2006) 427–432
8. Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: INFOCOM. (2001) 1548–1557
9. Varga, A.: The omnet++ discrete event simulation system. In: ESM. (2001)
10. Medina, A., Lakhina, A., Matta, I., Byers, J.W.: Brite: An approach to universal topology generation. In: MASCOTS, IEEE Computer Society (2001)
11. Margi, C.B., Obraczka, K., Manduchi, R.: Characterizing system level energy consumption in mobile computing platforms. In: WirelessCom, IEEE (2005)
12. Margi, C.B., Obraczka, K., Manduchi, R.: Characterizing energy consumption in a visual sensor network testbed. In: TridentCom, IEEE (2006)

Exploiting Local Knowledge to Enhance Energy-Efficient Geographic Routing[⋆]

Juan A. Sanchez and Pedro M. Ruiz

Dept. Information and Communications Eng.
University of Murcia
{jlaguna, pedrom}@dif.um.es

Abstract. Geographic routing is one of the most widely-accepted techniques to route information in large-scale wireless sensor networks. It is based on a greedy forwarding strategy by which a sensor node selects as next hop relay the most promising neighbor (according to some metric) among those being closer to the destination than itself. This decision is based solely on the position of its neighbors and the destination. Given that sensor nodes are usually operated by batteries, energy-efficiency is a very important metric to be considered by the routing protocol. In this paper we present Locally-Optimal Source Routing (LOSR), a new localized and energy-efficient geographic routing algorithm for wireless sensor networks. Unlike existing energy-efficient geographic routing algorithms, in which current node routing the packet only considers nodes closer to destination than itself, LOSR uses all nodes in the neighborhood to compute a local energy-optimal path formed only by neighbors of the current node towards the selected next hop. Then, *source routing* is used to force data packets to follow that locally optimal path until next hop is reached. Our simulation results show that the proposed algorithm outperforms the best existing solution, over a variety of network densities and scenarios.

Keywords: Unicast Geographic Routing; Energy efficient; WSN.

1 Introduction and Related Work

Wireless Sensor Networks (WSN) consist of a set of tiny components called sensors which are able to acquire data from its environment, process it and communicate with other sensors using low-range and low-power radio interfaces. WSNs are specially useful in scenarios in which data needs to be gathered and processed in a distributed way. In addition, their battery-operated nature and tiny size, allows them to be a good solution for those cases in which technologies need to be non-intrusive. Some of those scenarios for which sensors are considered include among others: disaster relief, habitat monitoring, wildfire detection, etc. Their wide applicability is one of the reasons for their increased popularity both in industry and academia.

[⋆] Partially supported by the Spanish MEC by means of the “Ramon y Cajal” program, SAVIA project (CIT-410000-2005-1) and SMART project (TIN2005-07705-C02-02).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 567–578, 2006.

Devices participating in a WSN sense their environment using special hardware called sensors. We will call those devices sensor nodes along the paper. In addition, they are also equipped with radio interfaces which allow them to form a multihop network. That is, they are able to transmit messages using as relays their neighbors. Sensor nodes have very limited resources, in terms of computing power, memory and battery life. In particular, the limited energy can be really problematic. Radio communications are the main cause of energy consumption of a sensor node [3]. Thus, it is of vital importance to reduce the energy consumption due to communications in order to extend sensors lifetime as long as possible. For this reason designing energy efficient routing algorithms for WSN is of paramount importance.

Geographic routing algorithms are well-suited to the special characteristics of WSNs. In these algorithms, routing decisions are solely based on the position of the destination and the current node. They are usually referred as localized algorithms because they only rely on local information directly available from neighbors. This makes them almost state-less and they only require a minimum memory space to maintain data structures. Their computational cost is normally low and they are very scalable which is important due to the large number of sensors a WSN could have. That is the reason why several routing protocols based on this technique were proposed in the late 80's such as MFR [11], Compass Routing [6] and GEDIR [9]. Those three algorithms use the notion of progress, first introduced by Finn in [2], to determine which neighbor to select in order to achieve the maximum advance toward the destination. However, packets may eventually reach a node with no neighbor providing advance toward the destination, making the algorithm fail. A variant of MFR described in [4] that proposes to adjust transmission power to reach only the selected neighbor has the same problem. To avoid that problem face routing was proposed in Greedy-Face-Greedy(GFG [1]). A similar scheme is proposed in the GPSR [5] protocol.

Based on GFG and GPSR and using different energy metrics, authors have proposed different energy efficient geographic routing protocols. The most common energy metric can be found in the work done by Rodoplu and Meng in [7]. They assume that the energy needed to send a message from a node u to one of its neighbors v located at distance d is proportional to d^α being α $(2 \leq \alpha \leq 6)$ the power attenuation factor.

GFG and GPSR usually select neighbors providing more advance towards destinations but this might not be an energy optimal decision. Stojmenovic defined a general framework called cost over progress [8] in where different approaches can be taken into account for selecting the best next forwarder. The same concept is used by the same author to design in [10] a location-based energy efficient algorithm for WSN called Iterative Power Progress (IPOW). This algorithm selects as best next hop neighbor the one minimizing cost over progress ratio, being the cost the energy needed to reach such neighbor. After the selection is done, an iterative process tries to optimize the decision. The optimization can be achieved if another neighbor can be used as relay to reach the previously selected and doing it in two hops needs less energy than a direct transmission. In this algorithm, when there is no neighbor providing advance towards the destination, standard face routing [5] is applied to get over the local minima. To the best of our knowledge, this algorithm is the best in this field.

In this paper we present LOSR which is a a new localized and energy-efficient geographic routing algorithm for wireless sensor networks. In LOSR the current node processing a message computes the energy shortest path to the neighbor closest to the destination and follows it until a node closer to the destination than itself is found. After the selection of the next forwarder is made, we use the Source Routing (SR) to force the message to follow that energy efficient path to reach it. That is, the source routing header includes the usually short list of nodes that need to be used hop by hop. One of the key aspects of our proposed scheme is the use of source routing to exploit the local knowledge in a node's neighborhood to save energy. LOSR uses Dijkstra Shortest Path algorithm considering as link weights the energy required to send a message from one of the endpoints to the other one. Thus, Dijkstra's algorithm computes the local energy shortest path using only local information (i.e. the position of the node's neighbors). However, as in any greedy algorithm, it may happen that the locally optimal decision might not be globally optimal. This is the reason why we take the decision of not reaching the node providing the greatest advance but the first one providing advance in the locally optimal path to reach it. Our scheme guarantees that it is possible to correct the routing direction in case the first decision was not good enough (normally due to the lack of global knowledge about nodes ahead of the neighborhood).

As we mentioned before, routing in greedy mode means that packets may reach a node which has not got any neighbors providing advance. In our case, we also use face routing in those situations, but we also use the minimum energy path to reach the next hop in face mode and use SR to reach it. We show by simulation that our algorithm outperforms IPOW algorithm, which was the best energy-efficient localized routing protocol to date.

The rest of the paper is organized as follows: section 2 defines the physical model used. Our proposed algorithm is described in section 3. In section 4 we show an analysis of the performance of our solution. Finally, section 5 provides some conclusions and discusses open issues.

2 Physical Model

2.1 Network Model

Following the generally accepted unit disk graph (UDG) model, we represent a WSN as an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. We assume that every node, represented by a vertex $v \in V$, is embedded in the plane, i.e. there are no great differences in height between nodes. Each node $v \in V$ has a maximum transmission range r that can be considered, without losing generality, the same for all nodes. Let $dist(v_1, v_2)$ be the Euclidean distance between two vertices $v_1, v_2 \in V$. An edge between two nodes $v_1, v_2 \in V$ exists $\iff dist(v_1, v_2) \leq r$ (i.e. v_1 and v_2 are able to communicate directly).

2.2 Energy Model

There are different energy models that can be used to estimate the energy required by a node n to send a message far enough to reach a specific neighbor placed at distance d. In this work we follow the model proposed by Rodoplu and Meng in [7]. In this model, the energy consumption for transmitting a fixed size message at distance d is:

$$E(n, d) = d^{\alpha} + C$$

Being α the media attenuation factor satisfying $2 \leq \alpha \leq 6$ and C a constant representing the power used to process the radio signal.

3 Routing with Locally Optimal Paths

The basic idea of this algorithm is to progress as much as possible in each step but using as low energy as possible. To do it, the algorithm is as conservative as possible about the goodness of the direction locally selected in each decision. Our intuition is that progressing as much as possible in each step might not be globally optimal. The routing algorithm works as follows. A node a currently holding the message uses Dijkstra's shortest path, to compute the minimum energy path towards b the neighbor placed closest to the destination among those which are closer than a. Then, the next hop selected is the first in that path being closer to the destination than a. Finally, Source Routing (SR) is used to force the packets to follow the computed path between a and c.

When the node a has no neighbor providing advance towards the destination, it switches to face routing. In face routing, nodes locally derive a planar subgraph of their neighbors and select the next hop applying the right-hand rule over it. Once that next hop is selected, the local energy shortest path between current node and the next hop is computed in order to reach it using as low energy as possible. If a shortest energy path is found, the SR header is inserted in the message. By doing that, we can also save energy in face mode. In the next subsections, we explain in detail the whole process.

3.1 Greedy Routing

The well-known Dijkstra algorithm, to find the shortest path between two nodes in a graph, can be used to find energy shortest path as long as edge weights reflect the energy needed to send messages between nodes connected by those edges. Instead of using the complete topology (which would not satisfy the locality requirements of the protocol) we use this algorithm locally to find out the shortest energy path to a neighbor of a node. The idea is to compute this path to one of the neighbors providing advance towards the destination and then follow the path. This behavior guarantees that each hop is saving as much energy as possible, but the problem is that globally that path may not resemble the energy shortest path between source and destination. The reason is that decisions are taken trying to get advance without global knowledge. In fact, in geographic routing providing advance is a must to avoid routing loops. However, trying to advance too much on each step can lead to a bad overall decision. In our solution, we choose the first node in the path being closer to the destination than current one. This guarantees the avoidance of loops and at the same time we are following the best possible path with the known (local) information. Once the selected relay is reached, a new decision can be taken with new information about its neighbors which was not available to previous nodes. Hence, a better decision can now be taken based on the new information known by the new current node. The message is sent including in its header the list of nodes in the shortest path that the message must traverse as in IP Source Routing (SR). Nodes receiving a message with a SR header on it do not compute the next hop. Instead, they remove themselves from the SR header and forward the message to the next hop in the SR header.

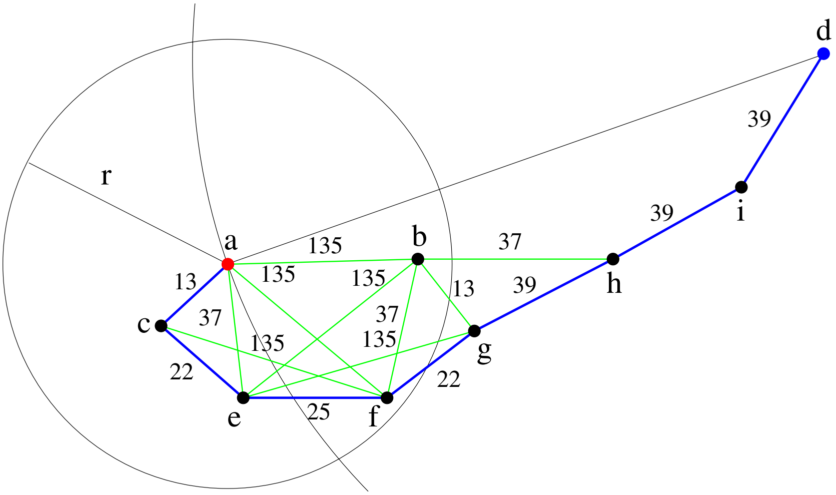


Fig. 1. Node a routing to d selects node c as next forwarder

Fig. 1 shows an example in which node a currently holding the packet has to select the next forwarding node to route the message to node d. Each link is labeled with the cost in energy of sending a message through it. From the point of view of a, b is the neighbor that provides most advance towards the destination d whereas nodes c and e do not provide any advance. But if node a computes the energy shortest path to reach b, the resulting path is a, c, e, f, b. Notice that, globally, the energy shortest path includes also node g but, g is not a neighbor of a, therefore, it does not know about its existence and can not include it in the computation of the path. Following this path completely is the local optimal decision to reach node b. That is, the best local decision with the knowledge that node a has. However, going to b using that path may not be the best global decision. Our algorithm selects f as next hop node because it is the first node in the local path whose distance to d is lower than the distance from a. The reason for doing that is that the deviation from the globally best path can be reduced. Therefore a creates a message including in the header the path that the message should follow to reach f using the previously computed energy shortest path. The list included in the message is c, e because it is not necessary to include the origin nor the destination of the SR as they are already included in the header of the message. When node c receives the message, it checks if it has a SR header or not. In our example, the header exists so, c removes himself from the list and forwards the message to the next node in the list (e) that repeats the process. At node f source routing ends and then, it can recompute the next best hop. Node f has more information than node a because it knows of the existence of node g. Here we can see how the conservative election of a allows us to take a better decision. In the next step, node f selects g and finally g passes the message to h which forwards it to i and finally to d.

Not being conservative would have lead us to reach b in a locally optimal way but after that, the shortest path to d from b would have gone through g which is not globally optimal. Our approach takes advantage of the progressive increment in knowledge that nodes located closer to the destination have. In fact, the further away the next hop selected, the greater the error that may be incurred. On the other hand, selecting nodes providing the lowest advance towards the destination does not represent much error but it is not useful because it only delays the moment of the real decision. In our case, we reach a trade-off. We only follow the path to the first point in which a new evaluation might be better than the current.There is also another reason for not following the complete energy shortest path towards most promising neighbor. Including large SR headers in the message implies augmenting the bandwidth usage and at the same time the energy consumption.

3.2 Face Routing

We now explain how to deal with the situation commonly-known as local minima. As mentioned above, routing in greedy mode can end up reaching a node where no neighbor provides advance towards the destination. As previous geographic routing algorithms, we use face routing to get over this situation. However, to further reduce energy consumption we compute the shortest energy path towards the node selected by face routing, and again we use source routing to reach it. In order to maintain face routing behavior as it is defined, it is necessary to extend the face routing specific header. This header includes the node where face routing started, the edge used to enter the current face being used, etc. We add to this header the node where SR started. Notice that in an standard face routing node a, selecting node b as its next face hop, would be the sender of the message arriving at b. In our case, when there exist an energy shortest path between a and b, the sender of the message eventually received by b might not be

a. As face routing needs to know the previous node in the face path to compute the next one, we need this information to be carried in the message.

4 Experimental Results

We have performed a simulation-based analysis to assess the performance of the proposed scheme. Simulations have been performed with a custom-made java simulator for geographic routing, which is able to simulate very large topologies consisting of thousands of nodes. Each simulation has been performed for different network densities, that is, mean number of neighbors per node. Nine different densities have been simulated between 9 and 45. Thus, to fix the simulation area, the total number of nodes is changed accordingly having as a result a value between 320 nodes for a density of 9 and 1600 nodes for a density of 45. In all the scenarios tested the radio range is equal for all the nodes and fixed to 50m and the attenuation factor and C constant considered to calculate the energy of radio transmissions are α=4 and C = 100000 as in [7]. The size of the scenarios is fixed at $500m^2$ and, in each scenario, the source is placed at top left corner and the destination at the bottom right corner. For each density 50, different scenarios have been evaluated giving a total of 450 simulation runs for each of the tested protocols. We have simulated our new algorithm LOSR and its variant without Source Routing as well as Iterative Power Progress (IPOW) and its variant with SR. We have also simulated a centralized version of the Dijkstra algorithm to find out the global energy shortest path called ESP.

The main performance metric has been the mean energy consumed but also the percentage of messages sent in perimeter mode. The results presented below have a confidence interval of a 95% not sohwing in the curves to improve readability.

4.1 Energy Saving Due to Source Routing

The use of Source Routing reduces the energy hop by hop. To measure the amount of energy saved by the use of this technique we have simulated our algorithm without using SR, i.e., the algorithm forwards the messages to the neighbor that provide the most advance towards the destination, using SR from the node to the selected neighbor and using SR only up to the first node in the path providing advance towards the destination. The first variant of the algorithm selects the next hop in the same way (the neighbor closest to the destination) but it does not compute the energy shortest path to reach the relay in greedy nor in face mode.

Fig. 2 shows that the higher the density the higher the improvement achieved using SR. The energy reduction achieved is between a 50% and a 86%. Moreover, stopping the SR in the one-hop neighbor that would be the first node, that is closer to the destination than the current node, on the minimum energy path to the one-hop neighbor selected as forwarder, reduces the energy consumption between a 1% and a7% more. As we can see, at lower densities the energy

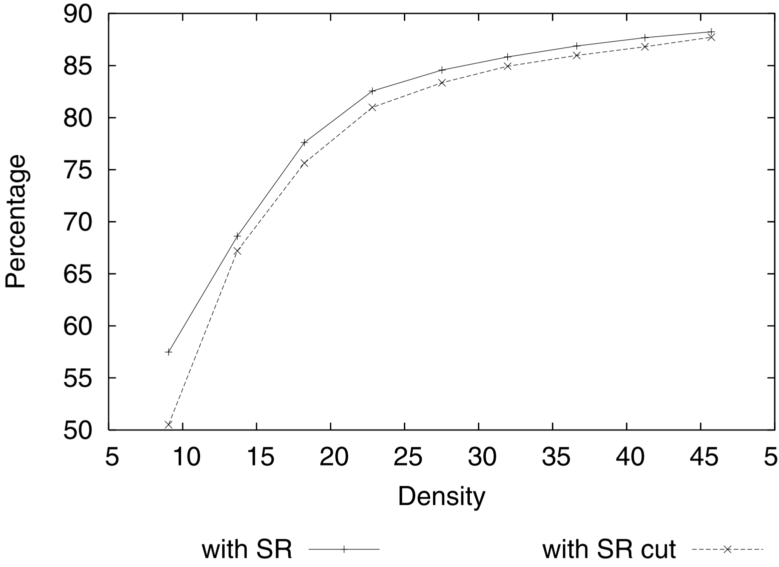


Fig. 2. Improvement by using Source Routing

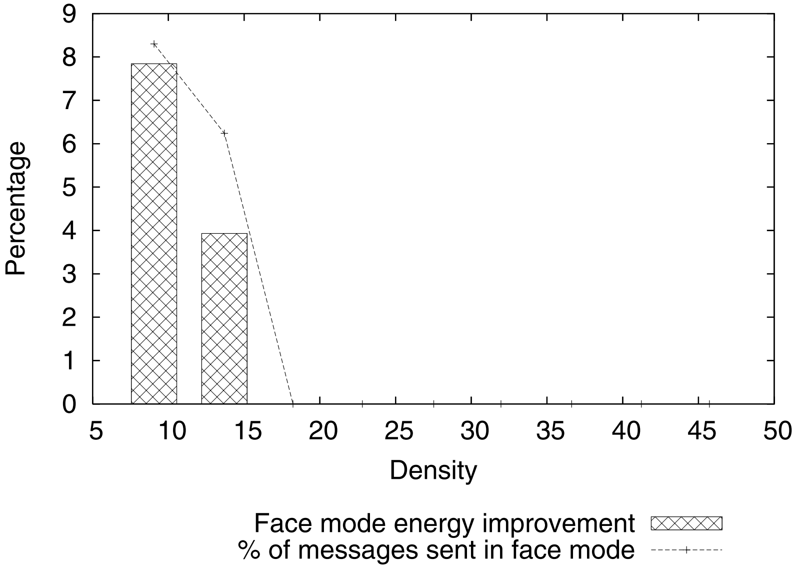


Fig. 3. Energy saving in face mode due to Source Routing

reduction is low. For lower densities it is difficult to find alternative energy efficient paths as the number of nodes is low whereas for higher densities the probability of finding nodes to build the Dijkstra path is higher.

In Fig. 3 we can see the reduction of energy consumption achieved applying SR to Face Routing against the ratio of messages sent in this mode. As it is shown, the percentage of messages sent in face mode decreases to 0 when the mean density goes over 14. Thus, only up to that density, the improvement can be of any significance. The reduction of energy achieved taking into account only the one used in face routing mode is up to a 8%. Taken into account the percentage of messages sent in face routing, the overall energy reduction due to the application of SR in face routing is under 1%. Obviously, the scenarios in which face routing is applied are the ones with lower densities. Building energy-efficient paths with only a few neighbors is difficult. That is the case in those scenarios, therefore, the improvement achieved is low but also important.

4.2 Performance Against Iterative Power Progress

Simulations of Iterative Power progress have been made using the same sets of graphs in order to compare the energy needed by each protocol. Fig. 4 shows the total amount of energy needed by each protocol at increasing densities. It also shows with column bars the percentage of energy that each protocol uses in perimeter mode with the exception of ESP that does not uses that mode. As it can be seen, the density has the same effect on the three protocols. The higher the density the lower the energy used. The reason is that at higher densities it is possible to find better paths through multiple nodes. These paths usually have more nodes than the ones built at lower densities whereas the total distance between source and destination remains very similar. Including more hops in a path means reducing the inter-hop distance and thus reducing the energy consumption up to a limit. That limit is given by the constant C of the formula for the energy in 2.2. Having a $C \neq 0$ guarantees that the best path is not the one made with infinite hops.

As expected, the higher the density the closer all protocols are to the best one (ESP). However, LOSR is better than IPOW when the mean density is higher than 14, and the difference increases with the density as expected. Having more neighbors per node allows us to locally compute a better energy shortest path saving much more energy than IPOW that only takes advantage of energy reductions achieved by adding a single node to the path. At lower densities, IPOW is better than LOSR because as figure 5 shows, the percentage of messages sent in perimeter mode by IPOW is lower than LOSR. Even though LOSR is reducing the energy of the messages sent in perimeter mode, the selection of the next forwarder based only in distance to the destination make the protocol enter more frequently in perimeter mode than the complex selection function of IPOW does.

With these results we decided to apply our energy reduction method to IPOW in order to test how much can SR reduce the energy if IPOW. In each step of IPOW algorithm an iterative process looks for a node whose cost of being reached directly could not be reduced adding another one in the middle and making two hops instead of only one. Obviously, it might exist a longer path with less energy consumption.

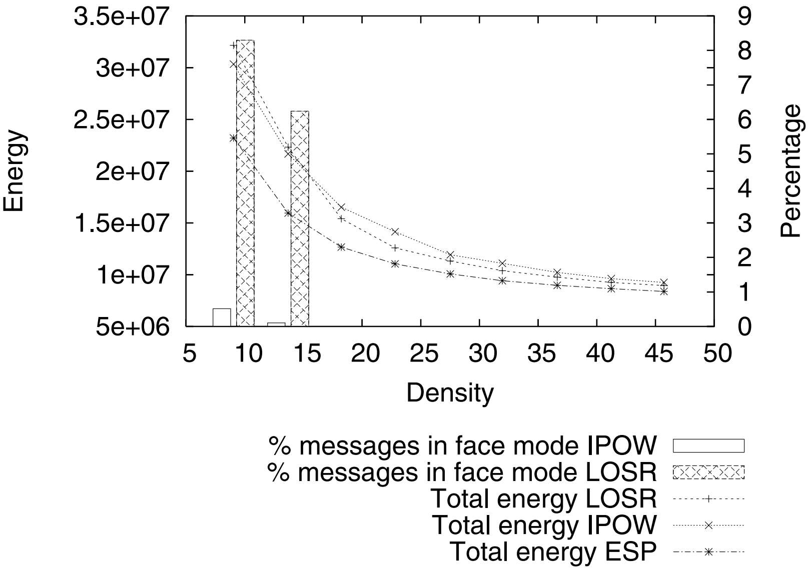


Fig. 4. Total Energy consumption

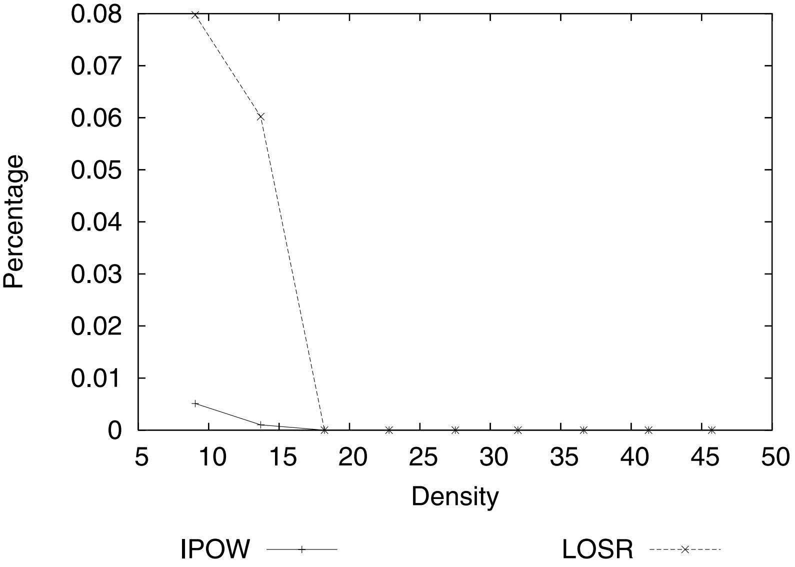


Fig. 5. Percentage of messages sent in perimeter mode

Fig. 6 shows the total energy consumed by IPOW and the SR variant of IPOW and the percentage of improvement that the SR variant achieves over the original one. The improvement is greater at lower densities because the original IPOW

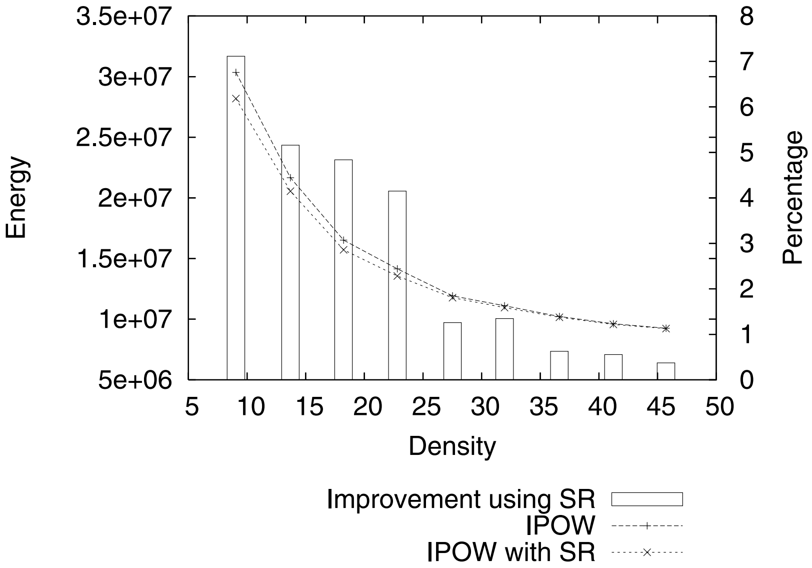


Fig. 6. Energy saving improvement over Iterative Power Progress

algorithm only selects as forwarders neighbors closer to the destination than the node taken the decision. Adding SR allows the protocol to use also the neighbors that does not provide advance towards the destination. Therefore, less energy-consuming paths are found to reach the original forwarder neighbor selected. At higher densities, SR has almost no effect because having enough neighbors allows IPOW to chose almost every time the one that provides advance and at the same time, the cost of sending a message directly to it is lower than through any other path.

5 Conclusions and Future Work

We have introduced a new localized and energy-efficient geographic routing algorithm for wireless sensor networks which as density increases outperforms Iterative Power Progress. We have also shown that for scenarios with lower densities the best approach is to apply our SR technique to Iterative Power Progress. Thus, we have shown that the use of the well-known Source Routing technique can save energy when it is applied in conjunction with a locally computed Dijkstra's energy shortest path regardles of the original next forwarder selection function used. The main conclusion is that locally made decisions have to be carefully taken as they can be wrong due to the lack of global knowledge. Thus, it is better not to follow initial selections till the end. Rather, new information obtained after a shorter advance might correct the initial decision. Our approach of taking the minimum part of the initial decision has really good results improving IPOW as density increases. For future works we are studying how to deal with more realistic scenarios in which errors play an important role as well as the possibility of adapting this algorithm to multicast.

References

1. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *Wireless Networks*, 7(6):609–616, 2001.
2. G. G. Finn. Routing and Addressing Problems in Large Metropolitan-scale Internetworks. Tech. Rep. ISI/RR-87-180, University of Southern California, Information Sciences Institute, March 1987.
3. G. Halkes, T. van Dam, and K. Langendoen. Comparing Energy-Saving MAC Protocols for Wireless Sensor Networks. *Mobile Networks and Applications*, 10(5):783–791, 2005.
4. T.-C. Hou and V. Li. Transmission Range Control in Multihop Packet Radio Networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.
5. B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proc. 6th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pages 243–254, New York, NY, USA, 2000. ACM Press.
6. E. Kranakis, H. Singh, and J. Urrutia. Compass Routing on Geometric Networks. In *11th Canadian Conference on Computational Geometry (CCCG '99)*, pages 51–54, Vancouver, August 1999.
7. V. Rodoplu and T. Meng. Minimum Energy Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, 1999.
8. I. Stojmenovic. Localized Network Layer Protocols in Wireless Sensor Networks based on Optimizing Cost Over Progress Ratio. *IEEE Network*, 20(1):21–27, 2006.
9. I. Stojmenovic and X. Lin. Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, 2001.
10. I. Stojmenovic and X. Lin. Power-Aware Localized Routing in Wireless Networks. *IEEE Tran. on Paralell and Distributed Systems*, 12(10):1122–1133, October 2001.
11. H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, 32(3):246–247, March 1984.

A Graph-Center-Based Scheme for Energy-Efficient Data Collection in Wireless Sensor Networks

Dajin Wang

Department of Computer Science
Montclair State University,
Upper Montclair, NJ 07043, USA
wang@pegasus.montclair.edu

Abstract. We consider the problem of sensor data collection in a wireless sensor network (WSN). The geographic deployment of sensors is random, with an irregular network topology. We propose a data collection scheme for the WSN, based on the concept of the *center of the graph* in graph theory. The purpose of the scheme is to use less power in the process of data collection. Because it is mostly true that the sensors of WSN are powered by batteries, power saving is an especially important issue in WSN. In this paper, we will propose the energy-saving scheme, and provide the experimental results. It is shown that under the energy consumption model used in the paper, the proposed scheme saves about 20% of the power collecting data from sensors.

Keywords: Energy efficiency, Graph center, Hierarchical structures, Wireless networks, Wireless Sensor Networks.

1 Introduction

A Wireless Sensor Network (WSN) is composed of a large number of sensor nodes, and one (or a few) "central" node(s). The sensors are deployed in various physical environments mainly for the collection of physical world data. The data are transmitted to, or gathered by, the central nodes for aggregation, analysis, and processing. The central nodes also play the role of manager of the WSN. The communication among nodes is all via wireless means. Therefore all nodes are equipped with radio transceivers/receivers. WSNs have very promising prospect in many applications, such as environment monitoring, traffic monitoring, target tracking, and fire detection.

Different models of WSN have been proposed. However some basic characteristics can be observed that are common in most proposed models.

- They are all composed of a large number of sensor nodes, and a small number of master nodes (a.k.a. central nodes, or base station);
- All sensor nodes are relatively low cost, perform relatively limited computational operation. Their main job in the whole system is to collect raw

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 579–587, 2006.

data, and render it to the master nodes, with or without some primitive preprocessing;
- The master node(s) collect the data from all sensors, and analyze/process them. They are more powerful, costlier processors than ordinary sensors. The master nodes are also the managers of the network.

A WSN can have either just one central node or a group of central nodes, depending on the network's scale of geographical coverage and/or cost effectiveness consideration. In a single-center WSN, the central node, a.k.a. *base station*, collects and processes data from all sensors. It is also the sole manager of the entire network system. In a multicenter WSN, the tasks of data collection, aggregation, processing, and network management are distributed among a group of nodes working collaboratively. The organization of these master nodes is one of the essential issues in the design of WSN architecture.

There are many different WSN models. Topologically speaking, a WSN can be of *regular* or *arbitrary* topology. One example of regular topological structure is the COSMOS model (Cluster-based heterOgeneouS MOdel for Sensor networks) proposed in [8]. COSMOS is a cluster-based, hierarchical model for WSN. It comprises of a large number of low power, low cost sensors, presumably distributed in a large physical environment. The distribution of sensors is close to uniform. That is, in each unit area there is a sensor with high likelihood. Sensors are organized into equal-sized, square-shaped clusters according to their spatial proximity. For each sensor cluster, there is a clusterhead. Sensors within a cluster communicate in a time synchronized manner, using single hop communication. The clusterheads form a mesh-like topology and communicate asynchronously. In an WSN of arbitrary topology, sensors are deployed in a random manner. The network can then be modeled by a graph $G = (V, E)$. Each node in V represents a sensor. Each edge in E linking nodes u and v represents communication between u and v via wireless means. One of the nodes is designated as the base station of the WSN.

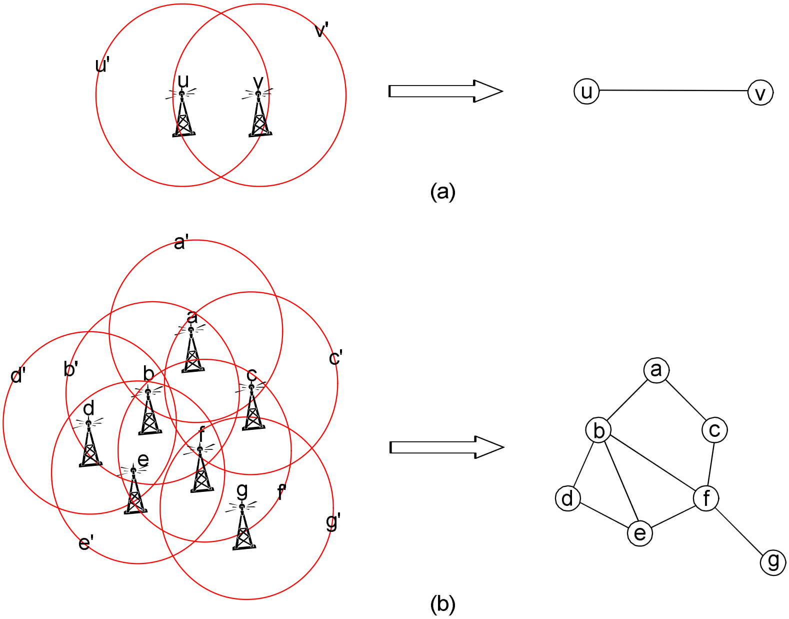
In this paper, we consider the problem of collecting data, from all sensor nodes to the base station, in an energy-saving manner. Using the COSMOS model's hierarchical idea, our proposed scheme applies a hierarchical, two-phase approach to the arbitrary topology. That is, the WSN is divided into logical hierarchies. The lower level sensors are grouped into *clusters*, and a *clusterhead* collects data within the cluster. The collected data are aggregated, preprocessed, and then forwarded to the base station. The purpose of the scheme is to use less power in the process of data collection. By the nature of WSN, all sensors are supposed to be powered by batteries. Therefore energy preservation is an especially crucial issue in WSN. The experiments show that under the energy consumption model used in the paper, the proposed scheme can save about 20% of the power collecting sensor data.

The rest of this paper is organized as follows. In Section 2, we describe the WSN model we will be working on. In Section 3, we will present the clustering scheme minimizing energy consumption. The scheme is based on the concept of the center of the graph in graph theory. Section 4 presents simulation results to demonstrate proposed scheme's gain in energy saving. Section 4 also gives concluding remarks and discusses possible directions the work of this paper can be extended.

2 The Sensor Network Model

A wireless sensor network resembles a conventional parallel and distributed systems in many ways. However, several unique characteristics standout to call for redefinition, or modification, of the network model. Those characteristics include energy efficiency consideration, communication reliability, and global awareness of individual nodes, among others. Because of the wide diversity of sensor applications, it is hard to capture all characteristics in one single model.

In this paper, we consider the WSN with its sensors randomly deployed, without following any proximal patterns. A sensor communicates with another one via radio transmitter/receiver. If a node needs to transmit to another node out of its radio range, the message has to be relayed by intermediate nodes. Such a WSN can be readily modeled with a graph $G = (V, E)$, illustrated in Figure 1. Each node in V represents a sensor. There is a link $(u, v) \in E$ if and only if sensors u, v are in each other's radio transmission range (Figure 1 (a)). In Figure 1, the primed letters (a', c' etc.) on the radio circle identify the sensors they belong to. Figure 1 (b) is an example WSN of seven sensors and its corresponding graph.



(a)

(b)

Fig. 1. (a) Node/edge definition in the graph model; (b) An example set of randomly deployed sensors, and its corresponding graph

For the purpose of power preservation, we assume that all sensors use as little power as possible for radio transmission, so that the transmission range covers just a few neighboring sensors. We also assume that there is only one transmission range, as opposed to multi-range models in some literatures. If a sensor wants to send message/data to the base node, it can only do so by relaying through intermediate sensors (routing scheme in this context is another issue, which will not be addressed in this work). We also assume a connected graph. That is, we do not consider isolated sensors or components in the WSN.

We quantify the energy dissipated by one hop of sensor transmission to a normalized unit. Refer to the example in Figure 1 again: If sensor a wants to send one piece of data to sensor g, at least 3 hops are needed; therefore 3 units of energy will be consumed, e.g. 1 unit for transmission from a to c, 1 from c to f, and 1 from f to g. In the discussion of the following section, we only consider data relaying via a shortest path.

3 A Hierarchical Scheme for Energy-Efficient Data Collection

3.1 The Designation of Base Station

We'd like to designate such a sensor as the base station, that it uses the least amount of total power to collect data from all sensors. To formulate the problems quantitatively, we first assume a model for calculating power consumption. It should be pointed out that the model is a simplified abstraction from vastly variable real scenarios. Refer to Figure 1 (b) again. We use the number of relaying hops to represent needed power to transmit data from sensor to base. The farther the sensor, the more hops are needed to relay the data, and the more power is consumed. Secondly, to measure the saving in power, we focus on the scenario of base station collecting one unit of data from each sensor. A natural choice would be to pick a "central node" of the underlying graph G. The central node(s) of an arbitrary graph can be established through the following definitions.

Definition 1. *In an undirected graph $G = (V, E)$, the eccentricity of a node v is the greatest distance between v and any other node.*

Definition 2. *The radius of a graph G is the minimum eccentricity of any node in G.*

Definition 3. *The center of a graph G is the set of nodes of G whose eccentricity is equal to the radius.*

In the examples of Figure 2, the number by a node is the node's eccentricity, and the grey nodes constitute the center of a graph. It can be observed that the notion of "center nodes" is based on the idea that these nodes have the shortest distance to all other nodes. So it would be appropriate to designate one of the center nodes to be the base station. In the examples of Figure 2, the circled nodes are designated as base station. We use the total number of hops

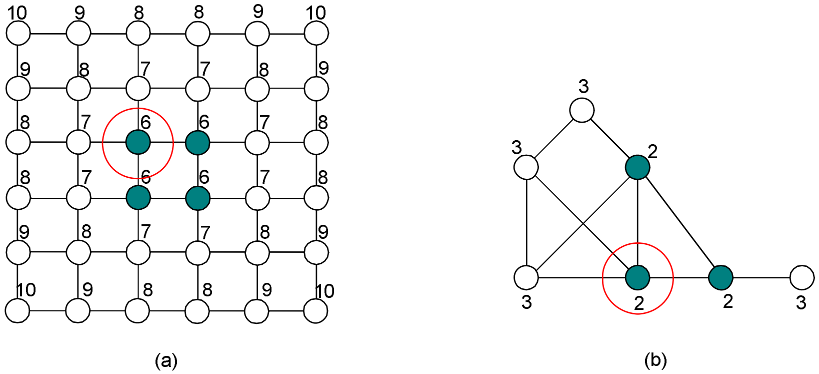


Fig. 2. Two example graphs: Eccentricities and centers

to represent power incurred in the process of data collection. To calculate power consumption for collecting one unit of data from all sensors, we just add up the distances from all sensor nodes to the base node. In Figure 2, the total power consumption for the two example WSNs are 108 in (a), and 8 in (b), respectively.

Having just one node performing the function of base for the entire WSN would be ideal. However it might not be feasible as the size of WSN grows larger, and the geographic range wider. Issues such as energy limitation, energy balancing, and scalability make a single-base WSN not only unfavorable, but also difficult to implement. The proposal of hierarchical organization of WSN [8,10] is to distribute computational and managerial tasks to a group of *clusterheads*. The approach will reduce the communication traffic in network, as well as the overall power consumption.

3.2 The Hierarchical Clustering Scheme

In the hierarchical approach, the whole WSN is divided into a set of smaller network clusters. There is still a base station for the whole WSN, chosen from the center nodes as defined in Definition 3. The data collection of base station is now performed in two phases. In the first phase, all clusterheads collect data from sensors in their own clusters. The data is aggregated and/or preliminarily processed in clusterheads. In the second phase, the WSN's base station collects data from all clusterheads. Figure 3 illustrates the structure of hierarchical WSN.

In Figure 3, the whole network is divided into $|C|$ subnetworks (clusters), where C is the set of center nodes as defined in Definition 3. *Each subnetwork of c_i consists of nodes that are closer to c_i than to any other center nodes.* Each subnetwork will then have a center node according to Definition 3, which will act as the clusterhead of the respective cluster. The $|C|$ clusterheads form a network at the upper hierarchy. At the center of upper hierarchy is the WSN's base station. Two examples are illustrated in Figure 4. In Figure 4 (a) and (b), the original networks have 4 and 3 center nodes, respectively. So they are divided

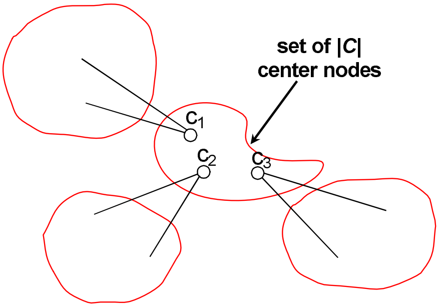


Fig. 3. Hierarchical division of WSN

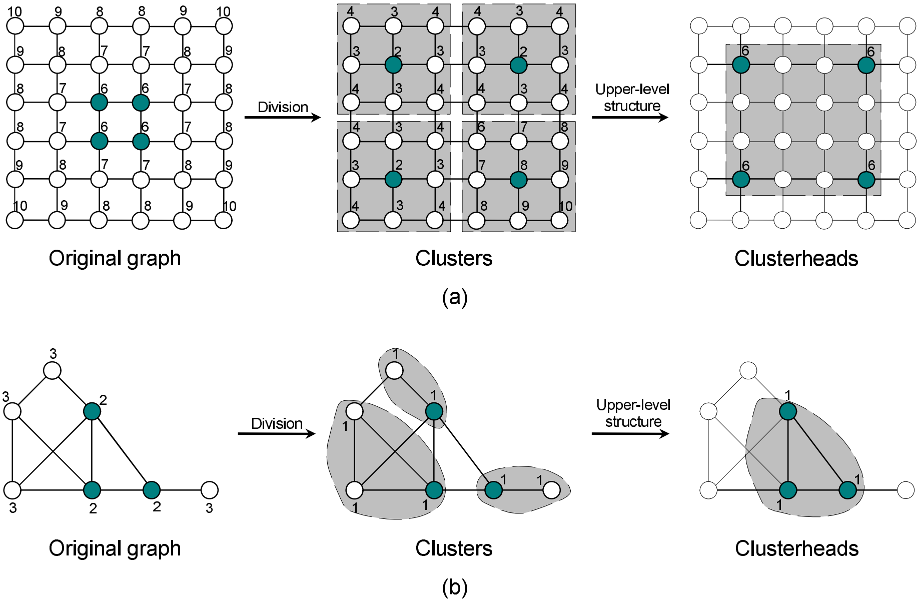


Fig. 4. Examples of hierarchical division

into 4 and 3 clusters. The numbers by the clustered nodes represent their new eccentricities *within* the cluster. The far-right of Figure 4 shows the upper-level of the hierarchy, composed of the clusterheads. The number by a clusterhead is its eccentricity in the upper-level graph. In both examples of Figure 4, the upper-level eccentricities all happen to be the same, but this is not true in general. An upper-level central node can be determined by those upper-level eccentricities, which will be designated as the base node for the whole WSN.

To calculate the total hops incurred in collecting one round of data from all sensors, first in all subgraphs, get the sum of hops from sensors to the corresponding clusterheads. Denote the total sums in all subgraphs as $Cost_I$. Then in the second phase, get the sum of hops from the upper-level nodes to the center node. Denote the sum as $Cost_{II}$. The total cost for one round of data collection is $Cost_I + Cost_{II}$. For the examples in Figure 4 (a) and (b), the total costs of the two-level approach are 60, and 6, respectively. Comparing with the single-level approach, the power-saving rates are 44% and 25%, respectively.

We summarize the steps of hierarchical scheme in the description below.

Cost computation for two-level approach

1. Calculate *eccentricities* for all nodes.
2. Find out all nodes with minimum eccentricities. The subgraph induced by these nodes is called the *center* of G, denoted $C(G)$.
3. Divide G into $|C(G)|$ subgraphs – a node v belongs to a center node $c' \in C(G)$ if $d(v, c') = \min\{d(v, c)|c \in C(G)\}$. If there are more than one such c', then pick any one.
4. After division, calculate eccentricities and centers in all subgraphs.
5. In each subgraph, calculate the cost of every node to a local center node (the sum of hops along the path), and then get sum of all costs. This is the cost for the subgraph.
6. Get the sum of costs of all subgraphs. Call it $Cost_I$ (level-one cost).
7. Calculate the center of C_I, where C_I is the set of all (level-one) center-nodes in subgraphs. Call the level-two center c_{II}. Calculate $\sum\limits_{v \in C_I} d(c_{II}, v)$. Call it $Cost_{II}$ (level-two cost).
8. Total cost: $Cost_I + Cost_{II}$.

4 Simulation Result and Concluding Remarks

For irregularly connected networks, simulation is an effective instrument to quantify the competence of a proposed hierarchy method. In the preceding section, we have proposed an energy-efficient clustering scheme for Wireless Sensor Networks, based on the center nodes of a network's underlying graph. Figure 5 shows the ratio of costs for the proposed hierarchical approach vs. the single base-station approach. The scenario being simulated is one round of data collection from all sensor nodes of the network. The cost is the total number of hops incurred in the process.

For the simulation, randomly connected graphs of different sizes are generated, and the corresponding costs for the two approaches are computed and compared. Graphs of 10 nodes through 70 nodes are simulated. The cost for a specific size is the average cost for many random graphs of that size. We can see from Figure 5 that the average cost for the hierarchical scheme is around 80% that of the single base-station approach. In other words, about 20% hops can be saved using the proposed hierarchical scheme for data collection.

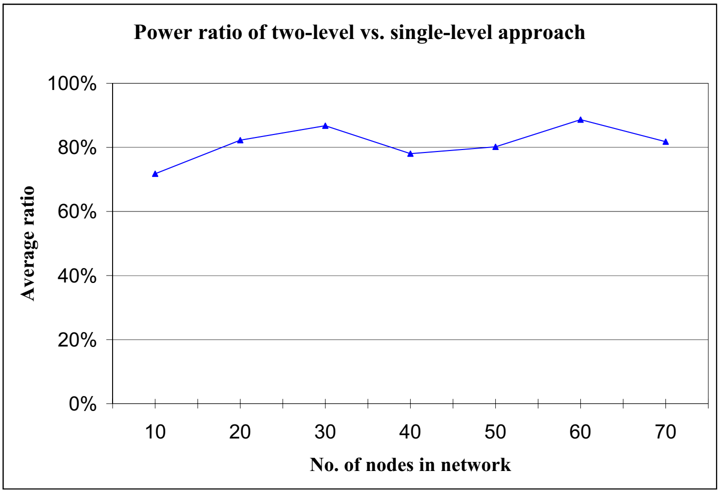


Fig. 5. Simulation result

It is worth pointing out that the proposed scheme is just a “soft” protocol for the task of data collection. It does not impose any hierarchical structure on the original WSN. For other applications, different protocols can be employed on the same network.

Power conservation is a problem that has been extensively addressed in the research of wireless networks, where many open problems exist regarding this issue. We can see some obvious directions to which the work of this paper can be immediately extended. For example, the proposed scheme only considers the saving of *total* transmission hop counts for the *entire* WSN. It does not address the issue of power balancing among sensors. That is, the closer a sensor to a clusterhead, the more power it consumes, because it relays more data packets to the clusterhead. Another worthwhile topic for further research is that in this paper, we assumed a rather simple communication model, not only for tractability reason, but also for the lack of a statistical model that better reflects the realistic transmission activities. Finding an appropriate communication model that’s more realistic as well as facilitating tractability would greatly increase the practical relevance of the hierarchical scheme.

References

1. J. Cao and F. Zhang, “Optimal configuration in hierarchical network routing,” *Proc. 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, Edmonton, Alberta, Canada, May 1999, pp. 249-254.
2. T. Chu and I. Nikolaidis, “Energy efficient broadcast in mobile ad hoc networks,” *Proceedings of Ad-Hoc Networks and Wireless*, 2002, pp. 177-190.

3. R. Marks II, A. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: optimizing using the viability lemma," *Proceedings of IEEE International Symposium on Circuits and Systems*, 2002, pp. 245-248.
4. O. Egecioglu and T. Gonzales, "Minimum-energy broadcast in simple graphs with limited node power," *Proceedings of lASTED International Conference on Parallel and Distributed Computing and Systems*, 2001, pp. 334-338.
5. X.-Y. Li and P.-J. Wan, "Constructing minumum energy mobile wireless networks," *ACM Journal of Mobile Computing and Communication Reviews*, Vol. 5, No. 4, 2001, pp. 55-67.
6. S. Lindsey and C. Raghavendra, "Energy efficient broadcasting for situation awareness in ad hoc networks," *Proceedings of International Conference on Parallel Processing*, 2001, pp. 149-155.
7. E. Lloyd, R. Liu, M. Marathe, R. Ramanathan, and S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *Proceedings of Annual Workshop on Mobile and Ad Hoc Networking and Computing*, 2002, pp. 123-134
8. Mitali Singh and Viktor K. Prasanna, "A hierarchical model for distributed collaborative computation in wireless sensor networks," *International Journal of Foundations of Computer Science*, 2004, Vol. 15, No. 3, pp. 485-506.
9. J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," *Proceedings of IEEE Infocom*, 2000, pp. 585-594.
10. M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," *Tenth ACM Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Fort Worth, TX, Oct. 2002.

An Energy Efficient Event Processing Algorithm for Wireless Sensor Networks

S. Selvakennedy

School of Information Technologies, University of Sydney,
NSW 2006, Australia
skennedy@it.usyd.edu.au

Abstract. Wireless sensor networks are being deployed in many monitoring scenarios as fundamental data collection protocols are becoming efficient in handling simple sense-and-send function. As the computation capacity of sensor nodes grows, these nodes are capable of performing more complicated functions. Moreover, the need to realize the complete loop of sense-control-actuate as the wired sensing facility demands for more in-network processing to be able to generate meaningful in-network actuation. One such useful primitive function for many applications is edge or boundary detection of a phenomenon. In this work, we propose a localized edge detection algorithm using basic geometry rules that just uses one-hop neighbourhood information. This algorithm is accordingly benchmarked against one of the best localised edge detection scheme available in the public domain. It is found that the proposed algorithm readily outperforms its counterpart. Moreover, its energy efficient operation is attractive as a primitive implementation for other more complex primitives or applications themselves.

Keywords: In-network processing, edge detection, localized algorithm, data processing, pattern recognition.

1 Introduction

Wireless sensor networks (WSNs) have been demonstrated to be useful in many different application domains, both for civil and military purposes. Typically, these applications entail simple sensing and relaying physical characteristics of interest back to a central server for processing and storage. This is functionally similar to the traditional wired sensing facility. Recently, there have been some proposals to introduce more *intelligence* into the network itself in form of data aggregation/compression function that is informed of the spatial and temporal correlation structure [1, 2]. This is inline with the advance of the hardware especially the node's processing capacity according to Moore's Law.

With the foreseeable increase of in-network processing, it is beckoning to provide automated response through suitable controllers with actuators to complete the sense-control-actuate loop as in the traditional control systems. A generic sensor-to-actuator coordination model for an event-driven application is proposed in [3]. Accomplishing such a feedback loop mostly requires application-specific knowledge and data

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 588 – 599, 2006.

semantic processing against statistical data processing required for data aggregation. Even though such semantic processing notionally requires application-specific knowledge, there are certain primitive functions that may be applied across many application domains. For instance, the notion of an event spanning across an area, instead of occurring at specific points, is readily envisaged in numerous applications such as in monitoring seismic disturbances, contaminant flows, chemical concentrations as well as events occurring in a relatively shorter time span like fire emergencies. For these applications, basic primitives like the extent of the *event region*, the number of nodes on the *event boundary* and the *event epicentre* might be useful for characterizing the phenomenon for appropriate feedback generation by the controllers and actuators. In any emergency applications involving human where the latency involved in the identification of an event to the response generation may imply life or death, it is paramount to know the event boundary as soon as possible. If available, it facilitates real-time response generation, such as activating the appropriate sprinklers for tackling fires before emergency squads arrive at the scene.

For such real-time needs, it might only be feasible to accomplish such task within the network itself in a distributed manner through collaborative processing. In the case where the event spans a large area, enumerating all sensors within the event region before initiating a response is likely to miss its response deadline limit. A more scalable solution might entail identifying just the nodes on the event boundary, whereby the associated geometry of the event could subsequently be deduced. Accordingly, the controller (i.e. one of the sensor nodes) could then activate the appropriate actuators to generate the required response. Such an event processing primitive is useful in many scenarios. The primitive for detecting nodes on the event boundary is liked to *edge detection* as in image processing by Cintalapudi et al. [4], and congruously termed *boundary estimation* by Nowak and Mitra [5] in their pioneering efforts.

An edge detection algorithm determines whether a node lies on or near an edge of an application-defined event of interest. There are two limitations to bound the estimation performance, namely sensor node spatial density and the noise in the readings. Since edge detection is a well-known primitive in image processing with efficient algorithms available for centralized processing, they could be naturally applied towards sensor data. Liu et al. [6] presented a centralized scheme based on Hough transforms to detect and track events. This transformation maps a non-local phenomenon to a point in the dual-space and the locations of the sensor nodes to a set of lines that partitions the dual space. Such a transformation reduces the detection problem to finding and tracking the point bounded by these lines. In [7], a generalized hierarchical architecture based on wavelet transforms is proposed for multi-resolution querying of regularly-placed sensor nodes. This architecture is demonstrated to be useful for queries involving non-local phenomenon such as edges. Inspired by [7], Nowak and Mitra [5] presented a multi-hierarchical scheme for estimating the boundary of a large-scale phenomenon by sharing of lower dimensional statistics (against all readings) with ancestral nodes within the hierarchy and pruning of irrelevant branches. Even though this scheme is shown to achieve optimal error-energy tradeoff, the model did not consider the cost of forming and maintaining such a hierarchical structure. In [4], three localized algorithms are proposed based on ideas in statistical thresholding, image processing and pattern recognition. The statistical scheme involves testing a Boolean function of neighbour values against a predefined application threshold. In their second scheme, they applied the Prewit filter algorithm from image processing by computing the gradients in *x* and *y* directions. If the composite gradient is larger than a threshold, an edge node is declared. The final scheme is based on the pattern recognition classifier (PR-Classifier), which aims to classify nodes into a bipartite set of similar and dissimilar values. The partition is validated against a *partition validity measure* [8]. A successful partition implies the presence of an edge. It is shown that PR-Classifier performed the best as it explicitly encodes the notion of location of the nodes in its estimation, and it does not require a threshold to influence its performance. However, for optimal performance, all the above schemes require neighbourhood information greater than direct neighbours. Using a similar approach to Prewit filtering as above, Devaguptapu and Krishnamachari [9] demonstrated that such a scheme is amenable for edge detection but it is highly sensitive to node density and node radio range.

In this paper, we propose a novel localized edge detection scheme based on basic geometric rules and trigonometry. Since it relies only on direct neighbours (i.e. one-hop nodes), its message overhead is just linear of the number of neighbours. It relies on *tangent fitting* between selected neighbours to decide whether a node is on edge. Due to the design of the proposed algorithm, it is also resilient against random isolated measurement errors, which are due to faulty equipment that are likely to be uncorrelated [10]. Unlike other localized algorithms, our approach is applicable to any convex regions irrespective of the event area, not just large elliptical regions only. To highlight its performance gain, it is compared against the *best* localized edge detection from [4]. We find our proposal readily outperforms PR-Classifier over a broad range of operating conditions and scenarios in terms of accuracy as well as energy budget.

In Section 2, model assumptions and terminology are given. The details of the proposed algorithm are given in Section 3. The benchmark PR-Classifier algorithm is also described there. The results from our extensive simulations are presented and discussed in Section 4, followed by some concluding remarks in Section 5.

2 Model Assumptions

Sensor nodes can be carefully deployed in a grid-like topology or randomly dispersed on a plane due to terrain inaccessibility. The main requirement for our purposes is that each node knows its location perhaps using some kind of localization technique [11]. Although we assumed a node location is specified by (*x*,*y*), our proposed algorithm is readily applicable in a 3-D space.

The notion of event edge is defined in [4]. Just for completeness and ease of exposition of the proposed algorithm later, we highlight the core definitions and concepts here. The initial step in event edge detection is for the nodes to discover which sensor readings are *interesting*. We assume that the range of interest is predefined by the application, or may be learnt by monitoring the normal conditions over time such that they can recognize interesting event readings [12]. Accordingly, a reasonable threshold is defined at each sensor node that enables it to ascertain whether its reading corresponds to an event. However, it is possible for the nodes with faulty sensors to report an event when they are outside the affected region. To counter such problems, we may resort to use a probabilistic mechanism to isolate uncorrelated faults, say using the Bayesian algorithm in [10]. In our case, this however entails dual broadcasts by each node with an *interesting* value (i.e. one for announcement of its reading, and the next for announcing its confidence), resulting in significant overhead. Instead, it is more energy efficient if the edge detection algorithm itself exhibits sufficient robustness against such errors.

All nodes within the event region are termed *affected nodes*. Otherwise, they are *unaffected nodes*. The edge of a phenomenon is the imaginary boundary between these sets, i.e. the set of all points (x,y) where the sets intersect. This is the *ideal edge*, and represents the ground truth that bounds the event region. However, due to the deployment artifact and finite limit on the sensor node density, this edge could only be approximated. A sensor node declares itself to be on edge, if it is closer to the ideal edge than any of its neighbours as shown in Fig. 1. All shaded nodes represent affected nodes and the numbered nodes are the edge nodes. Even though node 4 is not on the ideal edge, it is the nearest node to this edge than its neighbours. Here, the edge node set forms a convex region encapsulating all other affected nodes. We exploit the notion of convexity of some special class of event regions to perform efficient edge detection.

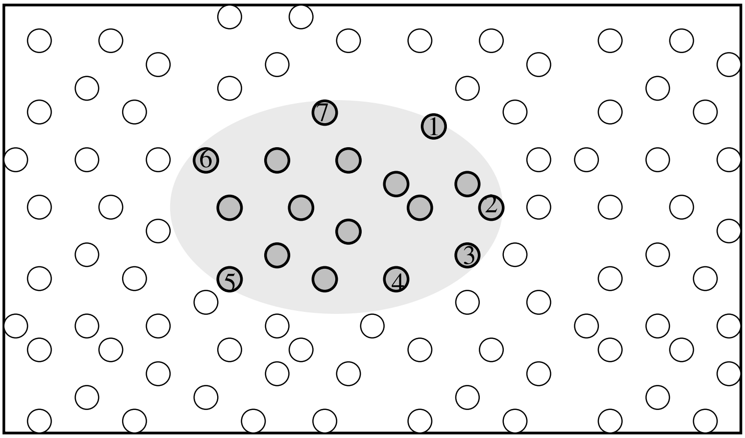


Fig. 1. The shaded area represents an event region of affected nodes. The edge nodes are numbered, and the other affected nodes are in the interior.

3 The Edge Detection Algorithm

When an event of interest breaks out on the monitored terrain, some nodes sensing within the affected region capture the phenomenon. However, to be able to draw a conclusive picture about the event, those affected nodes should exchange their information with the neighbours to gather more information about the event. Even though the sink could readily compute this information with complete network knowledge, some applications might have stringent latency requirements, while others might require quick response for in-network actuation. Moreover, more energy savings could be realized if we could trade off between communication and in-network processing.

As stated earlier, our proposed algorithm utilizes a form of tangent fitting to perform the edge test. Accordingly, we term it the *T-Fit* algorithm. In order to detect nodes on the edge of the affected region, each affected node broadcasts an urgent message upon detecting an event. From these messages, the nodes could gather the extent of the event coverage. Based on this direct neighbourhood information, an affected node classifies its affected neighbours into four quadrants on a Cartesian system with the current node as the origin. The node is on the edge of a convex region, if the angle between the reference node as the vertex and its farthest two neighbours is smaller than π radian, as shown in Fig. 2.

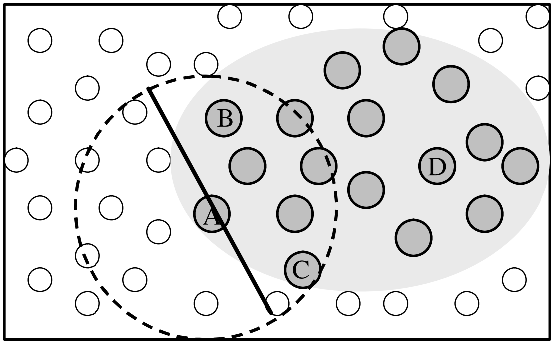


Fig. 2. A convex event-affected region is shown as a shaded area with affected nodes. Dashed circle centred at node A represents its radio range. A tangent segment is drawn on node A parallel to its farthest affected neighbours of B and C. From this tangent-fit test, node A is an edge node.

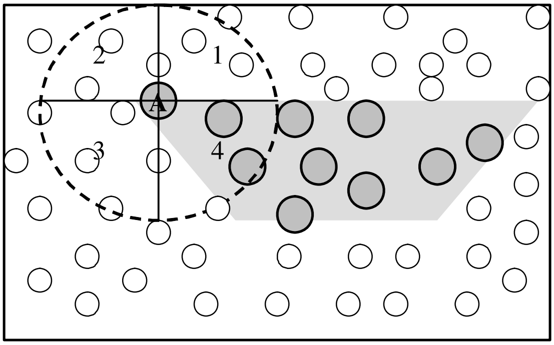
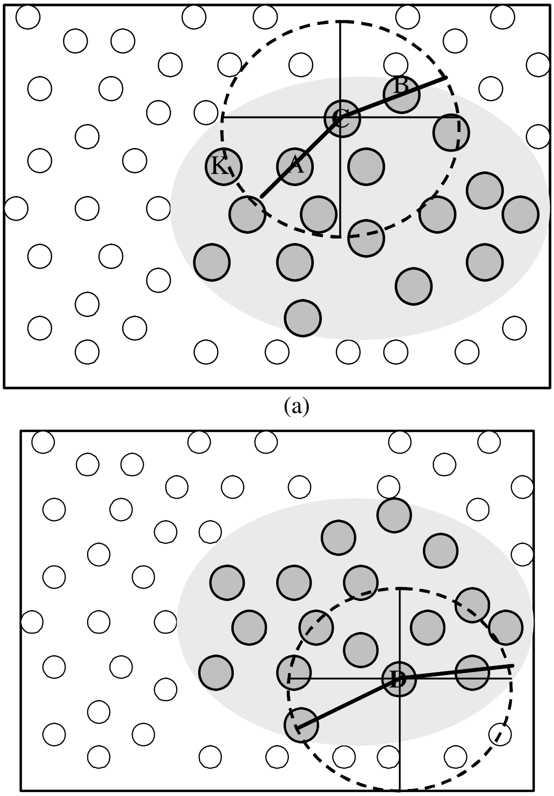


Fig. 3. A convex trapezoidal region that may have been bounded by a non-natural artifact. Node A is a valid edge node with all its affected neighbours populated only in the fourth quadrant.

As the nodes are classified into four quadrants, if the affected neighbours are only populated in two quadrants, this node is definitely on the boundary, like node A in Fig. 2. However, if the neighbours are populated in all four quadrants, such a node is surely not on the boundary, like node D. When the neighbouring nodes are populated in either one or three quadrants, more careful considerations are required. The former situation might represent an isolated erroneous sensor reading, whereas the latter might imply an edge node if the neighbours are inside the tangent. To clarify our solution for the first case, Fig. 3 depicts a situation where a convex region has a valid edge node in A whose neighbours are only populated in one quadrant. As such, as long as a node has another affected node as a neighbour, it is deemed eligible as an edge.



(a)

(b)

Fig. 4. Tangent-fit test for edge detection on nodes C and D with their affected neighbouring nodes populated in three quadrants {1,3,4} and {1,2,3}, respectively. (a) Node C is on edge as the angle is lesser than π radian; (b) Node D is not on the edge.

For the latter case where neighbours are populated in exactly three quadrants, we need to determine the node's *farthest* (in terms of angle) two affected nodes across diagonal quadrants. If the angle between these nodes is smaller than π radian, it is on the edge as depicted in Fig. 4(a). At first glance, it might appear odd that node C is deemed an edge node. One might think that if a tangent is drawn parallel to nodes B-K, node C might have failed the edge test. However, as previously stated, this is a localized algorithm that considers only direct neighbours, and node K is outside node C's radio range. In Fig. 4(b), a node that failed the edge test is depicted. To compute the angle in the tangent-test, a node has to determine its specific quadrant combination. There are four possible combinations, namely set {1,2,3}, {1,3,4}, {1,2,4} and {2,3,4}. Using basic trigonometry, we could then determine the angle between these nodes. The complete T-Fit algorithm is given in Algorithm 1.

Algorithm 1. The T-Fit Edge Detection Algorithm

```
for (all affected neighbours, $n_i$) {
// classify all affected neighbours into 4 quadrants
        // compute slope from current node
        Compute slope to $n_i$
        Classify $n_i$ into its quadrant
        if ($n_i$ in quadrant $j$) {
        // Store some information
          Increase neighbour count in quad $j$
          Check $n_i$ if it is the smallest or largest
              slope to current node

        }
        Store # quadrants has affected neighbours in
              quadsAffected
}
if (quadsAffected = 0) {
        Node has an isolated reading error
        onBoundary = false
}
else if (quadsAffected < 3)
        onBoundary = true
else if (quadsAffected == 3) {
        Compute angle among three nodes from
              non-empty quadrants
        if (angle < PI)
          onBoundary = true
}
else {
// has neighbours in all quadrants
        onBoundary = false
}
```

From Algorithm 1, it is obvious that this algorithm has time complexity as well as message complexity in linear of the number of affected nodes. For the resource-constrained sensors, these are highly desirable properties.

In [4], it is demonstrated that a linear classifier algorithm adopted from the pattern recognition literature achieved the best performance for diverse tested scenarios against statistical as well as image processing algorithms. As such, this algorithm would be an ideal benchmark for our algorithm. We will term it PR-Classifier. A classifier-based approach relies on the information provided by all neighbours irrespective whether they are affected or otherwise. Based on this information, the classifier attempts to partition data into two classes. The appropriateness of this classification may be assessed by a partition validity measure [8]. A valid partition implies the presence of an edge. In [4], a linear classifier is used to simplify the exploration of its parametric space. According to this classifier, if the valid partitioning line is within certain tolerance distance from the current reference node, this node is deemed an edge.

To investigate the performance of our algorithm against PR-Classifier, we developed a discrete-event simulation in C++. The simulations results and their analyses are given further.

4 Results and Discussions

Extensive simulations are performed to quantify the performance of the T-Fit algorithm as well as to benchmark against PR-Classifier. In most cases, there are 350 sensor nodes distributed randomly in a square $M \times M$ region with $M = 100$ m. To quantify the energy efficiency of these algorithms, the transceiver energy parameters are set as: $E_{elec} = 50$ nJ/bit and $\varepsilon_{fs} = 10$ pJ/bit/m^2 [13]. The data message size is fixed at 30 bytes. Initially, node's radio range is fixed at 15 m. Unless otherwise stated, all the following investigations adopt these values as their system parameters. For all simulation results in this paper, each experiment is repeated 20 times and a 95% confidence interval is obtained. We choose not to display this interval on the graphs to avoid clutter. The performance metrics utilized in our investigations are:

- *Mean distance to boundary*: This metric represents the average distance of the identified edge nodes to the virtual event boundary.
- *Total Energy*: This metric represents the energy dissipated by all sensors handling message exchanges in the edge detection protocols.

4.1 Network Visualisations

For ease of comprehension and appreciation of subsequent comparative results, a set of results are identified and visualised here. For the initial set of visualisations, the event region is assumed to be circular with a 35-m radius. Figure 5 depicts the visualisation for T-Fit with affected nodes shaded and the identified edge nodes further bounded by a square. In this visualisation, it is clearly visible how T-Fit is able to correctly identify the boundary of the event as one could visually trail the edge nodes. Such an accurate edge detection would assist many applications highlighted in the Introduction section to monitor the perimeter of the phenomenon of interest.

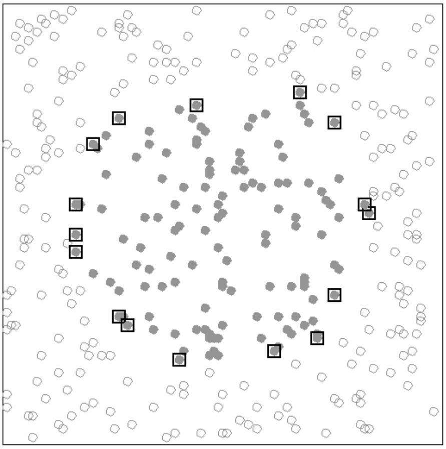


Fig. 5. Visualisation of T-Fit for network with 350 nodes at 15-m radio range of a circular event region with 35-m radius. The affected nodes are shaded, and the edge nodes are bounded by a square.

To understand the effectiveness of the above algorithm against other convex regions, we investigated its performance against rectangular event regions. Figure 6 depicts the edge nodes identified by our T-Fit algorithm. As this algorithm is designed for a general convex region, it is able to identify the appropriate edge nodes with a clear boundary visible from the selected edge nodes. Noticeably, there are no other affected nodes lying outside of the imaginary boundary formed by these edge nodes.

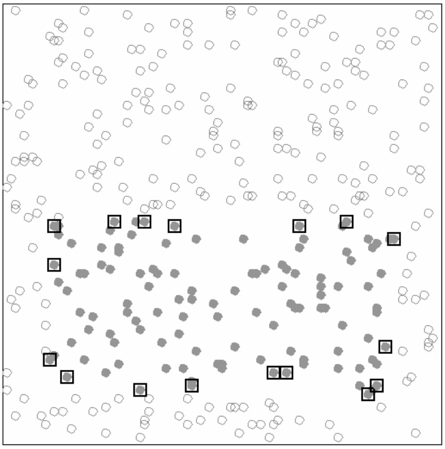


Fig. 6. Visualisation of T-Fit for a $80\text{m} \times 40\text{m}$ rectangular event region

4.2 Impact of Node Density

For the following results, the performance is obtained for circular event regions only. In this study, we examine the influence of node density on the edge detection behaviour in terms of both accuracy and energy usage. For the same monitored terrain area, we varied the number of sensors in range [80,400). In Fig. 7, the mean distance of edge nodes is plotted for both algorithms. As expected, both algorithms' accuracy improve with density and the rate is almost similar. However, T-Fit achieves 40% higher accuracy on average against PR-Classifier. T-Fit almost surely identified the nodes farthest from an event's epicentre as edges, whereas PR-Classifier sometimes identified a thick edge with many nodes within the tolerance range of the border, even when a small tolerance is used.

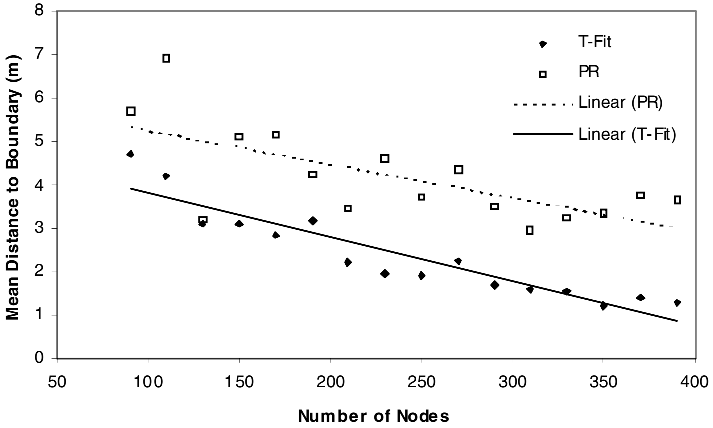


Fig. 7. Mean distance to event boundary against number of nodes for a 35-m circular event area and nodes with 15-m radio range

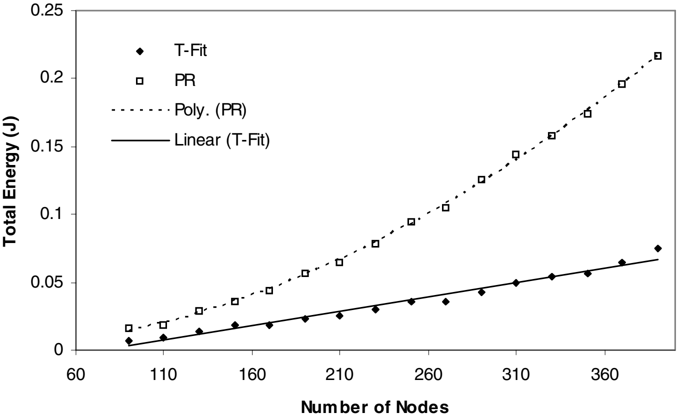


Fig. 8. Total energy dissipation against number of nodes for a 35-m circular event area and nodes with 15-m radio range

To appreciate the energy usage behaviour of both algorithms, Fig. 8 depicts the total energy usage plot for the same scenario. Since PR-Classifier requires up to 2-hop neighbourhood information for its best performance, its energy usage increases in quadratic to the radio range whereas only linear for T-Fit. Thus, it clearly shows that not only T-Fit is able to achieve higher accuracy, but it is also able to be energy efficient simultaneously meeting most needs of the sensor applications

5 Conclusions

As wireless sensor networks evolve from a simple sense-and-send technology towards more intelligent in-network processing, there seems to be more applications that could benefit from such networks. For certain applications whereby there is a need to respond in real-time towards a phenomenon of interest, it is also attractive to provide in-network actuation mechanism to minimise human intervention. One of the identified primitive functions that should be located in the network is edge or boundary detection. Even though edge detection is readily accomplished at the collection centre with a complete network knowledge probably using well-known image-processing algorithms, it is not trivial to perform them in a localized manner using only local information.

In this paper, we introduced an energy-efficient edge detection algorithm based on simple geometry. By exploiting convexity of a region, our algorithm uses tangent-fitting to identify edge nodes. It is also benchmarked against one of the best localized algorithm in the literature. Using extensive simulations, it is demonstrated that T-Fit readily outperforms its competition both in terms of accuracy as well as energy efficiency. It performs equally well against different convex regions. Moreover, its time complexity is just linear in number of neighbours against others.

As a primitive function, we believe this algorithm could readily be used by more complex primitives or by the applications themselves. One such primitive that may benefit employing this edge detection technique is the event area computation algorithm. The feasibility of such an extension is left for further work.

Acknowledgments. This work was supported in part by the ARC Discovery Project grant DP0664782.

References

1. Cristescu, R., Beferull-Lozano, B. ,Vetterli, M.: On network correlated data gathering. In: INFOCOM (2004) 2571-2582.
2. Gupta, H., et al.: Efficient gathering of correlated data in sensor networks. In: MobiHoc '05. ACM Press, Urbana-Champaign, IL, USA (2005) 402-413.
3. Melodia, T., et al.: A distributed coordination framework for wireless sensor and actor networks. In: 6th ACM international symposium on Mobile ad hoc networking and computing. ACM Press, Urbana-Champaign, IL, USA (2005) 99-110.
4. Chintalapudi, K.K. ,Govindan, R.: Localized edge detection in sensor fields. In: IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA (2003) 59-70.

5. Nowak, R. ,Mitra, U.: Boundary estimation in sensor networks: Theory and methods. In: Information Processing in Sensor Networks. Springer-Verlag, Palo Alto, CA, USA (2003) 80–95.
6. Liu, J., et al.: A dual-space approach to tracking and sensor management in wireless sensor networks. In: 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, GA, USA (2002) 131-139.
7. Ganesan, D., Estrin, D. ,Heidemann, J.: Dimensions: Why do we need a new data handling architecture for sensor networks? ACM SIGCOMM Computer Communication Review 33 1 (2003) 143-148.
8. Duda, R.O., Hart, P.E. ,Stork, D.G.: Pattern Classification. 2nd ed. John Wiley and sons (2000).
9. Devaguptapu, D. ,Krishnamachari, B.: Applications of localized image processing techniques in wireless sensor networks. In: SPIE Aerosense'03, Orlando,FL,USA (2003) 247–256.
10. Krishnamachari, B. ,Iyengar, S.: Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. IEEE Transactions on Computers 53 3 (2004) 241-250.
11. Niculescu, D.: Positioning in ad hoc sensor networks. IEEE Network 18 (2004) 24-29.
12. Maxion, R.A.: Toward Diagnosis as an Emergent Behavior in a Network Ecosystem. Emergent Computation (1991).
13. Heinzelman, W.B., Chandrakasan, A.P. ,Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1 4 (Oct 2002) 660 - 670.

A Novel Energy-Efficient Backbone for Sensor Networks★

Hekang Chen[1], Shuigeng Zhou[1], Bin Xiao[2], Jihong Guan[3], and Bo Huang[1]

[1] Dept. of Computer Sci. and Eng., Fudan University, Shanghai 200433, China
{hkchen, sgzhou}@fudan.edu.cn
[2] Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
csbxiao@comp.polyu.edu.hk
[3] Dept. of Computer Sci. and Techn., Tongji University, Shanghai 200092, China
jhguan@mail.tongji.edu.cn

Abstract. The construction of backbone is a fundamental problem in sensor networks. As being critical to routing, data fusion and query broadcasting, the backbone should contain as fewer nodes as possible. Meanwhile, it also should be power-efficient in order to prolong its lifetime. In this paper, we propose a novel design of backbone for sensor networks by minimizing its total transmission cost. We term this kind of backbone *Minimum-Weight Backbone* (simply *MWB*). The construction of MWB is proved to be NP-Complete. Two heuristic algorithms (one is centralized and the other is distributed) are developed for constructing such backbone. The centralized algorithm, executed at the base station, is to find an optimal backbone by using the information of all nodes and links in the network. The distributed algorithm run at each node, however, uses the information of its neighbors and two-hop neighbors to determine whether it should be included in the backbone. Simulated experiments are conducted to evaluate the novel backbone, and performance comparisons are also done with the existing schemes.

1 Introduction

A sensor network [1] is a wireless network that contains a large number of sensor nodes of cheap price, small size, low computation and communication capacity. It can be applied to many fields such as environment monitoring, military, and industry, etc. The sensor nodes are essential components of sensor networks. The most popular sensor mote Mica2 developed by UC Berkeley, for example, has a $7MHz$ CPU, $4KB$ RAM and $512KB$ flash. Its communication range and data transmission rate are about 100 feet and $38.6Kbps$ respectively. It uses AA batteries, and the current when working is $\sim 15mA$, and $\sim 10\mu A$ while sleeping.

★ This paper is supported by National Natural Science Foundation of China (Grant Nos. 60373019 and 90612007), partially supported by the Shuguang Scholar Program of Shanghai Education Development Foundation, Research Program of Science and Technology Committee of Shanghai Municipality (Grant No. 05dz22334). Bin Xiao is supported by HK POLYU A-PA2F and ICRG A-PG52. Jihong Guan is supported by National Natural Science Foundation of China (Grant No. 60573183).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 600–611, 2006.

In a sensor network, sensor nodes self-organize and collaborate with each other to send data to the base station (BS), from which the users can retrieve interested information. This is usually done by constructing a virtual infrastructure. However, due to the hardware and power constraints of the sensor nodes, the virtual infrastructure must be equipped with the following characteristics.

- *Power saving.* The sensor nodes usually be powered by batteries, and charging or changing batteries is always difficult (if not impossible) in an unattended environment. If a sensor node works continuously, it can last only about a week. In a large scale sensor network that consists of hundreds of or even thousands of sensor nodes, if each node works intermittently (e.g. 2% time is active), the entire network can operate over one year. To prolong its lifetime, the sensor network should finish the given task while minimizing both the number of active nodes and the working time of each active node.
- *Efficient data fusion.* When sensor nodes are active, communication is the major cause of power consumption. It is estimated that power consumption on data transmission is about 1000 times more than that on computation. Thus, a desirable infrastructure should support efficient in-network data fusion that can significantly reduce the transmission cost.
- *Low latency.* Many applications (e.g. industry monitoring and target tracking) require time-critical information. As a result, the sensed data should be sent to the users with as less latency as possible.
- *Robustness.* The harshness of the operating environment can cause serious impact on sensor nodes. The nodes or links may fail unpredictably. A sensor network should be robust enough to handle such failures in a distributed way, and guarantee the whole network work normally.

Considering the above characteristics, this paper proposes a novel backbone infrastructure to deploy sensor networks for packet routing, data fusion and broadcasting. Generally, a backbone infrastructure divides all nodes in a network into two categories: connected backbone nodes and leaf nodes (non-backbone nodes) that are only one hop to some backbone node. The backbone nodes and their links form the backbone, through which data from both the backbone nodes and the leaf nodes is transmitted to the base station. The traditional backbone [2,3] is a connected dominating set with minimum number of nodes. Here we consider the transmission cost of each link and construct the backbone with minimized total transmission cost in order to save battery power of the backbone nodes. We call the new backbone infrastructure *Minimum-Weight Backbone* (or simply *MWB*). On one hand, such backbone can prolong the lifetime of itself. On the other hand, it can save power of the leaf nodes that accounts for a large proportion of the network.

The construction of our new backbone infrastructure is proved to be NPC. For implementation, we propose two heuristic algorithms to obtain approximate answer. One is a centralized algorithm, and the other is a distributed algorithm. The centralized algorithm is implemented at the base station that knows the positions of all nodes in advance. To implement the distributed algorithm, each node keeps the information of its neighbors and two-hop neighbors, with which it is determinable whether the node is a backbone node. The distributed method is more flexible for building and updating the backbone, whereas the centralized algorithm can get solution of better performance, with fewer number of backbone nodes and less total transmission cost.

This paper is organized as follows. Section 2 surveys the related work. Section 3 introduces the new backbone and its construction algorithms. Section 4 presents the simulation results. Section 5 offers the conclusions.

2 Related Work

Roughly, there are two types of techniques for deploying sensor networks: infrastructureless approaches [4,5] and infrastructure based approaches [2,3,6,7,8,9,10].

Flooding and gossiping [4] are two classical routing mechanisms for sensor networks without infrastructure maintenance. In flooding, each sensor node broadcasts a received data packet to all of its neighbors. This process continues till the destination receives the packet. Gossiping, on the other hand, is an enhanced flooding-based routing mechanism, in which each node sends the packet to a randomly chosen neighbor. C. Intanagonwiwat et al. [5] propose an infrastructureless approach named *directed diffusion*, which is a data-centric routing method using a naming scheme for data. The base station broadcasts *interest* consisting of attribute-value pairs through the entire network. After propagation, several *gradients* path from the sources to the base station will be set up.

Recently, several kinds of infrastructure-based mechanisms have been proposed for deploying sensor networks. The most popular ones include cluster-based [6,7] and backbone-based [2,3,8,9,10] infrastructures.

Wendi B. Heinzelman et al. [6] propose Low-Energy Adaptive Clustering Hierarchy (LEACH) to deploy sensor networks. LEACH groups the nodes into clusters by selecting several cluster heads (CH), which are responsible for collecting data from their clusters and sending the aggregated data to the base station separately. The LEACH-based structure TEEN (Threshold-sensitive Energy Efficient sensor Network) [7] supports not only periodical data collection but also queries of critical data that exceeds a given threshold. It uses a multiple-hierarchy cluster-based infrastructure for both intra-cluster and multiple inter-cluster data fusion. As CHs consume more power than the other nodes, it is required that every node acts as CH by turns to balance the energy dissipation of all nodes.

Backbone-based infrastructure focuses on building a connected dominating set (CDS) [2,3,8,9] with minimum number of nodes. However, finding a minimum dominating set is proved to be NP-Complete. A number of centralized and distributed heuristic methods have been proposed to solve the CDS problem. In what follows, we will give a detailed introduction to the algorithm proposed in [3] since our distributed method is based on the algorithm. [3] gives a two-phase distributed algorithm named MPR-CDS for a node to determine whether it should be included in CDS. Here, each node has an unique ID and keeps in local memory the information of its neighbors and two-hop neighbors.

In the first phase, each node v_i computes a MultiPoint Relay (MPR) set of its own. The MPR set has the following two properties.

- Nodes in the MPR are selected from the neighbors of v_i.
- v_i can reach each of its two-hop neighbors via some node in the MPR set.

Finding a minimum MPR is an NPC problem [10]. [3] introduces a Greedy MPR computation method. At first, the MPR set of node v_i is set to be empty.

- **Step 1:** Find the two-hop neighbors of v_i that have only one neighbor (named connecting neighbor) connecting to v_i. Put in the MPR set these connecting neighbors.
- **Step 2:** Repeat adding in the MPR set the neighbor node that covers the largest number of two-hop neighbors of v_i that are not yet covered by the current MPR set.

In the second phase, v_i assigns itself to the CDS if it follows any one of the following two rules.

- **Rule 1:** It has a smaller ID than all its neighbors.
- **Rule 2:** It is in the MPR set of its neighbor with the smallest ID.

To make sure the correctness of the MPR-CDS algorithm, it needs a total order of the nodes which is achieved by using their IDs.

In this paper, we aim at extending the lifetime of the backbone and the entire network. Different from the related work mentioned-above, we propose a new backbone design that considers the power consumption weight of each link between two backbone nodes, and tries to minimize the total transmission cost of the target backbone.

3 Minimum-Weight Backbone (MWB): A New Backbone Infrastructure

In this section, we present a new backbone infrastructure to deploy sensor networks. We first introduce the main idea of the new backbone, and prove that the construction of this backbone is NP-Complete. Then, two heuristic algorithms are provided to construct the new backbone. Finally, we discuss how to balance the energy dissipation among all nodes.

3.1 Introduction to Minimum-Weight Backbone

Given a sensor network, in which all nodes are homogenous and the link between any two nodes is symmetric, denote r the maximum transmission range of two nodes. The network can be represented by a graph G=(V, E), in which V=$\{v_0, v_1, v_2, \cdots, v_n\}$[1] denotes the set of sensor nodes and E=$\{(v_i, v_j)||v_i, v_j| \leq r\}$ represents the set of edges[2] in the network.

[1] The base station is set to be v_0 by default.

[2] In the rest of this paper, we use the terms *link* and *edge* interchangeably.

Each edge is assigned with a weight w_{ij}, which is related to the power consumption of transmission along the edge, and the residual battery power of two nodes at both ends of the link. For simplicity of computation, we use the following model to evaluate the weight.

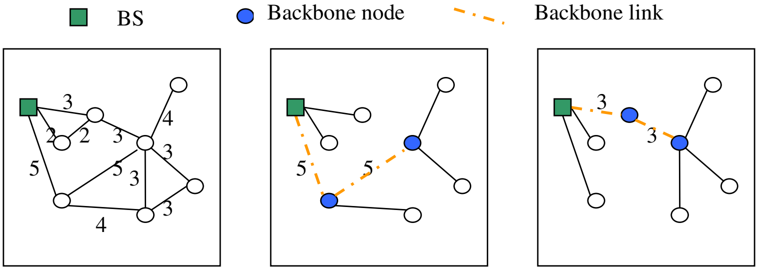
$$w_{ij} = \frac{|v_i, v_j|^k}{P_{residual}(v_i) \times P_{residual}(v_j)} \tag{1}$$

Above, $|v_i, v_j|^k$ denotes the transmission consumption of link (v_i, v_j), where $|v_i, v_j|$ represents the distance between nodes v_i and v_j, and parameter k measures the correlation between power consumption and distance. $P_{residual}(v_i)$ represents the residual battery power of node v_i, which is initially set to be 1. According to formula (1), a smaller w_{ij} implies that 1) the transmission cost between v_i and v_j is relatively lower, or 2) the residual battery power of v_i and v_j is relatively higher. Consequently, selecting pathes of smaller weights for data routing can reduce transmission cost and/or prolong the lifetime of network.

In the sensor network, we can build a spanning tree that is represented by T=(V, E_T) where $E_T \subset E$. We split the spanning tree T into two parts: 1) $G_{Backbone}$=$(V_{Backbone}, E_{Backbone})$, $V_{Backbone}$ and $E_{Backbone}$ are the set of all internal nodes of T and the links among them; 2)G_{leaf}=(V_{leaf}, E_{leaf}), V_{leaf}=V-$V_{Backbone}$ and E_{leaf}=E_T-$E_{Backbone}$. We try to find a spanning tree T_{Min} such that minimizes $\sum_{(v_i,v_j)\in E_{Backbone}} w_{ij}$. We call $G_{Backbone}$ corresponding to T_{Min} the *Minimum-Weight Backbone*, or simply *MWB*. That is to say, a MWB corresponds to a spanning tree in which the sum of weights of all edges between internal nodes is minimized.

Our design of MWB considers the weights of edges and minimizes the total transmission cost within the backbone. Such a mechanism can reduce the power consumption of backbone nodes, and subsequently prolong the lifetime of the backbone as well as the whole network.

Fig. 1 shows an example that compares MWB with CDS. The sensor network represented by graph G and the weights of all edges in the network are shown in Fig. 1(a). The backbone established by CDS that tries to minimize the number of nodes is shown in Fig. 1(b), which contains three nodes with the total value



(a) Original graph G (b) CDS (c) MWB

Fig. 1. Comparison of the backbones established by using CDS and MWB

of edge weights being 10. However, as shown in Fig. 1(c), the established MWB also contains three nodes, and the total value of edge weights is reduced to 6.

3.2 MWB Is NPC

In this subsection, we will prove that it is NP-Complete to find T_{Min}. We transform this optimal MWB problem into a decisive MWB problem. The decisive MWB problem is formalized as below.

Instance: Given graph $G = (V, E)$, in which w_{ij} is the weight of an arbitrary edge in E, and a positive integer W.

Question: Is there a spanning tree for G satisfying $\sum_{(v_i,v_j)\in E_{Backbone}} w_{ij} \leq W$?

Proof. We first show that the decisive MWB problem belongs to NP. Given an instance of the problem, we check whether the condition $\sum_{(v_i,v_j)\in E_{Backbone}} w_{ij} \leq W$ is satisfied. It is obvious that this process can be completed in polynomial time. Thus, MWB problem belongs to NP.

Note that the Maximum Leaf Spanning Tree (MLST) problem [11] has been proved to be NPC. The next step is to ascribe the MLST problem to the MWB problem.

Instance: Given a graph $G = (V, E)$, and a positive integer $K \leq |V|$.

Question: Is there a spanning tree for G in which K or more vertices have degree 1?

We now prove $MLST \leq_p MWB$. Suppose G=(V, E) is an instance of MLST, we construct an instance of MWB as follows. Construct the graph G'=(V, E'), where E'=E, and set the weight of each edge in E' to be 1 and the positive integer W to be ($|V|$-K-1).

If there is a spanning tree T for G in which K or more vertices are leaf nodes, then the number of non-leaf nodes will be $|V|$-K or smaller. That is, there exists a spanning tree T' for G' that satisfies $\sum_{(v_i,v_j)\in E'_{Backbone}} w_{ij} \leq$ ($|V|$-K-1)=W. Conversely, if a spanning tree T' for G' follows the condition $\sum_{(v_i,v_j)\in E'_{Backbone}} w_{ij} \leq W = |V|$-$K$-1, then the number of non-leaf nodes is smaller than or equivalent to $|V|$-K. Thus, there must be a spanning tree T for G in which the number of leaf nodes is larger than or equivalent to K.

Therefore, graph G has a maximum leaf spanning tree in which K or more vertices have degree 1, if and only if graph G' has a minimum-weight backbone that satisfies $\sum_{(v_i,v_j)\in E'_{Backbone}} w_{ij} \leq W$. Since MWB belongs to NP and satisfies $MLST \leq_p MWB$, the MWB problem is NP-Complete. ■

3.3 Algorithms

We provide two heuristic algorithms to build MWB. The first one is a centralized algorithm that is executed at the base station. Before the base station builds the backbone, each sensor node sends its position to the base station. The second algorithm is a distributed one. Each node uses the information of its neighbors and two-hop neighbors to determine whether it is a backbone node.

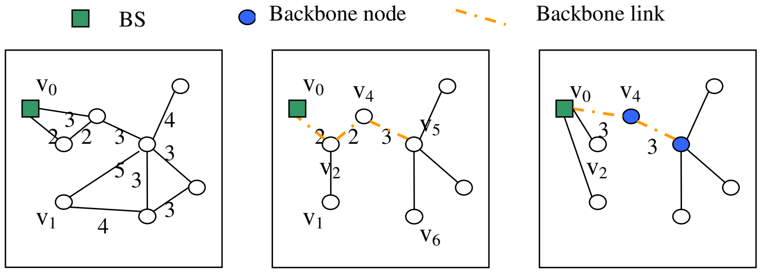
Centralized Algorithm. Our centralized algorithm consists of two phases. On the first phase, we construct a rough backbone by removing as many edges with large weight as possible. The second phase is to refine the rough backbone by pruning redundant backbone nodes. The first phase proceeds as follows.

- **Step 1:** Set $G_{Backbone}$ to be G and G_{Leaf} to be $\emptyset$ initially.
- **Step 2:** Select edge (v_i, v_j) which has the maximum weight in $G_{Backbone}$.
  - If $G_{Backbone}$ is still connected after removing (v_i, v_j), then remove it.
  - If $G_{Backbone}$ is not connected after removing (v_i, v_j), then define G_1 to be the connected subgraph containing v_0 (the base station) and define G_2 to be the set of remaining nodes and edges.
    - ∗ If each node in G_2 and G_{Leaf} can communicate with some node in G_1 directly, then delete (v_i, v_j) and G_2 from $G_{Backbone}$ meanwhile add the nodes in G_2 into G_{Leaf}.
    - ∗ If there exists at least one node that can not connect to any node in G_1, then keep (v_i, v_j) and G_2.
- **Step 3:** Repeat Step 2 till all the edges in $G_{Backbone}$ have been checked once. For each node in G_{Leaf}, set its nearest backbone node as its parent.

Now $G_{Backbone}$ is a rough backbone and G_{Leaf} consists of leaf nodes that are only one hop to the backbone. On the second phase, we do refinement over the rough backbone as follows.

- Traverse $G_{Backbone}$ from the base station using the deep-first strategy. Store the current node v_i, its parent node $P(v_i)$ and its grandparent node $GP(v_i)$. We denote $C_{Backbone}(P(v_i))$ to be the children of $P(v_i)$ that are in $G_{Backbone}$ and $C_{Leaf}(P(v_i))$ to be the set of children that are in G_{Leaf}. If there is an edge between v_i and $GP(v_i)$ in the original graph G and its weight is smaller than the weight sum of edges $(v_i, P(v_i))$ and $(P(v_i), GP(v_i))$, then
  - If $C_{Backbone}(P(v_i))$ contains only v_i and each node in $C_{Leaf}(P(v_i))$ can find another neighbor in $G_{Backbone}$, then move node $P(v_i)$ to G_{Leaf} and remove edges $(v_i, P(v_i))$ and $(v_i, GP(v_i))$ from $G_{Backbone}$. Furthermore, add edge $(v_i, GP(v_i))$ into $G_{Backbone}$ and set $GP(v_i)$ to be the parent of v_i. Finally, reset the parent for each node in $C_{Leaf}(P(v_i))$.
  - Else, $G_{Backbone}$ and G_{leaf} remain unchanged.

The refinement can achieve further reduction of both nodes and total weights. Fig. 2 depicts an example of backbone construction by using the centralized algorithm. The original graph G is shown in Fig. 1(a). The first phase is to find a rough backbone. $G_{Backbone}$ is initialized to be G and the progress starts at the edge (v_0, v_1) which has the maximum weight. According to Step 2 on the first phase, remove (v_0, v_1) since $G_{Backbone}$ is still connected, as shown in Fig. 2(a). After the first phase, we get the rough backbone $G_{Backbone}$ (refer to Fig. 2(b)) that contains four nodes (v_0, v_2, v_4, v_5) and three links ((v_0, v_2), (v_2, v_4), (v_4, v_5)). These three links cannot be removed, otherwise node v_6 cannot directly connect with any node in the rough backbone. Then we refine the rough backbone based on the procedure of the second phase. Node v_4 has a grandparent v_0 with which it can directly communicate. Replace the edges (v_0, v_2) and (v_2, v_4) by edge (v_0, v_4) and remove node v_2, since $w_{04} < w_{02} + w_{24}$ and v_1 can take v_0 as its parent. The final backbone consists of three nodes (v_0, v_4, v_5) and two links ((v_2, v_4), (v_4, v_5)). Fig. 2(c) shows the final spanning tree.



(a) Finding rough backbone (b) Rough backbone (c) Refinement

Fig. 2. Building MWB by the centralized algorithm

Distributed Algorithm. Our distributed algorithm is based on the MPR-CDS [3], which also consists of two phases. The first phase, which is the similar to that of the MPR-CDS method, is to compute MPR set for each node v_i. Step 1 remains unchanged. In step 2, we consider balancing the battery power of nodes.

- **Step 1:** Follows Step 1 of the MPR-CDS algorithm.
- **Step 2:** Repeat adding to the MPR set neighbor node of v_i (say v_j) that has the largest value of $N_{cover}(v_j) * P_{residual}(v_j)$, where $N_{cover}(v_j)$ is the number of two-hop neighbors of v_i that are covered by v_j but not yet covered by the current MPR set.

On the second phase, we set each node a priority using the following model.

$$Priority(v_i) = \frac{degree(v_i) * P_{residual}(v_i)}{AverageWeight(v_i)} \tag{2}$$

$$AverageWeight(v_i) = \frac{\sum_{(v_i,v_j)\in E} w_{ij}}{degree(v_i)} \tag{3}$$

With this model, the priority of a node is proportional to its degree and residual battery power, whereas inversely proportional to the average weight of the edges that link to it. A node determines whether it is to be included in the backbone in terms of the following two rules:

- **New Rule 1:** It has a higher priority than all its neighbors.
- **New Rule 2:** It is in the MPR set of its neighbor with the highest priority.

The correctness of the algorithm above can be proved by following the proof in [3]. Due to the space limitation, we omit the detail here.

3.4 Periodical Backbone Reconstruction

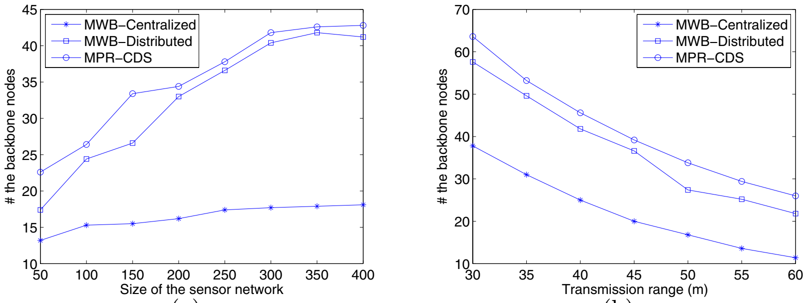
Once the construction of backbone is finished, the leaf nodes will switch to sleeping state till it is awakened by queries. The backbone nodes, however, remain active for data fusion, message relaying and query broadcasting. Although our MWB is to minimize the cost of the whole backbone, the backbone nodes cost more than the leaf nodes. After running for a certain time, the residual battery power of backbone nodes will be lowered. As a result, the backbone should be periodically rebuilt to balance the power consumption of all nodes.

We introduce a threshold θ $(0 \leq \theta < 1)$ to trigger the reconstruction of backbone. Each backbone node records its initial residual battery power $P_{residual}(t)$ when finishing the construction of backbone at t. If the ratio of its current residual battery power to the initial battery power $\frac{P_{residual}(t_{current})}{P_{residual}(t)}$ is smaller than θ, then it asks for establishing a new backbone by sending a reconstruction message to the base station. When using the centralized algorithm for reconstruction, the base station requires the current battery power of each node. Thus, each node needs to spend additional cost in sending this information to base station. In the distributed algorithm, on the other hand, each node should be informed of the current residual power of the neighbors and the two-hop neighbors. Since the reconstruction of backbone is periodical, such additional power consumption will not impact much on entire network lifetime.

4 Performance Evaluation

In this section, we present the simulation results of MWB. The simulations are conducted in an area of $200m \times 200m$, where the base station is placed at the center of the area. We will show the performance comparisons of backbone construction by using our centralized, distributed algorithms and the MPR-CDS algorithm respectively, in terms of the number of the backbone nodes, total transmission cost of the backbone, and the average number of hops from each node to the base station. The performance comparisons are carried out in two scenarios. In the first scenario, the transmission range is set to be $50m$ and the number of sensor nodes varies between 50 and 400. In the second scenario, we let the number of sensor nodes be 150 and transmission range vary between $30m$ and $60m$. Nodes are randomly distributed in both scenarios. From Fig. 3 to Fig. 5, (a) represents the first scenario and (b) represents the second one. We also present the results of the energy dissipation balance among nodes by examining the percentage of alive nodes during the lifetime.

Fig. 3 shows the number of backbone nodes included in the constructed backbone. The backbone constructed by our two heuristic algorithms has fewer nodes than that constructed by the MPR-CDS algorithm. Compared with the MPR-CDS algorithm, our centralized algorithm can significantly reduce the number of backbone nodes by about 50%. In general, the total transmission cost of the backbone is closely related to the number of backbone nodes. Thus, the backbone with low transmission cost always contains few nodes. When the number of

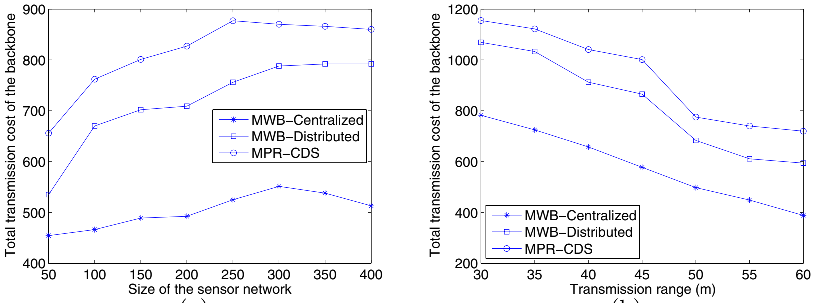


(a) (b)

Fig. 3. Comparison of the number of the backbone nodes

total sensor nodes increases, as shown in Fig. 3(a), the number of the backbone nodes also increases for all algorithms. However, the increasing rate slows down when the number of sensor nodes in the network reaches more than 300, since in our experimental setting the deploying area is fixed, and the increasing of the number of senor nodes will make the number of backbone nodes tends to a saturation status. In Fig. 3(b), the number of the backbone nodes decreases with the increasing of the transmission range because the increasing of transmission range makes each node be able to communicate with more nodes.

Fig. 4 presents the comparison results of the total transmission cost of the backbone. Both Fig. 4(a) and Fig. 4(b) show that the total transmission cost of the backbone constructed by using our algorithms is less than that by using the MPR-CDS algorithm. By the centralized algorithm, the backbone cost can be up to 40% less than that by the MPR-CDS algorithm. Fig. 4(a) shows that when the number of nodes in the network increases, the cost of backbone increases at first and then slightly decreases. The underlying reason is like this: as the number of nodes increases (fixed the deploying area), the number of the backbone nodes keeps almost the same, but the links of smaller weight (for the centralized algorithm) or the nodes of higher priority (for the distributed algorithm) will be chosen for backbone. Fig. 4(b) shows that the cost of backbone decreases when the transmission range increases because fewer backbone nodes will be selected.



(a) (b)

Fig. 4. Comparison of the total transmission cost of the backbone

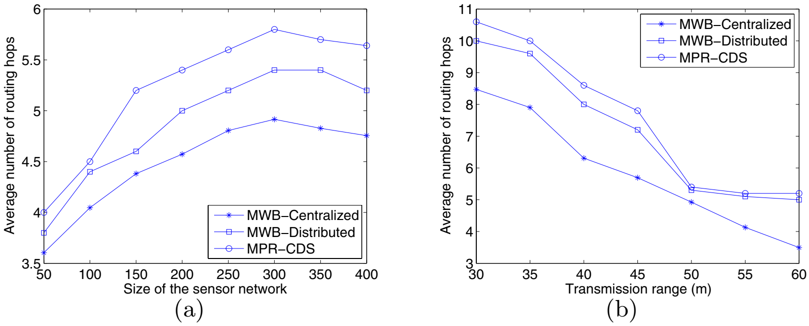


Fig. 5. Comparison of the average hops from nodes to BS

Fig. 5 presents the average number of routing hops from each node in the network to the base station. The trend is quite similar to that of the transmission cost. In the backbones constructed by using our methods, the nodes can reach the base station with fewer number of hops on average than in the backbone constructed by the MPR-CDS algorithm, which also means less latency. When the number of nodes increases, the average number of routing hops increases at first and then decreases a little as shown in Fig. 5(a). This is because the number of routing hops is closely related to the number of backbone nodes and the distance between the nodes and the base station. Fig. 5(b) shows when the transmission range increases, the number of routing hops decreases due to the decreasing of the number of backbone nodes.

We then show the results of the network lifetime and power consumption balance of all nodes. We randomly distribute 150 nodes in an area of $200m*200m$ and set the transmission range to be $50m$. Fig. 6 shows that the sensor network built and maintained by the centralized algorithm has a longer lifetime due to fewer number of the backbone nodes and less transmission cost of the backbone. It also illustrates that both of our methods can balance well the residual battery power among nodes. During most period of the network lifetime, the percentage of the alive nodes remains to be 100% when using the centralized algorithm, and more than 95% when using the distributed algorithm.

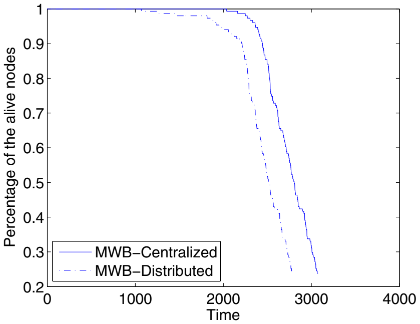


Fig. 6. Network lifetime

5 Conclusion

In this paper, we present a new backbone (MWB) to deploy the sensor networks. Our design is to minimize total transmission cost of the backbone, and thus reduce its power consumption. However, the construction of MWB is proved to be NPC. We propose two heuristic algorithms - one centralized and one distributed - to build MWB. The centralized algorithm is to prune as many large-weight edge as possible, and then refine the backbone by combining some edge-pairs. The distributed algorithm, based on the MPR-CDS algorithm, chooses the backbone nodes by a carefully-designed priority metric. These two algorithms have been shown by simulations to outperform the MPR-CDS algorithm in terms of the size and cost of the backbone and routing latency. We also deal with the issue of periodical backbone rebuilding. Simulation results show that our methods can balance the battery power of nodes well.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “a survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 – 114, 2002.
2. S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, pp. 374 – 387, Apr. 1998.
3. C. Adjih, P. Jacquet, and L. Viennot, “Computing connected dominated sets with multipoint relays,” *Ad Hoc and Sensor Wireless Networks*, vol. 1, january 2005.
4. S. Hedetniemi and A. Liestman, “A survey of gossiping and broadcasting in communication networks,” *Networks*, vol. 18, no. 4, pp. 319 – 349, 1988.
5. C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” *in Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.
6. W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, 2002.
7. A. Manjeshwar and D. Agrawal, “Teen: A routing protocol for enhanced efficiency in wireless sensor networks,” *Proc. First Intl Workshop Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2001.
8. X. Cheng, M. Ding, and D. Chen, “An approximation algorithm for connected dominating set in ad hoc networks,” *in Proc. of International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks*, 2004.
9. X. Chen and J. Shen, “Reducing connected dominating set size with multipoint relays in ad hoc wireless networks,” *in Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks*, 2004.
10. A. Laouiti, A. Qayyum, and L. Viennot., “An efficient technique for flooding in mobile wireless networks,” *In 35th Annual Hawaii International Conference on System Sciences (HICSS2001).*, 2001.
11. M. R. Garey and D. S. Johnson, “Computer and intractability: A guide to the theory of np-completeness,” *W. H. Freeman, ISBN 0-7167-1044-7*, 1979.

An Energy-Balanced Strategy for Clustering Protocols in Wireless Sensor Networks

Feng Zhao, Feng Xi, and Zhong Liu

Department of Electronic Engineering
Nanjing University of Science and Technology
Nanjing, Jiangsu 210094
The People's Republic of China
eezliu@mail.njust.edu.cn

Abstract. Energy is one of the most important factors in the design of routing protocols in wireless sensor networks. This paper first analyzes the energy consumption in typical clustering protocols and finds that the consumption is not evenly distributed among nodes. Thus some nodes die quickly with the reduction of the network lifetime. Then a new energy-balanced strategy is introduced in clustering protocols. The strategy assigns the head communication load to base station by detecting the energy consumption in the cluster heads. The evenly distributed energy among the nodes is realized by controlling the head consumption. Simulation results show that the lifetime of the network is significantly prolonged with the new energy-balanced strategy.

Keywords: Sensor networks, Clustering protocols, Energy-balanced strategy, Network lifetime.

1 Introduction

Recent advances of embedded systems, wireless communications and micro-electro-mechanical systems have motivated the development of tiny sensor nodes consisting of sensing, data processing and communicating components. These tiny sensors are constructed in a wireless sensor network (WSN) for a wide range of data-gathering applications in military and national security, environmental monitoring, and other fields [1~2].

An important aspect of WSNs is that the nodes are often unattended and their energy cannot be replenished. Therefore, it is essential for these sensor nodes to conserve energy to increase the WSN lifetime. Since wireless communications consume significant amounts of battery power, the nodes should spend as little energy as possible for receiving and transmitting data. Communication protocols are desired to maximize nodes' lifetime, reduce bandwidth consumption by using local collaboration among the nodes and tolerate node failures.

Different communication protocols have been developed, for example, flat-based routing [9~10], hierarchical-based routing [4~8, 13~14] and location-based routing

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 612 – 621, 2006.

[11~12]. A recent review can be found in [3]. In this paper, we pay attention on the hierarchical-based routing protocols, also called clustering protocols.

In the clustering protocols, the clustering divides the network into disjoint subsets, wherein a sensor (cluster head) from each subset is elected to represent that cluster. In each round of data-gathering, all nodes in a cluster transmit their data to the cluster head and the heads communicate the collected data with the base station (BS). The cluster head role is usually periodically rotated among the nodes to ensure the nodes consume energy more uniformly. Almost all clustering protocols aim to make the energy evenly distributed among all nodes in order to prolong the network lifetime. The protocols in [13~15] achieve this through the regular distribution of the cluster heads with almost same cluster size. The protocol in [4] focuses on the energy every cluster consumes evenly, so the energy is further balanced among the nodes and the lifetime is prolonged. However, to realize the evenly distributed energy among the nodes, we should focus on the energy every node consumes. If each node consumes the same energy, the energy can be completely balanced and thus the network lifetime is prolonged.

In this paper, we propose an energy-balanced strategy, which focuses on the energy every node consumes. The strategy first detects the energy consumption in the cluster heads, and then assigns the head communication to the BS. The evenly distributed energy among the nodes is realized by controlling the head consumption. Simulation results show that the lifetime of the network is significantly prolonged with the new energy-balanced strategy.

The remainder of the paper is organized as follows. In Section 2, we present our network and radio models. The existing clustering protocols are briefly analyzed in Section 3. In Section 4 we present new energy-balanced strategy in details. Section 5 is the energy consumption analysis. Simulation results are shown in Section 6. This paper is concluded in Section 7.

2 Network and Radio Models

In this section, we briefly introduce the network and radio models used in this paper. See [6] for details.

For our network, it is assumed that

- ♦ A fixed base station is located far away from sensor nodes.
- ♦ Sensor nodes are homogeneous and are equipped with a uniform initial energy.
- ♦ Each node senses the environment at a fixed rate and always has data to transmit to the base station.
- ♦ Every sensor node is immobile and is allocated an exclusive ID number.
- ♦ Sensor nodes can vary their transmitted power.

For the radio hardware energy dissipation, it is assumed that the transmitter dissipates energy $E_{Tx-elec}(l)$ to run the radio electronics and $E_{Tx-amp}(l,d)$ to run the power amplifier, and the receiver dissipates energy $E_{Rx-elec}(l)$ to run the electronics.

Depending on the distance between the transmitter and receiver, both the free space (d^2 power loss) and the multipath fading (d^4 power loss) channel models are used in this paper. Power control is used to invert this loss by appropriately setting the power amplifier, that is, if the distance is less than a threshold d_0, the free space model is used; otherwise the multipath model is used. Thus, to transmit a l-bit message in distance d, the radio expends

$$\begin{aligned} E_{Tx}(l,d) &= E_{Tx-elec}(l) + E_{Tx-amp}(l,d) \\ &= \begin{cases} lE_{elec} + l\varepsilon_{fs} d^2, d < d_0 \\ lE_{elec} + l\varepsilon_{mp} d^4, d > d_0 \end{cases} \end{aligned} \quad (1)$$

and to receive this message, the radio expends

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \ . \quad (2)$$

where E_{elec} is the electronics energy, ε_{fs} and ε_{amp} are the coefficients of the amplifier. The energy spent for aggregating n l-bit messages in a cluster is

$$E_{DA} = nlE_{da} \ . \quad (3)$$

where E_{da} is the energy spent for aggregating l-bit data.

3 Analyses of Clustering Protocols

A typical application of WSN is gathering of sensed data at BS. In each round, the data from all nodes need to be collected and transmitted to BS. Since the BS is located far away, high energy is required to transmit data to BS from sensor nodes. Clustering protocols are to alleviate this problem by allowing only partial nodes (heads) communicate with BS. We now analyze these clustering protocols from the point of view that the cluster heads communicate with BS.

The simplest way many clustering protocols adopt is that all the cluster heads directly (one-hop) transmit the data to BS. The LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol [5] is one of them, where a small number of clusters are formed in a self-organized manner. A head node in each cluster collects and fuses data from the nodes in its cluster and transmits the result to the BS.

To reduce the energy consumption in LEACH, multi-hop communication among the cluster heads [4] is proposed. In this protocol, the less number of nodes is needed to communicate with BS and hence the less energy is consumed in comparison with LEACH-like protocols. But in multi-hop communication, the nodes closest to BS are burdened with a heavy relay traffic load and often die first.

Another efficient way is that every cluster head takes turns to transmit data to BS. PEGASIS (Power Efficient Gathering in Sensor Information Systems) [7] is one adopting this idea. A data chain is first formulated from the farthest nodes to BS, a leader node is randomly selected, then the data gathered from the nodes are forwarded to the leader node along the chain and finally the leader transmits the collected data to

BS. Because in each round only one node transmits data to BS, PEGASIS may perform better than the protocols aforementioned. Enlightened from the idea of PEGASIS, we can improve LEACH by constructing a chain among the elected cluster heads and performing data transmission like PEGASIS. We name this improved LEACH as OH-LEACH protocol. The simulation results in Section V show that the network lifetime is prolonged in comparison with traditional LEACH. In this paper, we call this kind of protocols as one-head clustering protocol (OHCP).

However, in OHCPs there exists an inherent problem that energy consumption is not evenly distributed among the head nodes. The main reason is that the distance of the head node to BS is different from each other. Simulations in Section V show the energy consumption and network lifetime of PEGASIS. It is found that the energy consumption is different from node to node and some nodes die quickly. This not only affects the performance of the whole network, but also introduces additional cost for reconstructing the network topology.

In the next section, we develop a new energy-balanced strategy to solve this problem in OHCPs. By making every leader node consumes the same fixed energy, the evenly distributed energy among the nodes can be achieved and the network lifetime can be further prolonged.

4 A New Energy-Balanced Strategy

As described in the last section, the OHCPs can be divided into three stages: cluster construction, head topology formation and data transmission. In the first stage, the clusters are constructed and a head for each cluster is selected to fuse the data in the cluster. In the second stage, a head topology is formulated and a leader (head of heads) is created. In the third stage, all data from the heads will be transmitted to the leader. The leader transmits all data from heads to BS. PEGASIS is a special case of OHCPs, in which each node acts as a cluster. During the operation process, the second and third stages are dynamic, in which different heads will be periodically selected as leaders. The time that OHCPs take for all head nodes to act as leaders is called an *operation cycle* in this paper.

This section gives the details of the energy-balanced strategy used in OHCPs. The basic idea of the proposed energy-balanced strategy is to make every head node consume the same energy in an operation cycle. The strategy first detects the energy consumption in the cluster heads, and then assigns the head communication load. The evenly distributed energy among the nodes is realized by controlling the head consumption. The detection of energy consumption is realized by *working count*, working times in which a head can communicate with BS. The more working counts a head node works for, the more opportunities the head will be assigned to be a leader.

In one operation cycle of OHCPs, a head node will work either as head or leader. The energy that the head node consumes will be the summation of leader consumption and non-leader consumption. The leader consumption is mainly due to the data exchanges with BS while the non-leader consumption is due to the data exchanges with the neighbor heads. Suppose there are NH head nodes in one operation cycle. Denote the leader consumption and non-leader consumption as $A(j)$ and $B(j)$

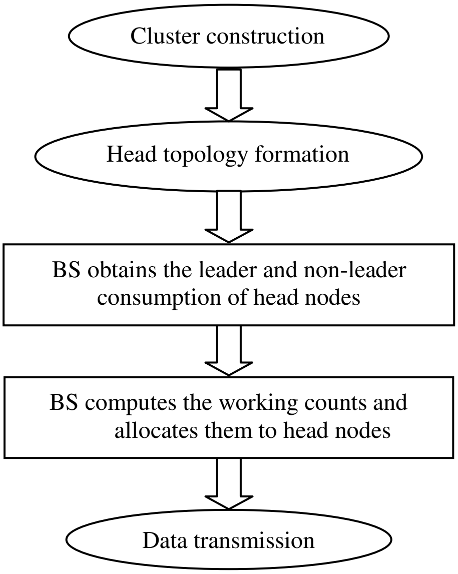

Cluster construction
Head topology formation
BS obtains the leader and non-leader consumption of head nodes
BS computes the working counts and allocates them to head nodes
Data transmission

Fig. 1. Flowchart of the protocol that the energy-balanced strategy is introduced

($j = 1, 2, \cdots, NH$) for one working count. Then the energy consumption $E(j)$ for each head node in one operation cycle can be expressed as

$$A(j)C(j) + \sum_{i=1, i\neq j}^{NH} C(i)B(j) = E(j) \; (j = 1, 2, \cdots, NH) \tag{4}$$

where $C(j)$ is the working count for the head node j. If $E(j)$ is the initial energy that each node has, $C(j)$ will be the actual working counts in which the head node can work. In our energy-balanced strategy, we force each head node consume the same (or almost) energy. To do this, we set a reference energy E_0 and let $E(1) = \cdots = E(NH) = E_0$. To ensure the network operation, E_0 should be larger than the energy that each head node consumes in one operation cycle, but less than the initial energy that each head node has. With $E(1) = \cdots = E(NH) = E_0$, we can solve Eq.(4) for $C(j)$ for each head node. In this way, we assign the possible working times for each head node in one operation cycle. In general, $C(j)$ is a non-integer number and can be set as an integer not larger than it.

The new energy-balanced strategy working with OHCPs can be described as follows. In one operation cycle, each head node will detect its energy and transmit it to BS. The BS will obtain the leader consumption. At the same time, the BS will also obtain the energy consumption for each head as non-leader. Then using Eq.(4), the BS will compute the working counts and allocate them to the corresponding head node according to the node identification (ID). One of the heads will be selected as leader node. After working in its working counts, the node will stop as a leader and move to the next head node in the head topology. After all head nodes have acted as leader nodes, the cluster will be reconstructed and the next operation cycle will repeat the operation.

Figure 1 shows a flowchart of the OHCP with the new energy-balanced strategy. The ellipses blocks describe the operation of the OHCPs, while the squares blocks detail the energy-balanced strategy. It is obvious that the introduction of the new strategy will not disturb the operation of the clustering protocol.

The new strategy only considers the evenly distributed energy in heads. The node consumption in a cluster was not included in the Eq.(4). This idea fits in PEAGSIS well. For other kinds of OHCPs, the node consumption in a cluster is much less than the head node. With the network operation, the node consumption will be approximately evenly distributed. The simulation results in Section 6 confirm this.

5 Energy Consumption Analyses

This section gives the energy consumption analyses on OHCPs with and without new strategy. For convenience, the new protocol is defined as Energy-Balanced OHCP and denoted as EBOHCP. We focus on the energy consumption of the whole network when every node has acted as head node once. The non-leader energy consumption is omitted for simplicity.

Suppose that the WSN has n nodes. Denote as T the total working counts (summation of all head working counts) after each node acting as head node. For comparison, it is assumed to be same for both OHCP and EBOHCP. The average working count for every head node is defined as T/n.

Lemma: If the same amount of data messages is received at BS, after every node acting as head node, the total node consumption by EBOHCP is less than that by OHCP.

Proof. The total node consumption by EBOHCP and OHCP is given respectively by

$$E_{total1} = nE_0 \ . \tag{5}$$

$$E_{total2} = \sum_{i=1}^{n} (\frac{T}{n} A(i) + (T - \frac{T}{n}) B(i)) \ . \tag{6}$$

Omitting $B(i)C(i)$ and letting $B(i) = kA(i)$, we have from (4)

$$A(i) = \frac{E_0}{C(i) + kT} \ . \tag{7}$$

The non-leader consumption $\frac{T}{n} B(i)$ is neglectable in comparison with the leader consumption. Then substituting (7) into (6), we have

$$E_{total2} = \sum_{i=1}^{n} \frac{E_0}{C(i) + kT} (\frac{T}{n} + kT) \ . \tag{8}$$

Since $(\frac{T}{n}+kT)=\frac{1}{n}\sum_{i=1}^{n}(C(i)+kT)$, we can get $E_{total1} < E_{total2}$ according to Cauchy Theorem.

This lemma shows for the same amount of data messages that the network consumption by the EBOHCP is less than that by the OHCP and hence the network life is prolonged.

6 Simulation Results and Evaluation

In this section, the new energy-balanced strategy is combined with PEGASIS and OH-LEACH and the simulation results are given to show the performance of the

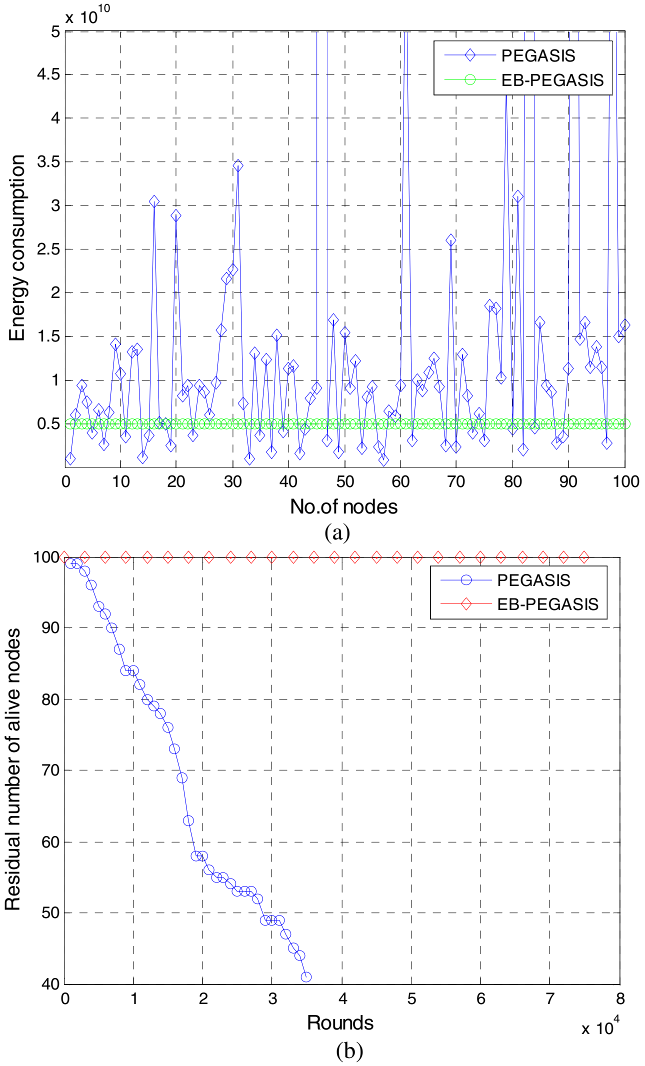
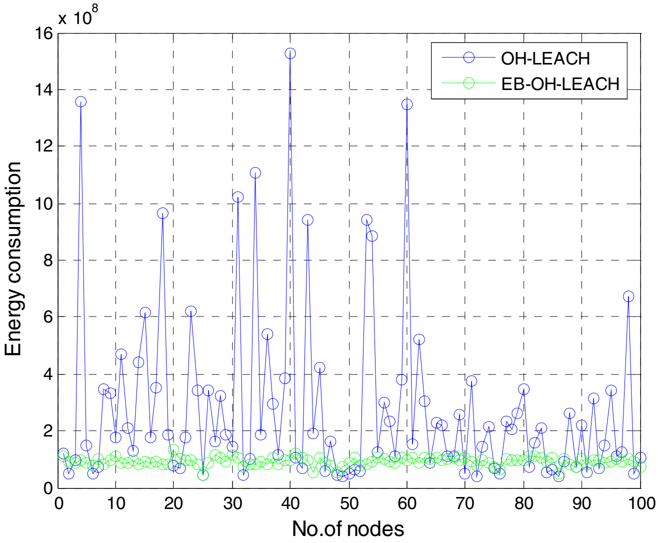
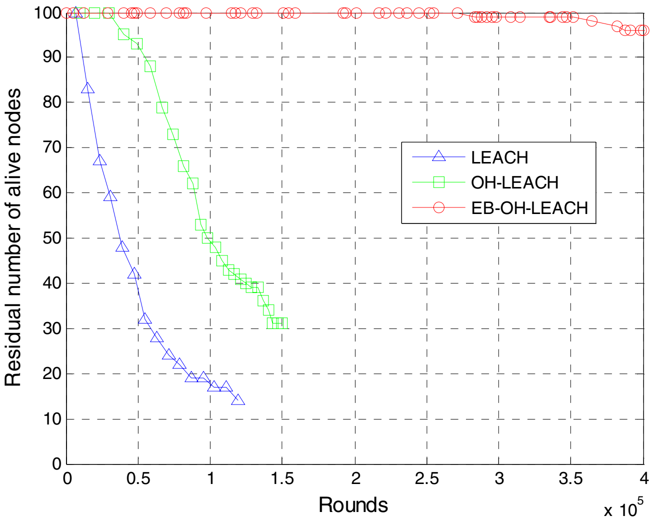


Fig. 2. Energy consumption (a) and Network lifetime (b) by PEGASIS and EB-PEGASIS



(a)



(b)

Fig. 3. Energy consumption (a) and Network lifetime (b) by OH-LEACH and EB-OH-LEACH

improved protocols. The two new protocols are named as EB-PEGASIS and EB-OH-LEACH, respectively.

We consider a 100-node network where nodes are randomly distributed between ($x = 0$, $y = 0$) and ($x = 500$, $y = 500$) with the BS at location ($x = 250$, $y = 175$). The bandwidth of the channel is set to be1 Mb / s , each data message is 100 bytes long. The communication energy parameters are set as $E_{elec} = 50nJ/bit$, $\varepsilon_{fs} =$ 10 nJ/bit $\varepsilon_{amp} = 0.0000013$ $pJ/bit/m^4$ and $E_{da} = 5nJ/bit/signal$. The initial energy of the node is set as $10J$.

The network lifetime is used as the performance metric. The definition of the lifetime varies with different applications, for example, if a high QoS is needed, the network will be considered useless when only one node die. In our simulations, we use the amount of the residual working nodes to reflect the lifetime of the network.

Figure 2 shows the network performance of PEGASIS protocols without and with the new energy-balanced strategy. Since PEGASIS fits the energy-balanced strategy perfectly, the energy every node consumes is the same in EB-PEGASIS, while the energy consumption is different from each other, as shown in Fig.2 (a). Fig.2 (b) gives the lifetime of the network. The performance of EB-PEGASIS is significantly superior to that of PEGASIS.

The same simulations are shown in Fig.3 for OH-LEACH protocols without and with the new energy-balanced strategy. Because there is a small approximation in EB-OH-LEACH, we can see that there is little fluctuation in the node consumption. But compared with OH-LEACH, EB-OH-LEACH achieves approximately the same energy consumption. The network lifetime in Fig.3 (b) shows that the new energy-balanced strategy can significantly prolong the lifetime.

7 Conclusion and Future Work

In this paper, we mainly discuss the clustering protocols in the network layer. By analyzing and comparing the typical protocols, we find that the energy consumption is not evenly distributed in OHCPs and thus some nodes will die quickly with the results that network is paralyzed. To prolong the network lifetime, we propose a new energy-balanced strategy to make the energy consumption evenly distribute among the head nodes after the head topology is formed. Simulations show the efficiency of the proposed strategy and the network lifetime is significantly prolonged.

The proposed energy-balanced strategy is depending on the network topology and is not tolerant to the network fault. In such case, we may adjust the reference energy to make the cluster heads work in less counts to alleviate this problem.

Future works include the extension of the proposed strategy to other clustering protocols.

Acknowledgments

This work was supported in part by the Chinese Natural Science Foundation (No. 60472059).

References

1. Estrin, D., Girod, L., Pottie, G, Srivastava, M.: Instrumenting the World with Wireless Sensor Networks. Proc. IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing, May 2001, 2033-2036.

2. Pottie, G. J., Kaiser, W. J.: Wireless Integrated Network Sensors. Communications of the ACM, 43 (2000), 51–58.
3. Jamal N. Al-Karaki, Ahmed Kamal, E.: Routing Techniques in Wireless Sensor Networks: A Survey. IEEE Wireless communications, 11 (2004), 1536-1284.
4. Soro. S, Heinzelman, W. B.: Prolonging the Lifetime of Wireless Sensor Networks via Unequal Clustering. Proc. IEEE Inter. Conf. on Parallel and Distributed Processing Symposium, April 2005, 4-8.
5. Heinzelman, W. R., Chandrakasan, A. P., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Proc. 33rd Hawaii Inter. Conf. on System Science, Jan. 2000, 10-20.
6. Heinzelman,W. R..,Chandrakasan,A. Balakrishnan,H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, 1(2002), 660–670.
7. Lindsey, S., Raghavendra, C., Sivalingam, K. M.: Data Gathering Algorithms in Sensor Networks using Energy Metrics. IEEE Transaction on Parallel and Distribute Systems, 13(2002), 924–35.
8. Culpepper, J., Dung, L., Mon, M.: Hybrid Indirect Transmissions (HIT) for Data Gathering in Wireless Micro Sensor Networks with Biomedical Applications. Proc. IEEE Computer Communications Workshop, October 2003, 124-133.
9. Kulik, J., Heinzelman, W. R., Balakrishnan, H.: Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. Wireless Networks, 8 (2002), 169–85.
10. Intanagonwiwat, C., Govindan, R., Estrin. D.: Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks. Proc. ACM Mobi-Com., Mar. 2000, 56–67.
11. Xu, Y., Heidemann, J., Estrin, D.: Geography Informed Energy Conservation for Ad-hoc Routing. Proc. 7th Annual ACM/IEEE Inter. Conf. on Mobile Computer and Net, 2001, 70–84.
12. Yu, Y., Estrin, D., Govindan, R.: Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. UCLA Comp. Sci. Dept. Tech. Rep., UCLA-CSD TR-010023, May 2001, 1—11
13. Yonis, O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. IEEE Transaction on Mobile Computing, 34(2004), 366—379.
14. Muruganathan, S.D., Ma, F.C.: A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. IEEE Radio Communications, 43(2005), 8-13.
15. Kim, J, Kim, S.: An Adaptive Cluster Radius Configuration Scheme for Topology Control in Wireless Sensor Networks. Proc. IEEE Vehicular Technology Conference, June 2005, 2546-2550.

QoS Topology Control with Minimal Total Energy Cost in Ad Hoc Wireless Networks*

Hai Liu[1], Deying Li[2], and Xiaohua Jia[1]

[1] Dept. of Computer Science, City University of Hong Kong
[2] School of Information, Renmin University of China, Beijing, China
{liuhai, jia}@cs.cityu.edu.hk

Abstract. This paper discusses the energy efficient QoS topology control problem in ad hoc wireless networks. Given a set of nodes in a plane, end-to-end traffic demands and delay bounds between node pairs, the problem is to find a network topology that can meet the QoS requirements and the total transmission power of nodes is minimized. We consider two cases of the problem: 1) the traffic demands are not splittable, and 2) the traffic demands are splittable. The first case is formulated as an integer linear programming problem. The latter case is formulated as a mixed integer linear programming problem. A greedy algorithm and an approximation algorithm with ratio n are proposed to solve the problem, where n is the number of nodes. Extensive simulations are conducted to evaluate the performance of proposed algorithms.

1 Introduction

An ad hoc wireless network is a special type of wireless networks that does not have a wired infrastructure to support communication among the wireless nodes. In multi-hop ad hoc networks, communication between two nodes that are not direct neighbors requires the relay of messages by the intermediate nodes between them. Each node acts as a router, as well as a communication end-point. There are many modern network applications that require QoS provisions in ad hoc networks, such as transmission of multimedia data, real-time collaborative work, and interactive distributed applications.

Extensive research has been done on QoS provisions in ad hoc networks, such as QoS routing or admission control [1-4]. Most of the existing works deal with resource allocation (e.g., scheduling or buffering) or routing for QoS requests. However, the construction of a network topology that can overall meet QoS requirements has not been studied in the literature. In multi-hop ad hoc networks, on-line QoS provisions, such as end-to-end bandwidth and delay, are highly dependent on the network topology. Without a proper configuration of the topology, some nodes in the network could be easily over-loaded and it might be impossible to find a QoS route during the operation of the network.

The topology of an ad hoc network can be controlled by some "controllable" parameters such as transmitting power and antenna directions. Topology control is

* Work partially supported by the Research Project CityU 1149/04E.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 622 – 637, 2006.

to allow each node in the network to adjust its transmitting power (i.e., to determine its neighbors) so that a *good* network topology can be formed. An issue often associated with topology control is energy management. In ad hoc wireless networks, each node is usually powered by a battery equipped with it. Since the capacity of battery power is very much limited, energy consumption is a major concern in topology control. To increase the longevity of such networks, an important requirement of topology control algorithms is to achieve the desired topology by using minimal energy consumption.

In this paper, we study the energy efficient QoS topology control problem. Given a set of wireless nodes in a plane and QoS requirements between node pairs, we assume the location information and QoS requirements of other nodes can be obtained, either by some central nodes or via the exchange of location information. Our task is to find a network topology that can meet the QoS requirements and the total transmission power of nodes is minimized. The QoS requirements of our concern are traffic demands and maximal delay bounds (in terms of hop counts) between end-nodes at the application level. With the network configured in such a topology, as many as possible QoS calls can be admitted at run-time and the network life time can be prolonged.

We assume that the network topology is controlled by the transmitting power at each node and the topology directly affects the QoS provisions of the network. If the topology is too dense (*i.e.*, nodes have more neighbors), there would be more choices for routing, but the power consumption of the system would be high. On the other hand, if the topology is too loose (*i.e.*, with less edges), there would be less choices for routing (hence, some nodes could be over-loaded) and the average hop-count between end-nodes would be high. Our goal is to find a balanced topology that can meet end-users QoS requirements and has minimal total transmission power. Note that the receiving power and the power needed to keep the electric circuits on are fixed parts in actual energy consumption. And we assume that transmission power is large enough such that receiving power is negligible [27]. We only focus on the transmission power of all nodes in the paper.

2 Related Work

There are some research works that have already been done on topology control for ad hoc wireless networks. The earlier works of topology control can be found in [5, 6]. In [6], Hou *et al.* studied the relationship between transmission range and throughput. An analytic model was developed to allow each node to adjust its transmitting power to reduce interference and hence achieve high throughput. In [5], a distributed algorithm was developed for each node to adjust its transmitting power to construct a reliable high-throughput topology. Minimizing energy consumption was not a concern in both works.

Recently, energy efficient topology control becomes an important topic in ad hoc wireless networks. Most of the works have been focused on the construction and maintenance of a network topology with good (or required) connectivity by achieving an objective on energy consumption. Lloyd *et al.* gave a good summary of the works in this type in [19]. They use a 3-tuple <M, P, O> to represent topology control problems, where “M” represents the graph model (either directed or undirected), “P” represents the desired graph property (e.g., 1-connected or 2-connected), and “O” represents the minimization objective (e.g., MinMax power or Min total power). The NP-completeness of this kind of problems has been analyzed and several algorithms have been proposed. In [7], two centralized optimal algorithms were proposed for creating connected and bi-connected static networks with the objective of minimizing the maximal transmitting power for each node. Additionally, two distributed heuristics, LINT (local information no topology) and LILT (local information link-state topology), were proposed for adaptively adjusting node transmitting power to maintain a connected topology in response to topological changes. But, neither LINT nor LILT can guarantee the connectivity of the network. Li *et al.* proposed in [9] a minimum spanning tree based topology control algorithm that achieves network connectivity with minimal power consumption. A cone-based distributed topology control method was developed in [8]. Basically, each node gradually increases its transmitting power until it finds a neighbor node in every direction (cone). As the result, the global connectivity is guaranteed with minimal power for each node. Huang *et al.* extended this work in [8] to the case of using directional antennas [10]. Marsan *et al.* presented a method in [11] to optimize the topology of Bluetooth, which aims at minimizing the maximal traffic load of nodes (thus minimizing the maximal power consumption of nodes). Using topology control to meet overall QoS requirements was first proposed in [18]. Work in [18] is to minimize the maximal transmitting power of nodes. Although minimizing the maximum transmitting power can balance workload on networks, energy consumption using this objective may be greater than that using minimizing total transmission power objective. This increase of energy consumption could result in reducing the network lifetime in long term. Different from the work in [18] that is to minimize the maximal transmitting power of nodes, this paper discusses QoS topology control problem that is to minimize total transmission power.

There are a lot more works on energy efficient communication in ad hoc wireless networks, such as in [12, 13]. Singh *et al.* studied five different metrics of energy efficient routing in [13], such as minimizing energy consumed per packet, minimizing variance in node power levels, minimizing cost per packet, and so on. Kawadia *et al.* proposed a clustering method for routing in non-homogeneous networks [14], where nodes are distributed in clusters. The goal is to choose the transmit power level, so that low power levels can be used for intra-cluster communication and high power levels for inter-clusters. In [15], Wieselthier *et al.* studied the problem of adjusting the energy power of each node, such that the total energy cost of a broadcast/multicast tree is minimized. Some heuristic algorithms were proposed, namely the Broadcast Incremental Power (BIP), Multicast Incremental Power (MIP) algorithms, MST (minimum spanning tree), and SPT (shortest-path tree). The proposed algorithms were evaluated through simulations. Wan *et al.* in [16] presented a quantitative analysis of performances of these three heuristics.

In this paper, we address the problem of topology control that can meet the QoS requirements and the total transmission power of nodes in the system is minimized.

3 System Model and Problem Specification

We adopt the widely used transmitting power model for radio networks: $p_{ij} = (d_{i,j})^{\alpha}$, where p_{ij} is the transmitting power needed for node i to reach node j, $d_{i,j}$ is the distance between i and j, and α is a parameter typically taking a value between 2 and 4.

The network is modeled by $G = (V, E)$, where V is the set of n nodes and E a set of directed edges. Let $p(i)$ denote the transmitting power of node i. We assume that each node can adjust its power level, but not beyond some maximal power P. That is, $0 \le p(i) \le P$ for $0 \le i \le n$. The connectivity between two nodes depends on their transmitting power. An edge $(i, j) \in E$ iff $p(i) \ge (d_{i,j})^{\alpha}$. Let $\lambda_{s,d}$ and $\Delta_{s,d}$ denote the traffic demand and the maximally allowed hop-count for node pair (s, d), respectively. Let $P_{total} = \sum_{i=1}^{n} p(i)$.

The QoS topology control problem of our concern is: given a node set V with their locations, $\lambda_{s,d}$ and $\Delta_{s,d}$ for node pair (s, d), where $s,d \in V$, our task is to find transmitting power $p(i)$ for each node i, $0 \le i \le n$, such that all the traffic demands can be routed within the hop-count bound. Our objective is to minimize the total transmission power of nodes P_{total}.

If $\lambda_{s,d}$ are selected to require the network should be strongly connected, and $\Delta_{s,d}$ are large enough, our problem is the topology control problem which was defined in [25]. It had been proved to be NP-hard. Thus, our problem is NP-hard. In this paper, we formulate the problem into ILP (MILP) that can be computed by some tools, and we further propose a heuristic algorithm and an approximation algorithm. We consider two cases: 1) end-to-end traffic demands are not splittable, i.e., $\lambda_{s,d}$ for node pair (s, d) must be routed on the same path from s to d; 2) end-to-end traffic demands are splittable, i.e., $\lambda_{s,d}$ can be routed on several different paths from s to d. We assume each node can transmit signals to its neighbors in a conflict free fashion. Thus, we do not consider signal interference in this paper. There are many MAC (medium access control) layer protocols [20, 21] or code assignment protocols [17, 22] that have been proposed to avoid (or reduce) signal interference in radio transmissions.

4 Topology Control with Traffics Non-splittable

In this section, we consider the case that traffic demands between node pairs are not splittable. That is, all traffic between a node pair should be routed on the same path. Our task is to determine $p(i)$ for each node i, such that the topology can meet the QoS requirements. Our objective is to minimize the total transmission power of nodes in the network.

4.1 Formulation

Given:
V, set of n nodes and their locations.
B, the bandwidth capacity of each node.
P, maximally allowed transmitting power of each node.
$\lambda_{s,d}$, traffic demands for each node pair (s, d).
$\Delta_{s,d}$, maximally allowed hop-count for node pair (s, d).
Variables:
$x_{i,j}$, boolean variables, $x_{i,j}$ =1 if there is a link from node i to node j; otherwise, $x_{i,j}$ = 0.
$x_{i,j}^{s,d}$, boolean variables, $x_{i,j}^{s,d}$ =1 if the route from s to d goes through the link (i, j); otherwise $x_{i,j}^{s,d} = 0$.
P_{total}, the total transmission power of nodes.
Optimize:
- Minimize the total transmission power of nodes.

$$Min\ P_{total} = \sum_{i=1}^{n} p(i) \tag{1}$$

Constraints:
Topology constraints:

$$x_{i,j} \le x_{i,j'} \qquad \forall i, j, j' \in V, d_{ij'} \le d_{ij} \tag{2}$$

Transmitting power constraint:

$$P \ge p(i) \ge (d_{i,j})^{\alpha} x_{i,j} \qquad \forall\ i,\ j\ \in V \tag{3}$$

Delay constraint:

$$\sum_{(i,j)} x_{i,j}^{s,d} \le \Delta_{s,d} \qquad \forall (s,d) \tag{4}$$

Bandwidth constraint:

$$\sum_{(s,d)} \sum_{j} x_{i,j}^{s,d} \lambda_{s,d} + \sum_{(s,d)} \sum_{j} x_{j,i}^{s,d} \lambda_{s,d} \le B \qquad \forall\ i \in V \tag{5}$$

Flow conservation:

$$\sum_{j} x_{i,j}^{s,d} - \sum_{j} x_{j,i}^{s,d} = \begin{cases} 1 & \text{if } s = i \\ -1 & \text{if } d = i \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{6}$$

Route validity:

$$x_{i,j}^{s,d} \le x_{i,j} \qquad \forall\ i, j \in V \tag{7}$$

Binary constraint:

$$x_{i,j} = 0,\ or\ 1,\ \ x_{i,j}^{s,d} = 0,\ or\ 1 \qquad \forall i, j \in V, (s,d) \tag{8}$$

Remarks
Constraint (2) ensures that nodes have broadcast ability. That is, the transmission by a node can be received by all the nodes within its transmitting range. This feature can be represented by the links in the network as: for node i, if there is a link to j (*i.e.*, $x_{i,j}$ = 1), then there must be a link to any node j' (*i.e.*, $x_{i,j'}$ = 1) if $d_{i,j'} \le d_{i,j}$, which is constraint (2).

Constraint (3) ensures that $p(i) = (d_{i,j})^{\alpha}$, where $d_{i,j} = \max\{d_{i,k} \mid 1 \le k \le n\}$.

Constraint (4) ensures that the hop-count for each node-pair does not exceed the pre-specified bound.

Constraint (5) ensures that the total transmission and reception of signals at a node do not exceed the bandwidth capacity of this node. The first term at the right hand side of inequality (5) represents all the outgoing traffics at node *i* (transmitting) and the second term represents all the incoming traffics (reception). Although this constraint does not preclude the case of simultaneous transmission and reception at a node, it is applicable to the usual case that a node is equipped with only one set of transceiver and cannot transmit and receive at the same time.

Constraint (6) is for flow conservation. Since traffics are not splittable, $x_{i,j}^{s,d}$ is either 0 or 1, representing either the entire traffics of (*s*, *d*) go through link (*i*, *j*) or none does. This constraint states that the entire traffics for (*s*, *d*) originate at node *s* and sink at node *d*, and at any intermediate node the (*s*, *d*) traffic entering this node must be equal to the traffic exiting this node.

Constraint (7) ensure that the validity of the route for each node-pair, stating that traffic flowing directly from node *i* to node *j* only when there exists a link (*i*, *j*).

Notice that the topology constructed by the above formulation is directed. To make the topology undirected (or bidirectional), we can simply add another constraint: $x_{i,j} = x_{j,i}$ for $\forall i, j \in V$. The QoS topology control problem for non-splittable case has been formulated as an integer linear programming problem (ILP) (1)-(8). There are several tools that can be employed to compute the solution to this problem. After computing out $x_{i,j}$ for $0 \le i, j \le n$, the transmitting power of node *i* can be determined by the distance to its furthest neighbor.

4.2 Numerical Results

To compute the ILP problem, we use a tool called lp_solve [23], written in ANSI C by Michel Berkelaar, to compute results. Due to the high complexity of the ILP problem, it can only compute the solution for the problem with a small size. Fig. 1 shows the topology of a network with 6 nodes and 6 requests. The simulation settings

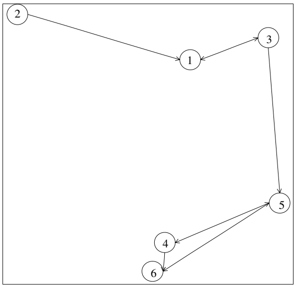


Fig. 1. Non-splittable case: topology of 6 nodes and 6 requests, P_{total} =6883

Table 1. The QoS requests and their routes for Fig.1

Request No	Source	Destination	Traffic demand	Route
1	2	3	22.5768	2→1→3
2	2	6	31.7372	2→1→3→5→6
3	3	4	40.0469	3→5→4
4	4	5	32.1713	4→5
5	4	6	41.0919	4→6
6	5	4	38.5352	5→4

are the same as in subsection 5.3. The 6 nodes are randomly dispersed in a 100×100 two-dimensional region. The bandwidth capacity of nodes, i.e., B, is set to 500. $\Delta_{s,d}$, maximally allowed hop-count for node pair (s, d) is set to 4. The source, destination, and traffic demand of the 6 requests are generated in the same way as described in subsection 5.3. The average traffic amount per request (i.e., λ_m) is 0.06B. The details of the requests and the computed routing information are showed in Tab.1.

5 Topology Control with Traffics Splittable

When the network is in operation, the traffics between a node-pair may take different routes due to congestion or failures in the network. In this section, we consider the case that the traffic demands can be split. That is, the flow can go through several different paths towards the destination.

5.1 Formulation

Given:
All the parameters in the formulation of non-splittable case remain the same.
Variables:
$x_{i,j}$ and P_{total} remain the same.
$f_{i,j}^{s,d}$, variables representing the amount of traffics of node pair (s,d) that go through link (i, j).
Optimize:
Minimize the total transmission power of nodes.

$$Min\ P_{total} = \sum_{i=1}^{n} p(i) \tag{9}$$

Constraints:
Topology constraints:

$$x_{i,j} \le x_{i,j'} \qquad \forall i, j, j' \in V, d_{ij'} \le d_{ij} \tag{10}$$

Transmitting power constraint:

$$P \ge p(i) \ge (d_{i,j})^{\alpha} x_{i,j} \qquad \forall\ i,\ j\ \in V \tag{11}$$

Delay constraint:

$$\frac{1}{\lambda_{s,d}} \sum_{(i,j)} f_{i,j}^{s,d} \le \Delta_{s,d} \qquad \forall (s,d) \tag{12}$$

Bandwidth constraint:

$$\sum_{(s,d)}\sum_{j} f_{i,j}^{s,d} + \sum_{(s,d)}\sum_{j} f_{j,i}^{s,d} \le B \qquad \forall i \in V \tag{13}$$

Flow conservation:

$$\sum_{j} f_{i,j}^{s,d} - \sum_{j} f_{j,i}^{s,d} = \begin{cases} \lambda_{s,d} & \text{if } s = i \\ -\lambda_{s,d} & \text{if } d = i \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{14}$$

Route validity:

$$f_{i,j}^{s,d} \le f_{i,j}^{s,d} x_{i,j} \qquad \forall\ i, j \in V, (s,d) \tag{15}$$

Variables constraints:

$$x_{i,j} = 0, or 1, f_{i,j}^{s,d} \ge 0 \tag{16}$$

Remarks

The objective and most of the constraints are the same as the non-splittable case. In the delay constraint (12), the delay is calculated as the average hop-count of multi-flows between two nodes. This representation of the delay constraint is reasonable, because in splittable case, traffics between a node pair can be routed via several different paths and a bound on average delay provides a good delay guarantee for network applications. Constraint (14) is for flow conservation along all the routes for node pair (*s*, *d*). Notice that the entire traffics for (*s*, *d*) (i.e., $\lambda_{s,d}$) is now split into multiple flows (i.e., $f_{i,j}^{s,d}$). The QoS topology control problem with traffics splittable has now been formulated as a mixed integer programming problem in (9) – (16).

5.2 Our Solution

Our problem is to find the network topology such that all traffics can be routed and the total transmission power is minimized. In the case where traffics are splittable, we first adjust the power of all nodes to the minimal level, i.e., $p(i) = 0$, for $1 \le i \le n$. Then we compute the QoS topology in two major steps: 1) pick a node and increase its power to reach a new neighbor. That is to add a new link to the network. 2) check if the traffics can be routed on the new topology obtained in step 1. If so, the QoS topology is found; otherwise repeat steps 1 and 2.

Because traffics are splittable, the problem in step 2 can be transformed to a variant of the multi-commodity flow problem, that is, for a given network topology, to route commodities on the network such that the maximal load of nodes is minimized. If we find the optimal solution in this topology, we can obtain a set of routes which meet the QoS requirements. Otherwise, we can conclude that the topology can not accommodate the traffics.

In the following, we first consider the step 2, the QoS routing problem for a given network topology.

A. QoS Routing Problem

Given a network graph *G* and traffic demands between node pairs, route these traffics in this graph, such that the maximal node-load in the system, denoted by L_{max}, is minimized. Node-load of node *i* is defined as the sum of all traffics that go through node *i*. This problem can be formulated as the following:

$$Min\ \ L_{\max} \tag{17}$$

$$\sum_j f_{i,j}^{s,d} - \sum_j f_{j,i}^{s,d} = \begin{cases} \lambda_{s,d} & \text{if } s = i \\ -\lambda_{s,d} & \text{if } d = i \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{18}$$

$$\sum_{(s,d)} \sum_j f_{i,j}^{s,d} + \sum_{(s,d)} \sum_j f_{j,i}^{s,d} \le L_{\max} \qquad \forall i \in V \tag{19}$$

$$\sum_{(i,j)} f_{i,j}^{s,d} \le \lambda_{s,d} \Delta_{s,d} \qquad \forall (s,d) \tag{20}$$

$$f_{i,j}^{s,d} \ge 0, \forall i,j \in V, (s,d), L_{\max} \ge 0 \tag{21}$$

Note that: $\forall (s,d), f_{i,j}^{s,d} = 0, if\,(i,j) \notin E(G)$

Function (17) is the objective, which is to minimize the maximal node load. Constraint (19) obtains the maximal node load in the network (note that all nodes have the same bandwidth capacity). Constraint (20) is delay constraint as (12). When $L_{\max} > B$, it means that the actual bandwidth usage of some nodes must have exceeded their capacities, which violates constraint (13). In this case, it indicates the given topology cannot accommodate the required traffic demands. In the following QoS topology control algorithm, we need to keep on adding more links into the topology until $L_{\max} \le B$, which means the topology can support the required traffics (i.e., no node has the actual bandwidth usage exceeding its capacity).

This is a linear programming (LP) problem. The optimal solution can be found in polynomial time. Let |*E*| denote the number of edges in graph *G*, and *t* denote the number of node pairs which have non-zero traffic. Time complexity to compute QoS routing problem is $O((|E| \times t)^{3.5})$, where $|E| \times t$ is the number of variables in formulations (17)~(21) [26]. We use Matlab 6.5 to compute the LP problem.

The next, look at step 1 on which link should be added to the topology, and integrate the QoS routing algorithm with this topology control method.

B. Energy Efficient QoS Topology Control Algorithm

We propose two methods to construct the required topology. The first method is to minimize the incremental power in each step. We name the method Least Incremental Power First algorithm (LIPF for short). The second method is to add link with the least power in each step. We call it the Least Power First algorithm (LPF for short).

LIPF algorithm

In LIPF, we first compute the least incremental power for each node to reach its new neighbor, and pick the node with minimal incremental power and increase its transmitting range to reach the new neighbor. Then we run QoS routing algorithm on this new topology to see whether the requested traffics can be all routed. The operation is repeated until the QoS topology is found, or all nodes already reach their maximal power *P* (the topology that can meet the QoS requirements does not exist in this case).

Input: node set *V* with their locations, $\lambda_{s,d}$ for node-pair (*s*,*d*), and bandwidth capacity *B*.

Output: transmitting power *p*(*i*) for each node *i* in *V*.

a) compute the least incremental power for each node to reach a new neighbor: $\Delta_i = \min\{(d_{i,j})^{\alpha} - p(i) \mid 1 \le j \le n, (i,j) \notin E\}, \forall 1 \le \mathrm{i} \le \mathrm{n}$.

b) pick the node k that reach a new neighbor j with the minimal incremental power: $\Delta_k = \min\{\Delta_i \mid 1 \le i \le n\}$, and add new link (k,j) to G.

c) run the QoS routing algorithm on G to obtain L_{max}. If $L_{\max} \le B$ or all nodes reach their maximal power P, then stop; otherwise, go to (a) and repeat.

LPF algorithm

In LPF, we first sort all node pairs (in fact, only the node pairs that can be reached within the maximal transmitting power P are considered) in ascending order according to their Euclidean distance. Each time the shortest link (the least power cost) which does not yet exist in the network is picked and the power of sender is increased until reach the other node. Then, the QoS routing algorithm runs on the network to see whether the requested traffics can be all routed. Similarly, this operation is repeated until the QoS topology is found, or all nodes already reach their maximal power P.

Input: node set V with their locations, $\lambda_{s,d}$ for node-pair (s,d), and bandwidth capacity B.

Output: transmitting power $p(i)$ for each node i in V.

a) sort all node-pairs in ascending order according to $d_{i,j}$ (note that p_{ij}= $(d_{i,j})^{\alpha}$ and $(d_{i,j})^{\alpha} \le P$).

b) add the minimal $d_{i,j}$ which does not yet exist in the network and get the new G.

c) run the QoS routing algorithm on G to obtain L_{max}. If $L_{\max} \le B$ or there is no available link left, then stop; otherwise, go to (b) and repeat.

In step (c) of both algorithms, it stops if all nodes already reach their maximal power P. An error of no solution is reported in this case. To reduce the number of times of calling the QoS routing algorithm in LPF algorithm, we use the binary search method to find the QoS topology, instead of adding a link each time and running the routing algorithm.

The following theorem states that approximation ratio of the solution found by LPF algorithm is at most n times of the optimal solution. To induce the theorem, we first introduce the following lemma.

Lemma 1. LPF algorithm finds the solution that meets the QoS requirements and the maximal transmission power of nodes in the network is minimized.

Proof. In LPF algorithm, each time the shortest link (the least power cost) is added into the network. Then the QoS routing algorithm is used to check whether the required traffic can be routed on the new topology. This process is repeated until the topology that meets the QoS requirements is formed. Note that the node power $p(i)$ is gradually increased. So the maximal node power in the solution is minimized.

Theorem 1. Approximation ratio of LPF algorithm is $O(n)$.

Proof. If LPF algorithm can not find solutions, it means that all edges are added into the network and QoS routing can not be found in this topology. That is, the required traffic can not be routed in the topology where all nodes use their maximal transmission power. So there does not exist solutions for this case.

If LPF algorithm finds solutions, we prove that the total transmission power of the network is at most *n* times of the optimal solution, where *n* is the number of nodes in the network. Let P_{total}, P_{max} denote the total transmission power and the maximal transmission power of nodes in the topology found by LPF algorithm, respectively. Let P^{opt}_{total} , $P^{opt}_{\max}$ denote the minimal total transmission power and the minimal maximal transmission power of nodes in the topology that meets QoS requirements.

According to Lemma 1, we have $P^{opt}_{\max} = P_{\max}$. Then $P_{total} = \sum_{i=1}^{n} p(i) \le \sum_{i=1}^{n} P_{\max} = nP_{\max} = nP^{opt}_{\max} \le nP^{opt}_{total}$.

Theorem 2. Time complexity of LPF algorithm is $O(n^2\log^n + \log^n(|E| \times t)^{3.5})$, where $|E|$ is the number of edges in the network, and *t* is the number of node pairs which have non-zero traffic.

Proof. In LPF algorithm, step a) costs at most $O(n^2\log^n)$ to sort all edges in the network according to their length. In step b), we adopt binary search method, it costs $O(\log^n)$. Note that step c) costs $O((|E| \times t)^{3.5})$ [26], then the total time complexity of LPF algorithm $O(n^2\log^n + \log^n(|E| \times t)^{3.5})$.

5.3 Experimental Results

The simulations are conducted in a 100×100 two-dimensional free-space region. The co-ordinates of the nodes are randomly and uniformly distributed inside the region. All nodes have the same bandwidth capacity B = 500. The value of α in the transmitting power function is set to 2, i.e., $p_{ij} = (d_{i,j})^{\alpha}$ for $\alpha = 2$.

The set of requests $R = \{(s, d, \lambda_{s,d}, \Delta_{s,d})\}$ are generated by using the Poisson function (i.e., the requests originating from a node follow the Poisson distribution. This is because the Poisson distribution is often used as a model for number of events in a specific time period [24]). For each node, we use the random Poisson function with the mean value λ= 1 to generate a number *k*, which is the number of requests originating from this node. The destinations of the *k* request are randomly picked from the other nodes. The traffic demand $\lambda_{s,d}$ for a pair of nodes (*s*, *d*) forms a normal distribution $N(\lambda_m, 0.25\lambda_m)$, where λ_m and $0.25\lambda_m$ is the mean value and the variance of the normal distribution, respectively (i.e., λ_m is the average bandwidth demand per request). $\Delta_{s,d}$ for all node pairs is uniform set to $\lfloor 2n/3 \rfloor$ to avoid excessive "no solution" cases, where *n* is the number of nodes.

In each run of the simulation, we randomly construct a network and a set of QoS requirements according to the above discussing. Then, we run LIPF algorithm and LPF algorithm on this network. Any topology that we can not find a solution is discarded. We present averages of 100 separate runs for each result shown in the figures.

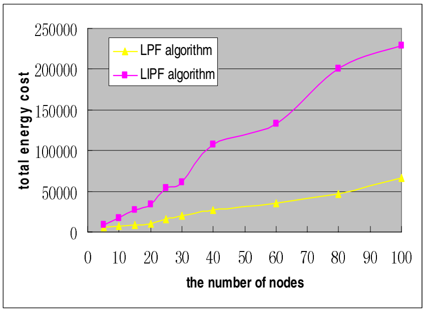


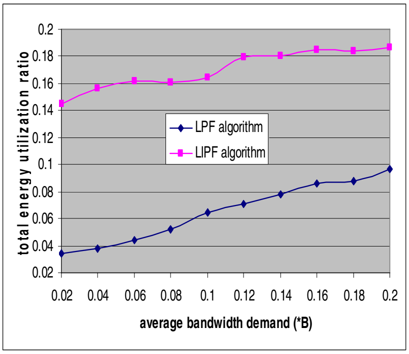
Fig. 2. Total energy cost versus *n*

The first experiment shows how the total energy cost increases by increasing the number of nodes, where total energy cost is measured by the total transmission power of nodes in the paper. We set λ_m =0.05×*B* and $P=(100\times\sqrt{2})^2$ (i.e. each node can reach any other node in its maximal power). From the curves in Fig.2, we can make the following observations:

The performance of LPF algorithm is significantly better than the performance of LIPF algorithm, especially when *n* is large. The reason is that LIPF algorithm minimizes the incremental power in each step, which may cause same node increasing its transmitting power to reach new neighbors in several sequential steps. It does not balance the transmitting power of nodes, that is, the transmitting power of nodes maybe much different. Compare with LIPF, LPF algorithm picks link with the least power in each step, which results in balance of transmitting power of nodes. Note that required traffics are distributed on the whole network. So the total energy cost of LPF will be less than the total energy cost of LIPF in average scale.

The total energy costs of both algorithms increase slowly as *n* increases from 5 to 20. The further increase of *n* would result in a quick increase of the total energy cost after *n* is greater than 20. The reason is that increase of *n* not only causes increase of the number of traffic demands, but also causes denseness of nodes. Denseness of nodes results in less energy cost to route traffic demands (because of more choices), while increasing number of traffic demands causes more total energy cost. So total energy cost increases slowly due to counteract of these two factors until *n* reach 20 in the experiment. The further increase of *n* results in a quick increase of total energy cost due to the saturation of nodes density after *n* is greater than 20.

In the simulation, we evaluate our objective by the total energy utilization ratio, which is defined by $P_{total}/(N\times P)$ (the total energy cost over the maximal total energy cost). The second experiment shows how the total energy utilization ratio and average out-degree of nodes change as the increase of λ_m. Note that the generated topology is a directed graph. We are specially concerned with the out-degree of a node, because the interference of transmissions highly depends on the out-degree of nodes in networks [9]. In the simulation, we found that when λ_m is greater than 0.2×*B*, the percentage of no solution cases (i.e., no topology can accommodate the requested



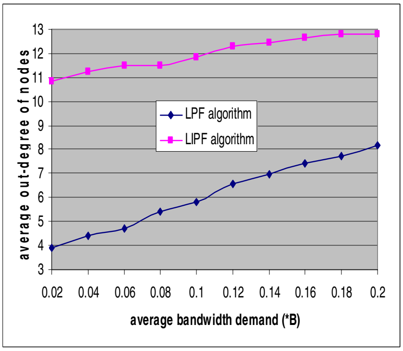


Fig. 3. Total energy utilization ratio versus λ_m **Fig. 4.** Average out-degree of nodes versus λ_m

traffics) reaches over 50%. So we increase λ_m from 0.02×*B* to 0.2×*B*, and set *n*=20, *P*= $(100\times\sqrt{2})^2$. From the curves in Fig.3 and Fig.4, we can make the following observations:

The performance of LPF algorithm is significantly better than the performance of LIPF algorithm in both figures. The reason is similar to the above analysis in (1) in the first experiment.

The total energy utilization ratio and average out-degree of nodes both increase as λ_m increases. This is because that requested traffics should be routed on more paths when λ_m increases, which results in more energy cost and larger out-degree of nodes.

Both the total energy utilization ratio and average out-degree of nodes in LPF algorithm increase faster than those in LIPF algorithm as λ_m increases. The reason is because that LIPF algorithm minimizes the incremental power in each step, which may cause same node increasing its transmitting power to reach new neighbors in several sequential steps. That is, the node has large out-degrees. This condition makes more benefit when the average bandwidth demand λ_m is large (since the traffic need to via several paths), while a lot of links are not used at all when λ_m is small. That is, LIPF performs better when λ_m becomes larger. So both the total energy utilization ratio and the average out-degree of nodes of LIPF increase slowly as λ_m increases.

The third experiment shows how the total energy cost and average out-degree of nodes change as the increase of *P*. In this experiment, we set *n*=10, λ_m =0.06×*B*. We let the maximum power *P*= $R_p\times(100\times\sqrt{2})^2$ (note that all nodes are distributed in a 100×100 region), and vary R_p from 0.5 to 1.0 (when R_p reaches 1.0, it means that each node can reach any other nodes in the region). From the curves in Fig.5 and Fig.6, we can make the following observations:

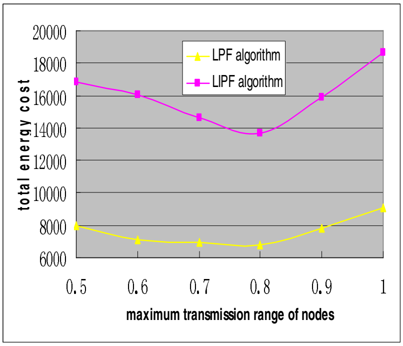


Fig. 5. Total energy cost versus R_p

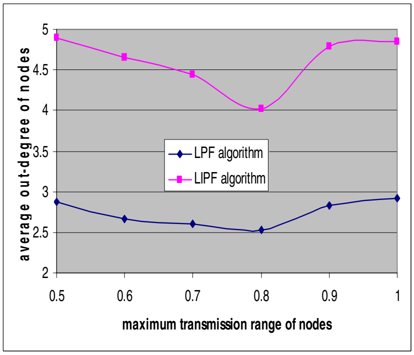


Fig. 6. Average out-degree of nodes versus R_p

The performance of LPF algorithm is significantly better than the performance of LIPF algorithm in both figures. The reason is similar to the above analysis in (1) in the first experiment.

Varying R_p has less effect on the performance of LPF algorithm than the performance of LIPF algorithm. As we pointed out before, compare with LPF algorithm, LIPF algorithm causes imbalance of transmitting power of nodes. The imbalance of transmitting power of nodes would make some nodes reach their maximal power P quickly. So varying R_p has more effect on the performance of LIPF algorithm.

An interesting phenomenon is observed that both the total energy cost and the average out-degree of nodes decrease as R_p increases from 0.5 to 0.8, but the further increase of R_p results in the increase of them. This is because some nodes reach their maximal power level and consume too much energy when R_p is too large, but it also consume much energy if there are a lot of hops between source-destination pairs when R_p is too small. In this experiment, setting R_p around 0.8 is a good choice.

6 Conclusions

We have discussed the energy efficient QoS topology control problem. Both cases of traffic non-splittable and splittable have been considered. For the former case, the problem has been formulated as an integer linear programming problem. For the latter case, a greedy algorithm LIPF and an approximation algorithm LPF were presented. We proved that the ratio of LPF algorithm is at most n, where n is the number of nodes in networks. Extensive simulations showed that the performance of LPF is much better than the performance of LIPF.

The problem discussed is a static configuration problem. The traffic demands are assumed to be known in prior. By configuring a good QoS topology, QoS requests can be best served in the system (i.e., less requests will be blocked). However, due to the dynamics and the unpredictability of network traffics, a QoS request can still be blocked no matter how good the topology is. In a dynamic environment where nodes are mobile and traffics are dynamic, the proposed topology control algorithm can be run periodically to keep a good topology in the sense that it minimizes the total energy cost, at the same time, meets users QoS requirements.

References

1. C. Zhu and M. S. Corson, "QoS Routing for Mobile Ad Hoc Networks", *IEEE INFOCOM'02*.
2. C.R. Lin and J.S. Liu, "QoS Routing in Ad Hoc Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol 17, No. 8, August 1999, pp. 1426-1438.
3. S. Chen and Klara Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, Vol.17, No. 8, August 1999, pp.1488-1505.
4. C.R. Lin, "Admission Control in Time-Slotted Multihop Mobile Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 19, No. 10, Oct 2001, pp. 1974-1983.
5. L. Hu, "Topology Control for Multihop Packet Radio Networks", *IEEE Trans. On Communications*, vol. 41, no. 10, 1993, pp. 1474-1481.
6. T. Hou and Victor O.K. Li, "Transmission Range Control in Multihop Packet Radio Netowrks", *IEEE Trans on Communications*, Vol. 34, No. 1, Jan 1986, pp.38-44.
7. R. Ramanathan, R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment", *INFOCOM'00*, pp.404-413.
8. R. Wattenhofer , L. Li, P. Bahl and Y.M. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks", *INFOCOM'01*, vol. 3, pp.1388-1397.
9. N. Li, J. Hou and Lui Sha, "Design and Analysis of an MST-based Topology Control Algorithm", *IEEE INFOCOM'03*.
10. Z. Huang, C.C. Shen, C. Scrisathapornphat and C. Jaikaeo, "Topology Control for Ad Hoc Networks with Directional Antennas", *IEEE 11th Conf on Computer Communications and Networks*, Miami, Oct 2002, pp.16-21.
11. M.A. Marsan, C.F. Chiasserini, A. Nucci, G.Carello, and L.D.Giovanni, "Optimizing the Topology of Bluetooth Wireless Personal Area Networks", *IEEE INFOCOM'02*.
12. V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks", *IEEE J. Selected areas in communications*, vol. 17, no.8, 1999, pp.1333-1344.
13. Suresh Singh, Mike Woo and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", *ACM MOBICOM'98*, Dallas, 1998, pp.181-190.
14. V. Kawadia and P. R. Kumar, "Power Control and Clustering in Ad Hoc Networks", *IEEE INFOCOM'03*.
15. J.E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks", *IEEE INFOCOM'00*.
16. P. J. Wan, G. Calinescu, X. Y. Li and O. Frieder, " Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks", *IEEE INFOCOM'01*.
17. L. Hu, "Distributed code assignments for CDMA Packet Radio Networks", *IEEE/ACM Trans. on Networking*, Dec. 1993, pp.668-677.
18. X. Jia, D. Li, and D. Du, "QoS Topology Control in Ad Hoc Wireless Networks", *IEEE INFOCOM'04*.
19. E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan and S. S. Ravi, "Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks", *ACM MobiHoc'02*.

20. A. Muqattash and M. Krunz, "CDMA-Based MAC Protocol for Wireless Ad Hoc Networks", *ACM MobiHoc'03*, pp.153-164.
21. V. Rodoplu and T. Meng, "Position based CDMA with Multiuser Diction (P-CDMA/MUD) for Wireless Ad Hoc Networks", *IEEE 6th Int'l Symp. On Spread Spectrum Techniques and Applications*, vol. 1, 2000, pp.336-340.
22. E. Sousa and J.A. Silvester, "Spreading Code Protocols for Distributed Spread-Spectrum Packet Radio Networks", *IEEE Trans. on Communications*, vol.36, no.3, Mar 1988, pp.272-281.
23. http://www.cs.sunysb.edu/~algorith/implement/lpsolve/implement.shtml**.**
24. http://www.math.csusb.edu/faculty/stanton/m262/poisson_distribution/Poisson_old.html.
25. A.E.F. Clementi, P. Penna and R. Silvestri, "The power range assignment problem in packet radio networks in the plane," in: *Proc. of the 17th Annual Symposium on Theoretical Aspects of Computer Science* (*STACS 2000*) (February 2000) pp. 651–660.
26. A. Schrijver, "Theory of linear and integer programming, " 1986, John Wiley&Sons.
27. X.Y. Li, W.Z. Song and W. Wang, "A Unified Energy Efficient Topology for Unicast and Broadcast, "*ACM MobiCom*, 2005.

An Energy Efficient TDMA Protocol for Event Driven Applications in Wireless Sensor Networks*

Haigang Gong[1], Ming Liu[1], Xiaomin Wang[1], and Li Xie[2]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, P.R. China
[2] Department of Computer Science and Technology, Nanjing University
State Key Laboratory for Novel Software Technology, China

Abstract. One of the key problems for Wireless Sensor Networks (WSNs) is the design of Medium Access Control (MAC) protocol. MAC protocol controls the activity of wireless communication module of sensor nodes, which is the major consumer of sensor energy. The energy efficiency of MAC protocol makes a strong impact on the network performance. TDMA-based MAC protocol is inherently collision free, and can rule out idle listening since nodes know when to transmit. In this paper, we present ED-TDMA, an energy efficient protocol for event driven applications in wireless sensor network. ED-TDMA improves channel utility by changing the length of TDMA frame according to the number of source nodes and saves energy with bitmap-assisted TDMA schedule. In addition, ED-TDMA employs intra-cluster coverage to prolong network lifetime and to improve system scalability. Simulation results show that ED-TDMA performs better for wireless sensor network with high-density deployment and under low traffic.

1 Introduction

Wireless sensor networks (WSNs) consist of a large number of very small, low-cost and battery-powered sensors with low-power radio, which can be used to collect useful information (i.e. temperature, humidity) from a variety of environment. WSNs have been envisioned to have a wide range of applications in both military as well as civilian domains [1][2] such as battlefield surveillance, machine failure diagnosis, and chemical detection.

Since sensor nodes powered by battery are often left unattended after deployment, e.g., in hostile or hash environments, making it difficult to replace or recharge their batteries, the protocols running on WSNs must be energy efficient. According to D. Estrin [2], the radio component of sensor nodes consumes most of nodes' energy when receiving or transmitting data, even in idle state. On the other hand, medium access control (MAC) protocol directly controls the activity of nodes' radio and decides when the competing nodes may access the shared medium to transmit the

* This work is partially supported by the Science Foundation of UESTC under Grant No. L08010601JX05031.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 638 – 649, 2006.

data. So, medium access is the major consumer of sensor energy and MAC protocols must be energy efficient to achieve longer network lifetime.

A lot of MAC protocols have been studied in recent years and could be categorized into two classes: schedule-based MAC protocols including TDMA, FDMA and CDMA, and contention-based such as S-MAC [3]. In schedule-based MAC protocol, TDMA is an important approach that is inherently collision free and avoids unnecessary idle listening, which are two major sources of energy consumption. For the inherently property of energy conserving, TDMA protocols have been recently attracted significant attention for many applications [4-6]. However, TDMA has poor scalability. Cluster-based protocols with traditional TDMA schedule, i.e. LEACH [7] and HEED [8], are more scalable than traditional TDMA protocol. In cluster-based TDMA protocol, sensor nodes are organized into several clusters and cluster heads are responsible for scheduling their members in a TDMA manner. Cluster-based TDMA protocol improves the scalability of the network and is suitable for large-scale wireless sensor networks. However, either traditional TDMA or cluster-based TDMA only applies for continuous monitoring applications, i. e. continuous collecting the temperature of the environments. They could achieve high channel utility because sensor nodes always have data to send in continuous data gathering applications. But when applying for another typical application in WSNs -- event driven applications such as earthquake monitoring and target tracking, in which sensor nodes only have data to send when a specific event occurs, they will waste more energy and achieve lower channel utility because sensor nodes still must be active when the event doesn't happen.

In this paper, we present ED-TDMA, an energy efficient TDMA protocol for event driven applications in wireless sensor network. ED-TDMA improves channel utility by changing the length of TDMA frame according to the number of source nodes and saves energy with a bitmap-assisted TDMA schedule. In addition, ED-TDMA employs intra-cluster coverage to prolong network lifetime and to improve system scalability. Simulation results show that ED-TDMA performs better for wireless sensor network with high-density deployment and under low traffic.

The rest of the paper is organized as follows. Section 2 presents the problem. Section 3 describes our ED-TDMA protocol in detail. Section 4 discusses the simulation results. Finally, Section 5 concludes the paper and presents future research directions.

2 Problem Statement

The operation of HEED which employs clustering and traditional TDMA schedule is divided into rounds. As shown in Fig.1, each round begins with a set-up phase, followed by a TDMA schedule phase and several TDMA frames. In the set-up phase, sensor nodes are organized into several clusters. And then the cluster heads broadcast a TDMA schedule to their members, allocating a slot to the members. In the following TDMA frames, the members send the data to their respective cluster heads during the allocated slot. There is only 1 TDMA schedule in each round and the length of TDMA frame is equal.

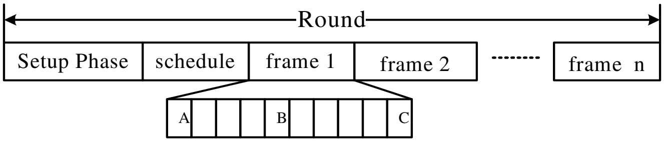


Fig. 1. Frame structure of traditional TDMA protocol

Obviously, this traditional TDMA schedule is effective for continuous monitoring applications while nodes have the data to send all the time. But for event driven applications, it has some disadvantages such as lower channel utility and unnecessary energy wastage of the cluster heads. As shown in Fig. 1, a TDMA frame contains 10 slots. If there has only several source nodes to transmit during a frame, there must be some empty slots. For example, node A, B and C transmit their data during the first, the fifth and the tenth slot, respectively, then 7 slots are empty which wastes network bandwidth and decreases the channel utility. Moreover, cluster heads do not know which members transmit in the current TDMA frame, so that cluster heads must be active during the round even if there have no data to transmit, which leads to unnecessary energy wastage of cluster heads.

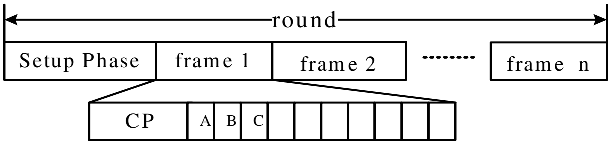


Fig. 2. Frame structure of BMA protocol

BMA [9] protocol improves traditional TDMA schedule in that there exists a contention phase (CP) in the beginning of each TDMA frame. In the contention phase during each frame, source nodes send 1-bit message to their cluster heads to reserve data slot so that cluster heads know which members will transmit in this frame and allocate successive data slot to these source nodes. When the source nodes finish their transmission, cluster heads could be asleep and will be active in the next frame. As in Fig.2, source node A, B and C transmit during the first three data slots and their cluster head could enter into sleep state in the forth data slot to avoid unnecessary energy wastage. However, like in traditional TDMA protocol, TDMA frames in BMA protocol have the same length, which couldn't improve channel utility of the network. In addition, there's a TDMA schedule in each frame and cluster heads will broadcast a TDMA schedule packet in each frame. The schedule packet includes the member's ID and the slot number allocating to the members, which introduces extra energy overhead. Broadcasting and receiving these schedule packets consumes considerable energy when the node density is high.

Our ED-TDMA protocol then improves channel utility by changing the length of TDMA frame according to the number of source nodes and reduces the length of TDMA schedule packets with a bitmap-assisted TDMA schedule to decrease the schedule overhead. Besides, it employs intra-cluster coverage scheme to prolong network lifetime and to improve system scalability.

3 ED-TDMA Protocol Design

3.1 Basic Protocol

Like BMA, the operation of ED-TDMA is divided into rounds. Each round begins with a set-up phase, followed by a steady phase. Set-up phase includes clustering and time synchronization. The steady phase consists of *n* TDMA frames that have different frame length. As shown in Fig. 3, each frame begins with a reservation phase, followed by a TDMA schedule and data transmission.

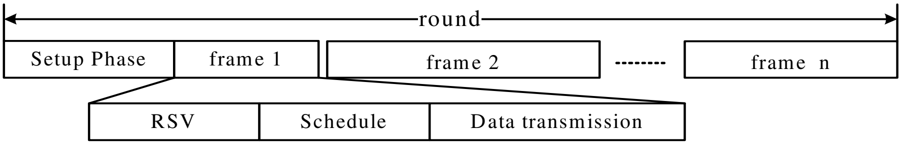


Fig. 3. Frame structure of ED-TDMA

The reservation phase consists of *m* mini-slot. *m* is the number of members in the cluster. The members occupy the mini-slot according to their *ID*. Node has the maximum *ID* occupies the first mini-slot while node has the minimum *ID* occupies the last mini-slot, and so on. A member sends a 1-bit *RSV* message to the cluster head if it has data to send in the current frame. Obviously, the length of the reservation phase is *m* bit.

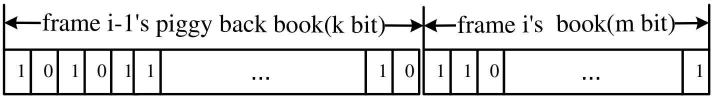


Fig. 4. TDMA schedule packet

In the TDMA schedule phase, the cluster head broadcasts a schedule packet according to the received *RSV* message in the reservation phase. The schedule packet format is a bit-map sequence as shown in Fig. 4. The sequence consists of two parts. The first *k*-bit part represents the piggybacking reservation of the previous frame, in which each bit corresponds to a source node in the previous frame. The second *m*-bit part represents the reservation of the current frame, in which each bit corresponds to a node in the cluster. Parameter *k* represents the number of the source nodes or the number of data slots in the previous frame and satisfies $0 \le k \le m$. The value of *k* is variable with the number of the source nodes and is set to 0 in the first frame of each round. In the schedule sequence, 1 means a source node has booked a data slot. If a source node reserves data slot in the i^{th} mini-slot, then it corresponds to the i^{th} bit of the last *m*-bit of the schedule sequence. If a source node reserves data slot by piggybacking reservation and it transmits data during the j^{th} data slot in the previous frame, it's reservation corresponds the jth bit of the first k-bit of the schedule sequence. A source node determines its data slot number according to the number of 1 in the substring of the schedule sequence ending at its corresponding bit. Obviously, the number of 1 is the number of data slots, k, in the current frame. All members in the cluster, including source nodes and non-source nodes, could get the knowledge of k from the schedule packet and then enter the reservation phase of the next frame after k data slots. If the number of source nodes is small, then the frame length will be too short, leading to frequent reservation and schedule and introducing more energy overhead. To avoid frequent reservation and schedule when the number of source nodes is small, we define a default minimum frame length $T_{frame\text{-}min}$. If the current frame length is less than $T_{frame\text{-}min}$, the frame length is set to $T_{frame\text{-}min}$.

For example, assuming that 4 source nodes A~D send the *RSV* message to the cluster head in the 1st, 2nd, 4th and mth mini-slot, respectively. The cluster head then broadcasts the TDMA schedule packet. In the schedule sequence shown in Fig.5, node A~D correspond the 1st, 2nd, 4th and mth bit of the schedule sequence, respectively. Note that the sequence has only the second m-bit part in the first frame. From the sequence, they know the current frame has 4 data slots. The corresponding substring of A is 1, then the slot number of node A is 1; the corresponding substring of C is 1101, then C occupies the 3rd data slot because the number of 1 in the substring is 3. Likewise, node B and node D occupy the 2nd and 4th data slot. In the second frame, assuming that node A, node C and node D reserve their data slot in the previous frame by piggybacking and node E and node F send the *RSV* message in the 3rd and 5th mini-slot in the reservation phase, the TDMA schedule packet then contains two parts in which the first 4 bit is the piggybacking reservation and the last m bit is the reservation of the current frame, as shown in Fig. 6.

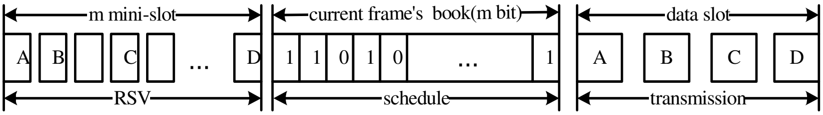


Fig. 5. The first frame structure of ED-TDMA

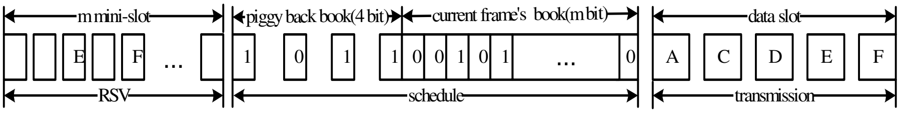


Fig. 6. The second frame structure of ED-TDMA

In the schedule sequence, node A, node C and node D correspond the 1st, 3rd and 4th bit of the sequence while node E and F correspond the 3rd and 5th bit of the last m-bit of the sequence. Then node A, node C and node D occupy the first 3 data slot in the current frame. The corresponding substring of E is 1011001 so that the slot number of E 4, and the corresponding substring of F is 101100101 so that node F occupies the 5th data slot.

In the transmission phase, the source nodes transmit the data to the cluster heads during its data slot. If they have more data to send in the next frame, they could reserve the data slot in the next frame by piggybacking a flag in the data packet.

Noticeably, if there are no any nodes have data to send, all nodes should be asleep for a default frame length to avoid frequent reservation and schedule. $T_{frame\text{-}def}$ is related to specific applications. $T_{frame\text{-}def}$ could be longer if the application has no real time requirements.

Obviously, the length of the schedule packet is $(k+m)/8$ bytes. With $0 \le k \le m$, the length of the schedule packet, l_s, satisfies $m/8 \le l_s \le m/4$. For BMA and traditional TDMA, the length of the schedule packet, l_s', is related to the number of the cluster members, *m*. Assuming that the schedule information includes the node's *ID* (2bytes) and the slot number (1byte), then l_s' is *3m* bytes.

The time of a round is predetermined and remains constant in the runtime, but the number of frames of clusters in a round is different from each other because the number of source nodes in each cluster is different. In order to enter into the next round at the same time, cluster heads are responsible for determine an appropriate length of the last frame.

3.2 Intra-cluster Coverage

Coverage is one of the most important issues in WSNs and has been studied in recent years [10]-[12]. And K-coverage can be descried as that every point in the monitored field is covered by at least K sensor. In [12], authors think it is hard to guarantee full coverage for a given randomly deployment area even if all sensors are on-duty. Small sensing holes are not likely to influence the effectiveness of sensor networks and are acceptable for most application scenarios. It's enough to meet the application's requirements if the active nodes in the network could maintain reasonable area coverage—coverage expectation. Coverage mechanism is to choose a subset of active nodes to maintain the coverage expectation.

We introduce this idea into clusters that is called "intra-cluster coverage", which selects some active node within clusters while maintaining coverage expectation of the cluster. Based on our previous work [13], cluster head randomly chooses *m'* nodes according to equation (1).

$$p_{\mathrm{cov}er} = \sum_{i=K}^{m'} C_{m'}^{i} \left(\frac{r}{R}\right)^{2i} \left(1-\frac{r^2}{R^2}\right)^{m'-i} \tag{1}$$

where P_{cover} is the coverage expectation of sensing field determined by specific applications; and *r* is sensing radius, *R* is cluster radius; *m'* is the number of active nodes. For example, distributing 200 nodes in a 100×100m^2 field, r = 12m, R = 30m, then the average number of cluster members is 60 or so. With intra-cluster coverage, if P_{cover} = 99% which means 99% of sensing field is expected to be monitored, 27 members should be active in each cluster to ensure 1-coverage of the cluster and 38 members to ensure 2-coverage. If P_{cover} =95%, only 16 nodes and 25nodes should be active to ensure 1-coverage and 2-coverage respectively.

Using intra-cluster coverage has two advantages. The first is to preserve energy consumption in each round by turning redundant nodes' radio off so that network lifetime is prolonged. The second is to reduce TDMA schedule overhead. Once clusters grouped, all cluster head broadcast a TDMA schedule packet in which contains the members ' *ID* and slot number allocated to the member. When node density is high, the number of cluster members turns higher so that the length of TDMA schedule packet turns longer that consumes more energy to transmit and receive. However, the length of TDMA schedule packet would not too long with intra-cluster coverage because the number of active node varies slightly when node density goes higher.

3.3 Energy Analysis

Assume that there are *m* nodes and m_s source nodes in each cluster and the event whether a node has data to send or not can be viewed as a Bernoulli process, in which the probability that a node has data to send is *p* and there is m_s=*mp*.

To ED-TDMA, the energy of the source nodes is consumed for sending the *RSV* message in the reservation phase, receiving TDMA schedule packet and transmitting data to the cluster head. It could be expressed as:

$$E_s = E_t(l_r, d_i) + E_r(l_s) + E(l_d, d_i) = (l_r + l_s + l_d)E_{elec} + (l_r + l_d)e_{fs}d_i^2 \qquad (2)$$

where d_i is the distance from source nodes to cluster head; l_r, l_s and l_d are the length of the reservation message, TDMA schedule packet and data packet, respectively.

Non-source nodes consume energy only for receiving TDMA schedule packet.

$$E_{ns} = E_r(l_s) = l_s E_{elec} \ . \qquad (3)$$

E_{CH} is the energy consumption of the cluster head, including listening or receiving in the reservation phase, broadcasting TDMA schedule packet and receiving data packet from the source nodes.

$$E_{CH} = mE_r(l_r) + E_t(l_s, r) + \sum_{i=1}^{m_s} E_r(l_d) \ . \qquad (4)$$

Then the total energy dissipated in a frame is:

$$\begin{aligned} E_{ED-TDMA} &= E_{CH} + \sum_{i=1}^{m_s} E_s + (m - m_s)E_{ns} \\ &= \left[(m + m_s)l_r + (m+1)l_s + 2m_s l_d\right]E_{elec} + \sum_{i=1}^{m_s}(l_r + l_d)e_{fs}d_i^2 + l_s e_{fs} r^2 \end{aligned} \qquad (5)$$

According to [9], the total energy consumption of BMA in a frame is:

$$E_{BMA} = \left[m(m+1)l_r + (m+1)l_s' + 2m_s l_d\right]E_{elec} + \sum_{i=1}^{m_s}(l_r + l_d)e_{fs}d_i^2 + l_s' e_{fs} r^2 \qquad (6)$$

For traditional TDMA protocol, the energy consumption could be expressed as:

$$E_{TDMA} = (m + m_s)l_d E_{elec} + \sum_{i=1}^{m_s} l_d e_{fs} d_i^2 \ . \tag{7}$$

The length of the reservation message l_s is only 1 bit. And there are $m/8 \le l_s \le m/4$ and $l_s^{'} = 3m$. Then we have

$$\begin{aligned} E_{ED-TDMA} - E_{BMA} &\le -(23m^2 + 22m - mp)E_{elec} - 22me_{fs}r^2 \\ &\le -(23m^2 + 21m)E_{elec} - 22me_{fs}r^2 \le 0 \end{aligned} \ . \tag{8}$$

From (10), the larger *m* is, the less energy consumption of $E_{ED\text{-}TDMA}$ than that of *E*-$_{BMA}$.

Besides, there is

$$E_{ED-TDMA} - E_{TDMA} \le \lfloor 2m^2 + 10m + 8mp - m(1-p)l_d \rfloor E_{elec} + 2me_{fs}r^2 \ . \tag{9}$$

It means that the relationship between $E_{ED\text{-}TDMA}$ and E_{TDMA} is related to the length of data packet, l_d.

3.4 Time Synchronization

Many applications of sensor networks depend on the time accuracy kept by nodes in the network. So sensor networks require a way for nodes to synchronize their clocks to a global time. Moreover, time synchronization is indispensable in the implementation of the commonly used medium access control protocols such as TDMA. There have been several time synchronization protocols for wireless sensor networks in recent years such as Reference Broadcast (RBS [14]) and Timing-Protocol for Sensor Networks (TPSN [15]). In RBS scheme, the nodes periodically send beacon messages to their neighbors using the network's physical layer broadcast. Recipients use the message's arrival timestamp as point of reference for comparing their clocks. However, the RBS scheme is effective only for a small cluster of nodes lying in a neighborhood. TPSN protocol is level-based time synchronization, which is more scalable than RBS. In TPSN, a hierarchical topology is created first, assigning each node a level in the hierarchy. Then, each node synchronizes itself to a node belonging to exactly one level above in the hierarchy. Eventually all nodes in the network synchronize their clocks to a reference node.

TPSN could be incorporated into our ED-TDMA after clustering in the setup phase. The cluster heads could be organized into a hierarchical structure. The time is at first synchronized among the cluster heads using TPSN, and then the cluster heads synchronize with their respective cluster members using simple local synchronization technique. Obviously, time synchronization introduces extra energy overhead especially in large hierarchical topology. The time to resynchronization (a round time) should be long enough to reduce the synchronization overhead, which is determined by the synchronization accuracy and clock drift of nodes.

4 Performance Evaluation

4.1 Simulation Setup and Parameters

We implemented ED-TDMA, ED-TDMA1, BMA and traditional TDMA protocols in the glomosim network simulator with the wireless extension, in which ED-TDMA1 is the extension of the basic ED-TDMA with intra-cluster coverage scheme. Simulation parameter are listed in table I. Assuming that data rate is 19.2kbps, which is the data rate of TR1000 [14] when using OOK modulation, then transmitting 100bytes data needs 42ms. A data slot is set to 45ms, which is long enough to send 100bytes data to the cluster head. For ED-TDMA, $T_{frame\text{-}min}$ is relevant to sampling frequency and sampling bits of the sensors [15] and should be long enough to generate a data packet. When the sampling frequency is 100Hz and 16bit sampling, $T_{frame\text{-}min}$ then is set to 495ms. The reservation phase and schedule phase could be accomplished in a data slot. Set $T_{rsv}+T_{schedule}$ = 45ms. Moreover, we assume that the packets are generated according to Bernoulli process. The transmission probability is *p*, which controls the network load. And the radio model for the radio hardware energy dissipation is the same as in [7].

Table 1. Simulation parameters

Parameters	Value	Parameters	Value
Sensor area (*M*×*M*)	100×100m^2	*n*	10
Node number (*N*)	300	T_{slot}	45ms
Sensing radius(*r*)	12m	$T_{frame\text{-}min}$	495ms
Cluster radius (*R*)	30m	$T_{frame\text{-}def}$	9.9s
P_{cover}	95%	$T_{rsv}+T_{schedule}$	45ms

4.2 Simulation Results

Fig. 7 and Fig. 8 plot the TDMA schedule overheads after 5000 data cycles under different node density and different traffic load, respectively. With the increase of the node density, which means the number of members in the cluster increases, the schedule overhead of BMA increases rapidly and is far more than ED-TDMA. For example, when node density is 0.04nodes/m^2, the schedule overhead of BMA is triple than ED-TDMA. When node density is constant and the traffic load turns higher, the number of source nodes increases which increases the length of the schedule packet so that schedule overhead also increases. For ED-TDMA, the max length of schedule packet is 2m bits so that its schedule overhead increases slowly. For ED-TDMA1, the number of working nodes is constant and is far less than others; its schedule overhead is very small and is independent on the node density and traffic load.

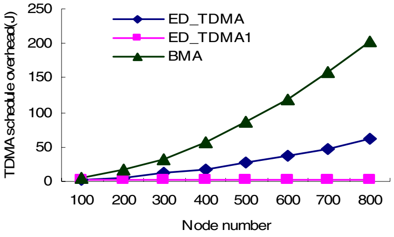


Fig. 7. TDMA schedule overhead .vs node number

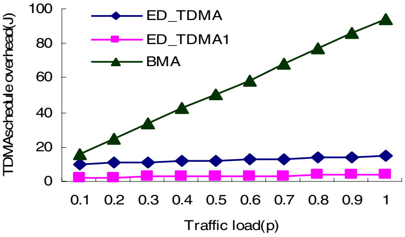


Fig. 8. TDMA schedule overhead under difff-errent traffic load

Fig. 9 and Fig. 10 show the energy consumption after 5000 data cycles under different node density and different traffic load. Obviously, there are more energy consumptions with the increase of node density or traffic load. And BMA consumes energy more quickly than ED-TDMA. For instance, BMA consumes about 25% more energy than ED-TDMA and about 91% more than ED-TDMA1, when node density is 0.03nodes/m^2. As shown in Fig. 10, the traditional TDMA wastes more energy due to the idle listening of cluster heads during a round, especially under light traffic load. When p is higher than 0.8, the energy consumed by traditional TDMA is less than ED-TDMA. The reason is that the working time of cluster heads is long but ED-TDMA has more energy consumption in TDMA schedule.

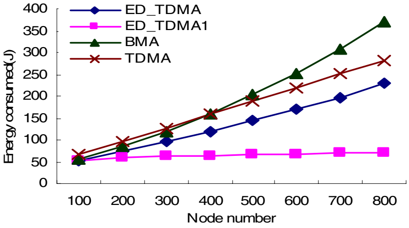


Fig. 9. Energy consumption under different node number

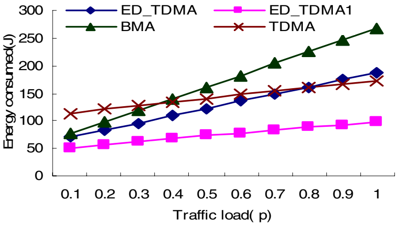


Fig. 10. Energy consumption under different traffic load

Fig. 11 shows the relationship between energy consumption and data packet length after 5000 data cycles. The energy consumption increases with the increasing of packet length. The energy consumed by traditional TDMA is faster than others, which reflects the essence of the equation (9). The more the packet length is, the more energy consumed by traditional TDMA than ED-TDMA.

5 Conclusion

In this paper, we presented ED-TDMA, an energy efficient TDMA protocol for event driven application in wireless sensor network. ED-TDMA improves channel utility by changing the length of TDMA frame according to the number of source nodes and

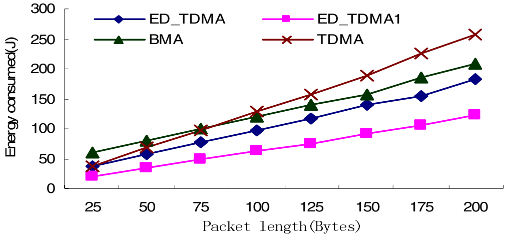


Fig. 11. Energy consumption under different traffic load

saves energy with bitmap-assisted TDMA schedule. In addition, ED-TDMA employs intra-cluster coverage to prolong network lifetime and to improve system scalability. Simulation results show that ED-TDMA performs better for wireless sensor network with high-density deployment.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, Vol. 38, pp. 393-422, March 2002.

[2] D. Estrin and R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks", in *Proc. of MobiCOM '99*, August 1999.

[3] W. Ye, J. Heidenmann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", *In Proceedings of IEEE INFOCOM*, New York, NY, June 2002.

[4] K. Arisha, et al. "Energy-aware TDMA-based MAC for sensor networks", *In IEEE Workshop on Integrated Management of Power Aware Communications, Computing and NeTworking(IMPACCT 2002)*, New York City, NY, May 2002.

[5] S. Kulkarni, et al. "TDMA service for sensor networks", *In 24th Int. Conf. on Distributed Computing Systems (ICDCS04), ADSN workshop*, pp. 604-609, Tokyo, Japan, March 2004.

[6] G. Pei and C. Chien, "Lower power TDMA in large wireless sensor networks", *in Military Communications Conference (MILCOM 2001)*, volume 1, pp. 347-351, Vienna, VA, Oct. 2001.

[7] W. R. Heinzelman, et al. "An Application -Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactions on Wireless Communications,* vol. 1, no. 4, Oct. 2002.

[8] O. Yonis, et al. "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks", *IEEE Transactions on Mobile Computing*, volume 3, issue 4, Oct-Dec, 2004.

[9] S. Kulkarni and M. Arumugam, "TDMA service for sensor networks," *In 24th int. Conf. on Distributed Computing Systems(ICDCS04), ADSN workshop*, pp. 604-609, Tokyo, Japan, March 2004.

[10] H. Zhang and J.C. Hou, "Maintaining scheme coverage and connectivity in large sensor networks," in *Proceedings of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc wireless, and Peer-to-Peer Networks*, 2004.

[11] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C.D. Gill, “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks,” in *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, pp 28-39, ACM Press, 2003.

[12] Y. Gao, K. Wu, and F. Li, “Analysis on the redundancy of wireless sensor networks,” in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (WSNA 03)*, September 2003, San Diego, CA.

[13] LIU Ming, et al. “A Distributed Energy-Efficient data Gathering and aggregation Protocol for Wireless sensor networks”, *in Journal of Software*, January, 2006.

[14] RF Monolithics. http:// www. rfm. com/, *ASH Transceiver TR1000 Data Sheet.*

[15] R. Govindan, and D. Estrin. A Wireless Sensor Network for Structural Monitoring. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, November 2004.

Self-organization Data Gathering for Wireless Sensor Networks

Hongli Xu[1,2], Liusheng Huang[1,2], Junmin Wu[1,2], Yang Wang[1,2]
Jichun Wang[1,2], and Xu Wang[3]

[1] Department of Computer Science and Technology, University of Science & Technology of China, Hefei 230027, P.R. China
hlxu3@mail.ustc.edu.cn
[2] Anhui Province key Laboratory of Software in Computing and Communication, Hefei 230027, P.R. China
[3] Department. of Research and Development, Anhui Machine Whole Set Bureau, 230001, P.R. China

Abstract. Sensor networks have attracted much attention in the recent years for its wide applications in biology, medicine, security and battlefield, etc. Data gathering is one of the most important operations in wireless sensor networks. In this paper, we present an energy-delay efficient data gathering protocol, SODG, for sensor networks while taking the transmission interference into considerations. The significance of this proposal is that it is fully localized and distributed, and only depends on the one-hop neighbors' information. To minimize the delay, the parallel transmissions are permitted with interference-free guaranteed. The experimental simulations show that our protocol is energy-delay efficient. Especially, SODG protocol improves about 64% over PEGASIS on energy consumption, and about 80% over chain-based protocols on delay.

Keywords: Data Gathering, Interference, Wireless Sensor Networks, Energy, Delay.

1 Introduction

With the development of low-cost processor and radio technologies, it becomes possible to make cheap wireless micro-sensor nodes. Though these sensors are not so powerful compared with those macro-sensor counter-parts, it is possible to build a high-quality and fault-tolerant sensor network by using hundreds of them. Wireless sensor networks (WSNs) can be used to collect the useful information from the monitored area, especially where physical environment is so harsh that the macro-sensors can not be deployed. WSNs have wide applications in the different fields, such as military, environment surveillance, biology and common security, etc [1, 2].

The wireless sensor network usually consists of a large scale of inexpensive sensor nodes. Each sensor node is equipped with transmission power control and an omni-directional antenna, therefore the communication range can be changed by adjusting its transmission power. A sensor node can sense the local environment by the certain

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 650 – 661, 2006.

sensor, and exchange the information with others. For example, a sensor network can be used to detect the presence of the potential threats in a military conflict. What's more important than individual data in the sensor network is data aggregation from the entire network. Since power is the most critical resource for each sensor node, the protocol should be energy-efficient for data gathering. Moreover, the delay is another important factor for some emergent applications, such as patience monitoring and fire detection, etc.

Data gathering is an important issue in the monitoring sensor networks, and some algorithms have been presented for this problem. These algorithms can be divided into three classes. One is cluster based, such as LEACH [3] and HIT [4]. Both of them first divide the sensor nodes into clusters, and fused the results to the base station. To mediate the conflict, the cluster-heads should support TDMA schedule and assign the time slots for the inner-nodes. The second class is chain-based, such as PEGASIS [5, 6] and DWZ [7]. These methods only permit one sensor node to send the data packet in each time, thus interference is avoided naturally. To minimize the latency, the hierarchical approach is introduced by Lindsey and Rahsavendra [6]. The third class is tree-based [8, 11-13]. All of them gather the sensed data from the network through the tree structure. These algorithms are centralized, and required to know the informa- tion of the entire network. Therefore, they are not suitable for self-organized sensor networks.

In this paper, we present a fully localized tree-based data gathering protocol for self-organized sensor networks. In order to preserve the energy, the proposed method constrains the transmission between Gabriel neighbors. To minimize the delay, the parallel transmissions among the sensors are necessary while guaranteeing interference free. We prove its correctness and compare with PEGASIS [5] and PEDAP [11] protocol through experimental simulations.

The rest of the paper is organized as following. Section 2 introduces the preliminary background for the proposal. The next section presents a self-organization data gathering (SODG) protocol to construct the data gathering tree and assign each node a time slot. In section 4, we prove the correctness of the algorithm. The experimental results are shown to compare SODG protocol with PEDAP [11] and PEGASIS [5] algorithms in section 5. Section 6 concludes the paper.

2 Preliminary Background

2.1 System Model

In this paper, the sensor network is treated as the wireless ad-hoc network, which contains a large scale of static sensor nodes with limiter energy. Each node can adjust the transmission's area by power control and sense the environment. Among them, some nodes need to relay data to the base station (*BS*), while the user can access the network through BS or Internet. The base station is fixed, and has sufficient power supply. In the following, $|uv|$ denotes the Euclidian distance between node u and v.

We use the same radio model as in [3], which is the first order radio model. In this model, each radio dissipates $E_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $E_{amp} = 100pJ/bit/m^2$ for the transmitter amplifies. The radio can expend the required energy to reach the intended recipients, and be turned off to avoid receiving unintended transmission. The transmission and receiving cost for a *k*-bit message and the distance *d* are shown below:

$$E_T(k,d) = E_{elec} \times k + E_{amp} \times k \times d^2 \tag{1}$$

$$E_R(k,d) = E_{elec} \times k \tag{2}$$

2.2 System Model

The interference concept has been introduced in some previous works [9, 10]. For example, Burkhart et al. [9] define the interference of a link *uv* as the number of the nodes covered by two disks centered at u and v with radius $|uv|$. In this paper, the definition of the interference is a little different from that in [9]. Assume that there is only one channel used in the network. The interference occurs on one node, if and only if this node is in the communication ranges of at least two transmitters simulta- neously. Consider the following scene where *u* and *s* send the packets to node *v* and *t* respectively, as shown in Fig. 1. if $|sv| \le |st|$, interference occurs on node *v*, because it is in the transmission ranges of both node *u* and *s*. Thus, the nodes *u* and *s* can not transmit simultaneously.

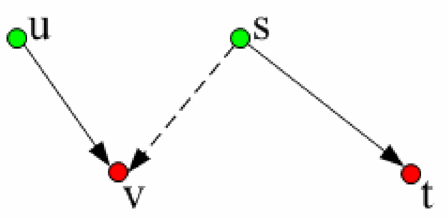


Fig. 1. Illustration for Interference

2.3 Computational Geometry

This sub-section mainly introduces the concept of Gabriel Graph (*GG*). For convenience, let $disk(u,v)$ be the closed disk with the diameter *uv*. The Gabriel Graph of a point set *V*, denoted as *GG*(*V*), consists of all edges *uv* such that node *u* and *v* are connected and $disk(u,v)$ doesn't contain any nodes from *V* except *u* and *v*. As shown in Fig. 2, link *uv* doesn't belong to *GG*, because $disk(u,v)$ contains node *s*. if the point set *V* is connected, then *GG*(*V*) keeps the original connectivity.

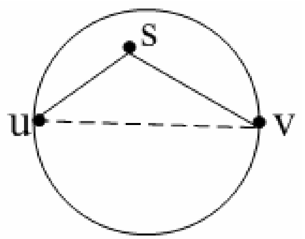


Fig. 2. Illustration for Gabriel Graph

2.4 Problem Formalization

The sensor network is modeled as a planar graph G={V, E}, where V is the set of nodes and E is the set of edges between the nodes. All the nodes are labeled from 0 to n-1 in turn, where n is the number of the nodes in the network. Two nodes u and v can communicate directly, if their planar distance is less than d, where d is the maximal transmission radius of the sensor nodes. Thus, the neighbor set of node u is denoted as $N(u)=\{v \mid |uv| \le d\}$. For simplicity, base station is regarded as node 0. Our task is to construct a data gathering tree for a given graph G, rooted at the base station. To mediate the interference, each node except the root will be assigned a time slot to transmit. The ultra goal is that data gathering on the constructed tree is energy and delay efficient with interference-free in the entire procedure.

Now, we formalize this problem into Integer Programming (IP). First, the variable $z_{i,j}^{k}$ is used to denote whether the link $\mathrm{l_{i,j}}$ is just on the path from node k to the base station. Here, we assume that $z_{i,j}^{k}$ is n if this link is a sub-path from node k to base station. That is:

$$z_{i,j}^{k} = n \text{ or } 0 \tag{3}$$

Each link $\mathrm{l_{i,j}}$ is assigned a time slot, $x_{i,j}$, which is denoted as an integer from 0 to n-1. Obviously, if the link $\mathrm{l_{i,j}}$ is not included in the tree, $x_{i,j}$ should be zero. Otherwise, $x_{i,j}$ is less than n. Thus,

$$x_{i,j} \le Max\{z_{i,j}^{k}\},\ \forall k: 1 \le k \le n-1 \tag{4}$$

One constraint on the data gathering tree is the transmission's orders of the nodes. That is, if there are two consecutive links $l_{i,j}$ and $l_{j,k}$ in the tree, then $x_{i,j} < x_{j,k}$. The other constraint is that all the nodes except the base station only send one packet to the neighbor by fusion, and may receive several packets from the neighbors. Thus, the formalization is:

$$Max\{x_{i,j} \mid 1 \le i \le n-1\} < Max\{x_{j,m} \mid 1 \le m \le n-1\},\ \forall j : 1 \le j \le n-1 \quad (5)$$

In the data gathering tree, each node k will transmit a sensed data to one of its neighbors, expressed in equation (6). And the final result must reach the base station, expressed by equation (7). While for the intermediate node, it receives the others' packets and forwards to the next-hop neighbor, as in equation (8).

$$\sum z_{i,j}^{k} = n\text{, } i{=}k \quad (6)$$

$$\sum z_{i,j}^{k} = n,\ j = 0, \quad (7)$$

$$\sum z_{i,j}^{k} = \sum z_{j,l}^{k}\text{, Otherwise} \quad (8)$$

To denote the interference, we construct an interference matrix C, where the element $c[l_{i,j}, l_{m,n}]$ is 1, if the simultaneous transmission on $l_{i,j}$ and $l_{m,n}$ won't conflict with each other. Otherwise, $c[l_{i,j}, l_{m,n}]$ is 0. Based on this,

$$x_{i,j} \ne x_{m,n}\text{, if } c[l_{i,j}, l_{m,n}] = 0 \text{ and } (x_{i,j} > 0 \text{ or } x_{m,n} > 0) \quad (9)$$

In order to compute the energy consumption, another variable $y_{i,j}$ is used to denote whether the link $\mathrm{l}_{\mathrm{i,j}}$ is included in the data gathering tree. It means that this link is included in the tree, if this link must be on the path from at least one node to the base station. Therefore,

$$y_{i,j} = \max\{z_{i,j}^{k} \mid 1 \le k < n\},\ \forall(i, j, k) \quad (10)$$

Our task is to minimize the total energy consumption and the delay in the data gathering. The total consumption *P* is derived as following:

$$P = \frac{1}{n} \times \sum (y_{i,j} \times E_{i,j}) \quad (11)$$

Where $E_{i,j}$ is the energy cost through the link $\mathrm{l}_{\mathrm{i,j}}$. Our object is described as:

$$Min\{P\} \text{ and } Min\{\max\{x_{i,0}\}\} \quad (12)$$

The problem of self-organization data problem has been formulated as an integer programming. There are several solutions to solve this kind of problem, such as cut-and-branch, etc. But these general algorithms are time cost and centralized. At the same time, these algorithms are all centralized, thus not fit for the wireless sensor networks. Next section will design a fully distributed and localized algorithm to solve this problem.

3 SODG Protocol Description

This section mainly described our contribution, self-organized data gathering (SODG) protocol. It is a fully localized and distributed protocol, which belongs to the tree- based data gathering protocols. Thus, SODG protocol is fit for self-organized sensor networks. We assume that the entire network is synchronized by some synchronization mechanisms.

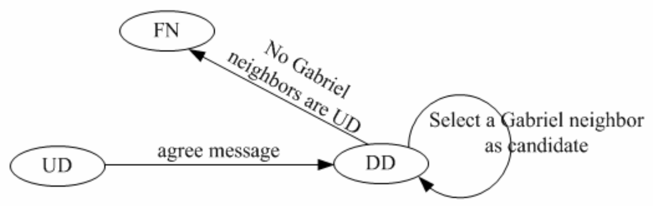

FN
No Gabriel neighbors are UD
Select a Gabriel neighbor as candidate
UD
agree message
DD

Fig. 3. State Transition Graph for Each Node in SODG Protocol

3.1 SODG Protocol

The protocol consists of three phases. In the first phase, each node collects the one- hop neighbors' information, and determines the Gabriel neighbors based on the local information. To conserve the energy, each node only transmits the sensed data or fused result to the Gabriel neighbors. By this phase, each node discovers its Gabriel neighbors.

The second phase is to construct a data gathering tree and assign each node an integer, called inverse tune skit (*ITS*). It is ensured that no interference occurs while the nodes with the same *ITSes* transmit simultaneously. Each node has three states: Undecided (*UD*), Decided (*DD*) and Finished (*FN*). This phase is divided into several rounds. Initially, all nodes are set to *UD* state except that the base station is set to *DD* state. At the beginning of the k-th round, $k \geq 1$, each node that becomes *DD* state from *UD* state in the last round broadcasts a *CLM* message to the direct neighbors about its new state. Next, each DD node selects the closest UD Gabriel neighbor as a candidate and broadcasts a *CNDT* message containing the identity of the candidate. If the candidates are not unique, the one with the minimal identity is selected. After a certain time *t*, which is more than the maximal delay between two nodes, each *UD* node decides whether it can win or not in the current round. The WIN algorithm is described in details in the next sub-section. The won node broadcasts *ACK* message in the vicinity. After another period *t*, if some *DD* node only receives the ACK message from the selected candidate, it sends an *AGREE* message to the candidate. On receiv- ing the *AGREE* message, its state is changed from *UD* to *DD* and set *ITS* as *k*. When one node detects that all the Gabriel neighbors are in the DD or FN state, the node turns to FN state. The state transition graph is shown in Fig. 3. An example of the result after this phase is shown in Fig. 4(a), where the numbers in the bracket denote the node identity, the parent identity and *ITS* respectively. The formal description of this phase is given in Fig. 6. In the description, UGN_i denotes the Gabriel neighbors with UD state of node *i*.

And the array variables *T* and *R* denote the candidate transmitters and receivers in the current round.

In the last phase, each node will be assigned a real time slot (*RTS*) for data gathering which denotes the transmission order for this node. Each leaf node on the tree sends its ITS to the parent node. If the relay node has received all the *ITSes* from the children, it sends the maximal one combined with own *ITS* to the parent node. When the root knows the maximal *ITS*, denoted as *MTS*, then broadcasts in the network. On receiving the *MTS*, each node computes its *RTS* as following: *RTS*=*MTS*- *ITS*. An example is shown in Fig. 4(*b*), where the numbers in the bracket denote the node identity, the parent identity and RTS respectively.

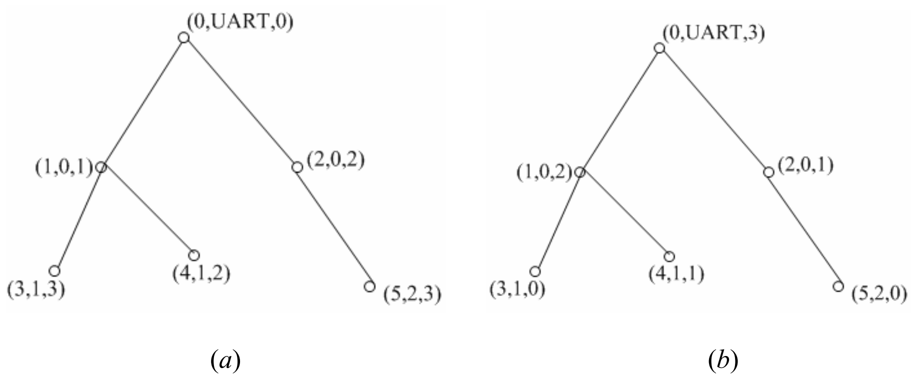


Fig. 4. The result on each node after phase 2 (*a*) and phase 3 (*b*)

```
Boolean Win($u$, $v$){
  For $i$=1 to $m$ do{
    if( $(|T[i],R[i]|,T[i]) \succ (|u,v|,u)$ )
    if( $|T[i],R[i]| \geq |T[i],v|$ )
      Return False;
  }
  Return True;
}
```

Fig. 5. The description of Win algorithm

3.2 Win Algorithm

The win algorithm is used as a called function in the SODG protocol. Given a set of the candidate transmitters and receivers, the algorithm is to determine whether node *u* can transmit to node *v* or not in this round. Without loss of generality, assume that all the transmitters are distinct. We first order all the candidate transmissions according to the certain rules. The priority of the candidate transmission from node u to v is denoted as $(|u,v|,u)$, where $|u,v|$ is the planar distance between node u and v. the priority (p_1,p_2) is higher than (p_3,p_4), denoted as $(p_1,p_2)\succ(p_3,p_4)$, if and only if $p_1 < p_3$ or $((p_1 = p_3)$ and $(p_2 < p_4))$. The node can win, if its transmission won't interfere with those transmissions of higher priority. The formal description of win algorithm is given in Fig. 5. The arrays *T*[1..*m*] and *R*[1..*m*] are the candidate transmitters and receivers, where *m* is the number of candidate transmissions. The returned result indicates that the transmission from node *u* to *v* is permitted (*True*) or forbidden (*False*) in this round.

4 Correctness Analysis

This section proves the correctness of the proposed algorithm. The first theorem is to show that SODG guarantees interference-free in the procedure of data gathering.

Theorem 1. SODG protocol guarantees no interference.

Proof: assume by contradiction that there is interference in the parallel transmissions. As shown in Fig. 1, when node *s* and *u* transmit the packets to *t* and *v* respectively, *v* can also overhear the message from *s* for $|st| \geq |sv|$. Thus, node *u* will overhear the candidate transmissions messages (*s*, *t*) and (*u*, *v*) from *t* and *v* respectively. We consider the following two cases:

1. $(|s,t|,s)\succ(|u,v|,u)$. By the Win algorithm, node *u* cannot win in this round because of its lower priority.

2. $(|u,v|,u)\succ(|s,t|,s)$. In this case, node *u* will win, and send *ACK* message to the neighbors. Thus node *t* will receive this message. Since $(|u,v|,u)\succ(|s,t|,s)$, node *t* won't send *AGREE* message to node *s* though node *s* win in this round or not.

Therefore, there is no interference on node *t*. The theorem is proved. □

Theorem 2. The algorithm terminates in the limited rounds.

Proof: We divide all the sensor nodes into two node sets, US and DS. US contains the nodes of *UD* state, while DS contains others. Consider all the Gabriel links that connecting DS and US. Among those, if there is the unique shortest link, the attached *UD* node will win in the current round. Its state is changed from *UD* to *DD*. Otherwise, the *UD* node associated with the shortest link and the minimal ID is selected. This node is changed to *DD* state too. By this way, all the nodes change to *DD* state after limited rounds. Thus the algorithm is terminable. □

```
Variable: Parent, T[1..m], R[1..m], Selected = True, index = 0;
//state: to denote the current state of this node;
//CurrentRx: to denote the transmitter who selects the local node as candidate;
At the beginning of round k on node i:
 If state=DD && Selected
   Then Broadcast CLM(i) to the neighbors
       Selected = False;
 If state=DD && $UGN_i \neq \phi$        //select the closest Gabriel neighbor
     Then Select the closest Gabriel neighbor $s \in UGN_i$,
              Broadcast CNDT(s) message to the neighbors
              Selected = False;
              Index=0, CurrentRx=-1
On receiving the CNDT (s) from node j  //receiving the candidate message
  If state=UD
     Then T[index] =s
        R[index++]=j
        If(s=i) then currentRx=j
After waiting for a certain time t        //check whether this node can win
   If (win(i,currentRx)) then broadcast ACK(CurrentRx) to neighbors;
After waiting for a certain time t
  If state=DD and only one ACK message is received from the candidate node s
    Then Send AGREE(s) to node s
On receiving AGREE(s) from node j
  If (s=i) then ITS = k
                     parent = j
                     state = DD
On receiving CLM message from Gabriel neighbor j
    $UGN_i = UGN_i - \{j\}$
   if $UGN_i = \phi$ then state=FN
```

Fig. 6. Self-organization Data Gathering Protocol Description

5 Experimental Results

This section compares the performance of SODG protocol with the previous algorithms, such as PEGASIS [5] and PEDAP [11]. In the experimental simulations, we generate the sensor network depending on the predefined number of sensor nodes and the area size. The sensors are randomly located among the rectangle area, and the base station is on the vertex of the rectangle. Also, we assume that the scale of each packet is 1000 bits.

Using this network configuration in the simulation, we run each protocol and track its progress in terms of the number of rounds in each cycle. To evaluate the performance of SODG protocol, we compare three protocols under the same network configuration. As we know, PEGASIS protocol is a hierarchical data gathering algorithm with interference free. And PEDAP protocol is based on tree structure and energy efficient. But this algorithm cannot guarantee free interference. In the following, the energy cost is the total consumption of all sensor nodes in each data gathering cycle, including receiving and delivering power cost.

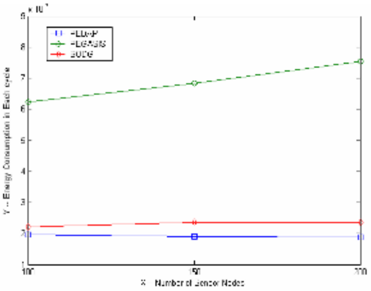


Fig. 7. Node number vs. Energy Cost

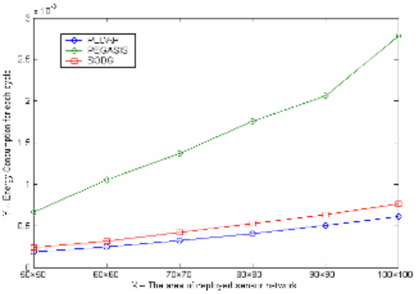


Fig. 8. Area size vs. Energy Cost

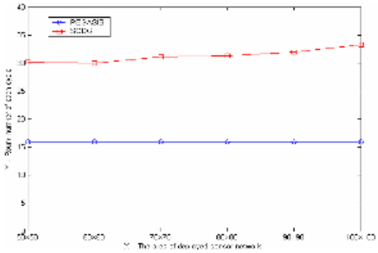


Fig. 9. Area size vs. Delay

First, we observe the impact of the number of the nodes on the performance of three algorithms. In the simulations, all the sensor nodes are deployed in the area of $50m \times 50m$, while the number of the nodes is variable as 100, 150 and 200. And the maximal transmission radius is 10m. The comparison of the energy cost is shown in Fig. 7. the figure shows that the performance of the proposal is similar to that of PEDAP, and improves about 65% on PEGASIS protocol. That's because SODG protocol only permits the transmission to occur between the Gabriel neighbors, thus avoiding the transmission among the long distance.

The next experiment mainly explores how the size of the deployed area impacts the energy cost and delay of each cycle. We deploy 200 sensor nodes in the different areas, and the experimental results are shown in Fig. 8 and Fig. 9. In Fig. 8, the energy cost of SODG protocol increases slowly similarly to PEDAP protocol. On the contrary, that of PEGASIS increases very fast as the area size enlarges. That's because PEGASIS protocol will result in the long-distance's transmission, thus cost much energy. Figure 9 shows that the delay of SODG protocol is only two times that of PEDAP protocol, and improves 64% on PEGASIS protocol. That's because SODG protocol will increase the delay to avoid the interference, compared with the PEDAP protocol. What's more, the delay of SODG is about 1/4 – 1/6 of chain-based algorithms [3, 5, 7]. Thus, the proposed algorithm is energy-delay efficient.

Finally, we observe how the maximal communication radius impacts the performance of SODG protocol. Here, 200 nodes are deployed in $50m \times 50m$ area, while the maximal transmission radius is varied from 8m to 12m. The final results are shown in Fig. 10 and Fig. 11. The both figures show that the communication radius has little impact on the performance of three protocols.

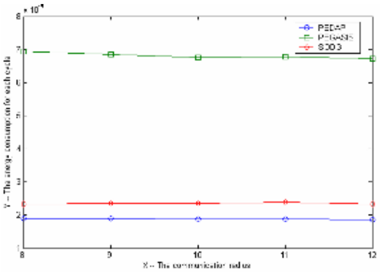


Fig. 10. Max. Commu. Radius vs. Energy Cost

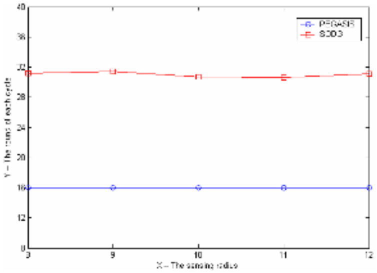


Fig. 11. Max. Commu. Radius vs. Delay

6 Conclusions

In this paper, we present a self-organization data gathering protocol for wireless sensor networks, which is fully localized and distributed. Multiple sensor nodes are permitted to transmit simultaneously with interference free guaranteeing. The experimental simulations show that SODG protocol has satisfactory performance compared with the previous algorithm. In the future, we will focus on the minimizing the maximal energy consumption on each node. Thus, the lifetime of the sensor network can be extended. Moreover, the more complex interference model should be studied for this problem.

Acknowledgments. This work is supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2006CB303006 and by the China Next Generation Internet Program (Grant No. CNGI-04-10-2A). Also, this work is partly supported by the Innovative Fund of USTC.

References

1. D. Estrain, R. Govindan, J. Heidemann, and S. Kumar, Next century challenges: scalable coordination in sensor networks, Proceedings ACM/IEEE, MobiCom, Aug. 1999;
2. A. Mainwaring, J. Polastre, R. Szewcayk, and D. Culler. Wireless sensor networks for habitat monitoring, ACM International Workshop on Sensor Networks and Applications (WSNA'02), Atlanta, GA, Sept. 2002;
3. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocol for wireless micro-sensor networks, Proceedings of the 33rd Annual Hawaii International Conference on System Science, Jan 2000, pp.3005-14;
4. J. Culpepper, Lan Dung, M. Moh, Hybrid indirect transmission (HIT) for data gathering in wireless micro-sensor networks with biomedical applications, In proceedings of 18th Annual workshop on Computer Communications, pp.124-33, 2003;
5. S. Lindsey, and C. S. Raghavendra, PEGASIS: Power-Efficient Gathering in Sensor Information Systems, Proceeding IEEE Aerospace '02, vol:3, 9-16, Mar 2002
6. S. Lindsey, C. Raghavendra, Data gathering algorithms in sensor networks using energy metrics, IEEE Transactions on Parallel and Distributed Systems, vol 13, No. 9. Sep 2002;
7. K. Du, J. Wu and D. Zhou, Chain-based protocols for data broadcasting and gathering in the sensor networks, Proceedings of the international Parallel and Distributed Processing Symposium, 2003;
8. V. Annamalai, S.K.S.Gupta, and L. Schwiebert, On Tree-Based Convergecasting in Wireless Sensor Networks, WCNC, 2003;
9. M. Burkhart, P. V. Richknbach, R. Wattenhofer and A. Zollinger, Does topology control reduce interference, In ACM Mobihoc, 2004;
10. K. M. Nejad and X. Y. Li, Low-interference topology control for wireless ad-hoc networks, Journal of Ad Hoc and Sensor Wireless Networks, Vol. 1, pp. 41-64;
11. H. Tan and I. Korpeoglu, Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks, SIGMOD Record, Vol. 32, No. 4, Dec 2003;
12. T.S. Chen, H.W. Tsai, and C.P. Chu, Gathering-Load-Balanced Tree Protocols for Wireless Sensor Networks, IEEE Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2006, Vol. 2, 05-07 Jun, 2006, pp. 8 – 13;
13. Y. Yu, K. B and P.V. K, Energy-latency tradeoffs for data gathering in wireless sensor networks, 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, March 7-11, 2004;

Continuous Monitoring of *k*NN Queries in Wireless Sensor Networks

Yuxia Yao, Xueyan Tang, and Ee-Peng Lim

School of Computer Engineering
Nanyang Technological University
Singapore 639798
{yaoy0003, asxytang, aseplim}@ntu.edu.sg

Abstract. Wireless sensor networks have been widely used for civilian and military applications, such as environmental monitoring and vehicle tracking. In these applications, continuous query processing is often required and their efficient evaluation is a critical requirement to be met. Due to the limited power supply for sensor nodes, energy efficiency is a major performance measure in such query evaluation. In this paper, we focus on continuous *k*NN query processing. We observe that the centralized data storage and monitoring schemes do not favor energy efficiency. We therefore propose a localized scheme to monitor long running nearest neighbor queries in sensor networks. The key idea is to establish a monitoring area for each query so that only the updates relevant to the query are collected. Experimental results show that our scheme outperforms the centralized scheme in terms of energy efficiency and network lifetime.

1 Introduction

The development of wireless technology and sensors have enabled wide use of sensor networks. In these networks, a large number of low-powered sensor nodes are distributed in an area of interest and wirelessly connected. The sensor nodes are equipped with computation and communication capabilities [13]. Sensor networks are popular for a variety of applications, e.g., habitat monitoring, pollution monitoring, and object tracking [2,9]. Data collection and query processing in sensor networks are challenging research topics in sensor network database management. Existing work has focused on non-spatial query processing [8,16]. To the best of our knowledge, there has been little work on spatial query processing [19,21], especially spatial query monitoring. In this paper, we consider monitoring *k*NN queries in an object tracking sensor network.

Energy efficiency is a critical design consideration in wireless sensor networks. The sensor nodes usually with low battery power have to be deployed unattended for a long time. To prolong the network lifetime, we need to reduce network-wide energy consumption. Energy is mainly consumed during communication [13]. Thus, to reduce energy consumption, we need to reduce the number of message transmissions. Meanwhile, it is also important to balance energy consumption across sensor nodes since the sensor network may be disconnected and fail to operate properly if some nodes run out of energy and fail to communicate. A straightforward *centralized scheme* for monitoring *k*NN queries is to continuously send the sampled locations of objects to a base station.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 662–673, 2006.

User queries are also routed to the base station for initial and continuous evaluations. However, the centralized scheme is likely to suffer from unnecessary update messages. This is because kNN queries are usually *localized* in that only the locations of objects close to the query points need to be reported for kNN query processing and the objects farther away can be exempted from location updates. Moreover, in the centralized scheme, the energy consumption is highly unbalanced among the sensor nodes. Sensor nodes closer to the base station consume much more energy due to message relay and this would reduce the network lifetime. To improve energy efficiency, it is desirable to store data locally at the sensor nodes in a distributed manner and process the queries *in-network* [6,18,21]. In this way, we hope to extract only the relevant data from the network and cut down the communication cost compared to the centralized scheme.

Motivated by the localized property of kNN queries, we propose a *localized scheme* to continuously evaluate kNN queries in sensor networks. Each query is characterized by a geographical location q called the *query point*, and a number k of the required nearest neighbors. The objective of a kNN query over a set of objects O is to identify the k objects with the shortest distances to the query point, i.e., to find an ordered subset of k objects $\mathcal{N} = \{o_1, o_2, ..., o_k\} \subseteq O$ such that $\forall o_i \in \mathcal{N}$ and $\forall o \in O - \mathcal{N}$, $d(o_i, q)$ $\leq d(o, q)$ and $\forall i < j$, d(o_i, q) $\leq d(o_j, q)$, where $d(o, q)$ denotes the Euclidean distance between an object o and the query point q. Our key idea is to establish a monitoring area for the query so that only the relevant updates are collected. In this way, we reduce the network-wide energy consumption and avoid hotspots in the network.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 introduces some preliminaries and Section 4 presents the localized scheme to continuously evaluate kNN queries. Section 5 describes the experimental setup and discusses the experimental results. Finally, Section 6 concludes the paper.

2 Related Work

With the growing needs for location-based services, continuous monitoring of kNN queries is becoming more popular in spatial databases [4,10,11,14,17,22]. Recently, some grid-based methods are explored in continuous monitoring of kNN queries. Examples include YPK-CNN [22], CPM [10] and SEA-CNN [17]. These methods assume that there is a centralized repository to store all object locations and all location updates are simply reported to the centralized repository. However, such centralized storage is costly for object tracking sensor networks due to their energy constraints. Therefore, these methods are not appropriate for kNN monitoring in sensor networks.

Other relevant works include the MobiEyes algorithm proposed by Gedik [4] and a threshold-based algorithm proposed by Mouratidis [11]. Similar to YPK-CNN, SEA-CNN and CPM, there is a centralized server in the system. But differently, the MobiEyes and threshold-based algorithms assume *smart* objects that have some storing and processing capabilities. When an object moves away from its current position, the object can decide whether to send the location update to the server or not. Both the MobiEyes and threshold-based algorithms aim at reducing the communication cost between the objects and the server by eliminating unnecessary location updates. MobiEyes [4] focuses on monitoring range queries by assigning a safe region to each query. The objects within the safe region periodically check whether they are in the query range. Only the objects within the query range report their locations to the server. The threshold-based algorithm [11] assigns a distance range to each object in the result set. The distance range for the ith nearest object is defined by two thresholds: the midpoint between the ith and the $(i-1)$th nearest objects and the midpoint between the ith and the $(i+1)$th nearest objects. Only when an object moves out of its distance range is the location update of the object sent to the server. However, in both [4,11], all the queries are still processed at one centralized server. In contrast, in this paper, we make use of the localized property of kNN queries to process them in-network.

3 Preliminaries

3.1 System Model

We consider a sensor network with sensor nodes distributed over a 2-dimensional space. The sensor nodes are aware of their locations through GPS [3] or other localization algorithms [12]. Each sensor node can communicate directly with the nodes (called neighbors) within the distance r_{tx} of radio communication. Through message exchange, each sensor node is aware of the geographical locations of its neighbors. We assume the network is connected, i.e., any sensor node can communicate with any other sensor node either directly or indirectly through a routing protocol. The sensor nodes detect moving objects within their sensing range r_s and sample their locations periodically. We assume a dense sensor network in which a geographical area of interest is fully covered by the sensing ranges of the sensor nodes. Instead of sending all collected location data to a central repository, we propose to store them locally at the detecting sensor nodes [18,20]. Recall that each kNN query specifies a query point q. A continuous kNN query issued by the user is injected into the sensor network at any sensor node and forwarded to q through GPSR routing [7,15]. The sensor node closest to q would receive the query. This sensor node is called the *query initiator*. Our objective is to continuously collect the kNN result at the query initiator which in turn returns it to the user.

3.2 One-Shot Remainder kNN Query Processing

We first consider the processing of a generalized one-shot kNN query called *remainder kNN query*. A remainder kNN query finds k nearest objects to a given query point q among the objects beyond a given distance r from q. When $r = 0$, a remainder kNN query reduces to the original kNN query. Remainder kNN queries are used for query reevaluation in kNN monitoring (refer to Section 4). The evaluation of a one-shot remainder kNN query proceeds in two phases: (i) *preliminary search* and (ii) *expanded search*. The purpose of the preliminary search is to find a *boundary object* and define the search space. In this step, the sensor nodes surrounding the query initiator are visited by message passing until at least k objects are collected. Among the k objects detected, the kth closest one to the query point is selected as the boundary object. A search space is defined based on the location of the boundary object to guarantee that it includes all sensor nodes possibly detecting an object closer to the query point than the boundary

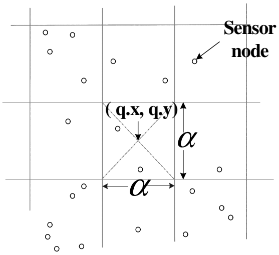


Fig. 1. Grid Structure in Sensor Networks

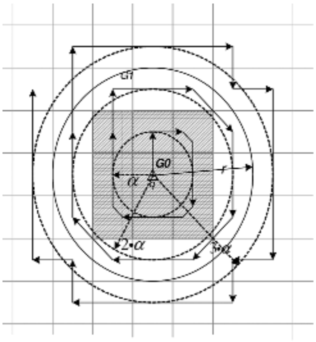


Fig. 2. Msg. Routing Path in Preliminary Search

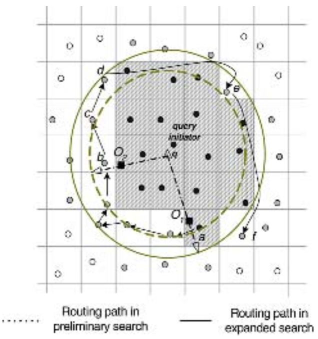


Fig. 3. Expanded Search

object. During the expanded search, the sensor nodes in the search space that are not yet visited in the preliminary search are visited to locate the k nearest objects. Finally, the query result is routed back to the query initiator.

To facilitate message traversal among the sensor nodes, the sensor network is partitioned into a set of grid cells. As shown in Figure 1, each grid cell is a square of size $\alpha \times \alpha$. For a query (q, k), the grid structure is constructed by designating q as the centroid of a grid cell. Then, given any sensor node located at (x, y) on the plane, the centroid of the grid cell containing (x, y) is given by $\big(q.x + \frac{\alpha}{2} + \frac{1}{2} \cdot (\lfloor \frac{x-(q.x+\frac{\alpha}{2})}{\alpha} \rfloor \cdot \alpha + \lceil \frac{x-(q.x+\frac{\alpha}{2})}{\alpha} \rceil \cdot \alpha), q.y + \frac{\alpha}{2} + \frac{1}{2} \cdot (\lfloor \frac{y-(q.y+\frac{\alpha}{2})}{\alpha} \rfloor \cdot \alpha + \lceil \frac{y-(q.y+\frac{\alpha}{2})}{\alpha} \rceil \cdot \alpha)\big)$.

The preliminary search and expanded search are carried out by visiting a series of grid cells. When visiting grid cell G, one sensor node (called the R-node) is responsible for collecting the object locations detected by the sensor nodes in the cell. Once receiving the query, the R-node broadcasts a one-hop *probe* message to the sensor nodes in G. To guarantee that all sensor nodes within G can hear the probe message, the diameter of the grid cell (i.e., $\sqrt{2}\alpha$) should be less than the transmission range r_{tx}. Therefore, we set $\alpha = \frac{1}{\sqrt{2}} \cdot r_{tx}$ and the R-node of a grid cell to be within $\frac{1}{2} r_{tx}$ to the centroid of the cell. The query point and the centroid location of G are included in the probe message broadcast by G's R-node. Each sensor node knows the value of α based on r_{tx} and thus knows the centroid of the grid cell containing itself autonomously. Only sensor nodes in G will reply to R-node with the detected object locations if any.

After collecting the data from the sensor nodes in G, the R-node processes the object locations accordingly (see Sections 3.3 and 3.4 for details) and continues forwarding the query message to the next grid cell. To select the R-node of the next grid cell G' to be visited, the R-node of G checks whether any of its neighbors is within distance $\frac{1}{2} r_{tx}$ to the centroid of G'. If multiple such neighbors exist, the one closest to the centroid of G' is selected as the R-node. If there is no such neighbor, the message is routed to the centroid of G' by GPSR routing. The sensor node S closest to the centroid would receive the query message and become the R-node. For simplicity, we assume a dense sensor network where there is at least one sensor node in each grid cell. So, S must be within distance $\frac{1}{2} r_{tx}$ to the centroid of G'. Due to space limitation, the handling of empty grid cells will be discussed in the extended version of this paper.

3.3 Preliminary Search

In the preliminary search, we need a rule to determine the visiting order of grid cells. Since the location of boundary object determines the search circle for expanded search, to reduce search cost, we would like the boundary object to be as close to the query point q as possible. Thus, it is intuitive to visit the grid cells based on their distances to q. We propose a *circle* approach to determine the visiting order of grid cells.

Specifically, the search is divided into rounds. In each round i ($i \geq 1$), the unvisited grid cells intersecting with the circle centered at q and with a radius of $i \cdot \alpha$ are visited in clockwise order (see the dash line circle in Figure 2). The query message contains the location of the query point q. Note that given the location of q, each sensor node can determine autonomously which grid cell to visit next. In case of $r = 0$, the query starts from the cell G centered at q. In case of $r > 0$, the query starts from a grid cell G' which is not within the circle centered at the query point q and with radius r. To determine this starting grid cell, we list all grid cells from round 1 to round $\lceil \frac{r}{\alpha} \rceil$ in a clockwise order. The first grid cell in the list whose maximum distance to q exceeds r is selected as the starting grid cell. Figure 2 shows the message routing path when $r = 0$ and $r > 0$. When $r = 0$, the preliminary search starts from G_0. When $r > 0$, the grid cells within the circle with radius r are labelled as shadow. They are exempted from being visited and the preliminary search starts from G_1. The query message is then routed to the centroid of G_0 or G_1 and received by the R-node. The R-node of each visited grid cell collects object locations from the sensor nodes in the cell and records them in the query message. The preliminary search completes when the number of collected objects is no less than k. Among these objects, the kth object closest to the query point q is chosen as the boundary object. The search space is then defined as a circle centered at q and with a radius of $d = r_b + r_s$, where r_b is the distance between the boundary object and q, and r_s is the sensing range. Intuitively, if the minimum distance between a grid cell and the query point q is smaller than d (the radius of the search circle), the sensor nodes in the grid cell are likely to detect objects less than distance r_b away from q.

3.4 Expanded Search

In expanded search, a *search list* is given by all grid cells within or intersecting with the search circle, excluding those already visited in the preliminary search. The query message passed between the grid cells in the expanded search contains the search list, the locations of the k recorded objects, and the query point q. When the R-node of a grid cell G receives the query message, it first removes G from the search list. After sending a probe message and collecting object locations from the other sensor nodes in G, one of the following three cases can occur at the R-node: (i) no object is detected by any sensor node in G; (ii) all objects detected are further away from the query point q than the boundary object; (iii) at least one object detected is closer to q than the boundary object. In cases (i) and (ii), the search list and the objects recorded in the message do not change. In case (iii), the detected objects nearer to q than the boundary object are used to update the k nearest objects recorded in the message. Meanwhile, the boundary object is updated as the new kth nearest object and the search circle is shrunk accordingly. The search list is then updated by removing all grid cells outside the new search circle. On finishing with a grid cell G, the query message is routed to the cell on the search list that is closest to G. The expanded search continues until the search list becomes empty. On completion of the expanded search, the message is routed to the query initiator and the locations of k recorded objects are returned to the user as the query result.

Figure 3 shows an example of 1NN query processing. The grid cells in shadow are visited in the preliminary search. Suppose the boundary object O_1 is found by R-node a. Node a determines the search circle (shown by the outer solid circle in Figure 3) and derives the set of grid cells in the search list (i.e., all the grey R-nodes in Figure 3). The query message is passed among the gray R-nodes of grid cells in the search list, and the object locations are collected. Suppose that at R-node b, a nearer boundary object O_2 is found. Then, the search circle is shrunk accordingly. The dotted circle in Figure 3 is the new search circle. The grid cells containing R-nodes c, d, e and f are now the only four unvisited grid cells left in the revised search list. After visiting the grid cell with R-node f, the search list becomes empty and the expanded search completes.

4 Localized Scheme for Continuous *k*NN Queries

4.1 Overview

The set of kNNs and their locations may change over time as the objects move. To continuously report the kNN result to the query initiator, we propose a localized scheme to monitor the updates of object locations in kNNs. The key idea of the localized scheme is to collect only the location updates that may potentially affect the kNN result at the query initiator. To this end, a *monitoring area* is setup for a continuous kNN query in the sensor network. It is defined as a circle centering at the query point q. The radius of the circle is set to $d(o_k, q) + r_s$, where $d(o_k, q)$ is the distance between the kth nearest object o_k and q, and r_s is the sensing range. Sensor nodes within the monitoring area will report all detected objects at each sampling interval to the query initiator. Upon receiving these reports, the query initiator reevaluates the kNN query. For simplicity, we assume the sampling interval is long enough for all necessary messaging to occur in order to complete the reevaluation of kNN query. On the other hand, the sensor nodes outside the monitoring area do not report their location updates.

This localized scheme for continuous kNN query processing requires the query initiator to centrally coordinate the maintaining of the monitoring area, collecting object updates from the monitoring area, and reevaluating the query results. In the first sampling interval, the query initiator evaluates one-shot kNN query, which is equivalent to a remainder kNN query for q with $r = 0$, using the scheme described in Section 3.2. The monitoring area is setup simultaneously during the query evaluation. Recall that during the query evaluation, i.e., preliminary and expanded search, all grid cells within or intersecting with the circle centering at q and with radius $d(o_k, q) + r_s$ are visited. When a grid cell is visited, the R-node will broadcast a probe message for data collection. Besides the location of the query point, the identity of the query initiator is also included in the probe message. To this end, the probe message is also an “include” notification message which helps setup the monitoring area. All sensor nodes in the grid cell can receive the probe message. They will start reporting the location updates to the query initiator from the following sampling interval.

At each subsequent sampling interval, the query initiator collects the locations of the objects detected and reported by the sensor nodes in the monitoring area. Objects beyond distance $d(o_k, q)$ are removed by the query initiator and only objects within distance $d(o_k, q)$ from the query point q are retained. We denote the number of these objects by k'. If $k' \geq k$, the query reevaluation is simply done at the query initiator. A set of new kNNs are derived by ordering the k' objects according to their distances to the query point and selecting the first k objects. If $k' < k$, we need to search $k - k'$ more objects outside the circle centered at q and with a radius of $d(o_k, q)$. This is equivalent to a remainder $(k - k')$NN query for q with $r = d(o_k, q)$. This query can be evaluated using the scheme described in Section 3.2.

4.2 Maintenance of the Monitoring Area

On computing the one-shot kNN result in the first sampling interval, the query initiator can set the radius of the monitoring area at $d(o_k, q) + r_s$, i.e., the distance between the kth nearest object and the query point plus the sensing range. The monitoring area is setup during the query evaluation.

At the subsequent sampling interval, the monitoring area may need to be updated due to the change in the kNN result. If the new kth nearest object is further away from the query point than the old one, the monitoring area should be expanded to include the new kth nearest object. That is, the radius of the new monitoring area is $d(o'_k, q) + r_s$, where $d(o'_k, q)$ is the distance of the new kth nearest object to the query point q. In case of monitoring area expansion, the difference between the new and old areas is a ring whose inner radius is the radius of the old monitoring area and whose outer radius is the radius of the new monitoring area. Sensor nodes in the ring are notified to report their location updates starting from the following sampling interval. Note that the monitoring area expansion only happens when the number of reported objects within distance $d(o_k, q)$ to q, i.e., k', is less than k. A new set of kNNs are derived through a remainder $(k-k')$NN query with $r = d(o_k, q)$. Since all grid cells within or intersecting with the ring are visited either in preliminary search or expanded search, it is guaranteed that all sensor nodes newly included in the monitoring area are notified.

On the other hand, if the new kth nearest object is closer to the query point than the old one, the monitoring area can be shrunk to reduce the number of sensor nodes reporting object locations to the query initiator. In this case, the difference between the new and old monitoring areas is a ring whose inner radius is the radius of the new monitoring area and whose outer radius is the radius of the old monitoring area. Sensor nodes in the ring need to be informed to stop reporting location updates to the query initiator. To do so, an “exclude” notification message is sent to all sensor nodes in the ring. The notification message takes the parameters of the query point q, the inner and outer radius of the ring and a flag (i.e., exclude) to indicate the action. The notification message is first sent to a sensor node within the ring. After reaching the sensor node in the ring, the message is flooded to all sensor nodes in the ring. When a sensor node within the ring receives the notification message for the first time, the sensor node further broadcasts the message to all its neighbors. Otherwise, if the sensor node receiving the message is outside the ring or it has already received the message before, the message is dropped. Sensor nodes receiving the message will stop reporting their location updates.

Since notification messages are sent to update the monitoring area, it incurs message overhead. There is in fact a tradeoff between the cost of updating the monitoring area and the saving in location updates from the sensor nodes in the area. We propose two methods to deal with the shrinking of the monitoring area. One *naive* method is to shrink the monitoring area to a minimum circle covering the new kNNs and their detecting sensor nodes whenever the new kth nearest object is found to have moved closer to q. The naive method is intuitive and it aggressively eliminates unnecessary location updates. However, it may lead to an expansion of the monitoring area soon after the kth nearest object moves further away from q again. As a result, the cost of updating the monitoring area may exceed the saving of location updates.

As an improvement to the naive method, we propose an *adaptive* method by considering the tradeoff between the saving of location updates and the cost of updating the monitoring area. Assume the radius of the monitoring area at current sampling interval is r_{old}. After updating the answer set, the kth nearest object moves nearer to q than the old one. Let $r_{new} = d(o'_k, q) + r_s$ be the radius of the new monitoring area if it is shrunk, where $d(o'_k, q)$ is the distance between the new kth nearest object and q. Let $\mathcal{S}$ be the set of sensor nodes in the ring defined by two circles with radii r_{old} and r_{new}. Let $\mathcal{O}$ be the set of objects detected and reported by the sensor nodes in $\mathcal{S}$. Note that the query initiator is aware of $\mathcal{O}$ by checking the location updates collected during current sampling interval. The expected saving of object location updates in the next sampling interval (in terms of message complexity) is $C_{saving} = \sum_{o_i \in \mathcal{O}} d(o_i, q)/r_{tx}$, where $d(o_i, q)$ is the distance between the object o_i and the query point q and r_{tx} is the communication range. The expected cost of updating (i.e., shrinking) the monitoring area consists of two parts: the cost of sending the "exclude" notification message from the query initiator to a sensor node within the ring; the cost of flooding the notification message to all sensor nodes in the ring. The expected cost of the first phase is r_{new}/r_{tx}. The expected cost of the second phase is approximated by the number of sensor nodes in the ring: $\pi(r_{old}^2 - r_{new}^2) \cdot f$, where f is the sensor node density and $\pi(r_{old}^2 - r_{new}^2)$ is the area of the ring. Therefore, the expected cost of updating the monitoring area is $C_{updating} = r_{new}/r_{tx} + \pi(r_{old}^2 - r_{new}^2) \cdot f$. In the adaptive method, if $C_{saving} \leq C_{updating}$, the new monitoring area is kept unchanged. Otherwise, if $C_{saving} > C_{updating}$, the monitoring area is shrunk.

5 Performance Evaluation

5.1 Experimental Setup

We simulated the sensor networks continuous kNN query processing using a simulator called J-Sim [1]. We simulated a connected sensor network geographically covering a $1000m \times 1000m$ area. A number of 2500 sensor nodes were deployed in the sensor network, implying that on average, there was one sensor node in an area of $400m^2$. The transmission and the sensing range for each sensor node were set at $35m$ and $28m$ respectively [5]. The maximum number of retransmissions at the MAC layer was set at 7. The sensor nodes were randomly deployed in the sensor network. On average, each sensor node can communicate directly with 9 neighbors. Since this paper focuses on query processing, we do not include the energy consumed in tracking objects. Only the

Table 1. Message Types and Sizes

type	size	type	size	type	size
query	>20 bytes	probe	16 bytes	query result	$k \cdot 24$ bytes
probe reply	24bytes	object update	24 bytes	monitoring area update	36 bytes

energy consumption of message exchanges was counted. The energy used to transmit and receive a bit was set at $2.64 \times 10^{-6}J$ and $1.58 \times 10^{-6}J$ respectively. The initial energy budget for each sensor node was set at $30J$. The default number of objects to be tracked was set at 250. The objects were initially placed in the network at random. The object movement follows a random walk model. In this model, the object repeatedly picks a random destination in the network and moves to the destination at a speed randomly chosen from a range $[0, V_{max}]$. V_{max} was set at $40m$ per time unit. The object locations were sampled by the sensor network at every time unit. In our experiments, we assume that the detecting sensor node of an object is the one closest to the object [23]. The interval of the GPSR beacon message is set at 1 time unit.

We compared the proposed localized scheme (naive and adaptive) with the centralized scheme. In the centralized scheme, we assume that there is a base station deployed at the centroid of the sensor network. All location updates are reported by the detecting sensor nodes to the base station at each sampling interval. Queries injected from any sensor node are routed to the base station for initial evaluation and are reevaluated at the base station continuously. If the new kNNs differ from the previous kNNs, the new result will be sent from the base station to the query initiator. Due to space limitation, we shall present only a set of most informative results. To evaluate the performance, we will show the average energy consumption in the sensor network as well as the message complexity. Table 1 shows the major message types and respective sizes.

5.2 Impact of the Number of NNs

In this section, we investigate the two methods for monitoring area update in the localized scheme, i.e., naive and adaptive methods. Figure 4 shows the average energy consumption of a single query for the localized and the centralized scheme with different k's. The average energy consumption is derived at the 65th time unit, i.e., the network lifetime under the centralized scheme with $k = 1$. From the figure we can see that for localized scheme, the adaptive method generally outperforms the naive method, especially for large k's. This is because the size of the monitoring area is generally larger with a larger k. The naive method simply shrinks the monitoring area whenever the kth nearest object moves closer to the query point. Thus, it incurs high volume of notification messages to update the monitoring area. On the other hand, the adaptive method considers the tradeoff between the cost of updating the monitoring area and the saving of object location updates, and shrinks the monitoring area only when it is beneficial. Table 2 presents the breakdown of the messages for the naive and adaptive method at the 65th time unit with $k = 8$. It shows that the naive method induces much higher number of messages in query processing (query message and probe message) and monitoring area update. Thus, we use the adaptive method throughout the remaining experiments.

Figure 4 also shows the impact of the number of NNs. With a given number of objects, the average energy consumption of the sensor nodes increases with k. This is

Table 2. Breakdown of Messages

Message	query	query result	probe	probe reply	object update	monitoring area update
k=8 (naive)	8813	525	1705	217	11957	224
k=8 (adaptive)	756	525	179	6	12297	13
Centralized (k=8)	6	443	0	0	258347	0

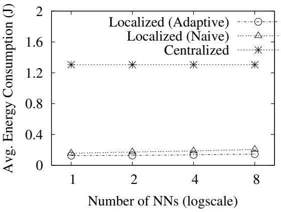


Fig. 4. Avg. Energy Consumption v.s. No. of NNs

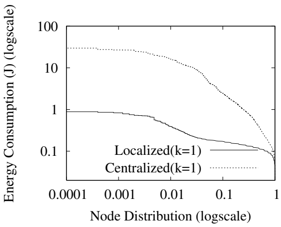


Fig. 5. Distribution of Energy Consumption v.s. No. of NNs

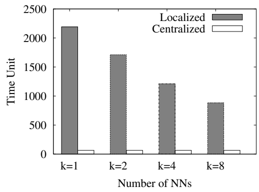


Fig. 6. Network Lifetime

because a larger monitoring area is established when more NNs are required. As a result, more location updates are sent to the query initiator at each sampling interval. Figure 4 also compares the average energy consumption for the localized and centralized schemes with different k's. The energy consumption is derived at the 65th time unit, i.e., the network lifetime under the centralized scheme with $k = 1$. We show only the performance of the centralized scheme with $k = 1$ in Figure 4. For the localized scheme, the average energy consumption for $k = 1, 2, 4, 8$ are given. Compared to the centralized scheme, the average energy consumption of the localized scheme is much smaller. This is because in the localized scheme, only relevant object location updates were reported to the query initiator, while in the centralized scheme, all location updates were sent to the base station. The number of sensor nodes involved in sending the updates was much smaller in the localized scheme than the centralized scheme which led to a lower average energy consumption. To gain more insight of how the localized scheme improves, we also present the breakdown of messages for the centralized scheme at the 65th time unit with $k = 8$ in Table 2. For the centralized scheme, the number of object updates was 20 times more than that of the localized scheme. For the localized scheme, the message overhead of query processing and monitoring area update was much smaller than the messages for object updates in the centralized scheme.

Figure 5 shows the energy distribution for the localized and centralized schemes with $k = 1$. The energy consumption is derived at the 65th time unit, i.e., the network lifetime under the centralized scheme with $k = 1$. A point (x, y) on the curve means that a fraction x of all sensor nodes consume more than yJ energy each. For the centralized scheme, among all sensor nodes, the top 1 percentile energy consumption is $16.3J$ which is 7 times higher than the top 10 percentile energy consumption (i.e., $2.3J$). For the localized scheme, the top 1 percentile energy consumption is $0.393J$ which is only 2.3 times higher than the top 10 percentile energy consumption (i.e., $0.17J$). This implies that the energy consumption in the centralized scheme is highly unbalanced compared to the localized scheme and thus, leads to a short network lifetime. Figure 6

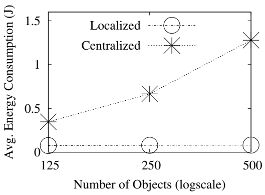


Fig. 7. Avg. Energy Consumption v.s. No. of Objects

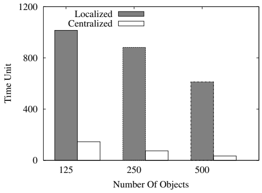


Fig. 8. Network Lifetime

shows the network lifetimes for both schemes. It shows that the localized scheme extends the network lifetime by 33 times over the centralized scheme when $k = 1$ and 13 times when $k = 8$. We also tested different numbers of queries in the experiments. The results, not included here due to space limitation, showed that the localized scheme can support up to 40 queries under the same network lifetime as the centralized scheme.

5.3 Impact of Number of Objects

Figure 7 compares the localized and centralized schemes with different numbers of objects when $k = 8$. The energy consumption is derived at the 34th time unit, i.e., the network lifetime under the centralized scheme with 500 objects. It is seen that the average energy consumption of the localized scheme is much smaller than that of the centralized scheme due to fewer location updates. The performance for the centralized scheme degrades rapidly with increasing number of objects due to the fact that more location updates are sent to the base station with more objects in the network. Figure 8 shows that when the number of objects increases to 500, the localized scheme extends the network lifetime by 18 times over the centralized scheme.

6 Conclusion

In this paper, we have proposed a localized scheme for continuous monitoring of kNN queries in wireless sensor networks. To avoid sending all the object location updates to a centralized repository, we store object locations locally at the detecting sensor nodes and monitor the queries in a localized manner. We setup a monitoring area for each query. Only the updates from the monitoring area are sent to the query initiator. Experimental results show that the localized scheme achieves low energy consumption than the centralized scheme over a wide range of system settings. Meanwhile, the energy consumption is more balanced among the sensor nodes in the localized scheme and therefore, the network lifetime is prolonged.

References

1. J-sim homepage. http://www.j-sim.org.
2. J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus. Tracking a moving object with a binary sensor network. In *Proceedings of Sensys'03*.

3. Hoffmann-Wellenhof B, Lichtenegger H, and Collins J. GPS theory and practice. *Springer-Verlag, New York*, 1997.
4. B. Gedik and L. Liu. Mobieyes: Distributed processing of continuously moving queries on moving objects in a mobile system. In *Proceedings of EDBT'04*.
5. C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of MobiCom 2004*, 2004.
6. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of Mobicom'00*, 2000.
7. B. Karp and H.T. Kung. GPSR: Greey perimeter stateless routing for wireless networks. In *Proceedings of Mobicom'00*, 2000.
8. S. Madden, M.J. Franklin, J.M. Hellestein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of OSDI'02*, 2002.
9. A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM Workshop on Sensor Networks and Applications*, pages 88–97, 2002.
10. K. Mouratidis, M. Hadjieleftheriou, and D. Papadias. Conceptual partitioning: An efficient method for continous nearest neighbor monitoring. In *Proceedings SIGMOD'05*.
11. K. Mouratidis, D. Papadias, S. Bakiras, and Y. Tao. A threshold-based algorithm for continuous monitoring of k nearest neighbors. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(11):1451–1464, 2005.
12. D. Niculescu and B. Nathi. Ad hoc positioning system. In *Proceedings of INFOCOM'03*, 2003.
13. G. Pottie. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.
14. S. Prabhakar, Y. Xia, D. Kalashnikov, W. Aref, and S. Hambrusch. Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. *IEEE Transactions on Computers*, 51(10):1124–1140, 2002.
15. S. Ratnasamy, B. Karp, L.Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: A geographic hash table for data-centric storage. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
16. M. Wu, J. Xu, X. Tang, and W.-C. Lee. Monitoring top-k query in wireless sensor networks. In *Proceedings of ICDE'06*.
17. X. Xiong, M. Mokbel, and W. Aref. SEA-CNN: Scalable incremental processing of continuous queries in spatio-temporal databases. In *Proceedings of ICDE'05*.
18. J. Xu, X. Tang, and W.-C. Lee. EASE: An energy-efficient in-network storage scheme for object tracking in sensor networks. In *Proceedings of IEEE SECON'05*, 2005.
19. Y. Xu, W.-C. Lee, J. Xu, and G. Mitchell. Processing window queries in wireless sensor networks. In *Proceedings of ICDE'06*, 2006.
20. W.-C. Lee Y. Xu, J. Winter. Prediction-based strategies for energy saving in object tracking sensor networks. In *Proceedings of MDM'04*, 2004.
21. Y. Yao, X. Tang, and E-P. Lim. In-network processing of nearest neighbor queries for wireless sensor networks. In *Proceedings of DASFAA'06*, 2006.
22. X. Yu, K. Pu, and N. Koudas. Monitoring k-nearest neighbor queries over moving objects. In *Proceedings of ICDE'05*.
23. W. Zhang and G. Cao. Optimizing tree reconfiguration for mobile target tracking in sensor networks. In *Proceedings of INFOCOM'04*, 2004.

History-Sensitive Based Approach to Optimizing Top-k Queries in Sensor Networks

Qunhua Pan, Minglu Li, and Min-You Wu

Department of Computer Science and Engineering, Shanghai JiaoTong University
1954 Huashan Road, Shanghai, China
{oct-pan, li-ml, wu-my}@cs.sjtu.edu.cn

Abstract. Sensor networks generate a large amount of data during monitoring process. These data must be sparingly exacted to conserve energy. There are two methods to obtain data: “push” and “pull”. When the sensory data satisfied a preset condition, they are “push”ed towards the base station. The “pull” method is to actively query the sensor networks for any interesting sensory data. The problem is how to plan the query and save the energy. When a query has been executed, there are some hints that can be kept to optimize the subsequent query processing. Energy consumption can be reduced by not contacting nodes whose values either can be predicted or are unlikely to be used. In this paper, we propose a history-sensitive based method to optimize top-k query processing in sensor networks. The top-k query looks for and utilizes the historical data in each sensor node. Subsequent top-k queries are guided by these historical data, therefore, to improve the entire query process. Simulation results show that the number of query hops can be reduced and the delays in response are improved.

Keywords: Query processing, Wireless sensor networks, Top-*k*, Historical data.

1 Introduction

Technology advances in wireless sensor networks have opened up new opportunities for collecting data from all sorts of environments. The task of effectively and efficiently querying these networks is an important and challenging problem. Because sensors are often battery powered, the lifetime of the network is tied to the rate at which it consumes energy. In particular, radio communication is a primary source of energy consumption in sensor networks. Hence, minimizing communication in query execution can save a significant amount of energy and prolong the lifetime of the network. Moreover, because of the extreme limited resource of sensors, the corresponding protocols and algorithms should be carefully designed.

Sensor networks generate a large amount of data during monitoring process. These data must be sparingly exacted to conserve energy. There are two methods to obtain data: “push” and “pull”. When the sensory data satisfied a pre-established condition, a pushing event is trigged and data are pushed to the base station. In this situation, the sensors should continue on working if needed unless their energy power exhausts. The “pull” method is to actively query the sensor networks for any interesting sensory data. The sensors can be in their sleeping or idle status when no query is requested.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 674 – 684, 2006.

Once they received a query, they will wake up, sense the environment and send the satisfied sensory data back. So, the query process is more flexible and efficient. The problem is how to plan the query to save energy.

Consider an example of temperature monitoring in a large exhibition. There are many exhibition rooms. To automatically monitor and adjust the temperature of the rooms, it needs to collect the real-time temperature of each room. Temperature sensors are deployed in all rooms. These sensors are self-organized into a wireless sensor network. The sensory data are sent back to the control room, where a manager can monitor the temperature of each room and particularly will be interested in knowing which room having the highest or lowest temperature. The controller may run a top-*k* query over the network to find out the target rooms in order to adjust their air conditioners.

Optimization of top-k sensor queries is significantly more complex than one for ordinary queries (e.g., return all readings greater than x). Top-*k* query must know all the sensory data in the sensor network. Flooding the query to the entire sensor network can obtain all information, but it will consume too much energy. Every sensor is visited no matter whether its data will be useful or not.

We propose a new approach which combines the push and pull methods. A base query specifies a time period for sensing and aggregating minimal data to continuously monitor the field. That is to simulate a push method but with minimal possible energy consumption. The important function of the base query is that it provides minimal but necessary information so that the later queries can be optimized. Other queries use the pull method to actively request information from the sensing field. In addition, the results from previous queries can be cached in sensor nodes for the subsequent queries. The cache can also include the information about data distribution. These hints can guide the subsequence queries to the right region.

Based on this idea, we analyze the features of top-*k* query dissemination and data aggregation. We optimize top-*k* query processing based on historical data in sensor networks. Our contributions are as following:

1. Based on historical data, we have developed a query optimization framework, for query dissemination and data aggregation in sensor networks. Useful information is extracted from the aggregated data. Historical queries are cached in sensors. A new incoming query checks if there is any matched historical query in the cached query table. If so, the query processing is stopped and the cached data result will be sent back. Otherwise, the query agent will find if some data with the tags can satisfy the query. Finally, the query will be sent to next sensor nodes by the routing schema which is also generated from the aggregated data. Numbers of query hops will be reduced in this framework. It is energy efficient. Moreover, the response time will be substantially improved.
2. We apply the above concepts and techniques to the top-k query in sensor networks. Top-*k* query processing always defines a threshold on the historical aggregated data. When the sequent query comes to one sensor, it will be forwarded when there are aggregated data that are higher than the threshold. However, the threshold is determined by the subjectivity of the user, and it is application aware. We optimize the top-k query without define a threshold. By following the algorithms of data aggregation and decision rules which judge if

the query should be forwarded to children nodes, queried nodes are pruned. We evaluate these algorithms using simulation.

The rest of this paper is organized as follows: Section 2 will provide an overview of related work in top-*k* query processing in traditional database management systems and current query processing in sensor networks. Section 3 will present the framework of our historical data based top-*k* processing. The algorithm will be discussed in Section 4, followed by its performance evaluation in Section 5. We will conclude with an outlook on open research problems in Section 6.

2 Related Works

A substantial amount of work has been done on querying sensor networks. Fjords [3] is a proposed architecture for managing multiple queries over many sensors to allow users to pose queries that combine streaming, push-based sensor sources with traditional pull-based sources. It also proposed power-sensitive Fjord operators called sensor proxies which serve as mediators between the query processing environment and the physical sensors. Reference [4] discussed the aspects of an acquisitional query language, introduced event and lifetime clauses to control when and how often sampling occurs. It discussed query optimization with the associated issues of modeling sampling costs and ordering of sampling operators. And it showed how event-based queries can be rewritten as joins between streams of events and sensor samples. This paper also demonstrated the use of semantic routing trees as a mechanism for efficiently disseminating queries. Query processing in sensor networks concerns with routing tree [5][6], aggregation[7][8] and semantics[9].

In the traditional database technology, the top-*k* query problem has been intensively studied. Reference [10] studied the advantages and limitations of processing a top-*k* query by translating it into a single range query that can efficiently processed by a traditional relational database management system. It studied how to determine a range query to evaluate a top-*k* query by exploiting the statistics available to a RDBMS. Donjerkovic and Ramakrishnan [11] proposed a probabilistic approach to query optimization for returning the top-*k* tuples for a given query. Chen and Ling [12] used sampling to define the range selection query that is expected to cover most of the top-*k* tuples. The result of the selection query serves as an approximate answer to the original top-*k* query.

The processing of top-*k* query in sensor networks is different from one in a traditional relation database. Deshpande et al. [2] proposes *model-driven data acquisition*, which suggests using models such as multivariate Gaussians to predict sensor readings. These models let us avoid visiting nodes whose readings can be accurately predicted or are unlikely to contribute to the final result. This approach can dramatically reduce the energy consumed by the network, but of course makes results approximate. Instead of using models explicitly, Silberstein et. al [13] proposes to use samples of past sensor readings. The samples are computationally efficient to use in query optimization. It demonstrates the power and flexibility of sampling-based approach by developing a series of top-*k* query planning algorithms with linear programming. Zeinalipour [15] presents the Threshold Join Algorithm (TJA), an efficient top-*k* query processing algorithm for distributed sensor networks. TJA uses a non-uniform threshold on the queried attribute in order to minimize the number of tuples that have to be transferred towards the querying node.

Because the data management systems in sensors, such as TinyDB and Courgar, are weak and the main target is the energy saving when querying the sensor networks, top-*k* query technologies on relation database can not be used in sensor networks. Prior works of top-*k* query optimization in sensor networks using the threshold based on history data. The threshold definition is application-dependent. However, the result accuracy using sampling-based approach cannot always be guaranteed. The processing of top-*k* query in our framework is distributed and its optimization is independent of the threshold. The communication cost and the delays will be low.

3 History-Sensitive Based Top-k Query Processing

The main goal of designing this querying system is to minimize the energy consumption of the system, that is, minimize the number of messages as well the size of the messages in the system including the number of queries and the number of data messages. In the application scenarios we described, detailed monitoring is only necessary for a subset of the data having corresponding numeric attributes whose values are among the *k* largest, where *k* is an application-dependent parameter. Therefore, the transmission, storage, and processing burdens in the monitoring infrastructure can be reduced by limiting the scope of detailed monitoring accordingly. Because of the extremely limited resource of a sensor node, purely flooding top-*k* queries to the sensor network to collect all sensory data is often unnecessary. A low-cost mechanism is needed for continually identifying the top-*k* data values in a sensor network. When a query has been executed, many sensory data are generated. They are sent back and aggregated in intermediate sensors. If the historical data still reside in the sensors that are near the base station. A repeated query can respond immediately if the result is still available. Moreover, the semantic meaning in resultant data can be used to guide the subsequent new queries. Based on this point of view, we can optimize the top-*k* query by utilizing the historical data.

3.1 Framework of Historical Data Based Query Processing

The framework illustrated in Figure 1 consists of several components: *query agent, cached historical data, cached queries, and routing schema.* When a new query arrives at a node, the cached queries table is to be checked to see if the query had been executed. If so, the query will not be forwarded. The query result can be extracted from the aggregated data. Even there is no matched query in the cache, the query result could also be generated from the aggregated data. If all these operations cannot satisfy the query, it will be forwarded to the next nodes through routing schema. Because there is information of data distribution in the aggregated data, the selection of next nodes also depends on the historical data.

This query framework consists of a base query and subsequent normal queries for an application. A *base query* is injected to the sensor network before the normal queries. The subsequent normal queries may further explore the field for details. They may utilize the data cached in the sensor nodes to minimize energy

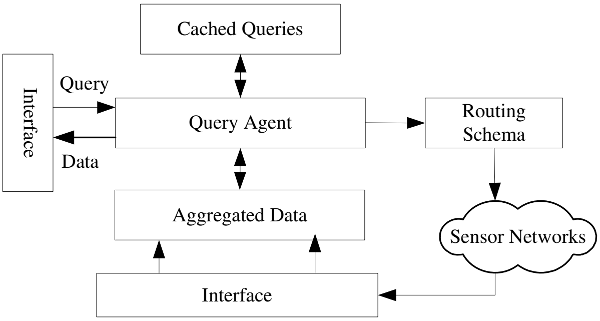


Fig. 1. Framework for History-based query processing

consumption. Historical data may provide guide so that the query will head to the most likely location to search for the top-*k* values.

In this paper, we assume that a global time synchronization mechanism has been implemented so all operations can be executed synchronously. Also, a fault-tolerance mechanism is implemented to handle possible failure of sensor nodes. Although dynamic change of the status of the field could be handled in this framework, we will focus on a relative simple problem in this paper, that is, how to obtain the top-*k* values between two consecutive query readings. The base query will be discussed below.

3.2 Base Query

Because of the variance of the monitoring environment, the base query requests all sensors in the sensor network to periodically sense the field. The principle of designing the base query is to reduce its energy consumption while required information of the field is continuously monitored. Normally, a base query can be an aggregation operation performed every time period of P.

An example can be a sensor network that monitors the forest fire. A base query may be designed as follows. A sensor in the field should wake up every P = 10 minutes, sense and transmit the temperature to its parent. The aggregation operation is to extract the maximum of values from its children as well as itself. The base station will receive information of the highest temperature in the field. To further minimize energy consumption, a threshold can be set so that only temperature higher than the threshold will be transmitted. Thus, when there is no fire in the field, the activity and energy consumption is negligible while the field is still monitored continuously. Once a potential fire is detected, normal queries could be generated to understand the degree and ranges of the fire. As an example, a top-*k* query may be sent to find out the most severe part of the fire.

In summary, a base query is designed as follows. First, it is a periodical query, that is, each node that executes the query will periodically execute the operation specified in the query with the specified period until another query explicitly cancels the operation. Second, it is a query that is sent to every node through a broadcast tree, that is, the query floods to the entire sensor network. Third, it is for surveillance purpose so it must be energy-efficient. To this extend, the length of the period must be carefully selected and should be application-aware. It must be frequent enough so the field is sufficiently monitored and must be longer enough to ensure long lifetime of the sensor network. Furthermore, the message sent to the base station must be minimized. The smallest possible message is sent upward to the base station, and normally, aggregation is performed instead of collecting all messages to minimize the size of the message. In the above example, aggregation of maximum of temperature values from every node is performed to monitor the fire in the field so that only N-1 messages with a single value is transmitted every time period. Finally, messages sent from a child to its parent are cached in the parent node. These cached data will provide a guide to subsequent queries.

The base query has its fundamental importance in the entire design of the querying system. It provides a basic surveillance mechanism with low-energy consumption which ensures long lifetime of the sensor network while continually monitoring is provided in the field. In addition, it provides a substrate of information for subsequent queries. Furthermore, as the query instructs every node to periodical sense the field and returns its reading, the sensor network keep monitor the dynamic changing of the global state of the field.

4 Algorithms for Top-k Query in Sensor Networks

We now describe our algorithm for historical data based top-*k* monitoring. The network initialization includes the query routing tree establishment. The root node broadcast a hello message to the sensor network. When node received the message forwarded by other node, it will add to the node as its children node. Then this children node will forward the message to its neighbor. After all the nodes have received the message, the initial routing tree is established. Because the nodes always received the message from the nearest node, the routing tree is a nearest-first tree.

After the base query is flooding in the sensor network by the routing of nearest-firs tree, responded nodes will send data back to their parent nodes. The query proceeding computes the answer bottom-up in one pass over the network. Each node simply collects the top-*k* values from each of its children, selects the top-*k from* all such values and its own, and passes them on to its parent. If the subtree rooted at a node has fewer than *k* nodes, then all values from the subtree are passed up to the node. Each parent node waits until it receives gathered data from all its children nodes, apply an aggregation operator on it and send the result to its parent. But since every node must be visited in order to guarantee an exact answer, the query hops are quite large. The base query procedure is shown as follows:

```
input : top-k query, N // N is the current query node.
output : result of top-k data

N.broadcast(query k)
Begin
  If (N has no child node) then
     N.visited = true
     NodeResult = localDataValue
     return (NodeResult)
  If each of my children node Ni.visited = true then
```

```
    AggregatedResult=MergeResult(localDataValue,NodeResult$_1$,…,NodeResult$_n$)
    NodeResult = FindtopK(AggregatedResult,k)
    N.visited = true
    return (NodeResult)
  else
    For Each of my children Ni.visited = false Do
    Begin
      NodeResulti = Ni.broadcastTopKQuery(query,k)
    End
End
```

Comments for the objects and methods in the above algorithm:

NodeResult: stored data-value, nodeID.

MergeResult: aggregate localDataValue with NodeResult$_i$ from its children nodes.

FindtopK: select top-*k* data as NodeResult, NodeResult will be sent to parent node.

After the base top-*k* query, there are most |Ni|*K data in each node N, {Ni} is the number of one level child nodes of N. But at least $(\jmath Ni\jmath$ - 1)$*k$ of them will not be in the final result, representing a significant waste of bandwidth. The utility of these historical data to guide the sequent top-*k'* queries can reduce the redundant data transmission. When the root node receives the sequent top-*k'*, recall from section 3, the k' is assigned {ki} to each children nodes. The child node i will execute top-*ki* query in its subtree. The algorithm when k' is less than k can be described as follows.

```
input : top-$k'$ query, N
output : result of top-k'

Initialization: N = root Node
N.broadcastTopKQuery(query,k')
Begin
  Sort OverallResult
  Find Top k' from OverallResult
  Find MinValue of Top k'
  Find {K$_i$}  //K$_i$ is the number of data sent by child node$_i$ and sum(k$_i$) = k'
  If K$_i$ = 1 AND top 1 data value = localDataValue then
    N.visited = true
    NodeResult = localDataValue
    return (NodeResult)
    If each of my children node Ni.visited = true then
      AggregatedResult=MergeResult(localDataValue,NodeResult$_1$,…,NodeResult$_n$)
      NodeResult = FindtopK(AggregatedResult,k$_i$)
      N.visited = true
      return (NodeResult)
  else
      For each of my child Ni do
      Begin
```

```
            NodeResult$_i$ = Ni. broadcastTopKQuery(query,k$_i$)
        End
End
```

When k' is larger than k, the k assignment algorithm is not suitable. Because when the k data are all come from one node i, the (k+1)th data in the root are from other nodes j. If we still forward the query to node j to find the (k+1)th data. The true data will hide in subtree of the node i. So, we can first find the (k+1)th data from the node i, then we compare it with the (k+1)th data in the root node. If the new received data is larger, the true top (k+1)th data is found, or the true top (k+1)th data is from the other node, such as node j. The top (k+2)th data can be also found by this way. The shortage is that there is long delay especially when the k' >> k. But it is energy efficient when k' is not very larger than k. The algorithm is as follows:

```
input : top-k′ query, N.
output : result of top-k′.
Begin
  Initialization: N = root Node
  Do until k = k′
    Find Nodei // Nodei is the node who sends the k-th top data value
    Ni.broadcastTopKQuery(query,k+1)
    If  > historical (k+1)th data in N then
    Updata (k+1)th data in N by (k+1)th data in NodeResult(N$_i$)
    k = k+1
  End do
 Return top-k′ data
End
```

5 Simulation and Results

In this section we report the simulation results to evaluate the effectiveness and efficiency of the algorithms in top-*k* query processing.

5.1 Data and Simulation Setup

We run our simulation on a 20×20 grid-topology network and there is one sensor in each grid. There are two different data distribution in our simulation. One is circle distribution and another is power law.

The circle distribution of data means there is a circle region in which the data values are higher than other region. The top-k query will be satisfied by the sensors in the region if it is efficiently guided. The transmission radius of sensor node is set 3 grid distance. The root node is located in left corner of the sensor network.

It has been observed that for several self-organizing networks the degree distribution follows a power law (or, equivalently, scaling) distribution of the form $P(k) \sim k^{-\alpha}$. In the simulation, the degree (k) of a sensory data is defined as the location of the sensor node. A higher degree exponent means the distribution goes to zero faster, i.e., there are very few nodes that have very high degrees. On the other hand, if the exponent is smaller, there are a relatively higher number of nodes with very high degrees. The power law distribution is normal in sensor network application.

A user-query is generated by a user who queries the sensor network for data. The user-query is started at the root node. Firstly, the base query is flooding to the whole sensor network. The nodes responded the query will send data to its parent node. The result data are aggregated and cached in intermediate nodes. When the sequence query arrived to one node, it will be determined whether to be sent to its children judged by the cached data. Here, the *query-hop* is used to measure the energy consumption. The number of query-hop means how many nodes will be visited in the sequence query.

5.2 Simulation Result

We first evaluate the performance the historical data based top k query when consequent k' < k. The base query in our simulation is to find top 5 data in the sensor network. After the base query, the sequence queries are top 4, … , top 1. In base query, each node is visited, the hops is 399. The result shown in Figure 2 plots that the hops of sequence query will reduce after the base query.

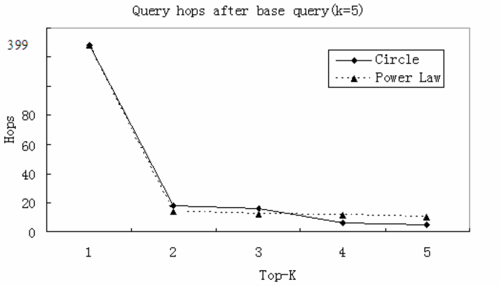


Fig. 2. Hops in static data distribution and k' < k

The optimized top-k query limits the scope of the region. So the visited nodes reduced, consequently, the hops of top k query will decrease. The delay is defined as the farthest visited node from the root node. Figure 3 shows the delays of top k' query (k' < k) in static data distribution. It is evidence that the delays are reduced by the hint of the base query.

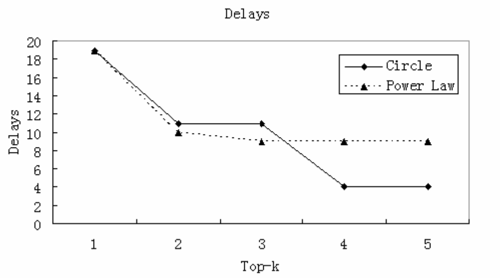


Fig. 3. Delays in static data distribution and k' < k

In Figure 4, we calculate the delays when the sequent query k' is larger than the k of base query. Here, the base query is to find maximum data of the sensor network. We increase the number of required top data by step 1 in the algorithm. The hops increase when the k' is increased. If the k' is not very large, the total hops will be less than the flooding query.

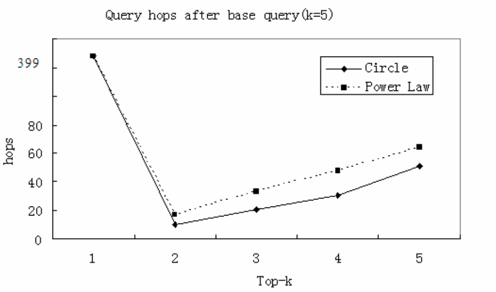


Fig. 4. Hops in static data distribution and k' > k

The delays will increase when the k' > k. Figure 5 shows the result of delays in each top k' query.

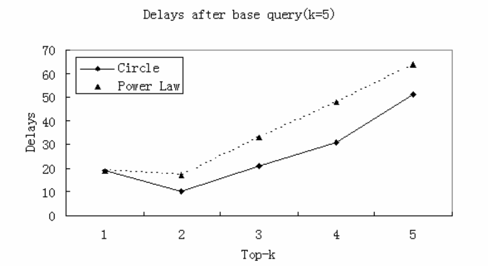


Fig. 5. Delays in static data distribution and k' > k

When the data center is moving or the data value varies, the historical data guided query can't get the accurate data. If the varying rate is slow, when the query reaches the stop node, it can extend one or more steps to query the neighbor nodes. This can expand the searching scope and get more accurate data. However, the direction and the rate of data center moving is application aware. If the query routing tree is static, it can't adapt to the data variance.

6 Conclusion and Outlook

Sensor network is data centric. This paper proposes a historical data based framework of query processing. The query result will be aggregated in intermediate nodes and then sent back, which is to conserve energy. However, these aggregated data will reside in the nodes for a period time. The sequent queries will be guided by these historical data. We apply this framework to the top-*k* query problem. There are base top-*k* query and sequent top-*k'* queries in the application. By the guide of the result data of base query, the top-*k'* query will get the target quickly and minimum the query hops from root to the target. Simulation results show the historical data base top-*k* query processing can save the energy and the delays are reduce when k' is less than k. When k' is larger than k, the query hops are reduced but the delays increased.

The problem of data storage in node is not discussed in this paper. Because the memory of a sensor is limited, an efficient compression algorithm should be designed in our future works. The query routing tree is very important to the framework we proposed, how to create an efficient query routing tree is our another direction. The design of query plan in dynamic environment is in our future work.

References

1. B. Babcock and C. Olston. Distributed top-k monitoring. In Proc. of the 2003 ACM SIGMOD Intl. Conf. on Management of Data, San Diego, California, USA, June 2003.
2. A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In Proc. of VLDB2004.
3. S. Madden and M. J. Franklin, "Fjording the stream: An architechture for queries over streaming sensor data", In Proc. Of ICDE2002.
4. S. Madden, M. Franklin, J. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In Proc. of ACM SIGMOD2003.
5. Jeffrey E. Wieselthier, Gam D. Nguyen, A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," In Proc. of IEEE INFOCOM 2000.
6. H. Yang, F. Ye and B. Sikdar, "A Dynamic Query-tree Energy Balancing Protocol for Sensor Networks", In Proc. of WCNC2002.
7. S. Madden, R. Szewczyk, Michael J. Franklin and David Culler. "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks", Workshop on Mobile Computing and Systems Applications, 2002.
8. W. Yu, T.Nam Le, Dong. Xuan, and W. Zhao," Query Aggregation for Providing Efficient Data Services in Sensor Networks", in Proc. of IEEE Mobile Sensor and Ad-hoc and Sensor Systems (MASS), October 2004.
9. Qunhua Pan, Minglu Li, Min-You Wu, "A semantic-based architecture for sensor networks", Annals of telecommunications, Vol.60 n°7-8, July-August 2005. pp.928-943.
10. N. Bruno, S. Chaudhurl, L. Gravano, "Top-k Selection Queries over Relational Databases: Mapping Strategies and Performance Evaluation", ACM Transactions on Database Systems, Vol. 27, No. 2, June 2002, Pages 153-187.
11. Donjerkovic, D, Ramakrishnan. R, "Probabilistic optimization of top N queries", In Proc. of VLDB'99.
12. 12 C. Chen, and Y. Ling. "A sampling-based estimator for top-k selection query", In Proc. Of ICDE2002.
13. A. Silberstein, R. Braynard, C. Ellis, K. Munagala, " A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks" , In Proc.of ICDE 2006.
14. Christopher R. Palmer, J. Gregory Steffan, "Generating network topologies that obey power law", In: Proc. of the IEEE GLOBECOM, San Francisco, 2000, pp.434–438.
15. D. ZeinalipourYazti, Z. Vagena, D. Gunopulos, V. Kalogeraki, V. Tsotras, "The Threshold Join Algorithm for Top-k Queries in Distributed Sensor Networks", in Proc. of DMSN'05, August 29, 2005, Trondheim, Norway.

Data Replication in Mobile Ad Hoc Networks

Samira Moussaoui, Mohamed. Guerroumi, and Nadjib Badache

LSI, USTHB University, BP 32 El Alia,
16000, BabEzzouar, Algiers, Algeria
moussaoui_samira@yahoo.fr, guerroumi@gmail.com,
Nbadache@wissal.dz

Abstract. Data replication is suitable to improve the response time, the global traffic, and the sharing of data since even in the case of disconnection of a server. The nodes can continue to have access to replicas of data. On an Ad hoc mobile network, the frequent partition of the network and the lack of fixed infrastructures complicate the data access and the sharing task.

In this paper, we propose a method of data replication in a mobile ad hoc network. The method is composed of two main phases. The first phase aims at creating replicas from new data in the network and at realizing the first distribution of these replicas. The second phase is devoted to the redistribution of replicas in order to overcome the impact of dynamic changes of topology and to satisfy the evolution of users' needs.

Keywords: Replication, Ad hoc mobile network, data availability, access.

1 Introduction

Replication allows better data sharing. It is a key approach for achieving high availability. It is suitable to improve the response time of the access requests, to distribute the load of processing of these requests on several servers and to avoid the overload of the routes of communication to a unique server. Since the access is generally carried out on the nearest replica of data, the global traffic is decreased.

The creation of replicas also allows to better data sharing since even in the case of disconnection of a node holder of data, the other nodes can continue to have access to a replica of data on another node.

On the static networks, connections failures are frequent and take part in the normal working of the mobile environment. The fragility of links is due from dynamic variations of the network topology. In mobile networks with infrastructure, static nodes may be useful for data replication. In mobile ad hoc networks, the absence of any infrastructure complicates replication task. Thus, it is possible to count only, on mobiles nodes which constitute the network. Since the links are less and less sure, these nodes have a limited storage capacity and limited energy.

The method proposed in this paper takes into considerations the characteristics of the mobile environments and it attempts to use at best its capacities. The method presents two main phases. The first phase aimed at creating replicas of data newly arrived or created on network, at realizing replicas of data and scatter them on the

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 685 – 697, 2006.

network in a global way. This distribution has two objectives. One objective is used for inform all the network of the existence of data (dissemination). Another objective is to distribute the replicas uniformly on the network in order to respond at best to the users' demands in which the interest degree to these data for each of the nodes has not been known yet. The second phase is a strategy of redistribution of replicas in order to remedy the dynamic changes of topology and to better respond to the evolution of users' needs.

2 Related Works

Several strategies of replication have been proposed for the static environments [5], [6], [13] where the access failures are not frequent and where resources are more important. These strategies improve the accessibility and reduce the load around one or several servers.

Other solutions of replication are proposed for the mobile environment [1], [10]. These latter usually suppose the existence of fixed servers. This hypothesis is constraining for autonomy of the Ad Hoc Mobile Network.

T. Hara proposed in [7] three methods to assign the replicas to mobile nodes in a mobile ad hoc network in order to improve the availability of shared data. The methods are founded on the hypothesis that all the nodes of the network know the access probabilities to data and these probabilities do not change. These methods consider accessed data only in reading. In the method SAF, several replicas of the same elementary data can be found on nodes neighbours. Thus, there are other data which are not assigned due to space storage lack. The method DAFN tries to remedy this redundancy. The algorithm of replication considers the frequencies of neighbours in order to avoid redundancies. The third method DCG manages the data within a group by taking into account the access frequencies of nodes of a group and limits the redundancy within the group. In this method, each node must periodically broadcast its access characteristics on the network. Consequently, we have a load in traffic besides the management traffic of groups. In spite of all these improvements, the problem of redundant replicas between the nodes neighbours is still asked.

In [9], T. Hara extends the three methods proposed in [7] assuming that the elementary data are periodically updated. In [10], T. Hara and S. Kumar proposed an extension of these methods by assuming that the updates are to be carried out in an arbitrary manner. In [12], T. Hara and others proposed a new method (DCG-S1), derived from the one proposed in [7]. In this method, the authors used the notion of stability of links, for constructing stable groups. This technique of replication allows to have a better availability of data. But the traffic produced remains high.

Yan and Cao [14] proposed two methods of replication in order to improve the access to data. The data are maintained by fixed servers and mobile nodes can create replicas for these servers. The first method CacheData allows the intermediary nodes between the access caller and the server to create replicas for serving the future demands. The second method CachPath allows the mobile nodes to store Paths towards the data and to use them for reorienting the future demands.

The proposed method distinguishes itself from prior by using a primary replication for information dissemination and a dynamic replication. The information dissemination increases the accessibility. Then the first phase aims at creating replicas from new data in the network. The second phase is devoted to dynamic replication of data in order to over come the mobility and to satisfy the evolution of users' needs. The frequency of access to each data and the number of hops between the user and the accessed replicas was considered.

3 Environment Model and Hypotheses

In our environment we consider a mobile ad hoc network, where each mobile node can cooperate to the construction of a common cache, by the sharing of its own storage space with the other nodes. A mobile user can create replicas and maintains them locally. As It can generate new data (original data) and share them with the other users. It can also put in cache the access paths to data ("PathData") which allow a quicker access for distant data. In this environment, a mobile node can eventually receive data from a static network in case of connection. We consider in this first phase of our work, the data only accessed in reading and not updated. We suppose:

- Each mobile node is designed by a unique identifier N_i; such that $1 \leq i \leq NT$, where NT is the total number of the mobile nodes in the mobile ad hoc network.
- Each data created by a mobile node is associated to a unique identifier D_{ij} where i is the identifier of the node which created these data and j is a sequence number. The data received from static network keep their original identifier and they mustn't be codified in this way.
- Each mobile node has a determined memory space, to maintain replicas locally, the original data and the access paths data.
- A mobile node periodically informs its neighbors of the local data access characteristics.

For each user of network, each accessed data is characterised by two types of frequency:

- A frequency of external access which represents the access rate of the user to external data which means outside of his cache and,

- A frequency of internal access represents the access rate of the user and all other users to internal data in its local cache.

4 Method

We propose a distributed and decentralized algorithm for dynamic data replication in a mobile ad hoc network. The aim of this replication is to offer high availability of access to data on such environments where the capacities of storage are limited and the communication links prone to disconnections. This replication aims at maintaining the data available on the mobile network in a way to give to each node a probability of a higher access. The method is augmented with a protocol of access to improve the access performances. The algorithm presents two behaviours according to data newly created on the network or data already presented on the network. These behaviours correspond respectively to two algorithms:

• Primary replication algorithm: Given a shared data which presents a certain interest for the most of nodes on the network, the_algorithm duplicates and distributes uniformly the replicas. This phase is accomplished when creating new data by one of the nodes or when loading the first replica of data from a fixed server to which the ad hoc network could be connected. This distribution tries to ensure access to data for each node and a compromise between the used storage space and the access time if the data require a remote access. This phase consist on data dissemination and replication on the mobile ad hoc network. The evaluation of distances between the replicas is carried out in term of the number of hops. The replication method is based on this distance. For example to avoid redundancy, two immediate neighbors must not have replicas of the same data. If a node B has two immediate neighbors A and C and these are not neighbors, it is more interesting that B retrieve replicas of different data on A and C. This optimizes the global time response and it increases the accessibility. But, connections failures are possible. Then, in this phase, a data is replicated on nodes separated by three hops. If a failure connection occur between B and A, B can retrieve, the data of A, on another node H at two hops.

• Dynamic replication algorithm: The mobile nodes change the position and certain connections failures may occur while others appear. This algorithm improves the response times of the access demands to data by placing the data to the proximity (in number of hops) of nodes which often manipulate it. The evaluation of distances between the replicas is carried out in term of the number of hops. This replication is based on the frequencies access data by a node and also on the distance between a node and a holder node of accessed data. The most accessed data is replicated locally or near (on one, two or tree hops neighbours).

4.1 Primary Replication

This algorithm considers the existing nodes and current topology of network.

Principle: We use a hop counter initialized to zero by the node which created or received, at first, the data. This replica of data is called ***original*** replica. Then, the node initiate a diffusion to its neighbours of a message **PrimaryCreat(NodeAdd, D_k, HopCpt)** of primary creation of replicas of the data $\mathbf{D_k}$. Each time the message arrives at another node, the hop counter is incremented by one, or reinitialized to zero, if a replica is already exist on the visited node. The node, which receives this type of message, processes it as follows:

- If the counter is equal to three hops and if a replica of data exists neither in the cache of the node nor in one of the neighbours, like for example, the node N_9 in figure 1. After the creation of a replica, the node reinitializes the counter to zero and sends the message of creation to the nodes neighbours.

- If the counter is equal to three hops, and at the same time, it exist a replica of data in one of the neighbours of node. In this case, the node "receiver" initializes the hop counter to one and sends the messages of creation to the ***following nodes*** (the nodes which don't send messages **PrimaryCreat**) and which don't hold a replica of data.

For instance, in the figure 1, the node N_{10} receives at first the message **PrimaryCreat** of N_6 and increment the hop counter to 3. But N_9 being a node neighbour of N_{10} holding a replica of data, N_{10} doesn't create replica of the same data in order to limit the redundancies of replicas on neighbours. The message **PrimaryCreat** is then transmitted towards N_{12} and N_{11}.

- If the counter is equal to two hops, and it doesn't exist the following links (links besides those by which the message has been received) thus, the node is chosen, to hold the replica as for the case for the node N_{18}.

A node which receives for the second time the message of primary creation of data must ignore it. When placing primary replicas, if there is a lack of space, the local data which have a low access frequency are deleted. The dynamic management of these replicas is presented in the following paragraph. This repartition offer to each node a higher availability.

If the data acceded doesn't locally exist, a high probability of finding it on a node neighbour is warranted. This reduces the response time of the access requests as well as the passing band and the consumption of energy. The load of work is repartitioned on the network in a uniform way, which balances the expenses in energy and reduces the traffic around the same mobile node. The space is used in an optimal way by avoiding the unjustified redundancy of replicas. The redundancy is measured by the number of replicas of the same data on the neighbours. Replicas on two neighbours must be of different data. It is interesting that a node can retrieve different data on his neighbouring. Then the number of accessible data increase. The number of hops between nodes is used for do this.

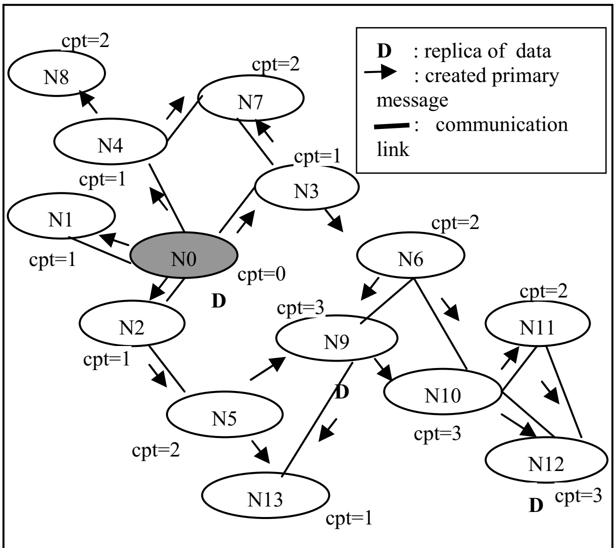


Fig. 1. Example of primary replicas generation

The role of **PrimaryCreat** message is: (1)to inform the nodes of the creation of a new data on the network, (2)to estimate the distance between a node and a replica, and (3) to distribute the data replicas on the network.

Algorithm:

1/ the functions and the parameters used in the algorithm are mainly:

- PrimaryCreat (NodeAdd, D_k, HopCpt): Message sent for a primary replication of data where a replica is found at the node level **NodeAdd**. **HopCpt** is the counter of the nodes covered since the last reinitialization of this counter.

- PrimaryReplica (D_k) : This function creates locally a data replica, sets the counter **HopCpt** to zero and broadcast the message **PrimaryCreat(NodeAdd, Dk, HopCpt)** to the following nodes neighbours to the identity of the local node. A node border is a node which hasn't a ***following link*** like N_8 (on figure 1). A node preceding a node N_i is a node "sender" of a message **PrimaryCreat** to N_i. A node following a node N_i is a node "non-sender" of a message **PrimaryCreat** to N_i.

2 / Algorithm of primary replication of data D_k:

```
If  N_J creates a new data D_k then    HopCpt = 0;  NodeAdd= N_J ;
                    send PrimaryCreat(NodeAdd, D_k, HopCpt ) to neighbours;   endif;
If reception of PrimaryCreat(NodeAdd, D_k, HopNb)  then
    If 1st reception then      HopCpt = HopCpt +1;
    If (HopCpt  =3)  then
        If (a replica of D_k does not exist on the neighbours) then PrimaryReplica(D_k );
            else If (a replica of D_k exists on one of the neighbouring N_i) then HopCpt = 1;
                NodeAdd= N_i ; Send PrimaryCreat(NodeAdd, D_k, HopCpt) to neighbours
                endif
            endif
    else If (HopCpt =2) and (this node has only one neighbour) then PrimaryReplica(D_k );
        else Send PrimaryCreat(NodeAdd, D_k, HopCpt) to the following nodes   endif
    endif
else  Ignore the received message;   endif;  endif ;
```

4.2 Dynamic Replication

The aim is to redistribute dynamically the replicas for each of the data in a way to maintain the access possibilities to data for each node, and to favour the best access time, and to limit the consumption in passing band and in energy. It is also about reducing the number of replicas unfolded because of the limitation in the capacity of mobile memory. Every time, two nodes neighbours can hold replicas for the same data in the case where it is very inquired by the two nodes. This redistribution is carried out by accomplishing/realizing the new replicas of data on certain nodes and by suppressing others on other nodes according to the access frequencies of nodes.

A user can frequently use data where he doesn't hold a replica. The traffic generated becomes then important even if it is not produced on the short paths, it means, a reduced number of hops. This traffic generated an overhead of these paths as well as a consumption of the passing band and an effort in energy from the part of the nodes included in the processing and the pathways of the requests. In order to overcome these drawbacks, we take into consideration the access rate to data at the time of replication of data. We define a variable $\mathbf{T_{ik}}$ which represents the access rate to data Dk by node $\mathbf{N_i}$, such as: $\mathbf{T_{ik} = NB/U}$, with

U: duration of time and, NB: the number of the access demands to $\mathbf{D_k}$ issued by $\mathbf{N_i}$.
Each time Units (U), the rates are dynamically re-evaluated.
For a better management of cache, we define for each node $\mathbf{N_i}$ two types of rates:

- $\mathbf{TE_{ik}}$: the access rate of user to external and data $\mathbf{D_k}$.
- $\mathbf{TI_{ik}}$: the access rate of nodes to internal data $\mathbf{D_k}$ local to user to $\mathbf{N_i}$.
- **S**: A threshold of an external access rate of node to data. Once this threshold is reached, the data are locally replicated in the node in order to avoid the performance degradation.

This replication raises the problem of lack of space. The solution is then to free the space by deleting the data replicas which have the lower external access rate and shouldn't be original data. This condition allows to ensure that the data should never be completely deleted from the network and at least a replica will always subsist.

➢ **Replication on demand:** When a just connected node wants to have access to none locally available data. It casts a message of the access demand to data. This demand will cover the nodes until their arrival on a node which holds a replica or the original of data. The demand includes besides the identity of the node "caller", the data identifier and the number of hops covered by this demand. The node holder of the replica consults this number. If the number of hop is greater or equal to three, instead of sending to it just a response to its demand, it transmits to it data replica. The condition of replicas generation will be explained in the following paragraph.

➢ **Replication on need:** In order to replicate the very required data, each **U** units of time, the following procedure is executed by a node Ni for each data $\mathbf{D_k}$:

- calculate the rate access $\mathbf{TE_{ik}}$ and $\mathbf{TI_{ik}}$
- if the **TEik** is higher than the threshold **S**, the data is considered very required by **Ni** and the replication of data **Dk** is initiated. A data replication request is sent to a holder node or cast on network. A holder node is known by **Ni**, if an access path data exist (see the following paragraph).

4.3 Access to Data

➢ **Safe-guard of access paths:** An access demand to data is processed as follows:

- The data is firstly researched locally at the node, If it exists at the node level, the access request is immediately satisfied.
- Otherwise, the request is cast to neighbours, a node which receives this request and which can not satisfy it, will in its turn, cast it to its neighbours.
- A node $\mathbf{N_j}$ which receives this request and which holds a data replica **Dk** inquired by **Ni** , is going to respond by $\mathbf{Rep(N_j, N_i, D_k, HopCpt)}$ where **HopCpt** is the number of hops necessary to reach Dk. This diffusion can charge or saturates the network and takes a higher execution time notably in an ad hoc mobile network with a big scale where the number of node is important. In order to avoid these drawbacks, and to reduce the consumption of energy and the bandwidth, a node which doesn't hold a replica of data saves the shortest access path which detects, it means, the identity of the nearest node in number of hops which holds a replica.

On a node $\mathbf{N_j}$, an access path is structured as follows: (**IdNode**, **IdData**, NbHop) where, **IdNode** is the identity of the node which holds a replica of the data identified by **IdData** and **NbHop** is the number of hops which separate $\mathbf{N_j}$ from **IdNode.**

Periodically, the messages of the neighbourhood discoveries are transmitted by the nodes. These messages are used to determine the immediate neighbours of the node. They are also used for the building of data paths. Each time, a node discovers a new neighbour; it sends to it the list of the local data replicas. The node 'receiver' of this list updates its data paths table. Likewise, when a node updates its table of data at the time of generation of local replica for instance, it diffuses this update to its direct neighbours. Then, these latter proceed to the update of the data paths. The algorithm of the updates of the paths is the following:

```
If (reception of data list of a node) Then For each data non local save the path. endif
   If $N_i$ don't possess a replica of $D_k$ then
      if $N_i$ receives the message PrimaryCreat (NodeAdd, $D_k$, HopCpt) of $N_j$ then
         if HopCpt =1 then $N_j$ holds a replica of Dk ; Save the access path: ($N_j$, $D_k$, 1)
      endif
         if HopCpt =2 then Save access path ($N_j$, $D_k$, 2) if it doesn't exist a shorter path
      endif
      endif
      If Ni receives Rep($N_j$, $N_k$, $D_k$, HopCpt) to an access demand to $D_k$ carried out by $N_k$
     then /*HopCpt : number of hops between $N_i$ and $N_j$ , $N_j$ the node holding the replica*/
      HopCpt= HopCpt+1;
      if Ni <> Nj then
         If it doesn't exist a shorter path then
               Save the access path: ($N_j$, $D_k$, HopCpt) ; Send Rep($N_j$, $N_k$, $D_k$, HopCpt)
          else /*exist shorter path ($N_p$, $D_k$, $HopCpt_1$) */ Send Rep($N_p$, $N_k$, $D_k$, $HopCpt_1$) to $N_k$
         endif
        else If HopCpt =3 then Create a replica of $D_k$; Update data paths tables "PathDatas"
                               Send Rep ($N_i$, $N_k$, $D_k$, HopCpt=0) to $N_k$; endif;
      endif
    endif
endif.
```

The mobility makes that the paths can become invalid, and so they need a permanent update. We associate a time TTL(Time To Live) to each path. When this time is reached, the path is considered invalid and it is deleted.

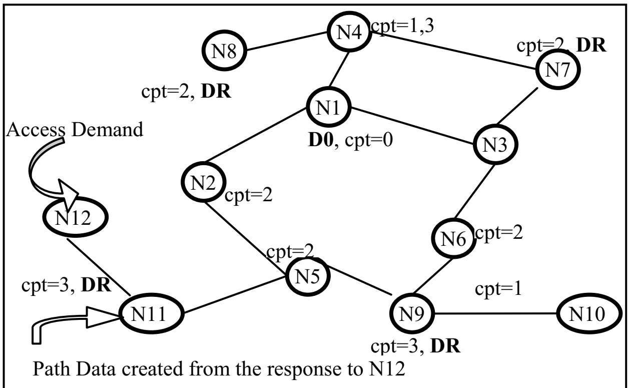


Fig. 2. Application of the replication

Example: We consider the following example (figure 2) where N1 is the node 'initiator' of the primary replication of data. It holds the original replica **D**. The nodes N11 and N12 are connected to the network after the primary phase of data creation. We suppose that N12 asked for the access to these data. We execute this method in order to notice its effects on the dynamic creation of replicas and on updates of the access paths to data.

➢ **Data access:** The algorithm of the data access procedure is the following:

```
If the node Nj calls for a data access then
   If Nj holds a replica (original or replica) of data then The access is locally carried
out
      else If Nj holds a valid path access towards this data then
               The demand is transmitted to the node indicated by the path.
              else The demand is sent to the neighbouring nodes. endif.
```

The receiver of an access request to a data behaves as a caller. The same previous processing is carried out to its level, except when the target data locally exist, a reply is transmitted to the requester node.

5 Performances Evaluation

This section presents results of the performance evaluation of the proposed method. The evaluation has been carried out by using the simulator Glomosim [3]. The table 1 shows the parameters of our environment. The nodes move according to the model 'Random Way Point' which is a model widely used and which seems the closest to real movement of the mobile nodes. In this model, the speed of the node mobility is varied between a minimal value précised by the variable MOBILITY-WP-MIN-SPEED and a maximal value given by the variable MOBILITY-MAX-SPEED. The dimension of the network is by default of 1000x1000m^2. And, in order to test the scalability, the number of nodes can reach 250 on a site of 5000x5000m^2.

The nodes are randomly chosen in order to create new data and send access demands. The proce-sses of creation of prim-ary replica and the sending of the access demands follow the model of Poisson with an average interval of 60 seconds.

This interval is varied between 1 second and 60 seconds for testing the load of the network.

Table 1. Configuration Parameters

Parameter	**D.V.**	**V.I.**
Number of nodes	50	50-250
Number of data	50	
Size of data (KB)	1	1 – 4
Size of cache (KB)	20	5 – 50
Max Speed(M/S)	1	1 - 10
Time of break (S)	10	
Threshold of access frequency (HZ)	0.5	0.01–1.5
Threshold of the size of data(KB)	2	
Bandwidth(Mbps)	2	
Interval of data creation (S)	60	1 - 60
Interval of the access demand (S)	60	1 - 60
(D.V.:Default Value, V.I.: Variation interval)		

The access frequency is calculated by using the relation of the moving averages, [4], [5], [14], [16] which is one of the formulae of well known technical analysis. The moving average is one of the oldest and widely used indicators. It allows to calculate an average value on a given period. The access frequency is then calculated by the formula: $\mathbf{MA}f\mathbf{ij} = \boldsymbol{\beta}\ \mathbf{MA}f\mathbf{ij^*} + (\mathbf{1}-\boldsymbol{\beta})\ f\mathbf{ij}$, Such as:

MA*f*ij: Moving average of the access frequency to the data j by the node i for the new period.

MA*f*ij*:Moving average of the access frequency to the data j by the node i for the old period.

***f*ij** : The access frequency of the node i to data j for the new period.

β: is a weight constant, in our simulation **β = 0.5,** with this value, we give more consideration to the new frequency value than to the historical background of the access of each node.

The **MA*f*ij** is calculated to each unit of time. If the value of the external **MA*f*ij** exceeds a certain threshold K, the data concerned must be replicated.

The metrics of our simulation are defined as follows:

- *The rate of availability*: This metric allows to know the rate of data accessibility during the simulation time. Formally, this rate is defined by the following formula: **TD=NDR/TDE**, Such as **NDR** and **TDE** are respectively, the number of successful demands and the total sent demands. The aim of all the protocols of replication is to increase the most possible TD.

- *The traffic* : The traffic is the number of messages transmitted by all the nodes during the simulation duration; formally, the traffic is defined as follows:

$$\mathbf{T} = \sum_{\mathbf{i=1}}^{m} \mathbf{NmesTri}$$, Such as n is the number of nodes and NmesTri is the number of messages transmitted by the nodes i during the simulation.

5.1 Accessibility and Storage Capacity

The figure 3 shows that the accessibility rate increases with the rise of the capacity memory. The expected result is due to the possib-ility of locally replicating a greater number of data. Also, the interval of success rates of access varies little in function of the maximum speed of nodes. The speed is one of the parameters of the mobility of nodes. We notice that the proposed method is not very influenced by the increase of the speed of the node mobility.

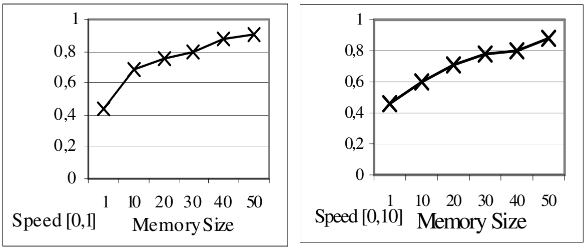


Fig. 3. Accessibility and storage capacity

5.2 Traffic and Storage Capacity

The figure 4 shows that the traffic decreases when the capacity of the memory becomes very big. With a very big storage capacity, the traffic decreases since the mobile nodes have higher chances to hold replicas for the most of data used (more local access) and the relocation of replicas becomes very rare.

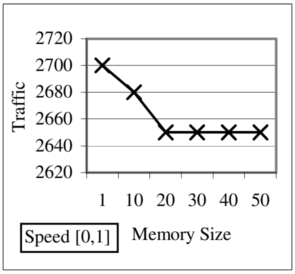


Fig. 4. Memory size and traffic

5.3 Accessibility and Access Frequency

This experiment aims at observing the influence of the increased number of access requests on the rate of data availability (figure 5). We notice that the rates of accessibility stay above the acceptable limit and the variation interval of these rates is not very influenced by the load. On the other hand, the speed variations influence the success of access too much. We observe that accessibility increases when the access frequency increases, because the data are locally replicated if their access frequency goes beyond the fixed threshold. The access demands are first sent to distance if the data concerned are not locally available. These requests can't all be satisfied. But after the creation of replicas, since their frequency access goes beyond the threshold, all following access will more quickly be satisfied.

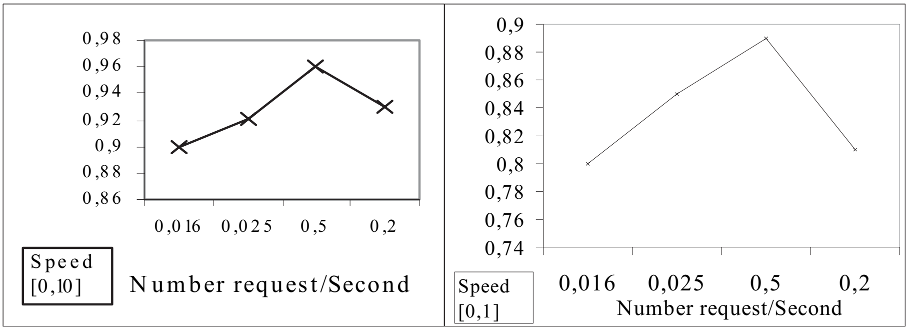


Fig. 5. Accessibility and access frequency

5.4 Traffic and Access Frequency

In figure 6, we notice that the traffic increases with the increase of the load expressed in frequency access. When the number of demands increases, the traffic increases too, notably in the case where the data required are not locally found and there is no path which leads towards these data. We also observe that the traffic is more influenced by the access frequency than by the speed of the node displacement because the traffic is mainly induced by the access and the frequencies of these accesses.

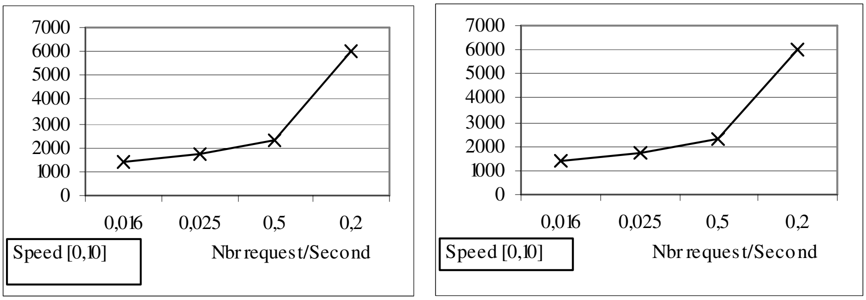


Fig. 6. Traffic and access frequency

5.5 Scalability

On the whole, we notice that the accessibility in figure 7 increases with the increase of the number of nodes because the network becomes more dense which favors the connections. The level of accessibility increases then decreases when

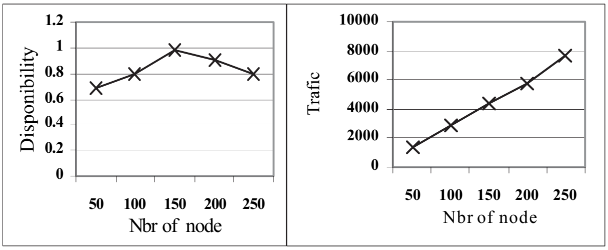


Fig. 7. Scalability

the number of nodes becomes greater, because in this method, every node can create new data, thus, it should execute the procedure of the creation of the primary replicas. Then an extra traffic is due to each phase of primary replication. But when the number of nodes becomes important, the number of new data increases too. These data consume the storage space. The more the size of the network is very important, the more the paths are subjects to disconnections.

6 Conclusion

The evaluation of the proposed method shows that the accessibility rates are interesting in most of the cases and the traffic induced stays acceptable. As the number of the creation of new data increases, the accessibility rate decreases, and the traffic increases because this method executes the procedure of primary replicas for each new created data. The primary replication allows to speed the information of data existence to all nodes of the network. Then, it distributes the replicas in an equal way on the network. If one of the nodes needs to be acceded, it has higher chances to find data at one or two hops. This method allows to put every information nearer the users. But it consumes the memory space. It is very effective when the mobile tools have a memory space more or less high or when the number of creation of new data is more or less small. One of the perspectives is to improve this method by considering the degree of interest which the network matches with data. Thus, data to which will be less in demand will not undergo a primary replication. Hence, we should limit the traffic and to still optimize the space especially if this latter is limited.

References

1. Barbara, D., Imielinski, T.: Sleepers and workholics: Caching strategies in mobile environments, Proc. ACM SIGMOD'94, pp.1-12, 1994.
2. Bagrodia, R., Zeng, X., Gerla, M. : GloMoSim - A Library for Parallel Simulation of Large-scale Wireless Networks. Computer Science Dep., Univ. of California at Los Angles, 1999.
3. Crowder, S. V.: A Simple Method for Studying Run-length Distributions of Exponentially Weighted Moving Average Charts. 1987, Technometrics, 29, 401-408.
4. Crowder, S. V.: Average Run Lengths of Exponentially Weighted Moving Average Charts. 1987b, Journal of Quality Technology, 19,161-164.
5. Fu, A.W, Cheung, D.W.L : A transaction replication scheme for a replicated database with node autonomy. Proc. 20th VLDB Conference, pp.214-225, 1994.
6. Hara, T., Harumoto, K., Tsukamoto, M., Nishio, S.: Dynamic replica allocation using database migration in broadband networks. Proc. IEEE ICDCS 2000, pp.376-384,2000.
7. Hara, T.: Effective replica allocation in ad hoc networks for improving data Accessibility. In Proc. of IEEE Infocom, Anchorage, Alaska, U.S.A., vol. 3, pages 1568-1576, April 2001.
8. Hara, T.: Replica Allocation in Ad Hoc Networks with Periodic Data Update. Proc. of the Third International Conf. on Mobile Data Management, 0-7695-1500-2/02 2002 IEEE .
9. Hara. T.: Replica allocation methods in ad hoc networks with data update. ACM-Kluwer Journal on Mobile Networks and Applications, vol. 8(4), pp 343-354, 2003.

10. Hara, T., Madria, S.K.: Dynamic Data Replication Using Aperiodic Updates in Mobile Adhoc Networks. 9th international conference. (Eds.): DASFAA 2004, LNCS 2973, pp. 869–881, 2004. © Springer-Verlag Berlin Heidelberg 2004.
11. Huang, Y., Sistla, P., Wolfson, O.:Data replication for mobile computer. Proc. ACM SIGMOD'94, pp.13-24, 1994.
12. Hara, T., Loh, Y.L., Nishio, S.: Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks. Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03) 1529-4188/03 2003 IEEE.
13. Krishnakumar, N., Bernstein, A.J.: High performance escrow algorithms for replicated databases. Proc. 18th VLDB Conference, pp.175-186, 1992.
14. Yin, L., Cao, G.:Supporting Cooperative Caching in Ad Hoc Networks. IEEE Infocom 04.
15. Wang, K., Li., B.: Efficient and guaranteed service coverage in partitionable mobile adhoc networks. In Proceedings of IEEE INFOCOM 2002.

Zone-Based Replication Scheme for Mobile Ad Hoc Networks Using Cross-Layer Design

Ke Shi, Rong Chen, and Hai Jin

Services Computing Technology and System Lab
School of Computer Science & Technology
Huazhong University of Science & Technology, Wuhan 430074, China
{keshi, crong, hjin}@mail.hust.edu.cn

Abstract. Accessing remote data is a challenging task in mobile ad hoc networks (MANETs). A common technique used to improve the performance of data access is replication, which improves the performance of data access in distributed systems at the cost of increased storage space and communication overhead. Due to strict resource constraint, node mobility and impairments of wireless transmission, applying replication schemes developed for distributed systems to MANETs directly leads to poor performance. In this paper, we develop a zone-based replication scheme for MANETs. In this scheme, every node proactively maintains replica distributing information within its replicating zone, which leads to the access from requesting node to the requested data item (including its replicas) distributed in its replicating zone can be satisfied directly. If requested data item is outside the replicating zone, reactive lookup process is invoked to find the node hosting the requested data item and the route to this node at the same time. Opportunistic data replicating is performed spontaneously with data transferring, and corresponding replica is allocated to nodes located in the replicating zone of the requesting node. Using cross-layer design, we illustrate how the hybrid adaptive routing technique, zone routing protocol, assists data lookup and replication to achieve high performance of data access. Simulation results have shown that our design is successful in a dynamic MANET.

Keywords: Ad hoc network, Zone, Zone Routing Protocol, Data Lookup, Data Replication, Cross-Layer Design.

1 Introduction

A MANET consists of many mobile nodes connected by wireless links[1].In such a network, each node operates not only as an end-system, but also as a router to forward packets. Two nodes can communicate with each other directly if they are within each other's wireless transmission range or via intermediate nodes if they are far away.

Although routing is a very important issue in MANETs, other issues such as data access are also very important since the ultimate goal of using MANETs is to provide information access to mobile nodes. Accessing desired remote data in MANETs is much more difficult than in fixed networks. Due to strict resource constraint, node

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 698 – 710, 2006.

mobility and impairments of wireless transmission, disconnections may occur frequently. This means that the creator of data is often unreachable when the data is needed. Even when the creator of data is reachable, multi-hop wireless connections cause long request delay.

Data replication has been widely used to improve data accessibility in distributed systems. By replicating data at mobile nodes, data accessibility can be improved because there are multiple replicas in the network and the probability of finding one copy of the data is high. Further, data replication can also reduce the request delay, since mobile nodes can get the data from some nearby replicas.

However, applying replication schemes developed for distributed systems to MANETs directly leads to poor performance since the underlying network connections are not reliable and stable anymore and the availability of nodes is not identically independent distributed anymore. The performance of data access heavily depends on the underlying routing service.

To achieve high performance in data access, we have developed a zone-based replication (ZBR) scheme for MANETs, integrated with a data lookup service, and assisted by a hybrid adaptive routing protocol, zone routing protocol (ZRP)[2]. In this scheme, every node proactively maintains replica distributing information within its replicating zone, which leads to the access from requesting node to the requested data item (including its replicas) distributed in its replicating zone can be satisfied directly. If requested data item is outside the replicating zone, reactive lookup process is invoked to find the node hosting the requested data item and the route to this node at the same time. Opportunistic data replicating is performed spontaneously with successive data transferring, and corresponding replica is allocated to nodes located in the replicating zone of the requesting node.

The rest of this paper is organized as follows: Section 2 presents background and related work. The system model and basic notion is introduced in Section 3. Proposed zone based replication scheme using cross-layer design is discussed in detail in Sections 4. Section 5 evaluates the schemes using simulations with different setups. We conclude with Section 6 and outline the future work.

2 Background and Related Work

Data replication is a traditional technique in distributed systems. Hara[3] proposed a data replication scheme in MANETs, which optimized the location of data replicas within a network periodically to achieve certain data accessibility. However, the assumption that access frequencies to data items from each node are known and are fixed limits the applicability of the schemes in practical systems. Periodical replicas reallocation also leads to heavy communication overhead.

Similarly to Hara’s work, Wang[4] considered the problem of replica allocation. It takes into consideration topological information, and data replication occurs only when necessary according to certain topology detection schemes. These schemes depend on the mobility model and assume that the locations and velocities of all mobile nodes are known. Yin and Cao[5] proposed data replication schemes that address both the query delay and the data accessibility. As both metrics are important for mobile nodes, their schemes need to balance the tradeoffs between data accessibility and request delay under different system settings and requirements. Their work adopted the same system model with Hara's work.

What is fully unseen from these works is the consideration of the underlying MANET routing services coupled with data access tightly. They do not address the issues that how to find an optimal path from the node requesting data to the node hosting the data or its replica either.

The existing routing protocols can be classified either as proactive or reactive. Purely proactive schemes continuously use a large portion of the network capacity to keep the routing information current, such as DSDV[6] and OLSR[7]. The widely used DSR[8] and AODV[9] are reactive protocols, in which global search procedure leads to significant control traffic and long delay. ZRP[2] provides efficient and fast discovery of routes by integrating these two radically different routing schemes.

Cross-layer design introduces stack wide layer interdependencies to optimize overall network performance. MobileMan project[10] introduces inside the layered architecture the possibility that protocols belonging to different layers can cooperate by sharing network-status information while still maintaining separation between the layers in protocol design. This project focuses only on providing a general architecture and does not look at the special problems such as data access.

Chen etc.[11] proposed a cross-layer framework to solve the data accessibility problem in MANETs. It utilizes advanced data advertising, lookup and replication services, as well as a novel predictive location-based QoS routing protocol in an integrated fashion to achieve high data access success rate. The performance of this scheme depends on the predictive accuracy of nodes' location and movement, which is the difficult task in practical environment.

Gruber etc.[12] develop a Mobile Peer-to-Peer Protocol (MPP) stack to support peer-to-peer file sharing in MANETs. It spans from the network layer to the application layer, and tries to reuse existing protocols as far as possible. The underlying routing protocol is DSR.

3 System Model and Basic Notion

Fig. 1 shows part of a MANET. Each mobile node in the network is assigned a unique identifier. Data is handled as a data item which is a collection of data. Each data item located in the network is also assigned a unique data identifier. The original copy of each data item is held by a particular mobile node. Each mobile node has certain storage space for creating replicas excluding the space for the original data item that the node holds. For example, node *S* has *n* original copies of data items $\{d_1, d_2 \ldots d_n\}$.

In this MANET, a data request is forwarded hop-by-hop until it reaches the node hosting the original copies of data items and then this node sends the requested data back. To improve the successful data access ratio and reduce the request delay, the length of this multi-hop path between the data provider and the requester should be as short as possible, and the time spending on find such an optimal path should be as short as possible too. Although data replication and

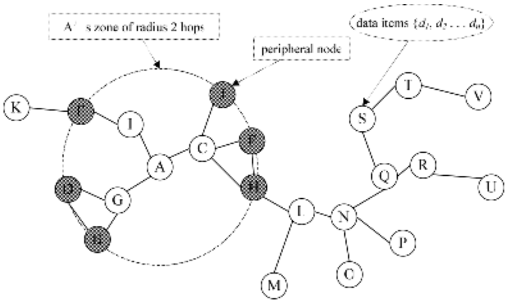

A' s zone of radius 2 hops
peripheral node
data items {d_1, d_2 . . . d_n}
K
I
A
C
D
G
S
T
V
Q
R
U
L
N
P
C
M

Fig. 1. A MANET

routing protocols can be used to achieve this goal separately, there is a limitation on how much they can achieve. In the following, we propose cross-layer zone-based replication scheme.

The basic notion is that data replication service in the application layer and zone routing protocol in the network layer can cooperate by sharing network-status information. Zone is the core concept in this scheme. A zone (of radius *r*) is defined for each node and includes the nodes whose minimum distance in hops from the node in question is at most *r* hops. Each node is assumed to maintain some kinds of information only to those nodes that are within its zone. In network layer, this zone is called routing zone, and routing information is maintained. In application layer, this zone is called replicating zone, and the information about replicas distribution and storage space are maintained. All these information are shared by the two layers. A node learns its zone through some sort of a proactive scheme.

Data lookup is incorporated with route discovery. Opportunistic replica distributing is performed with data transferring simultaneously when there is no available replica in the replicating zone of data requestor.

An example of a zone (for node *A*) of radius two hops is shown in Fig. 1. For the purpose of illustration, we depict zones as circles around the node. However, one should keep in mind that the zone is not a description of physical distance, but rather nodal connectivity (hops). Note that in this example, nodes *A*–*J* are within the routing zone of the central node *A*. Node *K*-*V* is outside *A*'s routing zone. Peripheral nodes are nodes whose minimum distance to the node is exactly equal to the zone radius. The remaining nodes are categorized as interior nodes. Thus, in Fig. 1, nodes *G*, *I* and *C* are interior nodes while *B*, *D*, *E*, *F* and *J* are peripheral nodes.

4 Cross-Layer Zone-Based Replication Scheme

In this section, we describe cross-layer zone-based replication scheme in detail. Data replication, lookup and ZRP work together to facilitate data access for various applications at the end-systems.

4.1 Zone Information Maintenance

In network layer, a node maintains routing information to those nodes that are within its routing zone through some sort of a proactive scheme, which is called intra-zone routing protocol (IARP) in ZRP. In this paper, we use OSLR [7]. However, any other proactive scheme would do. While the performance of the ZRP depends on the choice of IARP implementation, previous research[13] suggests that the tradeoffs are not strongly affected by the particular choice of the proactive scheme used.

In application layer, a node needs to maintain the information about the replica distribution and the nodes' capabilities within its replicating zone (its routing zone in network layer) to support data lookup and replica allocation. Using cross-layer design, it incorporates with routing zone maintenance. Every node has three tables, routing table, data lookup table and capability table.

Data lookup table describes the distribution of the data items (original copies or replicas) within the node's zone with a list of entries: <*data_id,o_flag, valid_period, description*>. The *valid_period* field indicating the freshness of the data is used to maintain consistency. We will discuss the detail in section 4.4. The *o_flag* field indicates whether the data item is original copy or not.

Mobile nodes may have very different capabilities ranging from small PDAs to laptops. Considering heterogeneity of the nodes, capability table provides useful information of nodes' resource within the zone supporting replica allocating decision. In this paper, only free storage space is considered since we assume single node's processing capability does not have significant effect on the performance of data access. Other parameters such as remaining power and processor utilization can be easily inserted into this table to support more comprehensive decision.

OLSR utilize periodical "*hello*" messages to sense neighbor and update routing table. To maintain the rest zone information, each node appends data lookup entries and capability entry to the "*hello*" message. Upon receiving the "*hello*" message, the data lookup table maintenance procedure renews the information associated with the existing data items, and augments the table if new data items become available, and capability table maintenance procedure renews the free storage space parameter associated with the nodes in this zone. Because the updates are only propagated locally, the amount of update traffic required to maintain a zone does not depend on the total number of network nodes (which can be quite large).

Since a node may contain a larger number of data items, the size of "*hello*" message may become very large. Periodical propagating full information may lead to unnecessary overhead. To reduce the size of the "hello" messages, we develop a method to include only the difference of contents from the previous "*hello*" message. The complete "*hello*" messages are propagated in longer time intervals. Between complete "*hello*" messages, smaller differential "*hello*" messages are sent out in shorter time intervals.

4.2 Data Lookup

In our scheme, data lookup is integrated with route discovery. In ZRP, IARP maintains routes for the nodes that are within the coverage of the routing zone. The inter-zone routing protocol (IERP) is responsible for reactively discovering routes to destinations located beyond a node's routing zone. In application layer, intra-zone data lookup based on IARP finds the data items located in the nodes that are within the coverage of the replicating; and inter-zone data lookup based on IARP is responsible for reactively discovering the data items located in the nodes beyond a node's replicating zone.

When a node wants to access a special data item, intra-zone data lookup takes place locally at this node. The process of intra-zone data lookup is as follows:

(1) Check the data availability through local data lookup table. If data item is available in the local zone, go to step 2; otherwise, inter-zone data lookup process is invoked.

(2) Check the path to the hosting node through local routing table.

(3) Send the request to the hosting node to retrieve the data.

(4) Hosting node transfer the data items to the requested node.

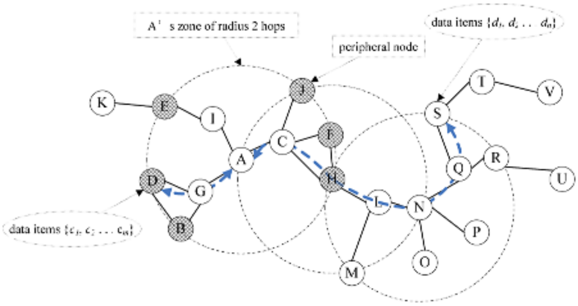

A' s zone of radius 2 hops
data items {d_1, d_2 . . . d_n}
peripheral node
K
E
I
J
C
F
A
D
G
H
B
L
N
M
O
P
S
T
V
Q
R
U
data items {c_1, c_2 . . . c_n}

Fig. 2. An example of data lookup

Since the lookup table has descriptions of all available data located in the nodes within the replicating zone, the node can get this kind of data items with a very low request delay. As shown in Fig. 2, if *A* want to access data items c_2, *A* can find its hosting node *D* and corresponding route *A*-*G*-*D* through intra-zone data lookup process since *D* is within *A*'s zone.

If the requested data item is not available in the replicating zone, inter-zone data lookup process is invoked to find the requested data item and corresponding path. In our scheme, it is implemented by extending AODV. The process of inter-zone lookup is as follows:

(1) The data requestor propagates a data request to all its peripheral nodes.

(2) Upon receiving the data request, the peripheral nodes execute the intra-zone data lookup: they check whether the data item is located in the nodes within their zone. If so, go to step 3. If not, the peripheral node forwards the request to its peripheral nodes, which in turn execute the same procedure.

(3) A data reply (including route reply) is sent back to the data requestor indicating the route to the node hosting the requested data items (original copies or replicas).

(4) The route between the data requestor and hosting node is established through route accumulation. AODV utilizes short-term storage at each relaying node to temporarily store a route in the form of next-hop routes back to the data requestor.

Comparing with DSR, route accumulation would occur during the route reply phase rather than the route query phase, resulting in less IERP traffic.

(5) Hosting node transfer the data items to the requested node along the established route. Opportunistic replication is triggered at the same time. The core idea of opportunistic replication is to replicate the data items beyond the zone of data requestor into the nodes within its zone with the normal data access.

An example of this inter-zone data lookup procedure is demonstrated in Fig. 2. If *A* wants to access d_2, *A* first checks whether d_2 is within its zone through intra-zone data lookup. Since *A* find no node within its zone hosts d_2, *A* propagates a query to its peripheral nodes; that is, *A* sends a query to nodes *B*, *D*, *E*, *F*, *H* and *J*. Now, in turn, after verifying that d_2 is not in the nodes within its routing zone, each one of these nodes forwards the query by propagating the query to its peripheral nodes. In particular, *H* sends the query to *N*, which recognizes d_2 as being in *S* within its zone and responds to the query, indicating the forwarding path: *A–C–H–L-N-Q-S*.

A nice feature of this distributed data lookup and route discovery process is that a single data query can return multiple route replies. The quality of these returned routes can be evaluated based on hop count (or any other path metric accumulated during the propagation of the query). The best route can be selected based on the relative quality of the route.

The inter-zone data lookup process based on IERP is distinguished from standard flooding-based query/response protocols by exploiting the structure of the routing/replicating zone. The zones increase the probability that a node can respond positively to a query. This is beneficial for traffic that is destined for geographically close nodes. More importantly, knowledge of the zone topology allows a node to efficiently continue the propagation of a query in the more likely case that destination can be found. At the same time, since the zones heavily overlap, the query will be forwarded to many network nodes, multiple times. To reduce the corresponding traffic, a query control scheme proposed by Haas [13] is adopted in our scheme.

4.3 Opportunistic Replication

Comparing with fixed networks, there exist many resource limitations in MANETs, for example, intermitted transmission, low bandwidth, poor connectivity, and unstable topology. At the same time, most nodes in MANETs also suffers from poor resources, for example, limited memory or storage space, short battery life, and unpredicted sleep or shutdown. Therefore, replication can not cause much additional network overload, which is determined by the characteristics of MANETs.

Our solution is opportunistic replication: data replication only performs with the occurrence of normal data accessing and transferring, which only consumes the energy and storage space of the nodes. Opportunistic replication is triggered by inter-zone data lookup as discussed in section 4.2. The data item that is not in the nodes within the zone of the requesting node is replicated into the node with enough free storage space within this zone when this data item is transferred from the hosting node to the requesting node. The following rules decide which node within the zone host the replica.

(1) The replica is first allocated to the peripheral nodes along the data accessing route.

(2) If the peripheral nodes along the data accessing route do not have enough free storage space, the replica is allocated to the interior nodes along the data accessing route.

(3) If the interior nodes along the data accessing route do not have enough free storage space, replica replacement procedure is invoked to evict the data replicas from the peripheral nodes when new data arrive. Replica replacement policy is the widely used LRU, which removes the least-recently-used data replicas.

In the intra-zone data lookup example shown in Fig.2, the replica of data item d_2 is first allocated to *H*. if *H* does not have enough free storage space, it is allocated to *C*. if *C* does not have enough free storage space, *H* replace old replicas with the replica of data item d_2.

Following these rules, no extra data transferring is needed to allocate data replicas. Due to the characteristics of MANETs, replica migration is restrictedly limited to reduce the network bandwidth and node energy consumption. When the node hosting the replica of a special data item is down or leaves the zone, we do not try to restore the replica until next access to this data item occurs. Every replica is created on demand. If there are more than one replica of a special data item existing in a zone, the redundant replicas can be removed and relieved storage space can host the replicas of other data items. However, this removing progress can only be triggered by the event that there is no enough storage space at any nodes in this zone to host the replica of new data items that does not exist at any nodes in this zone.

Since all the nodes are mobile, the nodes may join or leave a zone dynamically. IARP will sense the change in the network layer, and then trigger zone information maintenance process in the application layer to update corresponding zone information. If there are duplicated replicas after new nodes join, the duplicated replicas will be deleted later as mention above. We don't try to predict which node will leave and move the replicas that leaving node hosts to other nodes in the zone because mobility predication is difficult in the practical environment and replica movement may result in communication overhead.

4.4 Consistency

There is a consistency issue in our replicating scheme. Due to network bandwidth, power constraints and node mobility in MANETs, it is too expensive to maintain strong consistency among replicas. In our scheme, a weak consistency model called δ-consistency model [14] is adopted, which is a time-based consistency model. The intuition is based on the fact that replicas are consistent even if their versions are different but has not passed a predetermined time δ (the valid period) since they have been updated last. There are applications such as weather maps, etc., where updates arrive periodically and application only needs to know a consistent value in a certain period.

Every data replica is assigned a valid period δ_d, and a data requesting node considers a replica up-to-date if $\delta_d>0$. When opportunistic replication occurs, the owner decide the validation period δ_d of the data replicas based on updating interval and current data access time if the data requesting node gets the data from the owner.

If the data requesting node gets the data from other nodes hosting the replica, the value of δ_d of the new replica equals to the old one. Each node hosting the replica decreases its δ_d value in the same ratio.

The replicas with δ_d=0 can be removed from the hosting node to save storage space. The removing process is triggered later when there is no enough free storage space for new replicas. It is because the invalidated replicas may be still useful and indicate the users' interests for special data items.

5 Performance Evaluation

We use the OPNET network simulator, an event driven simulation package, to evaluate the performance of our scheme over MANETs. Cross-layer zone-based replication scheme is evaluated over a range of zone radius, ranging from purely reactive flooding-based method (r=1) to purely proactive table-driven method (r=∞). Performance is gauged by measuring successful data access ratio, request delay and the control traffic generated by this scheme. We also compare the performance of our ZBR scheme with MPP.

Our simulated network consists of 200 mobile nodes, whose initial positions are chosen from a uniform random distribution over an area of 2500m by 2500m. We utilize 802.11b with a maximum data rate of 11Mbit/s as MAC. The Two-Ray-Ground propagation model has a maximum radio range of 250m.

The nodes move according to random walk mobility model. A node moves from its current location to a new location by randomly choosing a direction and speed from pre-defined ranges, [0,v_{max}m/s] and [0,360^o], respectively. Each movement occurs in a constant time interval, 10 [s], at the end of which a new direction and speed are calculated. If a node reaches a simulation boundary, it bounces off the simulation border with an angle equal to the incidence angle. Two v_{max} values, 2m/s and 20m/s, are studied in the simulation. We do not adopt widely used random waypoint mobility model because it has been proved to fail to provide "steady state" in that the average node speed consistently decreases over time [15]. This could lead to unreliable results.

There are 200 data items in the simulated network. Each node can host 10 data items. For each data item, the fraction of nodes hosting this data item is defined as replication rate. The mean replication rate is 20%. Two values of the size of data item, 2KByte and 2MByte, are studied in the simulation. Each node generates a single stream of data requests. The request generating time follows exponential distribution with mean value $T_{request}$, 1 request/s. After a request is sent out, the node does not generate a new request until the request is served. The access pattern is based on uniform distribution. For MPP, the replication rate is also set to 20%, and the replicas are uniformly distributed in the nodes within the network.

The zone radius changes from 1 to 8 hops. The "*hello*" messages used to maintaining zone information are transmitted at random intervals of mean T_{hello}. T_{hello} is inversely proportional to the node speed, so networks with different mobility experience the same acceptable accuracy level of zone information.

Simulations were run on 50 randomly distributed node layouts, each for duration of 125s. No data was collected for the first 5s of the simulations while the initial intra-zone information maintaining process stabilized.

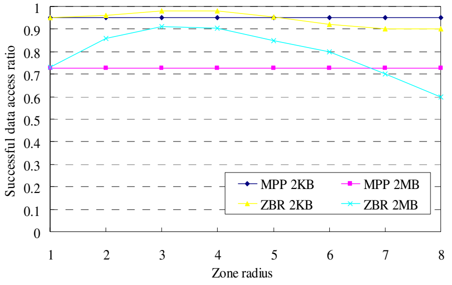


Fig. 3. Successful data access ration as a function of zone hops

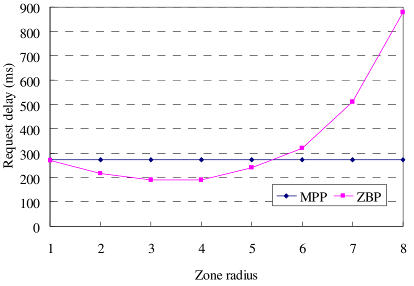


Fig. 4. The request delay as a function of zone hops

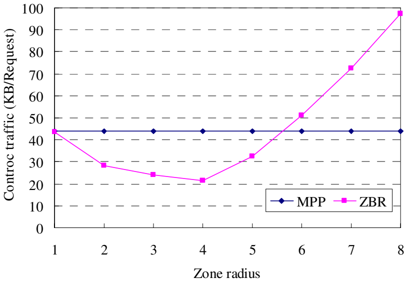


Fig. 5. The control traffic as a function of zone hops

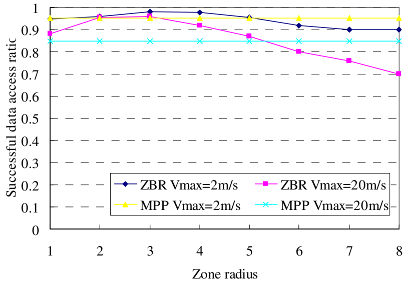


Fig. 6. The successful data access ratio against mobility, the size of data item is 2KB

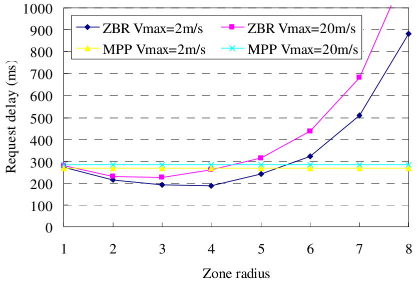


Fig. 7. The request delay against mobility, the size of data item is 2KB

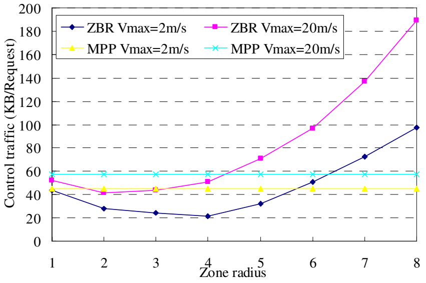


Fig. 8. The control traffic against mobility, the size of data item is 2KB

Fig. 3 shows successful data access ratio as a function of zone hops. Our ZBR scheme outperforms MPP scheme with appropriate *r* value, 2-5 in our simulation. When *r* increases further, the performance of our scheme degrades. Successful data access ratio of our scheme is lower than that of MPP scheme. The reason is that the number of data replicas distributed in the network decreases when *r* increases. Furthermore, the overhead caused by zone maintenance becomes very large, and the information updates may not arrive in each node in time, which leads to inaccurate zone information. The intra-zone lookup process based on these stale information in the routing table and data lookup table may fail to find the requested data .

We also find the performance improvement with appropriate *r* values is more significant when the size of data item is large. It is because our scheme allocates the replicas around the data requesting nodes dynamically. Comparing with the random allocation in MPP, it makes the requesting node can find its requested data in closer node, which lead to shorter transferring route. In MANETs, shorter path has much longer lifetime than longer path. The probability that path breaks during data transferring decreases significantly. When the size of data item is small, data transferring time is short, this improvement can be observed but not very obvious. When the size of data item is large, data transferring need more time, this improvement is more obvious.

Another important performance metric is request delay. In our simulation, request delay is measured as time span from the request generating in the requesting node to the route establishing between the hosting node and the requesting node. This definition excludes data transferring time. Although transferring data items with larger size may occupy more network bandwidth and increases the request delay, it is not a significant factor. Fig. 4 shows the request delay as a function of zone hops when the size of data item is 2KB.

Our scheme achieves lower request delay than MPP scheme with appropriate *r* value. The request delay decreases with the increasing of *r* when $r<6$. When *r* further increases, the request delay begins to increase. Since inter-zone data lookup happens less frequently when *r* becomes larger, it looks like data requesting node can get most data items through intra-zone data lookup and the request delay should be lower. However, as we mentioned before, the overhead of maintaining a large zone is very large, which leads to inaccurate zone information and makes intra-zone data lookup process fail. Before the requesting node finds its requested data, it may experience several failed intra-zone data lookup process, and then the request delay increases significantly.

The overhead is gauged as control traffic caused by our scheme. Transferring requested data items from the hosting nodes to the requesting node is not counted as control traffic, which eliminates the effect of the size of the data item on the overhead. Fig. 5 shows the control traffic as a function of zone hops when the size of data item is 2KB. Our scheme achieves lower control traffic than MPP scheme with appropriate *r* value. Opportunistic replication does not cause extra replica migrating traffic, and δ-consistency model does not need extra traffic to maintain the consistency among the replicas either. As we discussed above, when *r* is larger, zone information maintenance causes too much control traffic and make the control traffic of our scheme is higher than MPP scheme.

As shown in Fig. 3, 4, and 5, when $r = 1$, the performance metrics for ZBR and MPP are very similar because both the underlying routing mechanisms are reactive. The data requests are flooded among the nodes within the network. When mobility increases, we notice that the performance difference becomes a little larger. Under this high mobility circumstance, our scheme performs better than MPP because AODV performs better than DSR. The simulation results shown in Fig.6, 7, and 8 demonstrate this trend.

Fig. 6, 7, and 8 illustrate the effects of mobility on the successful data access ratio, request delay, and control traffic respectively. We see that the optimal zone radius at which better performance can be achieved decreases as mobility increases. Increased mobility causes the network topology to change more rapidly, resulting in an increased of zone information update traffic. Therefore, successful data access ratio decreases, and request delay increases.

6 Conclusion and Future Work

In this paper, we focus on the cross-layer design between two major layers of the mobile end-system, the routing layer and the application layer. They work together to facilitate data access for various applications at the end-systems.

Specially, a cross-layer zone-based replication scheme for MANETs is developed, which provides flexible and efficient data access service with low overhead by integrating data replication, lookup and underlying ZRP routing protocol. Our simulations show that the overall performance of data access is improved. Through simulation we also find for any particular network configuration and performance demand, each node has an optimal zone radius. Our future work is to determine the best choice for the zone radius based on local information directly available.

References

1. Homepage of IETF mobile ad hoc networks (MANET) working group, http://www.ietf.org/html.charters/manet-charter.html, updated July, 2005
2. Z.J. Haas and M.R. Pearlman, The Zone Routing Protocol (ZRP) for Ad Hoc Networks, Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002
3. Takahiro Hara. Replica allocation methods in ad hoc networks with data update. Mobile Networks and Applications, Vol. 8, Issue 4, August 2003, Pages: 343 – 354
4. K.H. Wang and B. Li, Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks, in Proc. of INFOCOM'02, 2002, pp. 1089–1098
5. Liangzhong Yin, Guohong Cao, Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks. SRDS 2004: 289-298
6. C. E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in Proc. ACM SIGCOMM 1994, pp. 234–244
7. P. Jacquet, P. Muhlethaler, and A. Qayyum, Optimized link state routing protocol, IETF MANET, Internet Draft, Oct. 2003.
8. D. B. Johnson and D. A. Maltz, Dynamic source routing in ad hoc wireless networking, in *Mobile Computing,* T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996.

9. C. E. Perkins and E. M. Royer, Ad hoc on-demand distance vector routing, in Proc. IEEE WMCSA'99, vol. 3, New Orleans, LA, pp.90–100.
10. M. Conti, G. Maselli, G. Turi, and S. Giordano, Cross-Layering in Mobile Ad Hoc Network Design. IEEE Computer Vol. 37, Num. 2, pg 48-51, Feb. 2004.
11. K. Chen, S. H. Shah, K. Nahrstedt, Cross-Layer Design for Data Accessibility in Mobile Ad hoc Networks, Journal of Wireless Personal Communications, vol. 21, pp. 49-75, 2002.
12. R. Schollmeier, I. Gruber, F. Niethammer Protocol for Peer-to-Peer Networking in Mobile Environments, In Proc. of ICCCN 2003, October 20-22
13. Z.J. Haas and M.R. Pearlman, The Perforamnce of Query Control Schemes for the Zone Routing Protocol, ACM/IEEE Transactions on Networking, vol. 9, no. 4, pp. 427-438
14. J. Cao, Y. Zhang, L. Xie, and G. Cao, Consistency of cooperative caching in mobile Peer-to-peer systems, Workshop on Mobile Distributed Computing, IEEE ICDCS, 2005
15. J. Yoon, M. Liu, and B. Noble, Random Waypoint Considered Harmful, Proc. Of INFOCOM 2003, pp. 1312-1321, Apr. 2003

Latency of Event Reporting in Duty-Cycled Wireless Sensor Networks

Yu Chen[1] and Eric Fleury[2]

[1] ARES/INRIA – INSA de lyon, France
ychen@cs.tamu.edu
[2] ARES/INRIA – INSA de lyon, France
Eric.Fleury@inria.fr

Abstract. We consider duty cycling in sensor networks for time-critical monitoring, where sensors monitor events and send the information to a data collection node (called *sink*). Latency is considered as the delay elapsed between the time when *an event occurs* and the time when *the sink gets the report*. Such networks are different from general purpose ad hoc networks, where latency is defined as the delay elapsed between the time when *a source node* initiates a packet and the time when *the destination node* receives the packet. We aim to prolong network lifetime by scheduling periodic sensors' duty cycles while preserving small latency from *events* to the *sink*. In our duty cycle schemes, sensors are grouped into sets; the collaboration between different sets is designed to enable the availability of event reports at active sensors before they reach the sink. Our ideas are simple and achieve desirable properties: the network lifetime is proportional to the number of sets and the performance is comparable to that of non-duty-cycled networks with the same number of active sensors, as shown in our analyses and simulations.

Keywords: Power management, Coverage, Connectivity, Event Detection.

1 Introduction

Wireless sensor networking has been a growing research area for the last years. Since sensor nodes are normally battery-operated and it is often not feasible to replace or recharge batteries, how to prolong the lifetime of sensor networks has become one important challenge [1,8]. Studies have shown that idle listening is a significant consumer of power [18,9,23,20,25,12] and energy efficiency can be achieved by periodic duty cycling of sensors, that is, scheduling sensors between active and sleep mode, in networks where the traffic load is light most of the time [23,24,2,12]. However, duty cycling also disrupts networks performance. Sensing coverage might not be guaranteed if too few sensors are active at a time. Duty cycling introduces extra latency: the data sampled by a node during its sleep period have to be queued until the active period; and, upon receipt of a packet,

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 711–723, 2006.

an intermediate node has to wait until the next hop wakes up to forward the packet. If the two end nodes of a link are not scheduled to be active at the same time, the link is disconnected in the sense of communication ability, which might cause network partitions. Furthermore, turning off sensors might cause, at least temporarily, the unavailability of certain information.

Because of the availability of a large number of low-cost sensors, one promising way to prolong network lifetime while preserving certain performance is to put individual sensors on a low duty cycle and add redundancy in sensor deployment to compensate for potential performance degradation. Therefore, questions of interest are how to design the collaboration among sensors to achieve a long network lifetime and whether a duty cycled network can achieve the same performance as a non-duty-cycled network with the same number of active sensors. They are nontrivial problems. As pointed in [11], most duty cycles suffer from a *data forwarding interruption problem* — in the progress of data forwarding, due to limited overhearing range, nodes on the multihop paths between the source and destination might be unaware of ongoing data forwarding and turn to sleep mode, even when explicit mechanism is used to adjust duty cycle since messages can only be forwarded limited hops in an active period; if such nodes exist on each path from the source to the destination, data forwarding will stop until the next active interval, which results in significant delay.

In this work, we consider duty cycling with the focus on sensor networks for time-critical monitoring of an area, in which each sensor monitors events and sends the information of events to a data collection node (called *sink*). In such networks, connectivity and latency are considered from *events* to the *sink* and latency is defined as the delay elapsed between the time when *an event occurs* and the time when *the sink gets the report.* They are different from general purpose ad hoc networks, where connectivity is considered between *nodes* and packet latency is defined as the delay elapsed between the time when *a source node* initiates a packet and the time when *the destination node* receives the packet. We aim to prolong network lifetime by duty cycling sensors while preserving small latency from each *event* to the *sink.* In our duty cycle schemes, sensors are grouped into sets; network lifetime is proportional to the number of sets. The collaboration between different sets is designed to enable the availability of event reports at active sensors before they reach the sink. Our analyses and simulation results show the performance of networks duty cycled by our schemes is comparable to that of non-duty-cycled networks with the same number of active sensors.

An efficient synchronization mechanism is required by most duty cycling schemes (e.g. [23,12]). Compared to a TDMA scheme, duty cycling schemes require a much looser synchronization, and message exchanges for synchronization can be reduced to save energy. Efficient synchronization protocols have been proposed for duty cycled networks (e.g. [22,6]). We assume an efficient synchronization scheme is available and describe system behavior in terms of slots.

2 Related Work

Duty cycling has been proposed to achieve energy efficiency [23,24,20,2,12]. Communication restricted to an established directed tree is considered in [11], where nodes are scheduled to sleep as soon as they transmit packets to the next level and wake up just in time to receive the next round of packets. For general communication, *adaptive listening* [23,20] reduces sleep latency for short paths at the expense of more energy expense due to extended activation.

Work has been done (e.g. [15,16,14]) on the number of sensors required to guarantee a whole coverage of the area (connectedness resp.) with a high probability when nodes are uniformly and independently deployed. Coverage in wireless sensor networks has been investigated in [13,21,17,10,5,14]; in particular, duty-cycled sensor networks are considered in [10,5]. Integration coverage and connectivity is studied in [21]. Most works consider the connectivity between *any pair of nodes.* It has been pointed out in [2] that full connectivity is unnecessary in networks for time-critical monitoring. However, the focus in [2] is on connectivity from an infinity component when the number of nodes approaches to infinity, instead of a whole coverage of the monitored area. Detection delay is considered in [3] for non-duty-cycled sensor networks.

3 System Model and Problem Definition

We consider a sensor network whose task is time-critical monitoring of a specific area where events can happen at any location at any time; each event is required to be reported to a data collection node (called *sink*) timely. The sink is always active and sensors are randomly deployed in the monitored area. The network is modelled by a geometry random graph $G(V, R_{tr}, R_{sen}, L)$ as follows. A set V of sensors are distributed randomly within area $[0, L]^2$. Each sensor is equipped with a radio transceiver with transmission range R_{tr}; two sensors are able to communicated with each other if and only if the distance between them is within R_{tr}. Sensors are equipped with sensory devices with sensing range R_{sen} and objects within a disc of radius R_{sen} centered at an active sensor are reliably detected by it. Since low-data-rate networks are the scenarios for which duty cycle schemes are primarily designed, and the latency due to congestion or interference (through some random access schemes) is not significant in light traffic, we assume *one slot* is long enough to forward packets to one-hop neighbor.

Our purpose is to prolong network lifetime by duty cycling. Strictly speaking, turning the sensory device off and turning the transceiver off are two different issues. Here we consider the scenarios in which sensory device and transceiver are simultaneously duty cycled, and by saying a node is active (sleeping resp.), we mean the sensory device and transceiver of this node is on (off resp.). A *schedule* of a node x is represented by an array S_x such that $S_x[i] \in \{0, 1\}$ for any $i \in [F - 1]$, where $F = |S_x|$; given any slot i, if $S_x[{\scriptstyle i\%F}] = 1$, the node is active, otherwise it stays in sleep mode. We call the F continuous slots: fF, $fF + 1$, ..., $fF + F - 1$ as the fth frame. For simplicity, we assume all the sensors have the same frame length, denoted by F.

In a duty cycled network, the shortest latency to forward packet from one node the other depends on the network topology, duty cycle scheme and the time when the forwarding starts. Given time T and any two adjacent sensors p and q, we denote the shortest delay for p to forward q a packet starting at time T by $Latency(p, q, T) = 1 + \min\{t \geq 0 | S_p((t+T)_{\%F}) = 1 \wedge S_q((t+T)_{\%F}) = 1\ \}$. Note the second term is the minimum time sensor p needs to wait for both p and q are active, which is $+\infty$ if no such slot exists. The latency to forward a packet along a path is defined as the sum of the latency to forward packet from one sensor to the next in the path; note the time when each of these forwardings starts depends on the latency in previous hops. Given any two sensors p and q, we denote by $Latency(p, q, T)$ the shortest latency for p to forward q a packet starting at time T along any path.

In networks for time-critical monitoring, the focus is on the time elapsed between the time when *an event occurs* and the time when *the sink gets the report of this event*; we call latency so defined as *event reporting latency.* Given a location X and time T, we denote by $A(T, X) \subseteq V$ the set of sensors that are active at T and whose sensing range covers X. The shortest latency to report an event that occurs at X at time T is $\min_{p \in A(T,X)} Latency(p, sink, T)$ if $A(T, X) \neq \emptyset$, otherwise is $+\infty$. Since events can occur at any time at any location, we consider the maximum shortest latency, called the *latency diameter.*

Definition 1. ***(Latency Diameter).*** *The latency diameter of a duty cycled network is defined as* $\max_{\forall location\ X, \forall time\ T} \{\min_{p \in A(T,X)} Latency(p, sink, T)\}$, *where we define* $\min_{p \in \emptyset} Latency(p, sink, T) = +\infty$.

Note Definition 1 is different from that for general purpose ad hoc networks, where latency is usually defined as $\max_{\forall p,q \in V, \forall T}(Latency(p, q, T))$ if the focus is on communication between any pair of nodes, or $\max_{\forall p \in V, \forall T}$ $(Latency(p, sink, T))$ if the focus is on communication to the sink. The difference can be illustrated by the example in Fig. 1, where monitored area is indicated by a square. For simplicity, we assume nodes are active all the time and they have the same sensing range and transmission range. The maximum event reporting latency is 3 since events that occurs in the light gray area can be reported to the sink within 1 slot, events that occurs in the dark gray area can be reported to the sink within 2 slots, and it takes 3 slots to report events that occur in the white area. Note the maximum delay between any pair of nodes is 6 (between nodes a and b) and the maximum delay from any node to the sink is 4 (from node a to the sink).

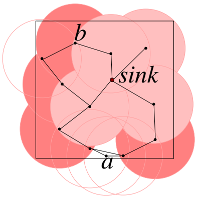


Fig. 1. An example of event reporting latency

Now we consider the requirements on duty cycling scheme in sensor networks for time-critical monitoring. First of all, the latency diameter should be finite, thus the report of each event can reach the sink. Definition 1 indicates $A(T, X)$ should not be $\emptyset$ for any time T and any location X, that is, a *whole coverage* of the network by active sensors is required at any time. Note the *connectedness* of active sensors at any time is not necessary. Furthermore, we want that, once an event happens, at any later time the report of this event is available at either some active sensors or the sink. Formally, we require

- Given $\forall$ location X and $\forall$ time T, for any time $T' \geq T$, $\exists$ a sensor q, such that, $S_q[T'] = 1$ and $\min_{p \in A(T,X)} Latency(p, q, T) \leq T' - T$.

We call this condition as *active information condition.* Informally, it enables the report of events flow through active sensors before they reach the sink. There are several reasons to propose this requirement. First, if the information of certain events is held only by sleeping sensors, this information will be lost if these sensors are not successfully activated again, due to the short battery lifetime or sensor failures. Secondly, if the information of each event is available at some active sensors, queues on events can be performed in the network, thus duty cycling is transparent to applications. From the aspect of duty cycling design, this enables sensors stay in sleep mode for long time without causing long latency, as active sensors can forward the information to the sink. Note this requirement is independent of bounded latency diameter requirement; neither of them can guarantee the other. It can neither be guaranteed by that at any time the active sensors form a connected network and cover the whole area.

Our approaches are based on the estimation of some system behaviors; this follows the engineering practice — a good estimation of system behavior is an importance reference in system design. From the theoretical aspect, given network area, sensing and transmission range, much has been done on computing the number of sensors required to guarantee a whole coverage (connectedness resp.) with a high probability ([15,16,14]); we denote this value by N_{cover} ($N_{connect}$ resp.). Here we define denotation D to represent a value that is, with a high probability, an upper bound on the diameter of networks when $N \geq \max\{N_{cover}, N_{connect}\}$ sensors are randomly deployed; such a value has been studied in [14]. Note this bound is not necessary tight; a loss bound is $\max\{N_{cover}, N_{connect}\}$. In our work, the number of active sensors is computed based on N_{cover} and $N_{connect}$; D is used in deciding the length of periods in which a sensor stays actively before it switches to sleep mode; a loss D will not affect the event reporting latency, provided that the power resources on individual sensors enable sensors stay active in at least D consecutive slots. As sink is always active, when $N_{connect}$ sensors are active, they are very likely connected to the sink. As we consider random deployment of sensors, in non-duty-cycled networks, at least $\max\{N_{cov}, N_{connect}\}$ sensors are required to detect and report each event with a high probability. For comparison purpose, we denote the latency in such networks by $Latency^{\star}$; note it is a random variable.

4 A Duty Cycle Scheme That Has Bounded Latency

Our duty cycling idea is intuitive and simple (Fig. 2). Sensors are grouped into several sets and randomly deployed in the monitored area. Sensors from the same set follow the same schedule. In the sequel, we will present our duty cycling scheme and explain how it guarantees coverage, active information condition and bounded latency. It will be easy to see the network lifetime achieved by this duty cycle scheme is proportional to the number of sets.

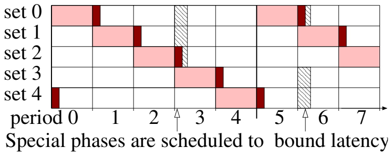


Fig. 2. A duty cycle scheme that has bounded latency

Since we consider a random deployment of sensors in the monitoring area, N_{cover} sensors are required to guarantee a whole coverage with a high probability. In our scheme, we group sensors into sets such that each set has N_{cover} sensors. In order to prolong network lifetime, multiple sets of sensors are deployed and each set is responsible to monitor the area in one period in each frame, called *duty phase* (light gray boxes in Fig. 2).

In order to enable information flow through sensors that are active in adjacent periods, at the end of their duty phase, sensors in each set stay active for an extra period, called *transitional phase* (the dark gray squares in Fig. 2). If the length of the transitional phase is large enough, sensors in one set will be able to forward the information to the sensor in the next set. For example, in Fig. 3, let light gray sensors be set S_1 and dark gray sensors be S_2. The transitional phase should be at least two slots since if sensor a detects an event at the end of its duty phase, it takes 2 slots to forward the information to some sensor in S_2; it is easy to verify a two-slot transitional phase is sufficient.

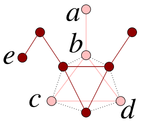


Fig. 3. An example of distance between two subgraphs

The length of transitional phase should be designed according to the "distance" between two adjacent sets. Informally, the distance between two sets is the maximum number of hops it takes for one sensor in one set to reach *any* sensor in the other. It is formally defined below.

Definition 2. *Given a graph G and subgraphs S_1, $S_2 \subseteq G$, the distance from S_1 to S_2 is defined as* $\max_{\forall u \in S_1} \min_{\forall v \in S_2}\{Dis(u,v)\}$*, where $Dis(u,v)$ is the number of hops in the shortest path from u to v in the subgraph formed by S_1 and S_2.*

In Fig. 3, the value of $\min_{\forall v \in S_2}\{Dis(u,v)\}$ for $u = a$, b, c, d is 2, 1, 1, 1 respectively, thus the distance from S_1 to S_2 is 2. Note the distance from S_2 to S_1 is 3 (the maximum $min_{\forall v \in S_1}\{Dis(u,v)\}$ is obtained when $u = e$).

It is easy to see active information condition is guaranteed if the length of the transitional phase is at least the distance between two adjacent sets, since each sensor is able to forward the report to the sensors in the next set before it turns into sleep mode. Thus the key is to decide the distance between two sets. Generally speaking, when sensors are randomly distributed, this distance depends on the density of active sensors. As for sensor networks for time-critical monitoring, we focus on sets of active sensors that cover the whole monitored area. We show below that the distance is *one* if the transmission range is no less than the sensing range. This implies that, in the transitional phase, each sensor that is on duty in last period is able to exchange information to a sensor that is on duty in the next period in *one hop*. In practice, sensing range might be larger than transmission range, but the number of active sensors can be computed by setting $R_{sen} = R_{tr}$ to achieve this property. Note this conclusion only depends on the property of coverage; it does not depend on the density of sensors.

Theorem 1. *Consider a sensor network $G(V, R_{tr}, R_{sen}, L)$ such that $R_{tr} \geq R_{sen}$. Given any two subsets $S_1 \subseteq V$ and $S_2 \subseteq V$, if S_1 (S_2 resp.) covers the whole network area, than the distance between S_1 and S_2 is 1.*

Proof. Consider any sensor $x \in S_1$, we need to prove x is adjacent to some node in S_2. Since sensors in S_2 cover the whole network area, including the location of x, x is covered by some sensor, say y, in S_2. That is, the distance between x and y is no more than R_{sen} (Fig. 4). Since $R_{tr} \geq R_{sen}$, there is a connection between x and y. Thus the distance between S_1 and S_2 is one.

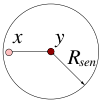


Fig. 4. One-hop distance when $R_{sen} \leq R_{tr}$

Now we consider event reporting latency. If $N_{cover} \geq N_{connect}$, sensors from the same set form a connected network in each period with a high probability. We require the length of each period is at least D. So the reports of events can be forwarded to the sink in this connected network if the events occur at least D slots before the end of the duty phase. Otherwise, at the end of each period, transitional phase enables the reports of events that are in forwarding progress to reach some sensors in the next set, and thus packet forwarding can be continued in the next period. Thus event reporting latency is no more than $2Latency^{\star}$.

If $N_{connect} > N_{cover}$, the connectedness of the network formed by one set cannot be guaranteed, that is, while data flow through active sensors, information cannot reach the sink due to disconnection. In order to make sure events are reported to the sink, a special phase, called *reporting phase*, is scheduled once a while, say at the end of every p periods, in which at least $N_{connect}$ sensors are active. In each of such phase, the $N_{connect}$ sensors include all the sensors that are on duty in the previous p periods. If $N_{connect} \geq (p+1)N_{cover}$, extra sensors are scheduled active to guarantee connectedness; in order to balance energy consumption among sensors, one way to select these sensors is to divide sensors into groups such that each group has at least $N_{connect} - (p+1)N_{cover}$ sensors and they are scheduled active in reporting phases in turn. In the example of Fig. 2, reporting phase is scheduled at the end of every 3 period. The shaped square indicates the sensors that are on duty in the previous 3 period; there might be other sensors that are active in this phase to guarantee connectedness with a high probability, which are not shown in the figure. Since the length of such phases should be large enough to forward each report to the sink, we set the length of such phases D. Event reporting latency is bounded by the interval between two consecutive reporting phases plus $Latency^{\star}$.

5 A Duty Cycle Scheme for Time-Critical Monitoring

In the approach presented in the last section, when $N_{connect} > N_{cover}$, the bound on latency depends on the interval between two consecutive reporting phases and $Latency^{\star}$. In this section, we consider a duty cycle scheme that guarantees short latency at the expense of more active sensors. A simple idea to reduce latency is to modify the approach in the last section by letting the number of sensors in each set be $N_{connect}$. In this idea, the number of active sensors is $2N_{connect}$ in transitional phase and $N_{connect}$ otherwise. In this section, we present an approach in which the number of active sensors remains $N_{connect}$ when $N_{cover} \leq \frac{1}{2}N_{connect}$.

In this scheme, sensors are also divided into sets, and each set is assigned N_u sensors, where $N_u = \max\left\{N_{cover}, \frac{1}{2}N_{connect}\right\}$. Note if N_u sensors are deployed, a whole coverage of the monitored area is guaranteed with a high probability, and if $2N_u$ sensors are deployed, connectedness of the deployed sensors is guaranteed with a high probability. In our scheduling, sensors in one set are active in two consecutive periods in each frame: set i is scheduled active in the ith and $(i+1)$th periods of each frame (Fig. 5).

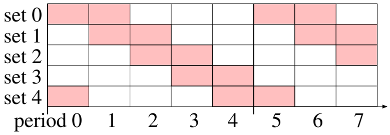


Fig. 5. A duty cycle scheme for time critical monitoring

Such a simple idea provides desirable service for time-critical monitoring. We can interpret this scheduling scheme as follows. Sensors in set i are responsible for monitoring the area in the ith period of each frame, and there are two purposes for the sensors in set i to stay active in the $(i+1)$th period: first, together with sensors in set $i+1$, the active sensors in the $(i+1)$th phase form a network that is connected to the sink with a high probability; secondly, since sensors in set i have the information of events that occur in the ith period, the reports of the events that have not reached the sink at the end of the ith period, e.g. those that occur close to the end of the ith period, can still be forwarded in the $(i+1)$th period. In order to bound event reporting latency, we set the length of each period D. Thus the event reporting latency of this approach is bounded by $2Latency^{\star}$: for each event that occurs in the ith period, if it reaches the sink in this period, the latency is $Latency^{\star}$; otherwise, it can reach the sink within $Latency^{\star}$ after the $(i+1)$th period starts.

In the approach presented in the last section, active information condition is guaranteed by scheduling transitional phases in which the number of active sensors is doubled. In this approach, the number of sensors that are scheduled to be active remains unchanged, and active information condition is guaranteed as follows. In each peorid, say the ith period of some frame, sensors in set i and set $i-1$ are active. We have showed in the above paragraph that each event is able to be reported to the sink within two periods, so we only need to consider active information condition for events that occur in the $(i-1)$th period and the ith period, which is true since information of events that occur in the ith period is available at the sensors in set i and information of those that occur in the $(i-1)$th period is available at the sensors in set $i-1$.

6 Simulations

We evaluate the performance of our schemes through high level simulations under the assumption that an appropriate synchronization mechanism is available. Since it is obvious that in our approaches, the network lifetime depends on the number of sets of sensors, we focus on the event reporting latency. Note the specific routing protocol that is used to forward packets also has impact on the event reporting latency. Since the focus is on the service provided by scheduling schemes, we evaluate the shortest latency to report events to the sink.

Much work has been done on computing the number of sensors required to guarantee the area is *almost surely covered* or the network is *almost surely connected* (e.g. [15,16,14]); here by saying "almost surely", we mean the probability of the network having certain property approaches to one when the monitored area approaches to infinity. However, little has been done on how fast the probability approaches to one; one exception is [16], where connectedness is considered. As such information is important in designing sensor networks and required in our simulations, we evaluate these values and present our results in section 6.1. In section 6.2, we present our evaluation on the latency of our schemes with various monitored area and applications' requirements on network lifetime.

6.1 Evaluation of N_s and N_c

Given L and radio range R_{tr} or R_{sen}, we evaluate the number of sensors that yield a high percentage of graphs that are connected or cover the whole network area, denoted by $N_{connect}$ and N_{cover} respectively. For each number of sensors N, we generate 200 networks and check the percentage of graphs that has the desirable property. We take the smallest value of N such that no percentage less than 98% or 99% is obtained with each larger N; at least 40 larger numbers than the taken value are checked. Our results are given in Table 1, where the value R_{tr} and R_{sen} are expressed as a fraction of L, varied from 1 to 8.

Results in [14] indicate $N_{connect}$ is much larger than N_{cover} when $\frac{L}{R}$ is large. Our results show that N_{cover} is larger than $N_{connect}$ when $\frac{L}{R}$ is small. This can be verified by a computation on the probability that the boundary of a network area is covered, which gives a lower bound on the number of sensors required to guarantee a whole coverage with a high probability. Since in practice, it is possible to deploy sensors slightly larger than the monitored area, we also evaluate the number of sensors $N^{\star}_{cover}$ that are necessary to guarantee the coverage of an inner area of the network area; in particular, we set the width of the boundary as $0.05R$ and consider the coverage of the center square $[0.05R, L - 0.05R]^2$.

Table 1. Evaluations of $N_{connect}$ and N_{cover}

$\frac{L}{R_{tr}}$		2	3	4	5	6	7
$N_{connect}$	98%	18	40	75	125	186	262
$N_{connect}$	99%	19	40	78	138	231	347

$\frac{L}{R_{sen}}$		2	3	4	5	6	7
N_{cover}	98%	26	58	122	194	266	379
N_{cover}	99%	29	72	122	216	340	481
$N^{\star}_{cover}$	98%	24	62	102	169	238	338
$N^{\star}_{cover}$	99%	29	62	122	215	274	350

6.2 Event Reporting Latency

We denote our schemes in section 4 and 5 by Approach I and II respectively. The length D of each period is set as $2\sqrt{2}\frac{L}{R_{tr}}$, and the length of transitional phase in Approach I is set as 2. We consider two-dimensional random geometric networks, where sensors are distributed in $[0, L]^2$ uniformly and independently. We assume the sink is at location $[0, 0]$ and always active. Given L, we generate 50 events located uniformly and independently in the monitored area $[0.05R_{sen}, L - 0.05R_{sen}]^2$; a randomly generated value t, $0 \leq t < 1$, is associated with each event to represent a random position in a frame when this event occurs. The number p of sets of sensors varied from 2 to 10; for comparison purpose, we also consider non-duty cycled networks with the same number of active sensors, marked by “Non-duty-cycled” in the figures. Given L and p, 20 geometric random networks are generated. For each network, we compute the latency it takes to report each of the 50 events to the sink, and get the maximum, average and minimum latency; the average of these values over the 20 generated graphs are presented.

We present in Fig. 6 the results for networks with $L = 300$, 500, 700 meters given fixed $R_{tr} = R_{sen} = 100$ meters; note $N^{\star}_{cover} \geq N_{connect}$ in these scenarios and the latency does not depend on the interval between reporting phases. We also consider the case $R_{tr} \leq R_{sen}$: Fig. 7 shows results for networks with $R_{sen} = 140, 175$ meters for fixed $L = 700$ meters and $R_{tr} = 100$ meters; since in this case $N^{\star}_{cover} \leq N_{connect}$ and the latency of Approach I strongly depends on the interval between two reporting phases, it is more interesting to evaluate Approach II. Our simulation results verify our analyses in section 4 and 5. Furthermore, it is shown that, since the size of connected component stays unchanged, event reporting latency is stable when the number of sets is increased.

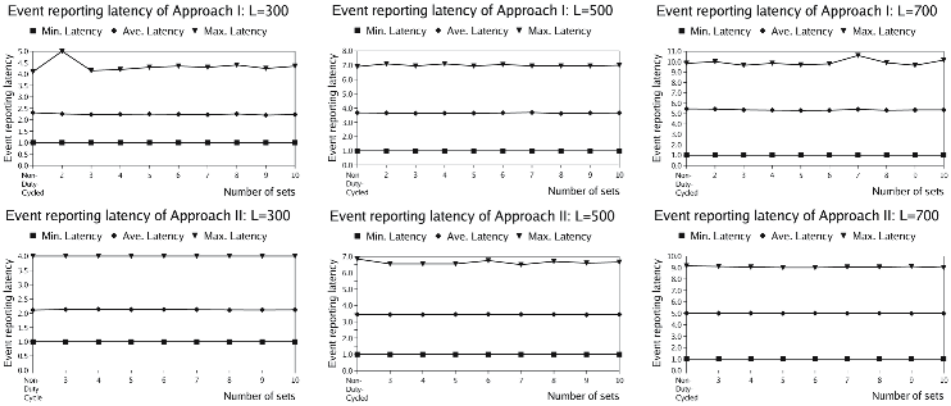


Fig. 6. Event reporting latency: $R_{tr} = R_{cov} = 100$ meters

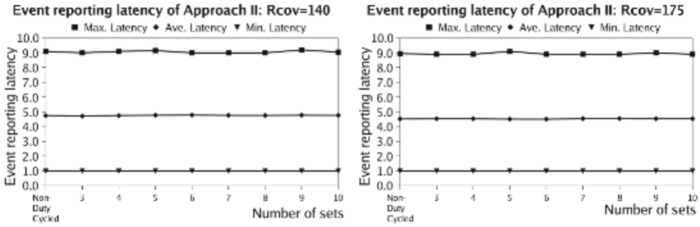


Fig. 7. Event reporting latency: $L = 700$ meters, $R_{tr} = 100$ meters

7 Conclusion

We present duty cycling with the focus on sensor networks for time-critical monitoring of an area, where sensors monitor events and send the information to a data collection node (called *sink*). In such networks, latency is considered as the delay elapsed between the time when *an event occurs* and the time when *the sink gets the report of this event*. We propose a formal definition of latency for such scenarios, and aimed to prolong network lifetime by scheduling periodic nodes' duty cycles while reserving small latency of event reporting. Our ideas are simple and achieve good performance, as shown in our analyses and simulations.

References

1. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
2. O. Dousse, P. Mannersalo, and P. Thiran. Latency of wireless sensor networks with uncoordinated power saving mechanisms. In *MobiHoc '04: Proc. 5th ACM international symposium on Mobile ad hoc networking & computing*, 2004.
3. O. Dousse, C. Tavoularis, and P. Thiran. Delay of intrusion detection in wireless sensor networks. In *MobiHoc '06: Proc. 7th ACM international symposium on Mobile ad hoc networking & computing*, 2006.
4. J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proc. 5th symposium on operating systems design and implementation (OSDI)*, 2002.
5. C. Hsin and M. Liu. Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms. In *IPSN'04: Proc. 3rd international symposium on Information processing in sensor networks*, 2004.
6. S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. Srivastava. Estimating clock uncertainty for efficient duty-cycling in sensor networks. In *SenSys '05: Proc. 3rd international conference on Embedded networked sensor systems*, 2005.
7. S. Ganeriwal, R. Kumar, and M. Srivastava. Timing sync protocol for sensor networks. In *SenSys '03: Proc. 3rd international conference on Embedded networked sensor systems*, 2003.
8. C. Jones, K. Sivalingam, P. Agrawal, and J. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.
9. E. Jung and N. Vaidya. An energy efficient mac protocol for wireless lans. In *Proc. IEEE INFOCOM*, 2002.
10. S. Kumar, T. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *MobiCom '04: Proc. 10th annual international conference on Mobile computing and networking*, 2004.
11. G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *Proc. 18th International Parallel and Distributed Processing Symposium*, 2004.
12. G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel. Delay efficient sleep scheduling in wireless sensor networks. In *INFOCOM 2005*, 2005.
13. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, 2001.
14. S. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties. In *Proc. the ACM-SIAM Symposium on Discrete Algorithms*, 2005.
15. T. Philips, S. Panwar, and A. Tantawi. Connectivity properties of a packet radio network model. *IEEE Trans. Inform. Theory*, 35(5):1044–1047, 1989.
16. P. Santi and D. Blough. The critical transmitting range for connectivity in sparse wireless ad hoc networks. *IEEE Trans. Mobile Computing*, 2(1):25–39, 2003.

17. S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. In *INFOCOM*, 2003.
18. S. Singh and C. Raghavendra. PAMAS: Power aware multi-access protocol with signaling for ad hoc networks. *SIGCOMM Comput. Comm. Rev.*, 28(3):5–26, 1998.
19. W. Su and I. Akyildiz. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13(2):384–397, 2005.
20. T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proc. 1st international conference on Embedded networked sensor systems*, 2003.
21. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *SenSys '03: Proc. 1st international conference on Embedded networked sensor systems*, 2003.
22. G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *SenSys '05: Proc. 3rd international conference on Embedded networked sensor systems*, 2005.
23. W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proc. IEEE INFOCOM*, 2002.
24. W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. Technical Report ISI-TR-567, USC, Jan. 2003.
25. R. Zheng, J. C. Hou, and L. Sha. Asynchronous wakeup for ad hoc neworks. In *ACM MobiHoc*, 2003.

Improving Bluetooth EDR Data Throughput Using FEC and Interleaving

Ling-Jyh Chen[1], Tony Sun[2], and Yung-Chih Chen[1]

[1] Institute of Information Science, Academia Sinica,
Taipei 11529, Taiwan
{cclljj, ycchen}@iis.sinica.edu.tw
[2] Department of Computer Science, UCLA,
Los Angeles, CA 90095, USA
tonysun@cs.ucla.edu

Abstract. Wireless communication is inherently vulnerable to errors from the dynamic wireless environment. Link layer packets discarded due to these errors impose a serious limitation on the maximum achievable throughput in the wireless channel. To enhance the overall throughput of wireless communication, it is necessary to deploy link layer transmission schemes that is robust to the errors intrinsic to the wireless channel. In this paper, we investigated the impact of three link layer enhancement techniques on the new Enhanced Data Rate (EDR) mode detailed in the new Bluetooth spec v2.0. We first studied the APT algorithm, and used it to obtain the optimal packet type for different bit error rates. We then evaluated FEC/IFEC coding schemes for the new EDR packet types, assessed their ability to alleviate the impact of burst errors, and discussed the tradeoffs. Using analysis and simulation, we show that the performance of the new Bluetooth EDR mode can be significantly improved when FEC and/or IFEC techniques are employed.

1 Introduction

Bluetooth [1] is a short-range radio technology specified in the IEEE 802.15.1 standard [2]. Bluetooth was first aimed as a cable replacement technology for the numerous proprietary cables, providing a universal wireless interface for different devices to communicate with one another. The low cost, low power, and small footprint of Bluetooth chips have fueled the popularity of Bluetooth technology, which has emerged as a good solution to interconnect different devices to form the so-called Personal Area Networks (PANs) [3].

Bluetooth operates in the unlicensed 2.4GHz ISM (Industrial-Scientific-Medical) frequency band, which is also utilized by various wireless and radio technologies, such as IEEE 802.11b/g standard [4], IEEE 802.15.4 standard [5], cordless telephones, and even microwave ovens. Although Bluetooth employs Frequency Hopping Spread Spectrum (FHSS) technique to alleviate the interferences caused by other wireless technologies, coexisting in the crowded 2.4GHz frequency band is still challenging for Bluetooth in general.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 724–735, 2006.

To improve Bluetooth's throughput performance in the crowded 2.4Ghz ISM frequency band, several approaches have been proposed in the literature. For instance, Golmie et al have proposed a Bluetooth Interference Aware Scheduling (BIAS) scheme to determine the frequency hopping pattern based on a *Frequency Usage Table* [6]. Since then, Adaptive Frequency Hopping (AFH), a BIAS-like approach, has been included in the Bluetooth Specification v1.2 [7][8]. In AFH, the Bluetooth channels are classified into two groups: one group is termed *unused* which had better not to be used (i.e., these channels might have been heavily interfered), the other is termed *used* which is to be used for transmission. A mapping function is then employed by AFH to uniformly map *unused* channels to the *used* channels. As a result, AFH can avoid the heavily interfered channels that is not to be used (the *unused* channels), and improve data throughput.

However, BIAS and AFH schemes would only improve Bluetooth link performance when there exists a portion of channels that are not interfered by other wireless technologies. If most channels are interfered by other radio sources, BIAS and AFH would still experience lower SNR (Signal to Noise Ratio), higher bit error rate, and overall lower data throughput performance. To resolve this problem, Chen et al have proposed an Adaptive Packet Type (APT) scheme [9], which adapts Bluetooth link layer packet type to the optimal one based on the measured bit error rate and the developed analytical model.

In addition to the methods described above, another link layer enhancement technique, called Interleaving FEC (IFEC) [10], has been proposed to enhance Bluetooth data throughput amid wireless burst errors. Since wireless errors are bursty in nature, by combining Forward Error Correction (FEC) and interleaving techniques, it became effective in correcting minor error in FEC codewords and in alleviating the impact of contiguous bit errors. Moreover, since IFEC interleaves the data in the bit level instead of the packet level, the additional latency caused by interleaving is minimized.

In this paper, our goal is to study the impacts of these three error alleviating schemes, namely APT, FEC, and IFEC, for the new Enhanced Data Rate (EDR) packet types, as defined in the Bluetooth Specification v2.0 [11]. It is the interest of this paper to illustrate/compare the potential benefits to Bluetooth V2.0 with the three enhancement schemes described. First, using APT algorithm, we analyzed and obtained the optimal EDR packet types under different bit error rates. We then applied FEC/IFEC coding schemes to the EDR packet types, and evaluated the packet error rate and the maximum achievable data throughput of the various FEC/IFEC enabled EDR packet types. The results confirmed that FEC coding does indeed provide more robust data throughput, and IFEC coding can effectively alleviate the impact of burst errors for the new Bluetooth EDR mode.

The rest of the paper is organized as follows. In section 2, we present an overview of Bluetooth technology. We present and evaluate three link layer enhancement strategies for Bluetooth EDR mode, namely APT, FEC, and IFEC in section 3, 4, and 5. Section 6 concludes the paper.

2 Background

2.1 Bluetooth Overview

Bluetooth is a short-range, low cost, and low power consumption radio technology operating in the unlicensed 2.4GHz ISM (Industrial-Scientific-Medical) frequency band. It employs FHSS (Frequency Hopping Spread Spectrum) technique and implements stop and wait ARQ (Automatic Repeat reQuest), CRC (Cyclic Redundancy Check), and FEC (Forward Error Correction) to achieve high reliability on the wireless links and to alleviate the interferences caused by other radio technologies, such as 802.11b [4], cordless phones, and microwave ovens. The FEC scheme used in Bluetooth is a (15, 10) shortened Hamming code, in which each block of 10 information bits is encoded into a 15 bit codeword, and it is capable of correcting single bit error in each block.

Bluetooth units can be connected to other Bluetooth units to form a piconet, which can support up to eight active units. One of the units in a piconet acts as a master and the other units act as slaves. All the data/control packet transmissions are coordinated by the master. Slave units can only send in the slave-to-master slot after being addressed in the preceding master-to-slave slot. Each slot lasts for 625 microseconds.

For real-time data such as voice, Synchronous Connection Oriented (SCO) links are used. For data transmission, Asynchronous Connectionless Link (ACL) is used. There are several ACL packet types, differing in packet length (and consequently, data transmission rate) and whether it makes use of FEC coding. Table 1 depicts the different ACL basic packet types and their respective properties.

Table 1. Basic Data Types in Bluetooth ACL Mode

Mode	FEC	Packet		Symmetric	Asymmetric	
		Size	Length	Throughput	Throughput	
		(bytes)	(slots)	(Kbps)	(Kbps)	
DM1	Yes	17	1	108.8	108.8	108.8
DM3	Yes	121	3	258.1	387.2	54.4
DM5	Yes	227	5	286.7	477.8	36.3
DH1	No	27	1	172.8	172.8	172.8
DH3	No	183	3	390.4	585.6	86.4
DH5	No	339	5	433.9	723.2	57.6

Note that, in the symmetric connection mode, both master and slave nodes will occupy the same amount of Bluetooth time slots (625 microseconds in each time slot); whereas in the asymmetric connection mode, the Bluetooth link will occupy 1/3/5 time slots (for DM1/DM3/DM5 or DH1/DH3/DH5 mode) in one direction of this link and only one time slot in the opposite direction.

2.2 Bluetooth Enhanced Data Rate (EDR)

The most recent version of Bluetooth specification proposes a set of new baseband packet types, called Enhanced Data Rate (EDR) [11]. EDR achieves higher data throughput by using Phase Shift Keying (PSK) modulation, instead of Gaussian Frequency Shift Keying (GFSK) modulation, which was used for Bluetooth basic packet types. Similar to other basic Bluetooth packet types, the new EDR packet types occupy 1/3/5 time slots, and each time slot is 625 microseconds length. For ACL mode, the new packet types are called 2DH1/2DH3/2DH5 when $\pi/4$-DQPSK modulation is employed, or 3DH1/3DH3/3DH5 when 8DPSK modulation is used. Unlike the basic packet types, Bluetooth EDR does not provide FEC enabled packet types (i.e. DM series packet types). Table 2 shows the properties and the maximum achievable data rates of EDR packet types in ACL mode.

Table 2. Bluetooth ACL Mode EDR Data Types

Mode	Modulation	Payload Size (bytes)	Symmetric Throughput (Kbps)	Asymmetric Throughput (Kbps)	
2DH1	$\pi/4$-DQPSK	54	345.6	345.6	345.6
2DH3	$\pi/4$-DQPSK	367	782.9	1174.4	172.8
2DH5	$\pi/4$-DQPSK	679	869.7	1448.5	115.2
3DH1	8DPSK	83	531.2	531.2	531.2
3DH3	8DPSK	552	1177.6	1766.4	235.6
3DH5	8DPSK	1021	1306.9	2178.1	177.1

It should also be mentioned that EDR packet types still use GFSK in the packet headers and use the same process for link establishment. Therefore, EDR devices are backward compatible with legacy Bluetoth devices. In fact, the symbol transmission rate (1 mega-symbol per second) remains unchanged in the new specification, the PSK modulation simply allows each symbol in the packet payload to carry more bits.

2.3 Wireless Error Model

In reality, wireless channel errors are usually burst and dependent in occurrences, rather than independently and identically distributed. To capture such behavior in the wireless channel, a Discrete Time Markov Chain (DTMC) model depicted in Fig. 1, commonly known as the Gilbert-Elliott model [12][13], is used to model the true nature of wireless channel errors. The Gilbert-Elliott model consists of two states, namely the *Good* state and the *Bad* state. Events originated from these states are denoted as g and b respectively. Four transition probabilities, P_{gg}, P_{gb}, P_{bg}, and P_{bb}, are then given and they specify the state transition probabilities. For example, P_{gb} defines the probability of transition from the good state to bad state, and P_{bb} defines the probability of remaining in bad

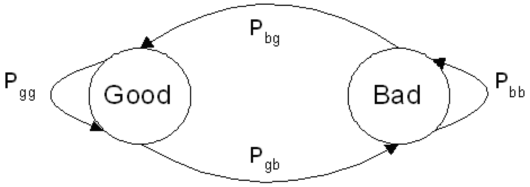


Fig. 1. Markov Model for Wireless Link

state, which actually reflects the degree of burst errors. The Markov chain is ergodic with stationary probabilities $P_g = \frac{1-P_{bb}}{1-P_{bb}+P_{gb}}$ and $P_b = \frac{P_{gb}}{1-P_{bb}+P_{gb}}$, and P_b is the average bit error rate (BER) [14].

3 Bluetooth EDR Enhancements (I): Adaptive Packet Type (APT)

We first perform analysis to determine the "optimal" packet type, which yields the maximum data throughput, using the APT algorithm described in [9]. We assume the random error model is used for the wireless channel, and the bit error rate is b. The packet error rate (PER, denoted as p) of Bluetooth DH packet types is given by Eq. 1 and the PER of Bluetooth DM packet types is given by Eq. 2, where s is the packet size (bits) [9].

$$p = 1 - (1-b)^s \tag{1}$$

$$p = 1 - ((1-b)^{15} + 15b(1-b)^{14})^{s/15} \tag{2}$$

The maximum achievable data throughput, T, of each Bluetooth packet type is then obtained by:

$$T = \frac{s(1-p)}{(n+1) \times 625\mu s} \tag{3}$$

where n is the length of packet in Bluetooth slots, and 625 μs is the length of a Bluetooth slot.

Using Eq. 3, we plot the maximum achievable throughput of Bluetooth packet types versus bit error rates in Figure 2. The "optimal" packet types for different BERs are then determined by selecting the packet type that gives the largest T for that BER. The thresholds for selecting the "optimal" packet type are shown in Table 3.

Since Bluetooth has a built-in system function call, *Get_Link_Quality*, to obtain the ongoing link quality information, which can be easily converted to bit error rate, the APT algorithm thus adapts the Bluetooth link layer packet type to the optimal one accordingly. As a result, systems with APT capabilities are able to achieve higher data throughput in wireless networks.

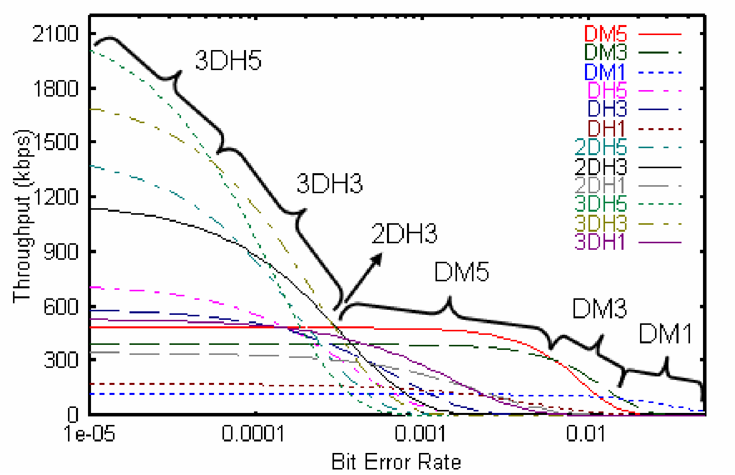


Fig. 2. Analytical results of Bluetooth EDR throughput of different ACL packet types

Table 3. Calculated threshold for Bluetooth EDR for Adaptive Packet Type

Packet Type	BER Range
DM1	0.0157813 < BER
DM3	0.0060695 < BER < 0.0157813
DM5	0.0003034 < BER < 0.0060695
2DH3	0.0002758 < BER < 0.0003034
3DH3	0.0000558 < BER < 0.0002758
3DH5	BER < 0.0000558

4 Bluetooth EDR Enhancements (II): FEC Coding

FEC coding has been well studied [15][16][17], and it has been implemented in many wireless standards, e.g. Bluetooth [1] and IEEE 802.11a [4]. FEC coding is the preferred error-control scheme to fight random losses, even though perfect recovery cannot be guaranteed. However, the main drawback of FEC technique is the incurred redundancy overhead, which may degrade the effective data throughput of a link.

FEC coding has been implemented by Bluetooth basic packet types, i.e. DM packet types, which employs a (15, 10) shortened Hamming code to protect packets from transmission errors. More specifically, for each DM mode packet, each block of 10 information bits is encoded into a 15 bit codeword, and the FEC scheme is able to correct any single bit error in each block. Analysis has shown that the deployment of FEC coding in DM mode enhances the transmission performance when the bit error rate surpasses a certain threshold [18].

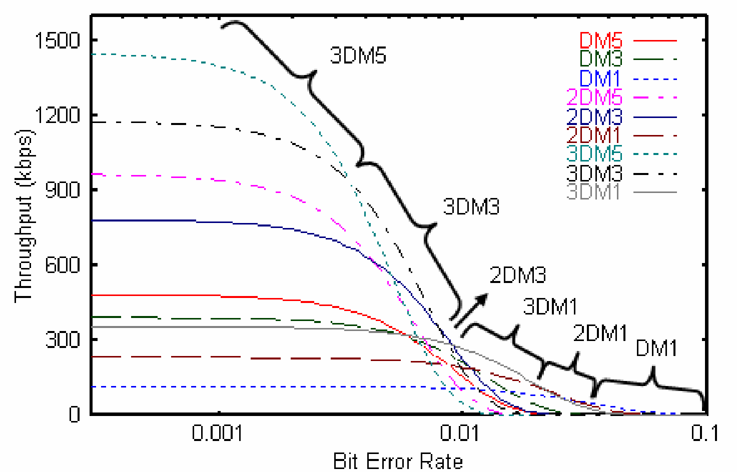


Fig. 3. Analytical results of Bluetooth EDR throughput of different ACL packet types with FEC coding enabled

Table 4. Calculated threshold for Bluetooth EDR (with FEC enabled DM packet types) for Adaptive Packet Type

Packet Type	BER Range
DM1	0.0285627 < BER
2DM1	0.0220129 < BER < 0.0285627
3DM1	0.0090206 < BER < 0.0220129
2DM3	0.0079932 < BER < 0.0090206
3DM3	0.0034665 < BER < 0.0079932
3DM5	0.0000496 < BER < 0.0034665
3DH5	BER < 0.0000496

Here, we propose to apply the same FEC coding scheme to Bluetooth EDR packet types, and name the resulting packet types 2DM1/3/5 and 3DM1/3/5 respectively. Since the (15, 10) FEC coding is employed in 2DM and 3DM series packet types, the effective data payload size becomes 2/3 of the corresponding 2DH and 3DH EDR packet types. Similar to the analysis presented previously for DM packet types, we model the packet error rate of the new packet types using Eq. 2 and maximum achievable data throughput using Eq. 3. We plot the maximum achievable throughput of FEC enabled packet types versus the bit error rate in Figure 3.

Combining the analytical results in Figure 2 and Figure 3, we apply APT algorithm to obtain the optimal packet type for different bit error rates. We show the calculated threshold in Table 4.

5 Bluetooth EDR Enhancements (III): Interleaved FEC Coding

So far, we have evaluated two link layer enhancement schemes (i.e. APT and FEC) for Bluetooth EDR mode. However, as pointed out in [10], APT and FEC techniques improve the effective data throughput of Bluetooth links only when the wireless errors are identically and independently distributed (i.e. using random error model). When wireless errors are mostly bursty in presence, these techniques fail to provide good data throughput unless interleaving technique is applied [10].

In this section, we used the Interleaved FEC (IFEC) coding technique [10] to Bluetooth EDR packet types, and the resulting packet types are called DMI1/3/5, 2DMI1/3/5, and 3DMI1/3/5 respectively. Figure 4 conceptually illustrates the difference between IFEC and FEC schemes.

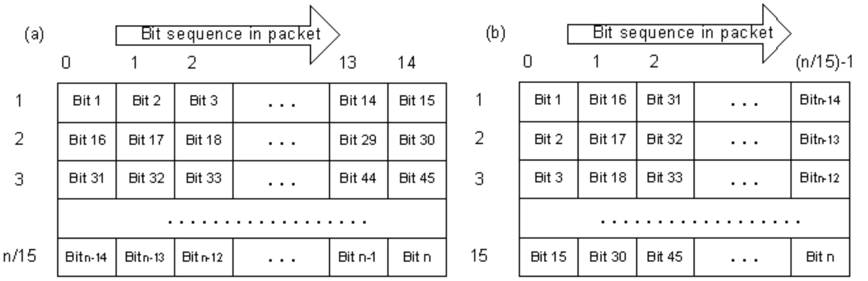
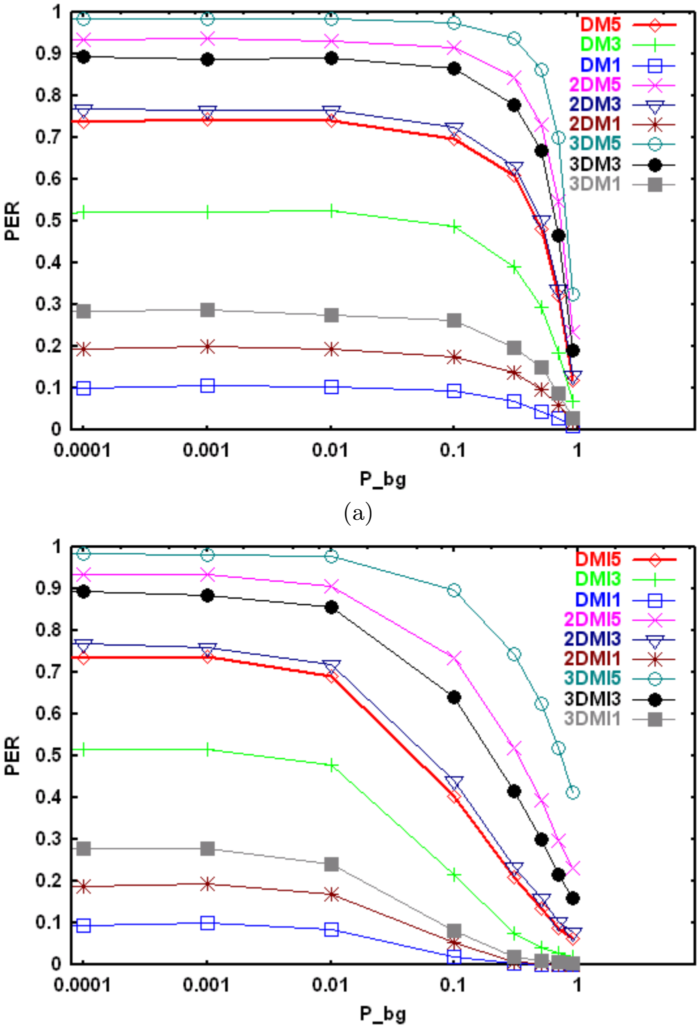


Fig. 4. The proposed link layer bit arrangement for Bluetooth EDR: (a) FEC coding (b) Interleaved FEC coding

Specifically, for the FEC scheme, a packet of size n bits is segmented into several $n/15$ bits blocks, and each block is a FEC codeword consisting of 10 data bits and 5 FEC coded bits. On the other hand, for the IFEC scheme, each packet is divided into 15 blocks with $n/15$ bits each. It then constructs the first 15 bit FEC codeword by applying FEC coding to the first bit of each block, the second 15 bit codeword is constructed by applying FEC coding to the second bit of each block, and so on and so forth. Therefore, each codeword in IFEC scheme inherits the ability to correct single bit error like it was in the FEC scheme, but it is more robust to burst errors since the 15-bit codeword contains no consecutive bits.

Using Gilbert-Elliott model (i.e. burst error model), we let $P_{gb} = 0.0005$ and vary P_{bb}, which determines the burst level of wireless errors, from 0.9 to 0.9999. We use Monte Carlo method to simulate the packet error rates of DM and DMI series packet types versus different bit error rates. The results are shown in Figure 5.

In Figure 5, the results clearly show that DMI packets are more robust against burst errors than DM packets. For instance, the packet error rates of DM packets



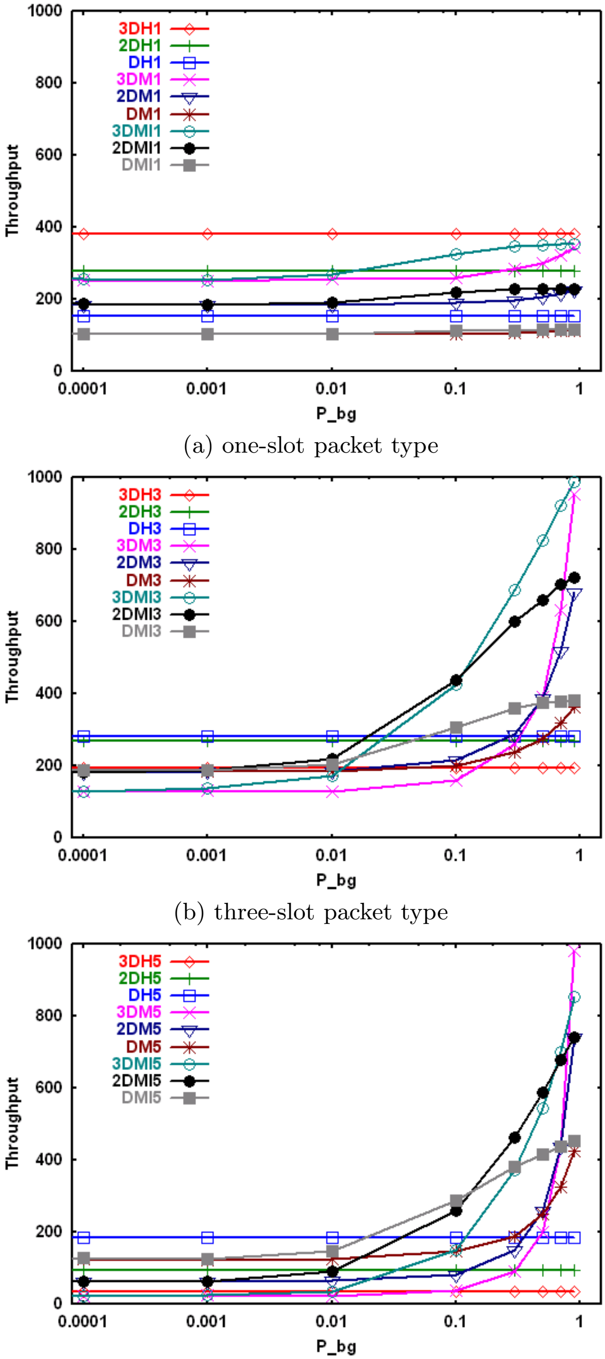
(a)

(b)

Fig. 5. Packet error rates versus different bit error rates: (a) FEC coding; (b) IFEC coding ($P_{gb} = 0.0005$ and $P_{bg} = 1 - P_{bb}$)

increase rapidly and become converged when P_{bb} becomes as large as 0.9 (i.e. $P_{bg} = 0.1$); whereas the PER of DMI packets increases more moderately as P_{bb} increases and becomes converged after P_{bb} becomes larger than 0.999.

We suppose P_{gb} is fixed at 0.0005, the first bit of each packet is 'good', and s is the effective payload size (i.e. excluding FEC overhead) of DM/DMI packets. Using Eq. 3 and the analytical PER results of Figure 5, we plot the maximum achievable data throughput of DM and DMI packet types of different burst levels in Figure 6.



(a) one-slot packet type

(b) three-slot packet type

(c) five-slot packet type

Fig. 6. Maximum achievable data throughput of DM and DMI packet types with different burst error levels, given P_{gb}=0.0005

Figure 6 shows that the data throughput of DH series packet types is very consistent regardless of burst error levels (P_{bb}). This is explained by the fact that DH packets are not protected by FEC/IFEC coding, any single bit error in a DH type packet will result in an error of the whole packet, and the packet would be dropped.

Moreover, the results also show that DM and DMI packet types outperform the corresponding DH packet types only when the burst error level is moderate. The results match our expectation since DM and DMI packets are able to improve link performance only when the wireless errors are recoverable (i.e. at most one bit error in each 15-bit codeword). If wireless errors are very slight or extremely serious, the overhead from FEC redundancy in DM and DMI packets degrades the effective data throughput instead.

In addition, the results also confirm previous results presented in [10], i.e. DMI series packet types outperform the corresponding DM series packet types when the burst error model is used for the wireless channel. This is due to the fact that, in IFEC scheme, most contiguous bit errors are distributed into different interleaved FEC codewords. As a result, since the deployed FEC scheme can correct single bit error in a codeword, DMI is able to recover the packet from burst bit errors as long as there are no multiple bit errors in any of 15-bit codewords.

6 Conclusion

In this paper, we studied the impact of three various link layer enhancement techniques on the new Enhanced Data Rate (EDR) mode detailed in Bluetooth spec v2.0. It is the interest of this paper to illustrate/compare the potential benefits to Bluetooth v2.0 with the three enhancement schemes described, namely APT, FEC, and IFEC. We first employed APT algorithm to obtain the optimal packet type for various bit error rates. We then evaluated FEC coding scheme for EDR packet types, which provided more robustness against wireless errors. Moreover, we evaluated IFEC coding, which is a technique combining the benefits of interleaving and FEC, and assessed its ability to alleviate the impact of burst errors in wireless channels. Using analysis and simulation, we show that the performance of the new Bluetooth EDR mode can be significantly improved when FEC and/or IFEC techniques are utilized.

References

1. “Bluetooth specifications v1.1,” http://www.bluetooth.com.
2. “Ieee 802.15 wpan task group 1 (tg1),” http://www.ieee802.org/15/pub/TG1.html.
3. P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla, “Bluetooth: An enabler for personal area networking,” *IEEE Network Magazine*, Sept/Oct 2001.
4. “Ieee 802.11, the working group setting the standards for wireless lans,” http://grouper.ieee.org/groups/802/11/.

5. "Ieee 802.15.4 wpan-lr task group," http://www.ieee802.org/15/pub/TG4.html.
6. N. Golmie, N. Chevrollier, and I. Elbakkouri, "Interference aware bluetooth packet scheduling," in *GLOBECOM*, 2001.
7. "Bluetooth specifications v1.2," http://www.bluetooth.com.
8. N. Golmie, N. Chevrollier, and O. Rebala, "Bluetooth and wlan coexistence: Challenges and solutions," *IEEE Wireless Communications Magazine*, December 2003.
9. L.-J. Chen, R. Kapoor, M. Y. Sanadidi, and M. Gerla, "Enhancing bluetooth tcp throughput via link layer packet adaptation," in *IEEE ICC*, 2004.
10. L.-J. Chen, T. Sun, M. Y. Sanadidi, and M. Gerla, "Improving wireless link throughput via interleaved fec," in *The Ninth IEEE Symposium on Computers and Communications*, 2004.
11. "Bluetooth specifications core v2.0," http://www.bluetooth.com.
12. E. O. Elliott, "Estimates of error rates for codes on burst-error channels," *Bell Syst. Tech. Journal*, vol. 42, September 1963.
13. E. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. Journal*, vol. 39, pp. 1253–1266, September 1960.
14. M. Zorzi and R. R. Rao, "Error control and energy consumption in communications for nomadic computing," *IEEE Trans. Computers*, vol. 46, pp. 279–289, 1997.
15. C. Barakat and E. Altman, "Bandwidth tradeoff between tcp and link-level fec," in *IEEE ICN*, July 2001.
16. J.-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive fec-based error control for internet telephony," in *IEEE Infocom*, 1999.
17. P. Frossard, "Fec performance in multimedia streaming," *IEEE Communication Letters*, vol. 5, pp. 122–124, 2001.
18. M. Ju, C. Park, D. Hong, K. Youn, and J. Cho, "Packet selection scheme based on a channel quality esti-mation for bluetooth systems," in *The 5th International Symposium on Wireless Personal Multimedia Communications*, 2002.

Information-Accuracy-Aware Jointly Sensing Nodes Selection in Wireless Sensor Networks

Huifang Li[*], Shengming Jiang, and Gang Wei

School of EIE, SCUT, GuangZhou, China
koalahfli@hotmail.com

Abstract. A key issue in wireless sensor networks (WSNs) is to select a set of sensors to join sensing task under some physical resource constraints while achieving a required information accuracy. This paper introduces a novel idea for information-accuracy-aware jointly sensing nodes selection based on a derived information accuracy model which formulates an explicit relationship between information accuracy and the number and position of jointly sensing nodes. We aim at eliminating the unnecessary transmission to minimize energy consumption while maximizing information accuracy, which is formulated as a joint optimization of information accuracy and energy consumption. In the proposed algorithm, a node is selected to join a sensing task based on its information accuracy gain and consumed energy. This allows a WSN to efficiently distribute sensing tasks given a limited energy supply. Simulation results have demonstrated that our algorithm improves the performance of joint optimization between information accuracy and energy consumption than a random node selection.

1 Introduction

Explosive growth in embedded computing and rapid advances in low power wireless networking technologies fuel the development of wireless sensor networks (WSNs). The primary task of a WSN is to collect information from a physical environment in order to answer a set of user queries or support other decision making functions. Typical high-level information processing tasks for a WSN include detection, tracking, or classification of physical phenomena of interest such as people, vehicles, fires and seismic events [1]. Although the tasks for a WSN can be varied, a common important problem is to find an efficient way to collect information and send it to the sink subject to some resource constraints such as limited on-board battery power and network communication bandwidth. The first step of this problem is to select a proper set of jointly sensing nodes for a given task, which is the focus of this paper.

Jointly sensing nodes selection belongs to the problem of sensor management which is a new research area that recently received some attention from a variety of perspectives [2]. However, most of them have focused on providing energy-efficiency only. For example, paper [3] demonstrates how to reduce unnecessary sensing data by intelligently selecting sensors for the quality of service with reliability. However, we think that sensor selection should consider the trade-off between information accuracy and

[*] Corresponding author.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 736–747, 2006.

energy-efficiency other than just trying to provide energy-efficient mechanism to prolong the lifetime of a WSN. Literature [4] and [5]) have reported some approaches on how to select a proper set of jointly sensing nodes for maximizing information collection while minimizing resource usage. However, they address this problem from information theory without addressing the constraint of information accuracy.

This article proposes an information accuracy model by exploiting the correlation between sensor observations to guide the sensor selection in WSNs. The contributions of this paper are twofold: (1) An information accuracy model is derived for a class of WSN applications such as event features detection, which includes an quantitive definition of information accuracy and a theoretical model on information accuracy versus jointly sensing nodes. One important result derived from this model is that the information accuracy may decrease as the number of jointly sensing nodes increase due to the unavoidable observation noise. (2) The paper formulates jointly sensing nodes selection problem as a joint optimization of information accuracy and energy-efficiency and describes a concrete algorithm of information-accuracy-aware jointly sensing nodes selection. With this algorithm, the tradeoff between information accuracy and energy consumption can be estimated without the need to communicate sensed data. Despite the fact that in this paper we focus on events' features detecting for illustration, The proposed notion of information accuracy is common to all kinds of sensing problems.

The rest of the paper is organized as follows. Section 2 gives a quantitive definition of information accuracy and derives a theoretical model on information accuracy versus jointly sensing nodes. The discussions on this model can help understand the relationship between information accuracy and different sensor selection parameters such as the number and position of jointly sensing nodes. Section 3 formulates the problem of information-accuracy-aware jointly sensing nodes selection, and develop a concrete algorithm for information-accuracy-aware sensor selection in cluster-based WSNs. Section 4 presents our simulation results with some analysis. Section 5 concludes and discusses possible extensions of the current algorithm.

2 A Model on Information Accuracy and Jointly Sensing Nodes

In this section, we first give a definition of information accuracy, and then derive a theoretical model on the relationship between information accuracy and jointly sensing nodes. Some notations used here are defined below.

- S: an event source to be sensed.
- $\hat{S}$: estimation of S.
- S_i: real value of S to be sensed by node i if no noise is present.
- $\hat{S}_i$: estimation of S_i.
- X_i: observed sample of S_i by node i.
- M: number of jointly sensing nodes.
- $d_{S,i}$: distance between S and node i.
- $d_{i,j}$: distance between nodes i and j.

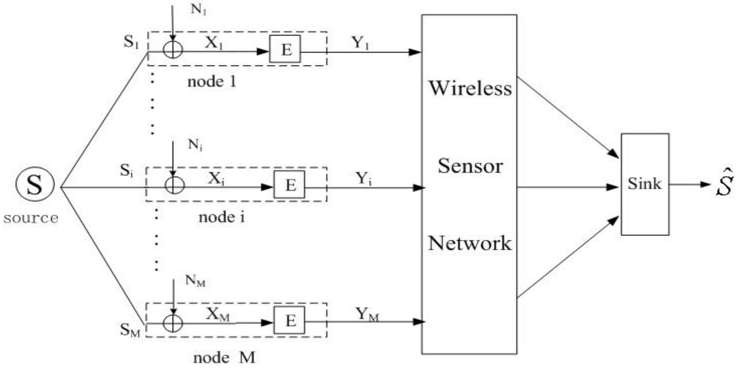


Fig. 1. Data gathering in WSNs: S_1... S_i...S_M are physical values of S at different spatial coordinates, N_i is the observation noise at node i, E denotes the encoding process at each node, $\hat{S}$ is the estimation of S at the sink

2.1 Information Gathering Problem Formulation and a Definition of Information Accuracy

The process of information gathering in a WSN can be illustrated in Fig. 1, which depicts a mathematical abstraction of a WSN consisting of M individual nodes that jointly make synchronous, correlated, noisy measurements of the same physical process S (e.g., temperature). Individual sensors are typically lightweight devices that have very limited battery power. In general, inter-sensor communication has heavy power and protocol overhead that strongly discourages information exchange between sensors. The goal of this WSN is to communicate sensor observations to a sink in a manner that allows the sink to form the best possible estimate of the physical process with respect to a required information accuracy of estimation. Note that the sink is only a logical entity and can also be one of the other sensors which acts as a "cluster-head" with different sensors taking turns. The sink is in the charge of collecting all the observations from M nodes to calculate the estimation of source ($\hat{S}$). Generally, since both S and $\hat{S}$ are random variables, the estimation quality can be measured by mean squared error, i.e.,

$$\overline{D} = E[d(S,\hat{S})] = E[\{S - \hat{S}\}^2]. \tag{1}$$

Here, E denotes the expectation operator. Then, a normalized information accuracy, I, is defined as follows:

$$I = 1 - \frac{\overline{D}}{E[S]} = 1 - \frac{E[(S - \hat{S})^2]}{E[S]}. \tag{2}$$

2.2 Information Accuracy Model Based on a Spatial Correlation Model of Sensed Data

As shown in Fig. 1, node i can observe a noisy version of S_i, i.e., X_i, which is given by

$$X_i = S_i + N_i, \tag{3}$$

where N_i is the observation noise at node i. We assume that the observation noise in (3) is white Gaussian noise, i.e., $E[N_i] = 0, var[N_i] = \sigma^2_{N_i}$. Note that each node observes the source independently, it is reasonable to assume the observation noise is independent of each other. We assume that the event source is a zero mean Gaussian source and the sensor nodes share a wireless additive white Gaussian noise (AWGN) channel. Since uncoded transmission is optimal for the point-to-point transmission of zero mean Guassian sources and AWGN channels [6], we adopt it here. Then, Y_i can be represented as a scaled version of X_i according to power constraint P:

$$Y_i = \sqrt{\frac{P}{\sigma^2_S + \sigma^2_{N_i}}} X_i, \tag{4}$$

where σ^2_S is the variance of S and $\sigma^2_{N_i}$ is the variance of N_i. Y_i is then sent through a wireless relay path to the sink. For the scope of this paper, we are mainly interested in the effect of jointly sensing nodes on information accuracy. Hence, only one hop transmission is considered here. In this case, each Y_i is sent directly to the sink. The sink has to calculate the estimation of S_i, i.e., $\hat{S}_i$ in order to get $\hat{S}$. For the uncoded transmission, the minimum mean square error (MMSE) estimation is the optimal decoding scheme [7]. Hence, $\hat{S}_i$ is simply the MMSE estimation of Y_i and S_i, which is given by

$$\hat{S}_i = \frac{E[S_i Y_i]}{E[Y_i^2]} Y_i = \delta_i (S_i + N_i), \tag{5}$$

where $\delta_i = \frac{E[S_i^2 + S_i N_i]}{E[(S_i + N_i)^2]}$. Since the sink decodes each Y_i using MMSE estimator, $\hat{S}$ can be calculated as the average of all $\hat{S}_i$ as follows:

$$\hat{S} = \frac{1}{M} \sum_{i=1}^{M} \hat{S}_i = \frac{1}{M} \sum_{i=1}^{M} \delta_i (S_i + N_i). \tag{6}$$

As shown in (6), the estimation of source $\hat{S}$ depends on the number of jointly sensing nodes and the characteristics of S as well as the observation noise.

Since the M jointly sensing nodes observe the same event source, these sensed data are spatially correlated. Such spatial correlation can be exploited to further analyse the relationship between information accuracy and jointly sensing nodes. Consider a Gaussian source with a zero mean and variance σ^2_S, i.e., $E[S] = 0$, $var[S] = \sigma^2_S$. Recall that S_i is the real value of S at the location of node i, so $S_1, \ldots, S_M$ are spatially correlated. These spatially correlated physical phenomena can be modeled as joint Gaussian random variables (JGRVs) as follows: $E[S_i] = 0$ and $var[S_i] = \sigma^2_S$, where $i = 1, \ldots, M$. The covariance between S and S_i is $cov[SS_i] = E[SS_i] = \sigma^2_S \rho_{(S,i)}$, where $\rho_{(S,i)}$ is the correlation coefficient between S and S_i. Similarly, $cov[S_i S_j] = E[S_i S_j] = \sigma^2_S \rho_{(i,j)}$, where $\rho_{(i,j)}$ is the correlation coefficient between S_i and S_j. Both $\rho_{(S,i)}$ and $\rho_{(i,j)}$ must be nonnegative and decrease monotonically with the distances $d_{S,i}$ and $d_{i,j}$, i.e., $\rho_{(S,i)} = 1$ when $d_{S,i}$=0, $\rho_{(i,j)} = 1$ when $d_{i,j} = 0$, $lim_{d_{S,i} \to \infty} \rho_{(S,i)} = 0$ and $lim_{d_{i,j} \to \infty} \rho_{(i,j)} = 0$. These correlation coefficients can be chosen according to the properties of S. Generally, the power exponential model, i.e., $\rho' = e^{-d/\theta} (\theta > 0)$, can be used as a correlation model for S and S_i as well as S_i and S_j, where d is the coherence distance denoting the correlation degree [8]. Then, $\rho_{(S,i)} = e^{-d_{S,i}/\theta}$ and $\rho_{(i,j)} = e^{-d_{i,j}/\theta}$.

Since WSNs are generally densely deployed and the observation region is usually small, it is reasonable to assume that the observation noise N_i is an independent identically distributed (i.i.d) Gaussian random variable with a zero mean and variance σ_N^2, i.e., $E\{N_i\} = 0, var\{N_i\} = \sigma_N^2$. Under the above assumptions, δ_i is found to be

$$\delta_i = \delta = \frac{\sigma_S^2}{\sigma_S^2 + \sigma_N^2}, \tag{7}$$

where $0 < \delta < 1$ is a constant. With δ and the above mentioned correlation model for S and S_i, the information accuracy in (2) is calculated as

$$I(M) = \frac{\delta}{M}\left(2\sum_{i=1}^{M} e^{-d_{S,i}/\theta} - 1\right) - \frac{\delta^2}{M^2}\sum_{i=1}^{M}\sum_{j\neq i}^{M} e^{-d_{i,j}/\theta}. \tag{8}$$

This equation shows that $I(M)$ is a function of M, $d_{S,i}$ and $d_{i,j}$, which means that $I(M)$ also depends on the position of jointly sensing nodes. Obviously, given an M, the achieved information accuracy decreases as $d_{S,i}$ increases and increases as $d_{i,j}$ increases, respectively.

According to (8), the increment of $I(M)$ is given by

$$\begin{aligned} \triangle\, I(M) &= I(M+1) - I(M) \\ &= \frac{\delta}{M+1}\left\{\frac{\delta(2M^2 - M - 1)}{M(M+1)}E_1 - 2E_2 - \frac{2\delta M}{M+1}E_3 + 2e^{-d_{S,M+1}/\theta} + \frac{1}{M}\right\}, \end{aligned} \tag{9}$$

where

$$E_1 = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{j\neq i}^{M} e^{-d_{i,j}/\theta}, E_2 = \frac{1}{M}\sum_{i=1}^{M} e^{-d_{S,i}/\theta}, E_3 = \frac{1}{M}\sum_{i=1}^{M} e^{-d_{i,M+1}/\theta}.$$

For $\triangle I(M) > 0$, we must have

$$2e^{-d_{S,M+1}/\theta} - \frac{2\delta M}{M+1}E_3 > 2E_2 - \frac{\delta(2M^2 - M - 1)}{M(M+1)}E_1 - \frac{1}{M}. \tag{10}$$

However, the above condition may not hold in certain cases. For example, consider the same statistical characteristics of S, N_i and W_i as used in Section 4 and four jointly sensing nodes are located at points $(6, 2)$, $(8, 4)$, $(6, 6)$, $(4, 4)$ and the source S at $(6, 4)$, respectively. If the fifth jointly sensing node is located at $(10, 4)$. Then, the left hand of (10) related to the new joint node is 0.0836 while the right hand related to the old nodes is 0.0947. As a result, $\triangle I(4) < 0$ in this case, which means $I(5) < I(4)$. This is mainly because the fifth node is so far away from S that the noise in its sensed data dominates its observation results and decreases the information accuracy. More numerical examples and discussion on this issue can be found in Section 4.

3 Information-Accuracy-Aware Jointly Sensing Nodes Selection Algorithm

With the relationship between information accuracy and jointly sensing nodes discussed in Section 2, we propose an information-accuracy-aware jointly sensing nodes selection algorithm in this section.

Consider a set of static sensor nodes deployed over a field and organized according to observation regions of events into non-overlapping clusters. The cluster header collects data from its cluster members, each of which can communicate with the cluster header directly. This is a reasonable assumption on WSN's applications since it is impossible or unnecessary for all the nodes in a large WSN to join a sensing task and only those close to the event source need to join data collection task. Hence, all the nodes in the observation region of an event source are closed to each others and can be grouped into the same cluster. Information-accuracy-aware jointly sensing nodes selection algorithm is performed within a cluster since only the nodes residing in the observation region can join sensing tasks. Cluster forming and cluster header selection are out of the discussion of this paper since there have a lot of algorithm on them such as [9].

In such cluster-based WSNs, the problem of jointly sensing nodes selection aims at finding a sensor set with maximum information accuracy gain at minimum energy consumption in each cluster. Thus, Information-accuracy-aware jointly sensing nodes selection can be formulated as finding a sensor set, $V = \{n_1, \dots, n_i, \dots, n_T\}$, in a cluster to minimize the total cost

$$C^T = \frac{\sum_{i=1}^{T} P_i}{I(T)}, \tag{11}$$

where $\sum_{i=1}^{T} P_i$ represents the total energy consumption of joining sensing task and $I(T)$ represents the total information accuracy contribution from the sensor set V. It is worth to point out that the information accuracy can be calculated only according to the number and position of jointly sensing nodes as shown in (8), hence, it can be estimated before each observation is actually observed. Therefore, the problem of sensor selection formulated by (11) can be settled before actual sensing tasks begins.

Although the individual energy consumption of each node in set V is independent of each other and the total energy consumption is additive, such characteristic of additivity does not hold in the case of total information accuracy due to the spatial correlation between observations as discussed in Section 2.2. In general, the increment of information accuracy of a new selected node, $\triangle I$, dependents on the position between the new selected node and previous selected jointly sensing nodes. For example, revisiting the increment of information accuracy equation (9) in Section 2.2, we find that the increment of information accuracy decreases as the average distance between the new joining sensing node and those already joining sensing nodes increases. Note that in WSNs, sensor observations are often spatially correlated. Intuitively, the observation of a sensor is not entirely new; it could be just part of what its neighbors have already reported. Hence, choosing nodes close to those already joining sensing nodes will result in little increment of information accuracy. This is an important property of WSNs, regardless of specific choices of information accuracy metric. The reason is that the closer the jointly sensing nodes are, the more redundancy in observations is and the fewer information is offered from observations.

We assume each node is aware of its own position through using a GPS device or other location services, and the cluster header has knowledge about the positions of its cluster members. Such knowledge can be established through local message exchange within a cluster during neighbor discovery and network initialization. Based on such knowledge alone, the sensor can compute its own information accuracy contribution. It is unnecessary for these nodes to take measurements and communicate the observation results with each other.

Now we describe information-accuracy-aware jointly sensing nodes selection algorithm in detail. The cluster header is in charge of the whole process of selection. Initially, the header sets $V = \phi$ and then selects node within the cluster to join V one by one with maximum information accuracy and minimum energy consumption. A global search algorithm is adopted within the whole cluster to achieve this goal. Negotiations between the cluster header and members are necessary in this algorithm. The process of such negotiations can be divided into several intervals and only one node will be selected during each interval. Hence, T intervals are needed in finding a set which is composed of T nodes. Particularly, in interval k, the cluster header first broadcasts a request of joining sensing task to all cluster members. The position of recently selected node in interval $k-1$ and the information accuracy $I(k-1)$ achieved by $k-1$ selected nodes are also attached to the request message. Then, each unselected node updates the information accuracy to $I(k)$ according to (8) by assuming that itself is the newly joining sensing node. Hence, the increment of information accuracy, i.e., $\triangle I^k = I(k) - I(k-1)$, can be work out at each unselected node. The updated information accuracy $I(k)$ will be sent to the cluster header together with the energy consumption of joining sensing task in each node if $\triangle I^k > 0$, which is described in detail in Fig. 2. Then, the header can calculated a total cost, C_l^k, for each unselected node l according to (11). At the end of this interval, the header will select one node n_k to join set V according to

$$n_k = arg\Big\{min\{C_l^k \leq C_0\}\Big\}, \tag{12}$$

where C_0 is a prespecified amount which controls the tradeoff between the energy consumption and the information accuracy. Low C_0 value favors energy conservation and energy-efficiency, while high C_0 allows more energy consumption in achieving more accurate information. This algorithm will be ended if the condition $C_l^k \leq C_0$ does not hold for any unselected node l within an interval k. Then, the cluster header reports the recently updated sensor set to the sink. This algorithm at each cluster member in interval k is summarized in Fig. 2.

4 Simulations and Analysis

Simulations were carried out to illustrate the relationship between information accuracy and jointly sensing nodes and validate the performance of information-accuracy-aware jointly sensing nodes selection algorithm. We simulated a sensor field of $2m \times 2m$ grid-based sensor topology with a fixed event source located in the center and a sink on the

```
$V_{pos}^k$: list of position information of $k$ selected nodes.
$L_k$: position information of the $k^{th}$ selected node.
$I(k)$: information accuracy with $k$ jointly sensing nodes.
$\triangle I^k$: increment of information accuracy in interval $k$.
1. Keeps idle until receives the request of joining sensing task $(k, L_{k-1}, I(k-1))$
      If (it has already been selected)
            keep idle ;
2. Updates the position information list to $V_{pos}^{k-1}$ with $L_{k-1}$;
3. Updates the information accuracy to $I(k)$ according to (8);
4. Lets $\triangle I^k = I(k) - I(k-1)$;
      If ($\triangle I^k > 0$)
            sends $I(k)$ attached with the energy consumption of joining sensing task
            to the cluster header;
5. go back to step 1.
```

Fig. 2. Information-accuracy-aware jointly sensing nodes selection algorithm at each node in interval k

edge. All the sensors in the network are within the observation region of the event source and each sensor can directly communicate to the sink. Such sensor layout is plotted in Fig. 3. We choose $\sigma_S^2 = 20$, $\theta = \{30, 70\}$ and $\sigma_N^2 = 2$ for the statistical characteristics of S, N_i.

Besides the number of jointly sensing nodes, their positions are also of interest. Hence, the formulated relationships between information accuracy and jointly sensing nodes are validated by effects of M, $d_{S,i}$ and $d_{i,j}$ on information accuracy. Our algorithm of sensor selection is evaluated in terms of the performance of reducing communication cost while achieving maximum information accuracy.

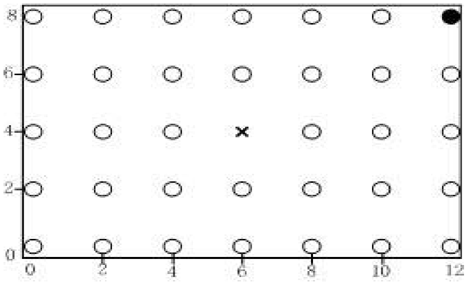


Fig. 3. Wireless sensor network topology in simulation: ○ means a senor, ● means a denoted sink, × means an event source

4.1 Relationship Between Information Accuracy and Jointly Sensing Nodes

Fig. 4 shows the information accuracy versus the number of jointly sensing nodes. Starting from 1 node, we increase the number of jointly sensing nodes one by one with random selection in Fig. 3. The numerical results are calculated with (8) for each value of

θ. As shown in this figure, for each value of θ the information accuracy stays practically constant when the number of jointly sensing nodes increases from 10 to 33. Hence, there is no necessary to choose many nodes joining sensing in achieving high information accuracy. It is also shown that information accuracy fluctuates when $M > 5$ for each value of θ, which confirms the theoretical analysis in Section 2.2 that the information accuracy may decrease as M increases. Moreover, information accuracy increases with increasing θ, this is because that higher θ means higher correlation between sensed data and S in the case of the same jointly sensing nodes.

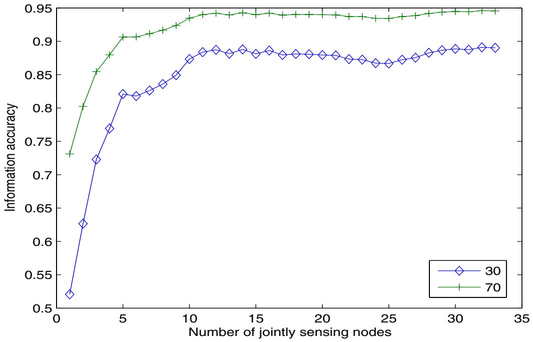


Fig. 4. Information accuracy versus the number of jointly sensing nodes with $\theta = \{30, 70\}$

Fig. 5(a) shows the information accuracy versus the average of $d_{S,i}$ (varying from 2 m to 4.7 m) with $M = 4$ and a fixed average $d_{i,j} = 3.2$ m. For each value of θ, the information accuracy decreases as average $d_{S,i}$ increases. This is because $d_{S,i}$ determines the correlation between S and the data sensed by node i. If node i is far away from S, its sensed data is less accurate, which results in lower information accuracy at the sink. Fig. 5(b) shows the effect of average $d_{i,j}$ on the information accuracy by varying the average $d_{i,j}$ from 2.2 m to 4.7 m and keeping average $d_{S,i}$ unchanged for $M = 4$. The results show that the information accuracy increases as average $d_{i,j}$ increases. This is because the further apart the jointly sensing nodes, the less correlated the sensed data are with less redundancy among the sensed data. In general, for a given number of jointly sensing nodes, the information accuracy decreases as $d_{S,i}$ increases and increases as $d_{i,j}$ increases, respectively.

4.2 Performance of the Proposed Algorithm

In order to perform the information-accuracy-aware jointly sensing nodes selection, a data collection cluster needs to be formed beforehand. Such cluster is usually composed of sensors located in the observation region of an event source. Based on the analysis of the relationship between information accuracy and jointly sensing nodes in Section 4.1, although some nodes apart from the event source still within the observation region, they offer little or even negative information accuracy gain in their sensed data. Hence,

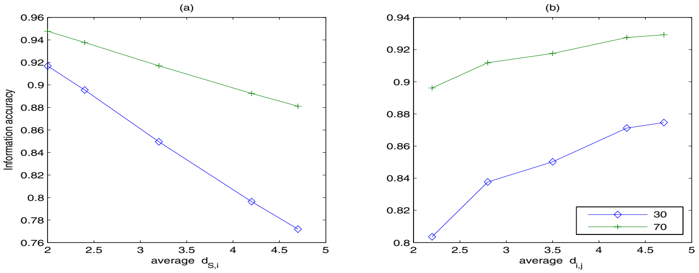


Fig. 5. Information accuracy versus the position of jointly sensing nodes with $\theta = \{30, 70\}$ in the case of 4 jointly sensing nodes

it is quite unnecessary to choose such sensors to join the data collection cluster. In general, knowledge about the distribution of an event source plays an important role in assisting our algorithm such as the decision of a cluster size. Therefore, a priori knowledge about a network or application should be exploited whenever possible to assist sensor selection in a WSN. Here, we choose a cluster of dimension $8m \times 8m$ with the event source being the center in our simulated WSN. The node at $(10, 8)$ is denoted as cluster header since it is the closest node to the sink as shown in Fig. 3. The energy consumption of each cluster member increases as the distance of the node and the sink increases, which is a reasonable assumption since the transmission energy is proportional to the transmission distance [10].

Table 1. Simulation results of proposed sensor selection algorithm with different constraints

	Proposed algorithm		Random selection	
C_0	total cost	information accuracy	total cost	information accuracy
300	223.6	0.8941	299.1	0.8366
500	432.6	0.9195	501.8	0.8871
700	647.3	0.9268	703.5	0.9111
1000	861.1	0.9319	952.5	0.9007

Recall from Section 3 that the jointly sensing node selection problem is essentially a tradeoff between the energy consumption and information accuracy, with the balance controlled by the constraint C_0. We vary the constraint C_0 in our simulations and adopt a random node selection as a benchmark for comparison. The numerical results are listed in Table 1. As shown in Table 1, although the information accuracy increases as C_0 increases, the increment decreases while the total cost increases constantly. This means greater cost is needed to achieve a little increment of information accuracy as the information accuracy increases, which is consistent with our analysis in Section 4.1 that there is no necessary to choose much nodes joining sensing in achieving a little higher information accuracy since the energy cost will be great. Therefore, one can choose an efficient tradeoff at an affordable cost by using a proper C_0 in the proposed algorithm. Fig. 6 visualizes the position of selected nodes. As shown in Table 1 and Fig. 6, the closer is a sensor to the event source, the further apart it is from those already been selected sensors, the closer it is to the sink, the higher priority it is to be selected with our algorithm. Compared with the random selection, our algorithm achieves more accurate information with fewer energy consumption.

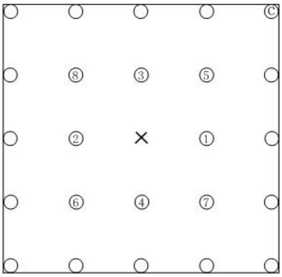


Fig. 6. Network topology of jointly sensing nodes in the cluster. × means an event source, c means the cluster header, each selected sensor is labeled with a serial number which represents the sequence of being selected.

Another important result from our simulation is that only eight nodes are selected to join sensing task even in the case of $C_0 > 1000$. This is because the information accuracy decreases if any other node is selected to join sensing tasks after set $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ has been selected. For example, we randomly select another node which is not belong to the set V in the cluster to join sensing and the information accuracy with the nine selected nodes is 0.9266. However, the information accuracy achieved by eight selected nodes is 0.9319 as shown in Table 1. While random selection may cause the decrease of information accuracy with increased jointly sensing nodes and total cost as shown in Table 1. Hence, the energy waist caused by unnecessary joining sensing nodes can be efficiently avoided with information-accuracy-aware jointly sensing nodes selection.

5 Conclusion

This paper aims at optimizing the maximum information accuracy and minimum energy consumption for cluster-based WSNs by proposing an information-accuracy-aware jointly sensing nodes selection algorithm based on a quantitative definition of information accuracy model. We first discuss the information accuracy versus the number and position of jointly sensing nodes by exploiting the spatial correlation between sensed data, which is used to guide the designing of sensor selection algorithm. The simulation results demonstrated the efficiency of proposed algorithm in joint optimization of information accuracy and energy consumption.

We presented the sensor selection algorithm for scenarios with a single static event source in a cluster. More studies are still required on this issue by considering more complex networking sceneries such as multiple event sources. Generalizing the algorithm to handle moving event sources while maintaining good tradeoff between information accuracy and energy consumptions also remains as a future research topic.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor networks: A Survey. Computer Networks. **38** (2002) 393- 422.
2. L.B. Ruiz, J.M. Nogueira, and A.A.F. Loureiro. MANNA a management architecture for wireless sensor networks. IEEE Communications Magazine. **41** (2003) 161-125.
3. M.A. Perillo and W.B.Heinzelman. Optimal sensor management under energy and reliability constraints. IEEE Wireless Communications and Networking, WCNC (2003). New Orleans, USA, March 16-20, 1621-1626.
4. F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration, IEEE Signal Processing Magazine, **19** (2002) 61-72.
5. J. Liu, J. E. Reich, and F. Zhao. Collaborative in-network processing for target tracking. EURASIP, Journal on Applied Signal Processing. **3** (2003) 378-391.
6. T. J. Goblick. Theoretical limitations on the transmission of data from analog sources. IEEE Trans. Inform. Theory. IT- 11 **4** (1965) 558C567.
7. V. Poor. An Introduction to Signal Detection and Estimation second ed., Springer, Berlin, (1994).
8. J.O. Berger, V. de Oliviera, and B. Sanso. Objective Bayesian analysis of spatially correlated data. J. Am. Statist. Assoc., **96** (2001) 1361-1374.
9. S. Lindsey, C. Raghavendra, PEGASIS: power efficient gathering in sensor information systems. IEEE Aerospace conference, (2002) 1125-1130.
10. T. Rappaport. Wireless Communications: Principles 4 Practice. Prentice-Hall, Inc., New Jersey, (1996).

Supporting Application-Oriented Kernel Functionality for Resource Constrained Wireless Sensor Nodes

Hyojeong Shin and Hojung Cha

Department of Computer Science
Yonsei University, Seoul, Korea
{hjshin, hjcha}@cs.yonsei.ac.kr

Abstract. A sensor network application requires diverse kernel supports to function properly. With its resource limits the sensor node cannot provide all the functionalities needed by many kinds of applications at the same time. The kernel's functionality therefore requires runtime reconfigurability, which can be achieved via modularizing the kernel. This paper presents a framework that dynamically reconfigures the kernel's functionality according to the needs of the application. In particular, the proposed mechanism handles the address resolution problem of a MMU-less processor. This framework has been implemented on a sensor network operating system, RETOS, which supports multi-threaded programming environments. It efficiently manages the modularized kernel's resources and works in an optimized condition. By providing modularized kernel programming, RETOS optimizes itself with functionalities that various kinds of sensor network applications require.

1 Introduction

Diverse applications exist in wireless sensor networks; for example, acoustic source localization, environmental monitoring, mobile object tracking, etc. System functions such as localization, time synchronization and data aggregation are required to make such applications work, so the sensor operating system should support them. A sensor OS should have general-purpose operating system features to provide these diverse services efficiently to applications. It is hard to provide all the functionalities necessary for a sensor network operating system at one time, because sensor nodes have limited energy, small memory and low computational power. Providing diverse functionalities to the application layer is therefore one of the important issues of a sensor network operating system. This issue is redefined as the problem of adequately selecting an operating system's functionalities, given the particular application. To implement such features efficiently, current approaches modify and re-distribute the operating system [1], or separate an application from the system to reduce the cost of reprogramming [2] [3]. Modifying and redistributing a system incurs the overhead cost of updating the whole kernel code. The modular approach, which separates the application from the system, can easily modify the functions for the applications' needs. However, this modification is done in the

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 748 – 759, 2006.

framework that the operating system supports. It implies that the modification is done only at the application level.

Considering the resource constraints of sensor network hardware, this paper proposes a framework that provides a mechanism to support diverse kernel functionalities that are needed only by the application at hand. The self-reconfiguration is achieved by selecting the appropriate kernel components in the operating system without changing the application. The proposed framework provides such a dynamic code update that means kernel level adaptability. This feature supports the multi-level network stack which provides application-aware routing protocols [4]. The network stack transmits a packet in various ways according to the application's requirement. The system has been implemented on RETOS [4] [5], which is a sensor network operating system that provides module-based multi-threaded programming. A loadable kernel module system is implemented as a part of the RETOS kernel to support modifying the kernel functionality.

The rest of this paper is structured as follows. Section 2 reviews the background related to our work. Section 3 presents the proposed system's key mechanisms. The system is evaluated in Section 4. Finally, Section 5 concludes the paper.

2 Background

2.1 Function Reconfigurability

Ideally a sensor network operating system should provide reconfiguration of the system functionalities as required by applications. TinyOS [1], the de facto operating system for sensor nodes, supports modularized programs, which are compiled into a monolithic image and distributed as a whole. Maté [3] implements a simple virtual machine (VM) on the system that allows the building of an executable and updatable code to enable modularized updating. Similarly, Magnet OS [6] builds a Java VM on the system. A good example of supporting modularized programming in sensor networks is SOS [2], which uses a text address and a stack base to indicate the addresses of variables and functions used in a binary code to handle the problem of MMU-less microprocessors. Contiki [7], a thread-based operating system for sensor networks, provides code update in terms of independent applications. The VM*[8] serves as middleware programming, which allows the building of a program for VM and a customized VM for a corresponding binary.

Reconfiguring the kernel functionality originates from traditional embedded systems. uClinux [9], an embedded version of Linux [10] ported on an MMU-less microcontroller, supports a variety of Linux functionalities and a loadable kernel module. With a relocation mechanism and Position Independent Code (PIC), uClinux builds a code that runs independently from a position in the allocated memory. uClinux successfully extends the system functionalities in general-purpose embedded systems. However, it cannot be used directly in sensor networks due to the sensor nodes' resource constraints.

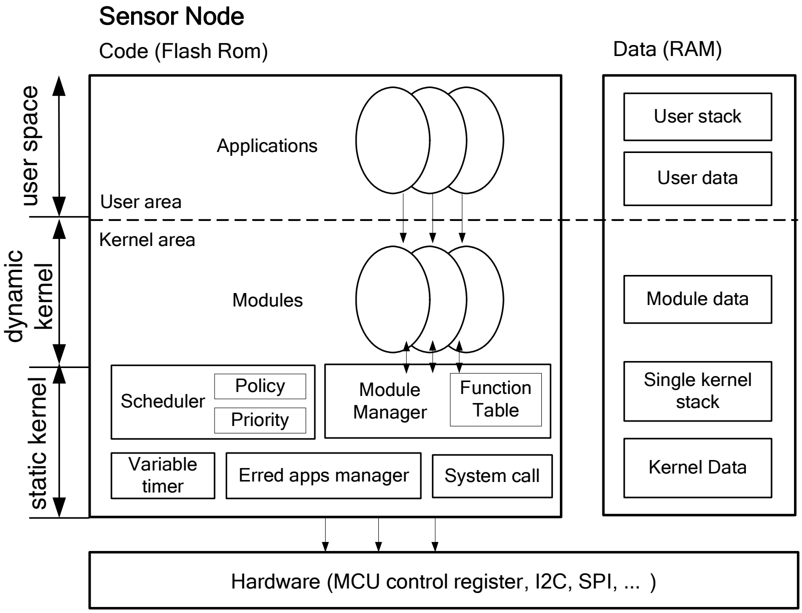


Fig. 1. RETOS overview

2.2 RETOS Overview

RETOS [5] is currently being developed to support a reliable and multi-threaded programming environment for sensor networks. It has three objectives: (i) to provide a multi-threaded programming environment, (ii) to serve as a reliable system which protects a kernel from errant applications, and (iii) to flexibly extend the sensor operating system's functionalities. For the multi-threaded programming environment, RETOS provides the POSIX 1003.1b real-time scheduler interface. With a dual mode operation and a protection mechanism, RETOS can protect the kernel from errant applications. The RETOS kernel shares the kernel stack among its threads to reduce memory usage incurred by dual mode operation. The application, kernel and kernel modules are developed in standard C language. RETOS is supported by several types of hardware including Telos Skymote [11] which is used in the evaluation part.

To build a flexible system, RETOS provides dynamic application installation. Moreover, RETOS supports a Loadable Kernel Module (LKM) that modularizes system functionalities as kernel modules and dynamically replaces them. This feature provides system level reconfiguration to support flexibility to diverse applications' operations. Figure 1 shows an overview of the RETOS system that features the Loadable Kernel Module. An application is loaded onto the system after the safety check, and system resources are then allocated. The operating system consists of two parts: a static kernel which includes a core part of the system and hardware dependent codes, and a dynamic kernel which includes common libraries used by applications and various kernel services. Section 3 presents the proposed RETOS LKM system and corresponding mechanisms.

3 Loadable Module Support

The LKM framework consists of the module manager that registers a modularized kernel on the system, and the function table that links functions of modules to applications. The following section describes the technical issues that arise in modularizing the kernel.

3.1 Kernel Memory Management

A binary code in general consists of text, heap, stack and data area. Some of the recent embedded systems support the eXecute In Place (XIP) technique to execute code on flash memory without loading it to the memory space, hence allocating memory for the text is not required. The amount of dynamic allocation is unpredictable and depends on the types of application, thus the heap size is hard to determine in the compiling time. This section presents a method for handling the stack and the global variable data.

The stack size is difficult to estimate and differs for each application; therefore, the stack space should be sufficiently reserved. Because much redundancy occurs in the stack space when using a global fixed stack for all applications, the stack size is specifically set depending on the application. The stack size is, however, eventually fixed, and results in some redundancy in the stack. To reduce this redundancy the RETOS LKMs share a common stack area with each other. The LKM uses the kernel stack as its common stack area. Every kernel module uses the single kernel stack to save intermediate results, thus the system benefits from saving redundant memory space. Figure 2 illustrates a snapshot of the module's memory usage in runtime. Each module has its own data area, but they all share the stack. During a module's execution the system prohibits another module from entering execution mode to prevent damage to previous contexts in the stack or any changes in the stack size.

To maintain the modules' information and status, memory resources are required to save them. Symbol table and memory allocation are usually used for storing the modules' information and status. A symbol table, managed by the kernel, is a manager that handles variables used in the modules. The symbol table's size depends on the number of variables used in the kernel module, which is unpredictable and should be controlled by the kernel. Also, overhead is incurred when accessing variables from the symbol table upon a system call. On the other hand, using memory space to allocate data used in the kernel module is a simple and general approach. This method is used in our work. The module's information and status are declared as global variables of the module and can be compiled with the module source. The global variables are allocated in the bss and data area of the memory space allocated to the kernel module.

The framework proposed in this paper allocates a memory space to maintain the module's information. The kernel module is compiled with a given stack base and data space address. The addresses of the functions and variables are decided a priori based on the address given. These addresses change when the kernel module is saved in flash memory and loaded into the RAM. Therefore, a mechanism is required to handle such indeterminate address changes. Figure 2 shows that each module has its own data space. Here, each kernel module is compiled separately with its own data space allowing direct access.

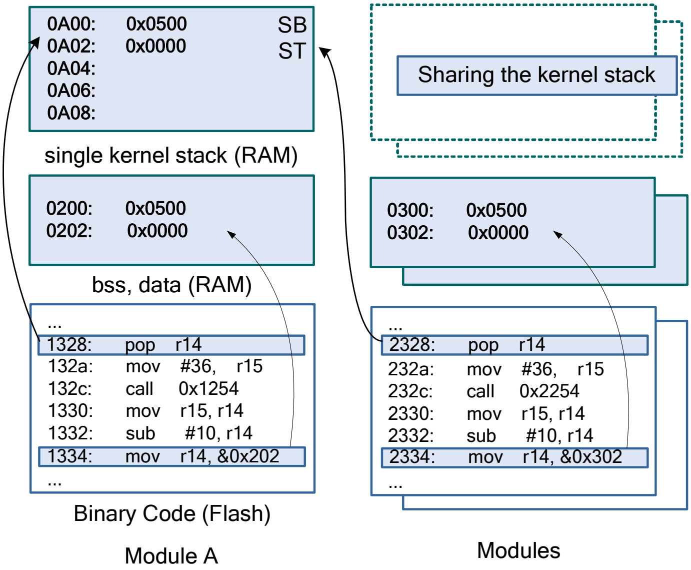


Fig. 2. Memory addressing

3.2 Dynamic Module Linking

Sensor node hardware typically uses a MMU-less microprocessor. The MMU translates a virtual address into physical memory and controls the memory's access privileges. A TinyOS-based application's binary image is always allocated in the same position in the memory space, thus no explicit memory management is needed. However, the location of the modularized code, which is used by module supporting operating systems such as SOS and RETOS, loaded in the memory cannot be expected. This means that the variables' and functions' addresses can be changed at load time, which causes problems when calling functions and accessing variables with an incorrect address. Figure 3 illustrates this problem. Figure 3 (a) shows a binary code after the compilation of the source code using the given stack base and text area address. The code accesses a function located in address number 0x0100 and a variable located in 0x1002. Figure 3 (b) shows the code whose text data are located in address number 0x7000 in the flash memory and whose data are located in address number 0x2000 in the RAM. Without any appropriate modification, this binary code may access the wrong address. This paper considers two approaches to preventing such a problem: PIC and address relocation.

Position Independent Code is a compiled code that does not have an absolute address and therefore can be run without knowing where the module is located. Function calls and jumps within the single binary code are compiled using PC-relative instructions, and variable access is compiled through indirect accessing. The compiled code runs without any code modification in run-time. PIC causes run-time overhead when modifying the Data Base Section Register by indirectly accessing a variable's specific address. SOS compiles the module using PIC. Meanwhile, the relocation method compiles the source code, assuming the location of the binary code is zero, and modifies every address accessed in the binary code when the assumed address changes.

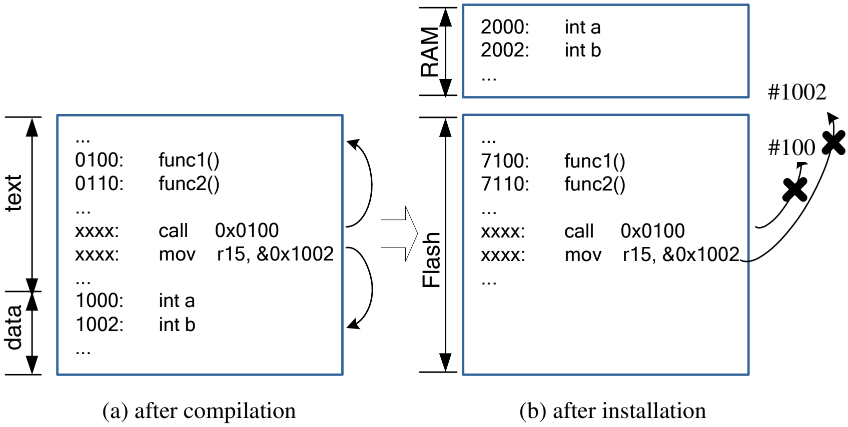


Fig. 3. Linking problem

Both PIC and the address relocation would cause system overhead in different time domains. PIC generates a relative-address code that does not need to be changed, although it has run-time overhead. Using relocation, however, an absolute-address code is generated, but overhead exists to maintain a relocation table and to modify the module. Our experiment shows that the PIC-based approach requires more overhead to access the functions of the module in a sensor node than the address-relocated approach. Under the assumption that run-time overhead is more important than loading overhead in wireless sensor network applications, RETOS uses relocation to handle the indeterminate memory address problem. RETOS acquires the kernel module via a relocation table from a host, and performs the relocation in run-time.

The uClinux system executes the relocation process when the binary stored in flash memory is loaded into the RAM. The code image on the flash memory contains a memory access that references an illegal position, so the code cannot be executed in flash memory. XIP technique is very important in terms of memory efficiency in memory limited sensor nodes, since it does not require extra memory for *text* area. RETOS stores kernel code in flash memory after the relocation process. Thus,

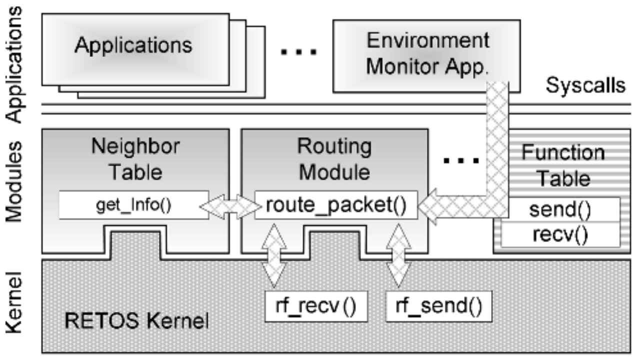


Fig. 4. Module accessing

RETOS can execute a code in flash memory while supporting relocation, whereas uClinux cannot provide such a service.

Relocation is also supported in uClinux. uClinux relocates the module when it is loaded into the RAM. Because the binary image on the flash memory contains illegal address accesses, uClinux does not support XIP using relocation. RETOS relocates the module both on the RAM and on the flash memory and the module runs on the flash memory. This reduces memory usage and the loading cost of an application. It is suitable for resource constraint systems such as wireless sensor networks.

3.3 Module Communication

RETOS has a function table that allows modules and applications to access other module's functions. The function table manages the function information of modules, such as function entry points, ownership, parameters and return types, which are accessible by other modules or applications. The module registers, un-registers and accesses the functions through the function table. The kernel and kernel modules are dynamically linked at run time even though they are directly linked. The dynamically loaded module is allocated with a kernel code in flash memory as a single executable code image, which is also dynamically removable. An application can also access a module's function, but it is done by calling a specific system call. The cost for invoking such a module's function is the same as other conventional system calls.

Figure 4 shows how a module and an application access a module's function through the function table. The application accesses the module to send a packet through radio. To perform the operation, the routing module gets the neighbor's information and accesses the kernel's hardware driver. These modules are already installed in a system and its functions are registered in the function table. In running the system, the kernel and modules work as a single image, so they access each other's functions directly. And the application accesses the module's function through a system call which references the function table and invokes the required functions. Here, the invoked system call performs a mode switch and verifies the validity of the corresponding function. This provides safety for the kernel and kernel modules from an application's illegal memory access. For kernel safety, RETOS supports dual mode operation. An application operates in user mode and a kernel module in kernel mode. While an application executes a function of the provided kernel module through a system call, the system switches the application stack to the kernel stack. These features protect the kernel from errant code [5].

3.4 Kernel Reconfiguration

A reconfigurable system reconfigures the system according to the application which runs on the system. For example, RETOS supports various routing mechanisms and selects a mechanism depending on the node's status and the current network's condition. When a node experiences low battery power, RETOS can switch the current protocol with a low-power routing algorithm. Also, when a node receives important and time-critical data, it adopts a time-sensitive routing algorithm to efficiently route the data to a sink node within the time constraint. Such a reconfiguration of the kernel functionality is supported by the modular kernel system of RETOS, LKM.

The RETOS LKM system supports kernel-level optimizing techniques. First, the system maintains a light-weight system to reduce overhead and resource usage. LKM unloads unnecessary kernel modules and only maintains modules that are required by the current working applications. The system can stop an unused system thread and reuse the resource for other applications later. Second, LKM reduces the cost for acquiring the required module over the network. RETOS supports dynamic application installment and each application requires different modules. Thus, the LKM system needs to acquire modules that are required by the installed application. Furthermore, to maintain an optimized system, the system must always keep a minimal requirement scheme by removing unused modules and acquiring newly required modules over the network with the change of application reconfiguration, which changes the requirements of the kernel functionality. This procedure needs extra network transmission and overhead.

When developing kernel modules for RETOS, the system's stability should be considered. First, the kernel module operates in kernel mode and therefore it should conduct garbage collection as resource leak in the module is critical to the whole system. Second, the module's function should return as quickly as possible. During the operating of the module function, no function in the module or application can be invoked. The module should avoid the use of loops or time-consuming logic. Last, the module should not use blocking functions because they lead to invoking context switches. By using a common shared kernel stack, context switching within the kernel module is apt to elicit unpredictable defects.

4 Evaluation

The RETOS version 0.86 is used to implement a loadable kernel module. RETOS provides multi-threaded programming, a dual mode operation and a code safety check. Each kernel, kernel's module and application is code-separated and resides in separate memory space. The experiment is conducted on the MSP430-based [11] Telos Skymote [12] hardware platform. We evaluate the performance of the RETOS LKM system in terms of system throughput and module updating cost.

For the system throughput measurement, we compared RETOS LKM against SOS, which is developed with a similar concept. For a fair comparison, two systems should be evaluated using the same hardware. RETOS is developed on msp430-based hardware. However, there is no SOS version available for msp430-based hardware. For a solution, we compared each linking method used by RETOS and SOS on Telos. Two sample applications were prepared. One was the original surge application of RETOS, whereas the other was a surge application built with position independent code, which is used by SOS. Both were executed using RETOS on msp430-based hardware. The comparison only includes the difference of overhead between the two types of linking methods, and not the operating system's overhead. With this comparison, the relocation-based system has better throughput against the PIC-based application. This comes from memory-accessing cost. Therefore, we also measured memory-accessing cost on two different types of hardware: msp-430 and AVR[13]. The module updating cost measurement was also evaluated using the surge application on three operating systems: TinyOS, SOS and RETOS. The cost for module transmission time and the installation time was measured. TinyOS uses Deluge [14] for code update, which is a well-known code updating protocol for wireless sensor operating systems. SOS and RETOS have their own module dissemination mechanisms. We installed each operating system with the surge application. To evaluate the cost for dynamic code updates, we modified a part of the application's network and measured the expenses for updating the modified parts over the network.

Table 1 shows the surge application's execution time for each operation, which is generated by a PIC compiler and a relocation linker. A relocation-produced application performs slightly faster than a position independent code based application. It only reflects binary execution overheads. The difference between the two systems is based on the difference of memory accessing and function invocation cost. A binary produced using the relocation method consists of direct addressing operations, whereas a binary using PIC consists of indirect addressing operations. As shown in Tables 2 and 3, PIC's function call and memory accessing require up to six more cycles than the relocated binary on msp430 and AVR. These overheads are generated due to modifying the base address of the position independent code and the execution of indirect addressing operations. RETOS implements direct addressing operations, hence it performs better.

RETOS supports a kernel optimization mechanism. The optimization is obtained by reconfiguring kernel compositions according to the new requirements of applications that are dynamically loaded to the deployed sensor motes. As a specific optimization operation, such a code updating mechanism is accompanied with some overheads. To measure the overhead, we modified a part of the network routing module in the surge application for each of the three operating systems. As shown

Table 1. Cost of the surge application

Cost	Relocation	PIC
Send the beacon message	0.3 msec	0.35 msec
Handle the received message	0.03 msec	0.05 msec
Updating the neighbor table	0.01 msec	0.015 msec
Active time (in a minute)	0.44%	0.58%

Table 2. Cost of accessing memory (msp430)

Operation	Relocation	PIC
Function call	5 cycles	9 cycles
Data Accessing	3 cycles	5 cycles
Global function accessing	9 cycles	9 cycles
Global data accessing	5 cycles	5 cycles

Table 3. Cost of accessing memory (AVR)

Operation	Relocation	PIC
Function call	4 cycles	10 cycles
Data accessing	2 cycles	3 cycles
Global function accessing	6 cycles	6 cycles
Global data accessing	3 cycles	3 cycles

Table 4. Data size (bytes)

Data size	Deluge	RETOS	SOS
Transmitted data size	31000	2614	2496
Flash Memory data size	31000	26542	33944

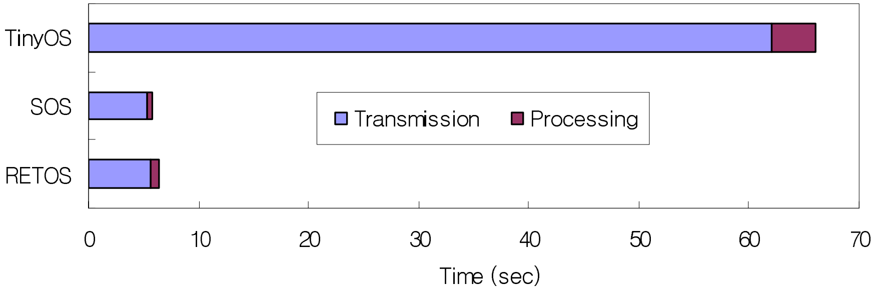


Fig. 5. Execution time of the code update

in Table 4, RETOS's updating size is smaller than that of Deluge and is similar to that of SOS, since RETOS also updates only the modified modules like SOS. Figure 5 shows the amount of time consumed for updating a module code. It consists of the transmitting time of the kernel code and the installation time for the linking process. As a small code transmission, RETOS has a smaller transmission time and installation time compared to Deluge. RETOS needs a relocation table for loading modules. The transmitted packet size is larger than the packets transmitted using SOS and also the relocation process needs extra linking process on a mote. Thus, RETOS takes slightly more time for updating modules compared to SOS. This overhead occurs only during loading a module and is negligible during run time.

5 Conclusion

The eventual goal of RETOS is to provide a general-purpose operating system for wireless sensor networks. Supporting modularized application programming suits such a goal. RETOS provides diverse kernel functionalities by supporting loadable kernel modules. The LKM system reconfigures the kernels' services according to the applications' requirements. Traditional reconfiguration approaches provide functional change by reprogramming the application; however, the RETOS LKM provides an OS level optimization for each node's applications, so it is a more general and abstract approach. The experiment showed that the LKM has a small overhead for operating the module manager. This overhead is due mainly to managing the loadable kernel system and to applying relocation to the module. In an environment where many applications work simultaneously, RETOS has the benefits of saving storage by sharing the kernel module, while reducing the LKM's operating cost.

The RETOS LKM is a first step toward an adaptable operating system. Its automatic reconfiguration mechanism includes protocols for code dissemination, and the handshaking of RETOS remains as a future work. Deluge [14], Trickle [15] and

MNP [16], which are well designed code dissemination protocols, are adaptable for RETOS's kernel module. The RETOS's module dissemination protocol must be suitable for small module kernels and customized to dynamic changes. A RETOS application may run without any consideration of the kernel's composition using the RETOS LKM system. The module resides in the kernel and can access and modify the kernel's information. The LKM system will have a mechanism to protect the system from unexpected kernel action.

Acknowledgements

This work was supported in part by the National Research Laboratory (NRL) program of the Korea Science and Engineering Foundation (2005-01352), and the ITRC programs (MMRC) of IITA, Korea.

References

[1] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister, "System architecture directions for network sensors," *Proceedings of the 9th international conference on Architectural support for programming languages and operating systems,* Cambridge, Massachusetts, USA, November 2000.

[2] Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler and Mani Srivastava, "A Dynamic Operating System for Sensor Nodes," *Proceedings of the 3rd international conference on Mobile systems, applications, and services,* Seattle, Washington, USA, 2005.

[3] Philip Levis and David Culler, "Maté: A Tiny Virtual Machine for Sensor Networks," *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems,* San Jose, California, USA, 2002.

[4] Suckwon Choi and Hojung Cha, "Application-Centric Networking Framework for Wireless Sensor Nodes," *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services,* San Jose, USA, July 2006.

[5] Hyoseung Kim and Hojung Cha, "Towards a Reliable Operating System for Wireless Sensor Networks," *Proceedings of the USENIX Annual Technical Conference: Systems Practice & Experience Track,* Boston, Massachusetts, USA, May 2006.

[6] Rimon Barr, John C. Bicket, Daniel S. Dantas, Bowei Du, T. W. Danny Kim, Bing Zhou, Emin Gün Sirer, "On the need for system-level support for ad hoc and sensor networks," *ACM SIGOPS Operating Systems Review,* vol.36 no.2, pp.1-5, April 2002.

[7] A. Dunkels, B. Gronvall and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks,* Tampa, Florida, USA, 2004.

[8] Joel Koshy and Raju Pandey, "VM★: Synthesizing Scalable Runtime Environments for Sensor Networks," *Proceedings of the 3rd international conference on Embedded networked sensor systems,* San Diego, California, USA, 2005.

[9] Linux, http://www.linux.org.

[10] uClinux, http://www.uclinux.org.

[11] msp430, http://www.ti.com/msp430

[12] Tmote Sky, http://www.moteiv.com.

[13] AVR, http://www.atmel.com/products/AVR

[14] Jonathan W. Hui and David Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, Maryland, USA, 2004.

[15] Philip Levis, Neil Patel, Scott Shenker and David Culler, "Trickle: A Self-Regulating Algorithm for Code Propagation and maintenance in Wireless Sensor Networks," *Proceedings of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004)*, San Francisco, California, USA, 2004.

[16] S. S. Kulkarni and Limin Wang, “MNP: Multihop Network Reprogramming Service for Sensor Networks,” *Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS)*. Columbus, OH, USA, June 2005.

XMAS: An eXtraordinary Memory Allocation Scheme for Resource-Constrained Sensor Operating Systems[⋆]

Sangho Yi[1], Hong Min[1], Junyoung Heo[1], Boncheol Gu[1], Yookun Cho[1],
Jiman Hong[2,⋆⋆], Hyukjun Oh[3], and Byunghun Song[4]

[1] System Software Research Laboratory
School of Computer Science and Engineering, Seoul National University
{shyi, hmin, jyheo, bcgu, cho}@ssrnet.snu.ac.kr
[2] School of Computer Science and Engineering, Kwangwoon University
gman@daisy.kw.ac.kr
[3] School of Electronic Engineering, Kwangwoon University
hj_oh@kw.ac.kr
[4] Intelligent IT System Research Center, Korea Electronics Technology Institute
bhsong@keti.re.kr

Abstract. The wireless sensor networks are sensing, computing and communication infrastructures that allow us to monitor, instrument, observe, and respond to phenomena in the harsh environment. Sensor operating systems that run on tiny sensor nodes are the key to the performance of the distributed computing environment for the wireless sensor networks. Therefore, sensor operating systems should be able to operate efficiently in terms of energy consumption and resource management. In this paper, we present *XMAS* to improve the time and space efficiency of memory management for the sensor operating systems. *XMAS* was implemented on *Nano-Qplus* which is a multi-threading sensor operating system. Our experimental results show that the *XMAS* performs efficiently in both time and space compared with existing memory allocation mechanisms.

1 Introduction

The rapid advances in computing technology, micro-electromagnetic systems technology and wireless communication technology have enabled smart, small devices to integrate micro-sensing and actuation with on-board processing and wireless communications capabilities. These wireless sensor networks become more popular over the past few years and have captured the attention and imagination of many researchers, encompassing a broad spectrum of ideas[1,2]. The

[⋆] This research was supported by the Ubiquitous Autonomic Computing and Network Project, 21st Century Frontier R&D Program in Korea, Brain Korea 21 Project in 2006, ICT at Seoul National University, and MIC & IITA through IT Leading R&D Support Project.

[⋆⋆] Corresponding author.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 760–769, 2006.

main purpose of the wireless sensor networks is to monitor the natural environment. Potential applications include scientific data gathering, e.g., environment monitoring and controlling systems, military systems, automatic manufacturing systems, surveillance systems, and medical systems. Wireless sensor networks can be used even in the harsh environments[3,4,5].

Generally, wireless sensor networks are composed of hundreds or even thousands of deployed sensor nodes that were designed to be very cost-efficient in terms of production cost. The sensor nodes have miniature computing devices, extremely small memory space, and very limited battery power. Key constraints of wireless sensor networks include low energy consumption and limited resources. Therefore, operating systems that run on the sensor nodes must also operate efficiently in terms of energy consumption and resources management to support low energy consumption. Therefore, the operating systems must minimize execution time and maximize resource utilization. For this reason, the sensor operating systems need a time and space-efficient memory allocation mechanism since each sensor node has very small memory space and limited battery power.

Many kinds of operating systems such as *TinyOS*[6], *Maté VM*[7], *SOS*[8], *Nano-Qplus*[2], and *MANTIS OS*[1] have been proposed for the wireless sensor networks. *TinyOS* and *Maté VM* do not use the dynamic memory allocation since there are no loadable modules and dynamic threads. Especially, *SOS*, *Nano-Qplus*, and *MANTIS OS* use dynamic memory allocation mechanisms to support dynamic loadable module and multi-threading techniques. However, the dynamic memory allocation mechanisms are not suitable for tiny sensor nodes but feasible for general purpose computing environment, due to the lack of memory space and limited power of the sensor nodes.

In this paper, we propose *XMAS*, which is an *eXtraordinary Memory Allocation Scheme* to improve the time and space-efficiency of memory allocation mechanism that aims at minimizing fragmentation ratio and average response time. Our experimental results show that *XMAS* performs efficiently in both time and space compared with existing memory allocation mechanisms.

The rest of this paper is organized as follows. In Section 2, we present some related works done on dynamic memory allocation mechanisms. Section 3 describes characteristics of wireless sensor networks and sensor nodes. We also present some examples of the practical usage of dynamic memory allocation mechanism of existing sensor operating systems. Then, we explain about the design and implementation of *XMAS* in more detail. Section 4 presents and evaluates the performance of *XMAS* compared with other existing memory allocation mechanisms for sensor nodes. Finally, conclusions are presented in Section 5.

2 Related Works

In this section, we briefly introduce previous works relating to dynamic memory allocation mechanisms. Considerable research efforts[9,10,11,12,13,14,15] have been done on dynamic memory allocation to improve the performance of the memory-related tasks on existing operating systems. Those can be categorized into following classes: *sequential fits*[9], *segregated fits*[10,11], *buddy systems* [12,13,14], and *hybrid mechanisms*[15].

The *sequential fits* manage the memory space by the linear-list of all the free blocks in memory, and it supports allocation and release of various-size blocks on the linear-list. Typically, the *boundary tags*[9] technique is used to support efficient coalescing and merging blocks. The famous variants include *first-fit*, *best-fit*, *next-fit*, and *worst-fit*[16]. In those mechanisms, no internal fragmentation occurs. Instead, the worst case response time is $O(N)$ because it must search for an appropriate block of the total memory space.

The *segregated fits* use free lists of each segregated memory space to improve the response time of the memory allocation and release mechanism. In this mechanism, each list holds free blocks of particular size, and there are two major variants: *simple power-of-n free lists* and *segregated sequential fit*. The *power-of-two lists* stores buffers of a particular size, and all the sizes are powers of two. If a user's request is not of a size for powers of two, the system rounds up the size of the request and allocate it appropriately. Thus, its response time is $O(1)$. But in worst case, the internal fragmentation may increase the fragmentation of the memory space up to 50%[16]. The *segregated sequential fit* uses the lists that are divided into particular sizes to reduce the time consumed to search the memory space. If the number of division is k, then the average response time is $O(N/k)$[11].

Peterson et al. proposed the *buddy systems*[13]. This approach creates appropriate buffers by coalescing adjacent free buffers and splitting the larger one. Thus, it provides flexibility, allowing memory to be reused for buffers of various different sizes. However, it has a problem of internal fragmentation, which may increase the fragmentation up to 25%[17].

In [15], Masmano et al. proposed a new dynamic memory allocator for real-time systems. It uses hybrid techniques of *segregated fits* with the *boundary tags* technique to minimize the worst case execution time.

3 Design and Implementation of *XMAS*

In this section, we describe the requirements for wireless sensor networks including the platform of sensor nodes and explain about the existing constraints. We also present examples related to the practical usage of memory allocation mechanisms on existing sensor operating systems and then explain about the design and implementation of the *XMAS* in detail.

3.1 Requirements for Wireless Sensor Networks

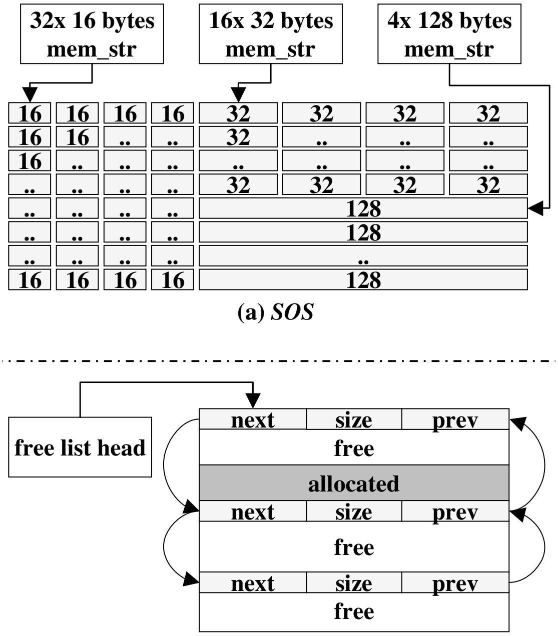
Typically, wireless sensor networks consist of many sensor nodes and it is very important to minimize the production cost to enhance networks efficiency. Such restriction limits computing power, memory space, and the batteries. For example, *Berkeley's MICA* motes have only 8-bit processor, 4 KB memory space, and 2xAA batteries[18]. Therefore, the limited memory and energy have to be efficiently utilized. In other words, the memory space must be managed in accordance with space and energy-efficient memory allocation mechanism. The following are the requirements that should be considered in the design process of an efficient memory allocation mechanism for wireless sensor networks.

- ◇ **Execution Time:** it is proportional to the energy consumption of the memory allocation mechanism, thus, the total execution time have to be minimized to reduce the amount of energy consumption.
- ◇ **Fragmentation Ratio:** the internal and external fragmentation may spoil the memory utilization, thus it must be minimized.

3.2 Practical Usage of Dynamic Memory Allocation on Sensor Operating Systems

Figure 1 shows some practical usage of the memory allocation mechanisms on existing operating systems for wireless sensor networks[1,2,8].

SOS uses a similar mechanism as the *power-of-two free lists*[8]. It consists of 32x16, 16x32, and 4x128 bytes memory blocks. The total memory block is thus 1536 bytes. Similar to the *segregated free lists*, it only needs $O(1)$ to find an appropriate block. Thus, in *SOS*, a serious internal fragmentation may be arose



(a) *SOS*

(b) *Nano-Qplus* and *MANTIS OS*

Fig. 1. Memory management data structures on *SOS*, *Nano-Qplus*, and *MANTIS OS*

and it decreases the efficiency in the utilization of the memory space. Figure 1(a) shows the memory space of the *SOS*.

Nano-Qplus, and *MANTIS OS* support preemptive thread scheduler for multi-modal sensing tasks. They need a dynamic memory management to allocate threads stacks, and use *sequential fits* with *best-fit* policy. Thus, they need $O(N)$ for linear searching the total memory space, but there is a possibility that they may over-use the limited power of batteries. Figure 1(b) shows the example of the *Nano-Qplus* and *MANTIS OS*.

3.3 *XMAS*: *eXtraordinary Memory Allocation Scheme*

XMAS is an *eXtraordinary Memory Allocation Scheme* that efficiently utilizes the tiny memory space in sensor operating systems. It adaptively uses the techniques of *sequential fits(bitmap fits)*, *segregated free lists*, and the *buddy systems* to fulfill the essential requirements for wireless sensor networks. Generally, the *sequential fits* are good at minimizing fragmentation ratio, and the *segregated free lists* perform well in terms of execution time. In addition, *buddy systems* can provide flexibility and scalability to the target operating systems.

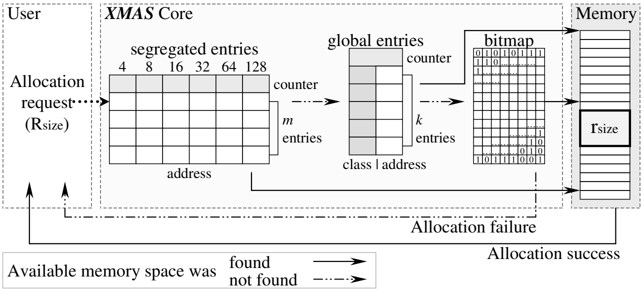
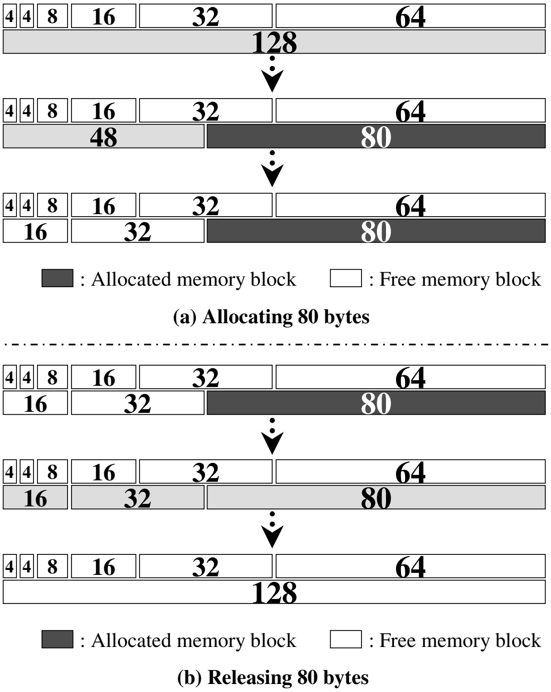


Fig. 2. Memory management data structures in *XMAS*

Figure 2 illustrates the relationship between physical memory space and the data structures used in *XMAS*. In Fig. 2, the data structures consist of the bitmap, k global entries, and the table of segregated entries. Each global entry has a class identifier and the address of a memory block. The table of segregated entries are combined by the power-of-two$(4, 8, 16, 32, 64, 128$ bytes) address entries, and each one has the m entries of address that points to the memory block. The bitmap is used for mapping and checking the used area of the physical memory space, and each bit is mapped to 4 bytes of the actual memory space. The global and segregated entries manage the free memory blocks. For example, the segregated entries can manage the free blocks with maximum m entries, and the remaining free blocks are saved on global entries. In sensor nodes, the memory space is very limited and thus cannot be used excessively. However, *XMAS* can give better utilization of the total memory space using limited the number of the



(a) Allocating 80 bytes

(b) Releasing 80 bytes

Fig. 3. An example of allocating and releasing 80 bytes in *XMAS*

global and the segregated entries, though it may slightly increase the average execution time of a task.

The following show the detailed description of the basic operations on *XMAS*.

◇ **Initialization:** The data structures are statically located on the global variables of the data region. The *counter* field and the *bitmap* are set to 0.

◇ **Allocation:** If an allocation request arrives with r_{size}, *XMAS* searches suitable block that greater than or equal to the size of $2^{\lceil log_2 r_{size} \rceil}$ on the global and the segregated entries. If a block is found, it returns to the user. Otherwise, *XMAS* searches on the bitmap with *fitst-fit* policy and returns the found block. Finally, it marks the bitmap of the allocated memory block.

◇ **Release:** If a release request arrives with r_{size}, *XMAS* re-registers the released block to appropriate entries and releases the bitmap of the released memory block.

◇ **Coalescing and Splitting:** As similar to the *buddy systems*, *XMAS* manages memory space in power-of-two blocks. *XMAS* coalesces and splits the memory blocks if needed. If an allocation request arrives with r_{size} and *XMAS* succeeds to find a free block of the $2^{\lceil log_2 r_{size} \rceil}$ bytes, then *XMAS* splits the remaining $2^{\lceil log_2 r_{size} \rceil} - r_{size}$ bytes block into multiple power-of-two blocks. The split blocks are re-registered to the global or the segregated entries. On the other hand, when a release request arrives with r_{size}, *XMAS* searches the adjacent $2^{\lceil log_2 r_{size} \rceil}$ bytes memory space for coalescing.

◇ **Management:** There are three-level data structures: segregated entries, global entries, and bitmap. It is similar to hierarchical cache in computer architecture. For example, *XMAS* searches to find an appropriate block in this order(segregated→global→bitmap). If the segregated entries become empty, it have to be replenished by the global entries. In this way, if the global entries become empty, then it must be filled up by searching the bitmap.

Figure 3 shows an example of way in which 80 bytes of memory space is allocated and released when using *XMAS*. If a user requests 80 bytes of memory space, it starts searching for an adequate memory block. As a result, a 128 bytes memory block is selected because the $2^{\lceil log_2 80 \rceil}$ is 128. It then splits into 80 and 48 bytes. The 80 bytes are allocated for the requested user, and the remaining 48 bytes are split and re-register to the power-of-two segregated entries(32 and 16 bytes blocks). It should be noted that if the user request that allocated 80 bytes block be released, *XMAS* starts searching adjacent 128 bytes memory space because the $2^{\lceil log_2 80 \rceil}$ is 128. Then, the 16 and 32 free blocks are coalesced with the released 80 bytes. As a result, 128 bytes memory block is generated and registered to the entries.

The following Algorithms 1 and 2 summarize the *XMAS*' memory allocation and release mechanisms.

Algorithm 1. Allocating and Splitting Mechanism

```
– when a user requests to allocate $r_{size}$ memory space:
  Search a free memory block that $\geq 2^{\lceil log_2 r_{size} \rceil}$.
  if a block was found on segregated entries then
    Allocate $r_{size}$ memory space to user.
    Split the residual memory space to powers-of-two.
    Register the split blocks to the entries.
  else if a block was found on global entries then
    Allocate $r_{size}$ memory space to user.
    Split the residual memory space to powers-of-two.
    Register the split blocks to the entries.
    Update segregated entries.
  else if a block was found on bitmap then
    Allocate $r_{size}$ memory space to the user.
    Update global and segregated entries.
  else
    Allocation fails.
  end if
```

In Algorithm 2, the time consumed to search for the coalescing memory block is only O(1) because the region of the search space is very limited. In *XMAS*, the space is bitmap and the size is only 4 bytes (one byte of the bitmap corresponds to 32 bytes of the physical memory space). Therefore, it does not consume a long time to search for the coalescing memory space.

Algorithm 2. Releasing and Coalescing Mechanism

```
– when a user requests to release $r_{size}$ memory space:
  Release $r_{size}$ memory space.
  Search the adjacent free memory space that $\geq 2^{\lfloor log_2 r_{size} \rfloor + 1}$ on bitmap.
  if free memory space was found then
    Coalesce the memory space and register to entries.
  end if
```

4 Performance Evaluation

In this section, we evaluate the performance of the *XMAS*. We implemented *XMAS* on *Nano-Qplus* to compare the performance of *XMAS* with that of *SOS*' memory allocator and *Nano-Qplus*' memory allocator[1].

In our experiment, we used *Octacomm's Nano-24* wireless sensor platform[19], which is similar to the *Berkeley's MICAZ* sensor platform. For our experimental evaluation, we modified kernel source of *SOS* and ported it to the *Nano-24* sensor platform, because there is no adequate kernel source of *SOS* for *Nano-24* sensor platform.

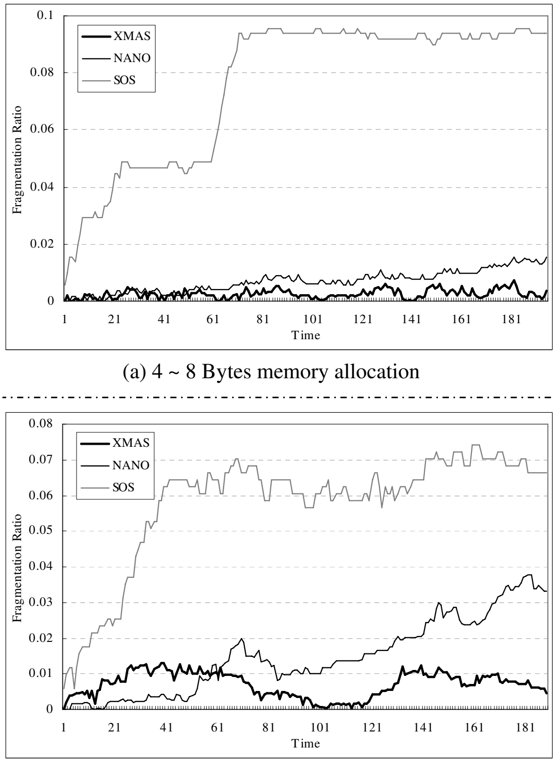
The memory allocation mechanisms are known to affect the time and space-efficiency of the operating system. Therefore, we focused on the fragmentation ratio and the total execution time of the memory allocation mechanisms for demonstrating the time and space-efficiency of *XMAS*. Table 1 shows the total execution time of each memory allocation mechanism over a various workload from 4 ∼ 8 bytes to 4 ∼ 16 bytes where the number of memory allocation and release requests was 200. In Table 1, the *SOS*' execution time is smaller than those of others because *SOS* uses fixed memory allocation mechanism. In addition, the *XMAS*' execution time is slightly smaller than that of *NANO*, and larger than *SOS*.

Table 1. Total execution time(ticks) of the memory allocators

Workload	*XMAS*	*NANO*	*SOS*
4 ∼ 8 bytes allocation	6,073	6,694	5,523
4 ∼ 16 bytes allocation	6,142	6,511	5,664

Figure 4 shows the fragmentation ratio of each memory allocation mechanism where the x-axis shows the fragmentation ratio while the y-axis is time. In the result, the performance of *SOS* is poorer than those of others. This is because the *SOS*' memory space is divided by the fixed-size of blocks, which means the large amount of the internal fragmented blocks raises the fragmentation ratio. Based on these results, we can convince that the proposed *XMAS* performs significantly better than the existing memory allocation mechanisms.

[1] *Kernel version of each operating system; SOS: 05-july, Nano-Qplus: 1.6.0e.*



(a) 4 ~ 8 Bytes memory allocation

(b) 4 ~ 16 Bytes memory allocation

Fig. 4. Fragmentation ratio of the memory allocators

5 Conclusions

Wireless sensor networks consist of hundreds or even thousands of deployed sensor nodes that are designed under constraints of cost efficiency. Therefore, an operating system that runs on tiny sensor nodes needs a time and space-efficient memory allocation mechanism because each sensor node has only small memory space with limited batteries. In this paper, we proposed *XMAS* to improve the time and space-efficiency of memory management for sensor nodes. It is designed to minimize internal and external fragmentation ratio and to improve response time of the memory management for sensor nodes. Our experimental results showed that *XMAS* outperforms existing memory allocation mechanisms of sensor operating systems.

6 Availability

The whole source codes of *XMAS* are available on this homepage: `http://ssrnet.snu.ac.kr/~shyi/`

References

1. Bhatti, S., Carlson, J., Dai, H., Deng, J., Rose, J., Sheth, A., Shucker, B., Gruenwald, C., Torgerson, A., Han, R.: Mantis os: An embedded multithreaded operating system for wireless micro sensor platforms. ACMKluwer Mobile Networks and Applications (MONET) Journal, Special Issue on Wireless Sensor Networks (2005)
2. Lee, K., Shin, Y., Choi, H., Park, S.: A design of sensor network system based on scalable and reconfigurable nano-os platform. In: IT-Soc International Conference. (2004)
3. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine (2002) 102–114
4. Lundquist, J.D., Cayan, D.R., Dettinger, M.D.: Meteorology and hydrology in yosemite national park: A sensor network application. Lecture Note in Computer Science **2634** (2003) 518–528
5. Hirafuji, M., Fukatsu, T., Hu, H., Kiura, T., Laurenson, M., He, D., Yamakawa, A., Imada, A., Ninomiya, S.: Advanced sensor-network with field monitoring servers and metbroker. In: CIGR International Conference. (2004)
6. Levis, P., Madden, S., Gay, D., Polastre, J., Szewczyk, R., Woo, A., Brewer, E., Culler, D.: The emergence of networking abstractions and techniques in tinyos. In: First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004). (2004)
7. Levis, P., Culler, D.: Mate: a virtual machine for tiny networked sensors. In: International Conference on Architectural Support for Programming Languages and Operating Systems. (2002) 85–95
8. Han, C.C., Kumar, R., Shea, R., Kohler, E., Srivastava, M.B.: A dynamic operating system for sensor nodes. In: MobiSys. (2005) 163–176
9. Knuth, D.E.: The art of computer programming, vol. 1: Fundamental algorithms. Addison-Wesley (1973)
10. McKusick, M.K., Karels, M.J.: Design of a general purpose memory allocator for the 4.3bsd unix kernel. In: Proceedings of the San Francisco USENIX Conference. (1988) 295–303
11. Lea, D.: A memory allocator. Unix/Mail, 6/96 (1996)
12. Knowlton, K.C.: A fast storage allocator. Communications of the ACM **8** (1965) 623–625
13. Peterson, J.L., Norman, T.A.: Buddy systems. Communications of the ACM **20** (1977) 421–431
14. Page, I.P., Hagins, J.: Improving the performance of buddy systems. IEEE Transactions on Computers **C-35** (1986) 441–447
15. Masmano, M., Ripoll, I., Crespo, A., Real, J.: Tlsf: a new dynamic memory allocator for real-time systems. In: Euromicro Conference on Real-Time Systems(ECRTS'04). (2004)
16. Vahalia, U.: Unix internals: The new frontiers. Prentice Hall (1996)
17. Johnstone, M.S., Wilson, P.R.: The memory fragmentation problem: solved? ACM SIGPLAN Notices **34** (1999) 26–36
18. Crossbow: http://www.xbow.com/. (website)
19. Octacomm: http://www.octacomm.net/. (website)

An Adaptive Distributed Resource Allocation Scheme for Sensor Networks

Hock Beng Lim[1], Vinh The Lam[2], Mao Ching Foo[3], and Yulian Zeng[2]

[1] Intelligent Systems Center, Nanyang Technological University
[2] Singapore-MIT Alliance, National University of Singapore
[3] DSO National Laboratories, Singapore
limhb@ntu.edu.sg

Abstract. A major research challenge in the field of sensor networks is the distributed resource allocation problem, which concerns how the limited resources in a sensor network should be allocated or scheduled to minimize costs and maximize the network capability. In this paper, we propose the Adaptive Distributed Resource Allocation (ADRA) scheme, which specifies relatively simple local actions to be performed by individual sensor nodes in a sensor network for mode management. Each node adapts its operation over time in response to the status and feedback of its neighboring nodes. Desirable global behavior results from the local interactions between nodes.

We study the effectiveness of the ADRA scheme for a realistic application scenario; namely, the sensor mode management for an acoustic wireless sensor network to track vehicle movement. We evaluated the scheme via simulations, and also prototyped the acoustic wireless sensor network scenario using the Crossbow MICA2 motes. Our simulation and hardware implementation results indicate that the ADRA scheme provides a good tradeoff between performance objectives such as coverage area, power consumption, and network lifetime.

Keywords: Sensor Networks, Distributed Resource Allocation.

1 Introduction

Sensor networks have generated much research interest because they present many challenging issues and they have a wide range of potential applications. In the field of sensor networks, one of the research issues that remain open is the problem of distributed resource allocation - how to allocate, without a central coordinator, the limited sensing, processing or communication resources in the sensor network to monitor a dynamically changing environment.

In this paper, we propose the Adaptive Distributed Resource Allocation (ADRA) scheme, which addresses the distributed resource allocation problem from a different angle compared to that of existing work in the literature. The ADRA scheme specifies relatively simple local actions to be performed by individual sensor nodes in a wireless sensor network for mode management. Each node adapts its operation over time in response to the status and feedback of its neighboring nodes. Desirable global behavior results from the local interaction between nodes. The ADRA scheme is scalable since

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 770–781, 2006.

the coordination of the actions of neighboring nodes requires little communication. It is adaptive and robust with respect to the dynamic environment that the sensor network operates in.

Our scheme provides a general framework for efficient resource allocation in sensor networks. It can actually be applied to many sensor network applications and problems. In this paper, to evaluate the effectiveness of the ADRA scheme, we apply it to a realistic application scenario. The scheme is used to perform sensor mode management in an acoustic wireless sensor network that tracks vehicle movement in an open terrain. We simulated the scheme under the scenario, and also prototyped it using the Crossbow MICA2 motes. Our simulation and hardware implementation results indicate that the scheme provides a good tradeoff between performance objectives such as coverage area, power consumption, and network lifetime.

The rest of this paper is organized as follows. In Section 2, we review the related work and discuss our contributions in this paper. Section 3 presents the problem statement and the ADRA scheme. We describe the application scenario in Section 4. Section 5 discusses our algorithms to implement the ADRA scheme for the scenario. The methodology and results of our simulation study are presented and discussed in Section 6. For the hardware implementation of the scheme, we discuss the hardware platform, and the performance results in Section 7. Finally, we conclude this paper in Section 8.

2 Related Work

Different aspects of the distributed resource allocation problem have been investigated by researchers. In [1], resources are allocated for applications in a distributed real-time system by characterizing the specifications (e.g. hardware platform, quality of service constraint), and then developing appropriate heuristics to maximize the performance goal. An initial static allocation is determined to maximize the allowable workload increase, followed by the dynamic resource reallocation process to avoid QoS violation.

In [2,3], a systematic formulation to map the distributed resource allocation problem into the distributed constraint satisfaction problem (DCSP) was proposed. The mapping is sufficiently generalized and reusable to tackle some specific difficulties such as ambiguity and dynamism. The problem is then solved by finding a solution to the DCSP, which is actually the assignment of values for distributed variables to satisfy all distributed constraints.

In market-based techniques [4,5], the distributed system is modelled as the interaction between agents taking economic roles. Resources are allocated through buying and selling activities between agents. A seller seeks to maximize its earnings whereas a buyer seeks to minimize its spending. Resource requests and price notifications are communicated among the agents. Certain heuristics or strategies are used to control agent behaviors through the propagation of price information.

Another interesting approach to the multi-agent resource allocation problem is the auction and bidding techniques for allocating resources to tasks, such as combinatorial auction [6,7] and coalition formation [8].

Much of the work in the multi-agent systems community has focused on multi-agent negotiation over the allocation of resources. In [9], a ”contract net” framework for communication and control in a distributed system was proposed. It functions as the common medium for contract negotiation, which is an essential form of task distribution. The protocol for the negotiation process should help determine the content of exchanged messages, and it is not just a means of physical communication. For example, [10] employs a finite state machine as the heart of the negotiation protocol, while [11] uses an underlying strategy of combinatorial auction.

Our work differs from the previous work and makes two important contributions. First, our ADRA scheme is a scalable and adaptive distributed resource allocation scheme for sensor networks. It is scalable since it relies only on near-neighbor communications between nodes in a sensor network. It is adaptive since each node reacts to the environment (such as the presence of targets) as well as the status and feedback of its neighbors. These local node interactions produce desirable global system behavior.

Second, the ADRA scheme provides a general framework that is applicable to many applications. Unlike previous work which seldom consider realistic application scenarios, we have applied the ADRA scheme to a realistic application scenario. Also, the performance evaluation in previous work is usually done via analytical modeling or simulation of a generic sensor network. In our work, apart from evaluating the ADRA scheme via simulation, we have implemented the scheme on real sensor hardware and evaluated its effectiveness.

3 Adaptive Distributed Resource Allocation Scheme

3.1 Problem Statement

In the remainder of the paper, the basic entity in a sensor network is a sensor node which has sensing, processing, and communication capabilities. The sensor network has a set of stationary sensor nodes. Each node has a set of modes (or actions) that it is able to partake, and it can only choose one mode at any particular time instance.

During the operation of the sensor network, each sensor node's *utility* value is a function of several factors including the sensing coverage (or target detection), target localization, and target error minimization. For a sensor node, the rate of energy consumption affects its useful lifetime. When the sensor node is out of power, it is no longer able to sense, process, or communicate, and thus its utility value will be zero.

The sensor nodes have no prior information on the targets and their movement. They can sense and detect targets, and are able to obtain directional information of the targets from sensor measurements. It takes two sensor nodes detecting a target to localize the target. Minimizing the error in localization requires more than two sensor nodes detecting the target. The sensor nodes can communicate with their neighbors. However, from available communications and environment sensing, the nodes would not have full knowledge of the entire network.

3.2 The ADRA Scheme

We propose the Adaptive Distributed Resource Allocation (ADRA) scheme as a framework or methodology to guide sensor nodes for efficient resource allocation. The ADRA scheme is shown in Algorithm 1.

Algorithm 1. Adaptive Distributed Resource Allocation

```
Phase 1: Initialization
Query neighbors' mode status.
Get information about detected targets (if any).
Update local variables (e.g. utility, battery life).
Send information on detected targets to neighbors.

Phase 2: Processing
Receive information on targets from neighbors.
Fuse own detected target info with neighbors' detected target info.
Compute change in utility based on information from neighbors.
Compute own plan regarding sensor mode.
Optional : compute plan for neighbors.
Send information on the plan to neighbors.

Phase 3: Decision
Receive information on neighbors' plans.
Resolve own plan with neighbors' influence.
Execute the plan to change own sensor mode.
```

Under the ADRA scheme, a sensor node goes through many operational cycles repetitively in its lifetime. An operational cycle represents a complete and self-contained activity period during which the node gathers sufficient information regarding the targets from the ambient environment and its neighbors for decision making. It determines the necessary actions to adapt itself to the environment while aiming for maximal performance of the whole network. Each cycle is split into three phases. Within each phase, the ADRA scheme specifies the necessary local actions to be performed to achieve efficient mode management and sensor resource allocation.

In Phase 1 (Initialization), each node initializes its internal states and prepares itself by querying its neighbors' mode status and the environment information such as the targets within range. At the end of Phase 1, each node shares the gathered preliminary information with its neighbors. During Phase 2 (Processing), each node collects all preliminary information from neighbors. The information would be analyzed and combined with its own information to yield its behavioral plan, i.e. the likely action to be executed. Again, the plan will be shared among neighbors. Phase 3 (Decision) is the stage to make a final decision. With all necessary information and action plans from neighbors, a node is able to determine how it should act to maximize the overall performance of the network.

4 Mode Management in Acoustic Sensor Network

We consider an acoustic wireless sensor network deployed for the purpose of monitoring vehicle movement in an open terrain. In this network, the acoustic sensor nodes are powered by batteries. The scheme should provide a good tradeoff between the ability to provide coverage for the area of interest and localize the targets, and the battery power conservation to prolong the network lifetime.

Each acoustic sensor node has two modes: *on*, *standby*. When the acoustic sensor node is in the "on" mode, it has full sensing, processing, and communications functionalities. When the node is in the "standby" mode, it stops sensing the environment and has limited communications capabilities. The amount of battery life consumed by the node per time unit in this state is assumed to be ten times lesser than when the node is in the "on" mode. A node in "standby" mode can still communicate and exchange messages with its neighbors according to the ADRA scheme, and switch to the "on" mode to sense a target when necessary.

The acoustic sensor's sensing capability is omnidirectional in nature, i.e. it can detect a target's acoustic signal from any direction, with an error variance of one radian. A target is considered to be detected when it is within range of the sensor. Sensor measurements or target detections are in the form of bearing (or angular) values of the target with respect to the sensors monitoring it, which are combined to form a positional fix of that target. The target bearing values and messages from the ADRA scheme will be transmitted among neighboring nodes.

5 Algorithms

5.1 Stansfield Algorithm

For the ADRA scheme, we adopt the Stansfield algorithm [12] to combine the bearing values of a target detected by multiple sensor nodes to localize the target, i.e. to obtain a positional fix of the target. The Stansfield algorithm computes the positional fix of the target in the form of the best point estimate of the target coordinates, and an uncertainty ellipse that bounds the likely location of the target.

5.2 Mode Management in Acoustic Sensor Network

Our algorithm to perform the ADRA scheme for the acoustic scenario is shown in Algorithm 2. In the first phase (initAndSend), each node obtains its own sensor measurements of the targets' bearing values, and computes the targets' positional fixes using the Stansfield Algorithm. It also updates its own `potential`, which is the utility value used for deciding the mode of the node (on or standby). Then, it sends the information on the detected targets and its own mode to the neighbors.

Our algorithm to perform the ADRA scheme for the acoustic scenario is shown in Algorithm 2. In the first phase (initAndSend), each node obtains its own sensor measurements of the targets' bearing values, and computes the targets' positional fixes using the Stansfield Algorithm. It also updates its own `potential`, which is the utility value used for deciding the mode of the node (on or standby). Then, it sends the information on the detected targets and its own mode to the neighbors.

In the second phase (rcvProcessSend), each node receives the bearing values and positional fixes of targets from its neighbors. Then, it fuses its own and the neighbors' bearing values to obtain the new positional fixes of the targets. The node updates its own `potential`, and sends its `potential` and battery life to the neighbors.

In the third phase (rcvExe), each node receives the potential and battery life information from its neighbors. Based on the difference in battery life between itself and

Algorithm 2. Mode management in acoustic sensor network

```
main()
Constants : battPri, /* priority value for battery life conservation */
  covPri, /* priority value for coverage */
  locPri, /* priority value for localization */
  threshold /* threshold value */
Variables : potential, /* potential for on or standby mode */
  battLife, /* battery life of node */
  battLifeDiff /* battery life difference between self and neighbor */
repeat
    initAndSend();
    rcvProcessSend();
    rcvExe();
until termination of operation, or if node depletes its battery life

procedure initAndSend()
Query neighbors' mode status.
Get own sensor measurement of target(s) bearing value(s).
Compute target(s) positional fix(es) using Stansfield Algorithm.
Update own potential.
Send to neighbors : target(s) bearing value(s) and existing positional fix(es), own mode (on
or standby).

procedure rcvProcessSend()
Receive from neighbors : targets' bearing values and positional fixes, neighbors' modes.
Update potential.
Fuse and update current set of bearing value and positional fix with new values from self and
neighbors.
for each bearing value from self and neighbors do
    Increase own potential (by covPri).
end for
for each positional fix from self and neighbors do
    Increase own potential (by locPri).
end for
Send to neighbors : own potential and battLife.

procedure rcvExe()
Receive from neighbors : potential values and battLife info.
for each neighbor do
    Compute battLifeDiff.
    if (neighbor_battLife > battLife) then
        Decrease potential by (battPri * battLifeDiff).
    else
        Increase potential by (battPri * battLifeDiff).
    end if
end for
if (potential < threshold) then
    Switch to "standby" mode.
else
    Switch to "on" mode.
end if
```

its neighbors, the node computes its new `potential` value. After computing its new `potential`, the node decides whether to be ”on” or ”standby” by comparing the `potential` with a threshold value.

6 Simulation Evaluation

We conducted simulation studies of the application scenario using the Recursive Porous Agent Simulation Toolkit (Repast 3.0) [13], an open source agent-based simulation and modeling toolkit. It is written in Java, and is originally developed at the University of Chicago. In the simulation setup, we model attributes such as the simulation world size, number of sensor nodes, node deployment topology, number of targets and their paths, sensor modes, sensor measurements, and communications capabilities. We assume that the sensor nodes are aware of their locations in their deployment area, and they are time-synchronized.

6.1 Experimental Methodology

We simulate this scenario by modeling an array of sensor nodes deployed in a grid-like manner with several rows and columns. A sensor node’s sensing range and radio communication range overlaps with that of its neighbors. The spacing between two neighbors is the smallest distance such that a circle representing the sensing coverage area of an internal node only intersects with those coverage circles of its four neighbors and no other nodes. By simple geometric rule, the node spacing d is related to the sensing range sr by the equation: $d = sr\sqrt{2}$.

We test two configurations of different network sizes to investigate the scalability of the ADRA scheme: Net16 with 16 nodes (4×4 grid) and Net256 with 256 nodes (16×16 grid). The sensing range is set as 150m, and so the spacing between nodes is $d = 150\sqrt{2} = 212$m. As each node needs to exchange messages with its neighbors, the radio communication range must be larger than the node spacing d. In the simulation, the communication range is set to be 300m. The corresponding dimensions of the grid areas for Net16 and Net256 are 1060m × 1060m and 3604m × 3604m respectively. We also model a number of targets (8 and 24 targets for Net16 and Net256 respectively) moving across the terrain.

We study three cases of the acoustic sensor network operation. In the baseline (”WithoutAlgo”) case, the network does not use the ADRA scheme, i.e. all the nodes would be “on” until they exhaust their battery life. In the other two cases ”WithAlgoWithoutTarget” and ”WithAlgoWithTarget”, the network uses the ADRA scheme to control its operation. There are no targets in the former case, while there are targets to be tracked in the latter case.

Figure 1 shows a screenshot of the Net16 simulation. The sensing coverage radius of an active node is delineated by a circle. The absence of such a circle indicates that a node is in “standby” mode. The figure also shows the target bearing lines of sensors that have detected the targets, and the uncertainty ellipses surrounding the targets.

We use the *network coverage area* and the *sensor network lifetime* as performance metrics. The coverage area is defined to be the largest area such that any inside point

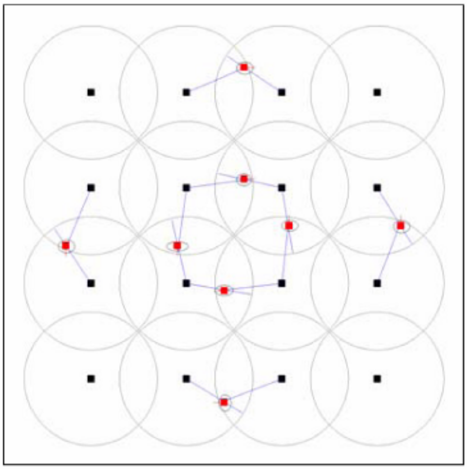


Fig. 1. Screenshot of acoustic sensor network simulation (Net16)

is covered by at least one circle, without double counting the regions where the circles overlap. Each sensor node starts off with a predefined battery life. As simulation time passes, each node consumes battery life at a varying rate according to the changes in its modes, until its battery life is depleted. We measure the coverage area of the Net16 and Net256 networks against time. As more and more nodes eventually use up their battery life, the trend is that the sensor network coverage area declines with time. We define the sensor network lifetime as the amount of time for the coverage area to drop to zero.

6.2 Results and Discussion

The coverage area against time for Net16 and Net256 are shown in Figure 2 and 3 respectively. Also, the results for the average coverage area and the network lifetime for each network under the three cases are shown in Table 1.

The baseline (”WithoutAlgo”) case is the simplest to understand. As all the nodes are always “on”, the maximum possible coverage area is provided until all nodes deplete their battery life. In our simulation, we set the nodes’ initial battery life such that the network lifetime will be 200s for both the Net16 and Net256 baseline cases.

In the ”WithAlgoWithoutTarget” case, the network converges into two steady state configurations. In one configuration, the nodes at alternating diagonals are “on” and the rest are in “standby” mode. In the other configuration, the modes of the “on” and “standby” nodes are reversed. Triggered by the adaptive nature of the ADRA scheme, the network periodically switches back-and-forth between these two configurations by reversing the modes of the nodes. In this manner, the battery life consumption of the nodes is balanced as much as possible across the network as time progresses. Consequently, half of the nodes are “on” at steady state, and the network life time in this case is double that of the ”WithoutAlgo” case. However, as not all the nodes are “on” at all times, the tradeoff is that the coverage areas for the Net16 and Net256 networks have dropped to 75.4% and 86.3% of the maximal coverage in the baseline case respectively.

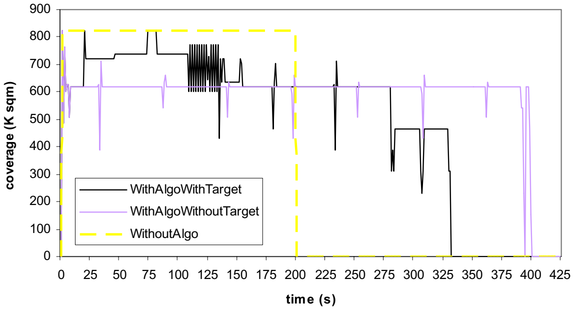


Fig. 2. Coverage area versus time (Net16)

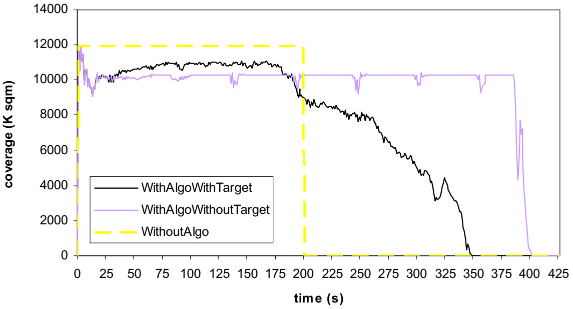


Fig. 3. Coverage area versus time (Net256)

The ”WithAlgoWithTarget” case shows the effect of target tracking. The coverage area rises above the ”WithAlgoWithoutTarget” case at the beginning since the ADRA scheme turns on more nodes to help track the targets. With more nodes turned on, the power consumption is higher too. Eventually, the coverage area starts to drop as more and more nodes deplete their battery life. Thus, the network lifetime in this case is shorter than that of the ”WithAlgoWithoutTarget” case, but still longer than the ”WithoutAlgo” case. In the Net16 network, the network lifetime of ”WithAlgoWithTarget” is 166% that of the ”WithoutAlgo” case, while its coverage area is 76.6% that of the ”WithoutAlgo” case. The corresponding numbers for the Net256 network are 174.5% and 72.9% respectively.

Table 1. Coverage area and network lifetime

Net16 cases	Avg cov area (K m^2)	Network lifetime (s)
WithoutAlgo	821.8	200
WithAlgoWithoutTarget	619.3	401
WithAlgoWithTarget	630.9	332
Net256 cases	**Avg cov area (K m^2)**	**Network lifetime (s)**
WithoutAlgo	11929.3	200
WithAlgoWithoutTarget	10295.0	401
WithAlgoWithTarget	8702.3	349

In general, the ADRA scheme provides a significant improvement in network lifetime at the cost of a small decrease in the coverage area in both the Net16 and Net256 networks. Our results also shows that the ADRA scheme is scalable, and it can work well for larger networks too.

7 Hardware Implementation

7.1 Hardware Platform

To assess the actual performance of the ADRA scheme on real sensor hardware, we prototyped the acoustic sensor network scenario using the Crossbow MICA2 motes [14]. The motes are programmed in nesC [15] under the TinyOS development environment [16]. nesC is an extension of the C programming language. TinyOS is an event-driven operating system designed for sensor nodes. Our hardware testbed deploys 16 MICA2 motes in a 4x4 grid resembling that of the Net16 simulation in Figure 1. We also use the Crossbow MTS310CA sensor boards, which are plugged onto the MICA2 motes.

The total power consumption of a mote is an aggregation of the power consumption of its components, including the processor, radio, logger memory, and sensor board. Each component can operate in different functional modes. The power consumption of each component is different when operating in different modes. For example, the microcontroller draws around 8mA during full operation but only 8 μA during sleep mode [14]. Therefore, the overall power consumption is the sum of all component-based consumptions, averaged by the duty cycles of operational modes for each component. In our testbed, we empirically measured the power consumption of a MICA2 mote as approximately 25mA in active mode and 11mA in standby mode.

Our testbed only aims to prototype the acoustic sensor network scenario to demonstrate a proof-of-concept hardware implementation of the ADRA scheme. We disable all sensors except the acoustic sensor for power saving. The acoustic sensor on the MTS310CA sensor board is a microphone capable only of providing the magnitude reading of an acoustic signal. It is unable to provide the direction of arrival of an acoustic signal. Thus, we simplify our implementation of Algorithm 2 so that it performs only target detection but not target localization. Fortunately, this simplification does not have any big impact on demonstrating the efficacy of the ADRA scheme because our key performance metrics of coverage area and network lifetime are still relevant.

We use the beeping sound of the MTS310CA sounder (at acoustic frequency of 4KHz) to emulate the noise from a target. The spacing between two motes is related to the sensing range of the acoustic sensor, in a similar manner as in the simulation. From empirical measurements, we determine that a good spacing distance between the motes in our testbed is 50cm, as it is a suitable distance for detecting the MTS310CA sounder signal with a reasonable internal threshold.

7.2 Results and Discussion

Figure 4 shows the coverage area against time for the three cases measured on our 16-node testbed. The unit of time in the x-axis is in terms of time cycles. A short time

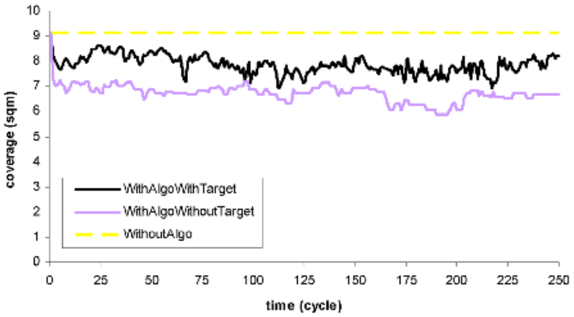


Fig. 4. Coverage area versus time for 16-node MICA2 testbed

cycle duration makes packet collision reduction and power management difficult to control, whereas a long time cycle hampers the target detection. In our implementation, we empirically determined that a time cycle duration of 5 seconds gives acceptable performance. Each MICA2 mote is powered by a pair of AA batteries which can last for days. To expedite the data collection and analysis process, we consider only the first 250 cycles as shown in Figure 4.

As expected, the baseline (”WithoutAlgo”) case is very simple: all nodes are always “on” and hence the coverage area is constant at 9.1 m^2 over time. In the ”WithAlgoWithoutTarget” case, the coverage area dropped to an average value of 6.7 m^2. When targets are introduced in the ”WithAlgoWithTarget” case, more nodes are triggered to turn on and hence we get a higher coverage area than the case of ”WithAlgoWithoutTarget”. In Figure 4, the graph representing ”WithAlgoWithTarget” is above that of the ”WithAlgoWithoutTarget” case during the duration of 250 cycles. The average coverage area in the presence of targets is 7.9 m^2.

During the duration of 250 cycles, the coverage area of the ”WithAlgoWithoutTarget” case and the ”WithAlgoWithTarget” case are 73.6% and 86.8% that of the ”WithoutAlgo” case respectively. However, if we were to run this experiment for a longer time, the coverage area graphs for both these cases should drop as more and more motes deplete their batteries, just like in the simulation.

8 Conclusion

We have proposed the Adaptive Distributed Resource Allocation (ADRA) scheme, which specifies the tight coordination amongst neighboring nodes in a wireless sensor network for action and decision making in mode management. The ADRA scheme helps sensor networks adapt to changes in the ambient environment dynamically and responsively. We demonstrated the ADRA scheme’s efficacy by studying a realistic applications of an acoustic sensor network that adopts the scheme for sensor mode management. The results from our simulations and hardware prototype show that the ADRA scheme can provide good coverage area and target tracking, while achieving significant power saving and prolonging the network lifetime.

References

1. Ali, S., Kim, J., Siegel, H., Maciejewski, A., Yu, Y., Gundala, S., Gertphol, S., Prasanna, V.: Greedy heuristics for resource allocation in dynamic distributed real-time heterogeneous computing systems. In: Proc. of the 2002 Intl. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA 02), Las Vegas, NV (2002) 519–530
2. Modi, P., Scerri, P., Shen, W.M., Tambe, M.: Distributed Resource Allocation: A Distributed Constraint Reasoning Approach. In: Distributed Sensor Networks: A Multiagent Perspective. Kluwer Academic Publishers (2003)
3. Salido, M., Barber, F.: Distributed constraint satisfaction problems for resource allocation. In: Proc. of the AAMAS 2003 Workshop on Decentralized Resource Allocation, Melbourne, Australia (2003)
4. Mainland, G., Kang, L., Lahaie, S., Parkes, D., Welsh, M.: Using virtual markets to program global behavior in sensor networks. In: Proc. of the 11th ACM SIGOPS European Workshop, Leuven, Belgium (2004)
5. Wellman, M.: Market-Oriented Programming: Some Early Lessons. In: Market-Based Control: A Paradigm for Distributed Resource Allocation. World Scientific (1996)
6. Ostwald, J., Lesser, V.: Combinatorial auctions for resource allocation in a distributed sensor network. Technical Report 04-72, Univ. of Massachusetts CS Department (2004)
7. Nisan, N.: Bidding and allocation in combinatorial auctions. In: Proc. of the 2nd ACM Conf. on Electronic Commerce, Minneapolis, MN (2000) 1–12
8. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence **101**(1-2) (1998) 165–200
9. Davis, R., Smith, R.: Negotiation as a metaphor for distributed problem solving. Artificial Intelligence **20**(1) (1983) 63–109
10. Mailler, R., Lesser, V., Horling, B.: Cooperative negotiation for soft real-time distributed resource allocation. In: Proc. of the 2nd Intl. Joint Conf. on Autonomous Agents and Multiagent Systems, Melbourne, Australia (2003) 576–583
11. Frank, M., Bugacov, A., Chen, J., Dakin, G., Szekely, P., Neches, B.: The marbles manifesto: A definition and comparison of cooperative negotiation schemes for distributed resource allocation. In: Proc. of the 2001 AAAI Fall Symp. on Negotation Methods for Autonomous Cooperative Systems, North Falmouth, MA (2001) 36–45
12. Stansfield, R.G.: Statistical theory of df fixing. Journal of the IEE (London), Part IIIA **94**(15) (1947) 762–770
13. (Repast 3.0 - recursive porous agent simulation toolkit, http://repast.sourceforge.net)
14. (Mica2 user's manual, http://www.xbow.com/support/support_pdf_files/mts-mda_series_users_manual.pdf)
15. Gay, D., et. al.: The nesc language: A holistic approach to networked embedded systems. In: Proc. of the 2003 ACM SIGPLAN Conf on Programming Language Design and Implementation (PLDI 2003), San Diego, CA (2003) 1–11
16. Hill, J., et. al.: System architecture directions for networked sensors. In: Proc. of the 9th Intl Conf on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), Cambridge, MA (2000) 93–104

Sequential Approach for Type-Based Detection in Wireless Sensor Networks

Dmitry Kramarev[1], Insoo Koo[2], and Kiseon Kim[1]

[1] Department of Information and Communications,
Gwangju Institute of Science and Technology (GIST),
Gwangju, Republic of Korea
{kramarev, kskim}@gist.ac.kr
[2] Department of Electrical Engineering,
University of Ulsan
Ulsan, Republic of Korea
iskoo@ulsan.ac.kr

Abstract. In this paper, we consider a sequential approach for decentralized detection problem in wireless sensor networks, and propose a new scheme for data fusion in the case of spatially and temporally identically and independently distributed observations. In addition, we investigate the performances of the proposed scheme in terms of average number of observations and total energy consumption, and further compare the results with those of a non-sequential scheme. As a result, we determined the region of individual node power where the proposed scheme outperforms the non-sequential scheme in terms of both average number of observations and total energy consumption.

Keywords: decentralized detection, sequential test, method of types.

1 Introduction

Problems related to decentralized detection in wireless sensor networks have received a lot of attention in the past couple of decades. Many different algorithms and approaches to solve these problems have been proposed in literature [1]-[4]. Typically, a number of distributed sensors monitors the state of the environment and carries out the observations of environmental parameters. If necessary, the observed data can be initially processed in the sensor nodes, and then are sent to the fusion center, which processes a final detection. Some recent works in this area assume that all observation data are available at the fusion center without any losses or changes. This idealistic type of detection is referred to as centralized detection, and has the best accuracy [2],[3]. However, in a real situation each sensor node has limited energy resources, and it is impossible to achieve perfect quality data transmission. As a result, information received by the fusion center is corrupted by channel noises and the accuracy of the decentralized detection is consequently degraded.

Another difficulty in decentralized detection is related to the abilities of each sensor to process data and make local decisions. A number of recent works [1]-[5] assume that full information regarding the source statistics is known at each

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 782–792, 2006.

node;that smart sensors produce local decisions based on their own observations. Transmission of local decisions to the fusion center can be organized with high reliability and low energy consumption, as compared to the transmission of all observed data. However, the storage of statistical information and ability to make local decisions require more complex, expensive and powerful sensors. Additionally, network-scale updates of source statistics for different tasks are necessary in each node, which can be quite a burdensome operation for some applications.

To avoid the difficulties mentioned above, one method for decentralized detection was proposed by Liu and Sayeed [6], and is called "type-based detection." This scheme employs simple, or not smart, sensors and local decisions are not made in each node. In this case, the nodes are used as counters of observation type (histogram) statistics. The scheme allows for the transmission of relatively small amounts of data from sensors to the fusion center. In addition, no statistical information is used in the sensors. At the same time, the accuracy of this type-based scheme asymptotically achieves the accuracy of centralized detection even under total power constraints. However, this scheme only considers fixed-sample-size detectors with a fixed number of observations.

Another approach, referred to as sequential detection, is described in [5]. Usually, sequential tests require a lesser number of observations as compared to non-sequential tests [5]. At the same time, it is shown in [6] that simultaneous data transmission through a multiple access channel dramatically reduces the influence of noise, and significantly improves the accuracy of detection. However, sequential data transmission requires a significantly greater amount of energy to send the data from the sensors to the fusion center [7]. With these two antagonisms, we are motivated to propose a new scheme for data fusion referred to as the "sequential approach for type-based detection", where type-based detection and sequential data transmission are combined. Furthermore, we construct the sequential detection rule and derive the performance formulas of the proposed scheme in terms of two performance metrics: the average number of observations, and the energy consumption. The parameter region where the proposed scheme outperforms other detection schemes is also determined.

The rest of this paper is organized as follows. The system model and proposed scheme are described in Section 2. Next, Section 3 overviews the related work on this topic. The theoretical analysis is provided in Section 4. The results of simulations are presented in Section 5, and Section 6 contains the conclusions of this work.

2 System Model and Proposed Scheme

In this paper we consider a sensor network with a simple structure where all sensors are directly connected with the fusion center. K nodes of this network are deployed in the environment. The observations of each sensor at discrete time step t are given in the n-length sequence $\bar{x}_k(t) = \{x_{k,i}(t)\}_{i=1}^{n}, k = 1 \ldots K$. The data are quantized to $\chi + 1$ levels, and $x_{k,i}(t)$ obtains values from the discrete alphabet $\widetilde{A} = \{a_0, a_1, \ldots, a_\chi\}$. We assume that all the observations are identically and independently distributed according to one of two possible discrete probability distributions on $\widetilde{A} : Q_1$ with probabilities $p_{Q_1}(a_m) = p_{1,m}$ and Q_0 with probabilities $p_{Q_0}(a_m) = p_{0,m}$, where $m = 0 \ldots \chi$. Then, the k-th node produces the type-information of the observations, $\bar{T}_k(t) = \{T_{k,m}(t)\}_{m=1}^{\chi}, k = 1 \ldots K$ [8]. The value of $T_{k,m}(t)$ can be represented as the relative frequency of $a_m \in \widetilde{A}$ in the sequence $\bar{x}_k(t)$:

$$T_{k,m} = \frac{N_{\bar{x}_k(t)}(a_m, t)}{n} , \tag{1}$$

where $N_{\bar{x}_k(t)}(a_m, t)$ is the number of occurrences a_m in $\bar{x}_k(t)$. The type-information is subsequently sent through a noisy multiple-access channel to the fusion center with individual power assigned for every transmission P_{ind}. As a result, the received signals $R_1(t), R_2(t), \ldots, R_\chi(t)$ take the form [6]:

$$R_m(t) = \frac{1}{K} \sum_{k=1}^{K} T_{k,m}(t) + \frac{\omega_m(t)}{K} , \tag{2}$$

where $m = 1 \ldots \chi$, and $\omega_m(t)$ is channel gaussian noise with a zero mean, and variance $1/P_{ind}$. Let us denote $\bar{R}(t) = \{R_m(t)\}_{m=1}^{\chi}$. The size of $\bar{R}(t)$ and $\bar{T}_k(t)$ is χ, whereas the size of used alphabet is $\chi + 1$.

Based on the received signals, the fusion center processes the hypothesis testing $H_1 : Q = Q_1$ versus $H_0 : Q = Q_0$ using a sequential decision rule, which consists of a stopping rule and a final decision rule. If the number of observations is not sufficient for making a final decision, the sensors send one more series of observations, made in the same manner as the previous one. When the conditions of the stopping rule are satisfied, the observations are stopped and the hypothesis testing is held by the final decision rule.

The desired accuracy is given by the probabilities of detection and false alarm, P_d and P_{fa}, respectively. The performance of type-based sequential detection can be evaluated in terms of average number of observations and average amount of energy consumption. The average numbers of observations under each hypothesis, $E(L_\theta)$, are counted as:

$$E(L_\theta) = nKE(l_\theta) . \tag{3}$$

The average amount of energy consumption for data transmission, $E(P_\theta)$, is:

$$E(P_\theta) = K\chi P_{ind} E(l_\theta) . \tag{4}$$

In both cases, $E(l_\theta)$ is the average number of data transmissions. Here $\theta = 0, 1$ is the indexes of hypothesis H_1 or H_0. Since we are going to obtain the parameters given by (3) and (4), our task is to find the average numbers of transmissions $E(l_\theta)$ under each hypothesis.

For comparison of our scheme with another detector, we selected the non-sequential type-based scheme with a fixed number of observations, proposed in

[6], with the same accuracy P_d and P_{fa}. The principles of data collection and transmission are the same as described for the sequential case. However, type information is sent to the fusion center only one time. Therefore, in this fixed-size scheme the sequence length, observed by each sensor, is n_{fs}, ($n_{fs} > n$). Subsequently, the number of observations and the energy consumption are the same under each hypothesis, such that:

$$L_{fs} = n_{fs}K \ , \tag{5}$$

and

$$P_{fs} = K\chi P_{ind,fs} \ , \tag{6}$$

where $P_{ind,fs}$ is the individual power that is used to send type-information in non-sequential type-based scheme.

Our second task is to show the superiority of the sequential scheme over the non-sequential scheme, and to explain the influence of parameters such as P_{ind} and n on the performance of the proposed scheme.

3 Related Work

In this section we briefly overview the theoretical background which will be used for the performance analysis of the proposed scheme in Section 4. First, we discuss the theory of sequential detection, and then the non-sequential Neyman-Pearson approach.

3.1 Sequential Detection

According to the Wald-Wolfowitz theorem [5], the rule which requires the smallest average number of observations for pre-given accuracy (P_d and P_{fa}) is the sequential log-likelihood ratio test, given as:

$$\begin{cases} \qquad\quad \Lambda(t) \leqslant \log(A) & \text{Accept } H_0, \\ \log(A) < \Lambda(t) \leqslant \log(B) & \text{Take more observations,} \\ \log(B) < \Lambda(t) & \text{Accept } H_1; \end{cases} \tag{7}$$

where $\Lambda(t)$ is the sequential log-likelihood ratio:

$$\Lambda(t) = \Lambda(t-1) + \log \frac{p_{Q_1}(\bar{R}(t))}{p_{Q_0}(\bar{R}(t))} \ . \tag{8}$$

Here, $\Lambda(0) = 0$. As the next step in the construction of the sequential decision rule, we need to define two thresholds A and B. If these are chosen as:

$$A = \frac{1 - P_d}{1 - P_{fa}} \ , \tag{9}$$

and

$$B = \frac{P_d}{P_{fa}} \ , \tag{10}$$

then independent of probability distributions Q_1 and Q_0, the actual probabilities of misdetection and false alarm can be given as:

$$P_m^a \leqslant \frac{1-P_d}{1-P_{fa}} , \tag{11}$$

and

$$P_{fa}^a \leqslant \frac{P_{fa}}{P_d} . \tag{12}$$

For the rule described above, the average number of observations L^s under each hypothesis is given by:

$$E(L_1^s) \approx \frac{1}{E(\Lambda(1)|H_1)} \left[(1-P_d) \log \frac{1-P_d}{1-P_{fa}} + P_d \log \frac{P_d}{P_{fa}} \right] . \tag{13}$$

and

$$E(L_0^s) \approx \frac{1}{E(\Lambda(1)|H_0)} \left[(1-P_{fa}) \log \frac{1-P_d}{1-P_{fa}} + P_{fa} \log \frac{P_d}{P_{fa}} \right] , \tag{14}$$

3.2 Non-sequential Detection

Next let us consider a non-sequential detector with a fixed number of recorded observations, which will be used in the comparison with the proposed scheme. Here, the Neyman-Pearson criterion is chosen [1], where the probability of false alarm is constrained to an acceptable value, and the probability of detection is maximized. By the Neyman-Pearson lemma, the best decision rule is also given by the log-likelihood ratio test:

$$\Lambda_{fs} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda , \tag{15}$$

where

$$\Lambda_{fs} = \log \frac{p_{Q_1}(\bar{R}_{fs})}{p_{Q_0}(\bar{R}_{fs})} . \tag{16}$$

The threshold λ is selected to constrain the probability of false alarm, such that:

$$\int\limits_{\Lambda_{fs}>\lambda} p_{Q_0}(x_1, x_2, \ldots, x_\chi) dx_1 dx_1, \ldots, dx_\chi \leqslant P_{fa} . \tag{17}$$

Hence, for the selected threshold, the probability of detection is given as:

$$P_d = \int\limits_{\Lambda_{fs}>\lambda} p_{Q_1}(x_1, x_2, \ldots, x_\chi) dx_1 dx_1, \ldots, dx_\chi . \tag{18}$$

4 Performance Analysis of the Proposed Scheme

In this section, we analyze the performance of sequential type-based detection in terms of average number of observations and average amount of energy consumption. For analysis, we first obtain the log-likelihood ratio for the detector and then the thresholds for pre-given accuracy. According to (1) and (2), the signals $R_{m,t}$, received by the fusion center at time step t, can be given as:

$$R_m(t) = \frac{1}{K}\sum_{k=1}^{K}\frac{N_{\bar{x}_k(t)}(a_m,t)}{n} + \frac{\omega_m(t)}{K} = \frac{N_X(a_m,t)}{Kn} + \frac{\omega_m(t)}{K} , \qquad (19)$$

where $N_X(a_m,t)$ is the number of times, the symbol a_m appeared in all observations at time step t. At every time step $N_X(a_m,t)$ is a random variable with binomial distribution, which has parameters Kn as the number of experiments and $p_{\theta,m}$ as the probability of event a_m in one experiment under hypothesis H_θ. Since $N_X(a_m,t)$ and ω_m are mutually independent, the joint pdf takes the form:

$$f_{\theta,m}(R_m(t)) = \sum_{i=0}^{Kn}\binom{Kn}{i}p_{\theta,m}^{i}(1-p_{\theta,m})^{Kn-i}\frac{\exp\left\{-\frac{\left(R_m(t)-\frac{i}{Kn}\right)^2}{2\sigma^2}\right\}}{\sqrt{2\pi}\sigma} , \qquad (20)$$

where

$$\sigma = \frac{1}{K\sqrt{P_{ind}}} . \qquad (21)$$

Now we can derive the sequential log-likelihood ratio based on (7)-(12). In the case of continuous probability density functions, the sequential log-likelihood ratio are given as:

$$\begin{aligned}\Lambda(t) &= \Lambda(t-1) + \log\frac{p_{Q_1}(\bar{R}(t))}{p_{Q_0}(\bar{R}(t))}\\ &= \Lambda(t-1) + \log\frac{\prod_{m=1}^{\chi} f_{1,m}(R_m(t))}{\prod_{m=1}^{\chi} f_{0,m}(R_m(t))}\\ &= \Lambda(t-1) + \sum_{m=1}^{\chi}\log\frac{f_{1,m}(R_m(t))}{f_{0,m}(R_m(t))} .\end{aligned} \qquad (22)$$

By combining (20)-(22), we can calculate the value of the log-likelihood ratio $\Lambda(t)$ since the values of the received signals are known.

As the next step, let us define thresholds $A(t)$ and $B(t)$ in order to perform the test in (7). It is straightforward to show using (9)-(12), that if the thresholds are given as:

$$A(t) = A = 1 - P_d , \qquad (23)$$

and

$$B(t) = B = \frac{1}{P_{fa}} \ , \tag{24}$$

then the actual probabilities of detection and false alarm are greater and less than P_d and P_{fa}, respectively.

To analyze the performance parameters of the detector, we further need to determine the values of (13) and (14). To do this we have to find the mean of $\Lambda(1)$ under each hypothesis, such that:

$$\begin{aligned} E(\Lambda(1)|H_1) &= \int\limits_{\mathbb{R}^\chi} \prod_{j=1}^{\chi} f_{1,j}(x_j) \sum_{m=1}^{\chi} \log \frac{f_{1,m}(x_m)}{f_{0,m}(x_m)} dx_1 \ldots dx_\chi \\ &= \sum_{m=1}^{\chi} \int\limits_{-\infty}^{\infty} f_{1,m}(x_m) \log \frac{f_{1,m}(x_m)}{f_{0,m}(x_m)} dx_m \\ &= \sum_{m=1}^{\chi} D(f_{1,m} \| f_{0,m}) \ , \end{aligned} \tag{25}$$

where $D(f_{1,m}\|f_{0,m})$ is a Kullback-Leibler distance [8] between the probability distributions given by $f_{1,m}$ and $f_{0,m}$. In the same manner, $E(\Lambda(1)|H_0)$ is given as:

$$E(\Lambda(1)|H_0) = -\sum_{m=1}^{\chi} D(f_{0,m}\|f_{1,m}) \ . \tag{26}$$

Then the average number of observation series or transmissions can be approximately found with (13) and (14):

$$E(l_1) \approx \frac{\dfrac{(1-P_d)(1-P_{fa})}{1-P_{fa}(1-P_d)} \log(1-P_d) - \dfrac{P_d}{1-P_{fa}(1-P_d)} \log P_{fa}}{\sum\limits_{m=1}^{\chi} D(f_{1,m}\|f_{0,m})} \ , \tag{27}$$

and

$$E(l_0) \approx \frac{\dfrac{P_d P_{fa}}{1-P_{fa}(1-P_d)} \log P_{fa} - \dfrac{1-P_{fa}}{1-P_{fa}(1-P_d)} \log(1-P_d)}{\sum\limits_{m=1}^{\chi} D(f_{0,m}\|f_{1,m})} \ . \tag{28}$$

Thus, we derived the approximate performance parameters of the sequential type-based detector. Moreover, with (3) and (4) we can estimate the approximate average number of observations and average amount of energy consumption under each hypothesis. In the next section, we will provide the results of the numerical analysis.

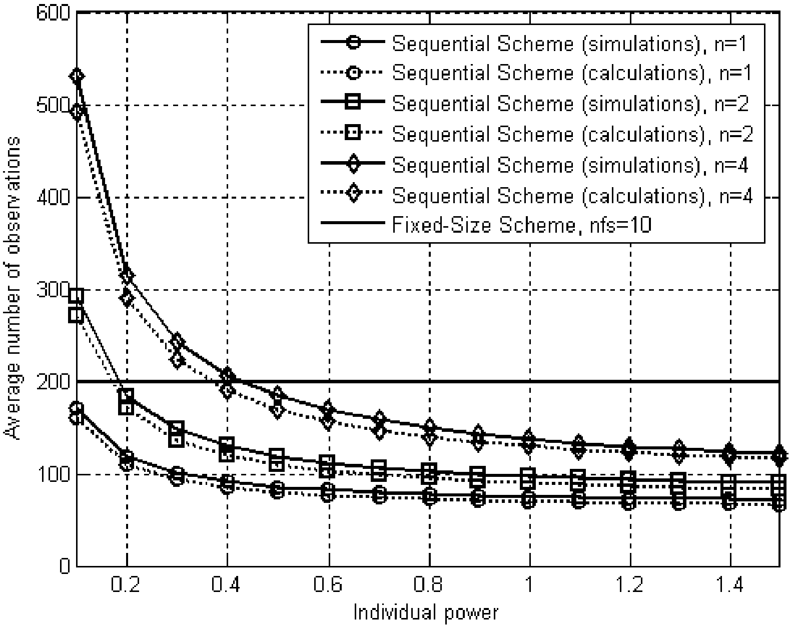


Fig. 1. Average number of observations

5 Numerical Results

In this section, we investigate the performance of the proposed scheme using both simulations and numerical analysis. Here, we examine the gain of sequential detection with respect to the average number of observations, and then investigate the average energy consumption. To do this, in our simulations we change the length of the observation sequence n and the individual power P_{ind}. For comparison, the non-sequential fixed-size detector with the same number of sensors and $n_{fs} > n$ is considered. We use a binary alphabet $\widetilde{A} = \{0, 1\}$; hence $\chi = 1$. The observations are distributed according the Bernoulli distribution $p_{Q_1}(1) = 0.7, p_{Q_0}(1) = 0.4$. The number of sensors $K = 20$, and the length of observations sequence for non-sequential detector is $n_{fs} = 10$.

For performance analysis, we first numerically solve equation in (17), and then for our chosen threshold we calculate the probability of detection P_d in (18). After that, both of these probabilities are set as a desired accuracy for the sequential detector. For simulations we set $M = 10^5$ as the number of rounds.

Figure 1 shows the average number of observations according to the individual power of each sensor, where the solid lines and the dotted lines represent the simulation results and calculation results, respectively. Observations are made with three different lengths of observation sequence $n, (n = 1, 2, 4)$. As a result, it is observed that the sequential scheme requires lesser number of observations in certain region of individual power. As can be seen from the figure, shorter

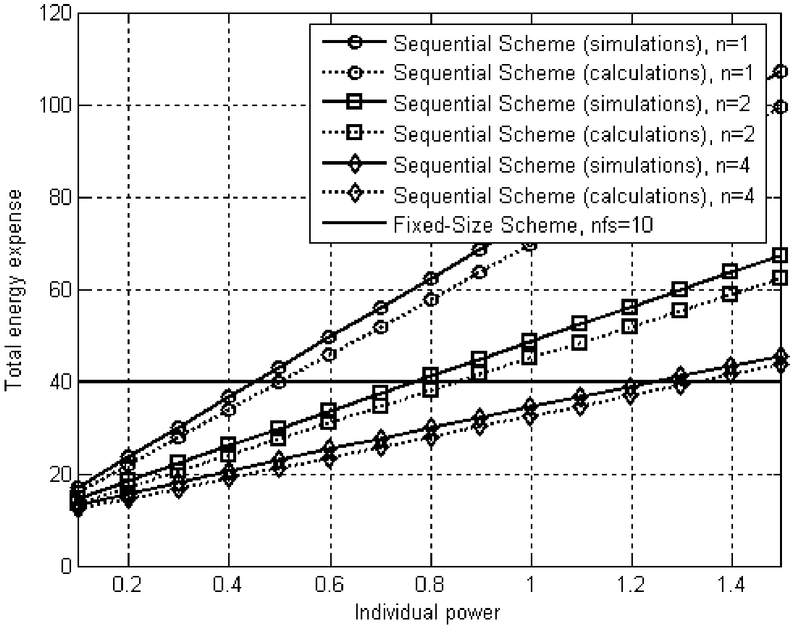


Fig. 2. Average energy expenses for data transmission

series ($n = 1$) provide better results. Additionally, it can be seen that detection with longer series such as $n = 4$ do not have superiority over the non-sequential scheme. Intuitively, this can be explained in that in the last sent series of observations more data can be regarded as redundant, after the log-likelihood ratio has exceeded one of the thresholds. Another observation is that in the area of relatively high individual power, additional energy does not provide significant gains in the number of observations. This is mainly due to the fact that the Kullback-Leibler distance in (27) is an upper-bounded function. Therefore, we can conclude that using relatively great values of P_{ind} is not effective.

Figure 2 shows the total energy consumption according to the individual node power, where the solid lines and the dotted lines represent the simulation results and calculation results, respectively. The values of energy expenses for the sequential detection are compared with those of the non-sequential detection. From Fig. 2, it can be seen, that total energy expense is an almost linear function of individual power. This is mainly due to the fact that the average number of observations tends to constant, as previously shown in Fig. 1. Figure 2 also shows that the sequential scheme achieves superiority over the fixed-size scheme in the region of relatively small individual power.

From Figs. 1 and 2 we can determine the region of individual node power in which sequential detection outperforms the fixed-sized scheme in terms of both number of observations and energy consumption.

6 Conclusion

In this work we have studied a type-based sequential detection scheme in wireless sensor networks, and a new approach of data transmission and detection was proposed. Here, the decision rule was constructed and the estimates of performance parameters were derived. Furthermore, the performances of this scheme were compared with those of a non-sequential type-based detector in terms of energy consumption and average size of observations. It was shown that by changing the parameters e.g. individual power of each sensor node and observation sequence length, we can obtain the performance which is better than that of a fixed-size detector. In addition, the average time necessary for decision-making can be reduced because a fewer number of observations is required. At the same time, sensors do not require the event's probabilistic model, and as such their construction becomes relatively simple, and initial settings for each new task are not necessary. In addition, the proposed method uses a smaller number of data transmissions. In our scheme, the sensors do not require ability for complex computations; however, a more powerful fusion center is required. As a second result, it was shown that for high values of individual power, additional increases in power do not significantly improve the performance, to the point that with high individual power the scheme becomes ineffective in terms of total energy expenses.

In this work, our scheme has been considered under some simplifications. First, the correlation between sensors' observations was ignored. Also, the parameters of the observation field were assumed to be temporally and spatially independent. However, in spite of these facts, these results can be used in practical applications which employ sensor networks and assume both sequential data observations and hypothesis testing.

Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment). Insoo Koo was particularly supported by Korea Research Foundation Grant funded Korean Government (KRF-2005-003-D00189).

References

1. P. K. Varshney, *Distributed Detection and Data Fusion*, Springer-Verlag, 1997.
2. J. N. Tsitsiklis, "Decentralized Detection", *Advances in Statistical Signal Processing*, Vol. 2, pp. 297-344, JAI Press, 1993.
3. R.S. Viswanathan, P.K. Varshney,"Distributed Detection with Multiple Sensors: Part1-Fundamentals", *Proceedings of the IEEE*, vol. 85, No. 1, pp. 54-63, Jan, 1997.
4. R. Blum, S.A. Kassam, H.V. Poor, "Distributed Detection with Multiple Sensors: Part2-Advanced topics", *Proceedings of the IEEE*, vol. 85, No. 1, pp. 54-63, Jan, 1997.

5. H.V. Poor. *An Introduction to Signal Detection and Estimation*, Springer-Verlag, New York, 1988.
6. K. Liu and A. Sayeed, "Optimal Distributed Detection Strategies for Wireless Sensor Networks", *42nd Annual Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, Oct. 2004.
7. L. Yu. A. Ephremides, "Detection Performance and Energy Efficiency of Sequential Detection in a Sensor Network", *Proceeding of 39th Hawaii International Conference on System Sciences-2006.*
8. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley New York, 1991.

System Support for Cross-Layering in Sensor Network Stack

Rajnish Kumar[1], Santashil PalChaudhuri[2], Charles Reiss[1],
and Umakishore Ramachandran[2]

[1] College of Computing, Georgia Institute of Technology, Atlanta, GA
[2] Department of Computer Science, Rice University, Houston, TX

Abstract. Wireless Sensor Networks are deployed in demanding environments, where application requirements as well as network conditions may change dynamically. Thus the protocol stack in each node of the sensor network has to be able to adapt to these changing conditions. Historically, protocol stacks have been designed with strict layering and strong interface between the layers leading to a robust design. However, cross-layer information sharing could help the protocol modules to make informed decisions and adapt to changing environmental conditions. There have been ad hoc approaches to facilitating cross-layer cooperation for adaptability. However, there has been no concerted effort at providing a uniform framework for cross-layer adaptability that preserves the modularity of a conventional protocol stack. This paper presents a novel service, information exchange service (IES), as a framework for cross-module information exchange. IES is a centrally controlled bulletin-board where different modules can post available data, or request for useful information, and get notified when the information becomes available. IES is integrated into the proposed *SensorStack* architecture that preserves the benefits of layering while facilitating adaptability. IES has been implemented in TinyOS and Linux, to show both the feasibility of the design as well as demonstrate the utility of cross-layering to increase application longevity.

1 Introduction

The explosive growth of the Internet has been spurred to a great extent by the modularity of the network protocol stack influenced by the OSI model. Adherence to the strict interfaces in the different layers, has enabled the independent development of robust protocols and their validation. While the focus on *modularity* (in the OSI model) has been a useful design guideline for Internet protocols, it is becoming clear that the decisions taken at runtime in the different layers could be better optimized with cross layer information. This is particularly true in dynamic settings when the network conditions can change quite dramatically. For example, researchers have shown the utility of explicit congestion notification from the routers to the transport layer [16], and link status information to the IP layer in a wireless setting [20].

While modularity is a key to protocol development and deployment, *adaptability* is emerging as a key determinant of performance, especially in a wireless setting. The design decisions in the protocol stack have to adapt to changing network conditions

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 793–807, 2006.

to maintain high performance. Such adaptability would be facilitated by the use of information available in different layers. Wireless Sensor Networks (WSN) amplify the need for sharing cross-layer information even further. In addition to the vagaries of the wireless network itself, the inherent resource constrained nature of the nodes pose additional challenges for the protocol stack. Nodes may join or leave the network to save their individual battery power, or environment conditions may vary, thus resulting in dynamic changes to the network topology. To allow for adaptability in the face of such dynamism, many WSN protocols have proposed piecemeal use of cross-layer information. For example, information from link layer may be used by the routing layer, and routing table information may be used by the application layer. However, it is difficult to foresee all the adaptation needs. Hence it is a challenge to standardize protocol interfaces that expose all useful cross-layer information. Optimizing energy, the single most important resource for WSN nodes, requires a holistic view of the stack instead of a layer-specific view available with such piecemeal solutions.

It is interesting to note that in spite of the increasing importance of cross-layering, it is still viewed with skepticism by the system community [12]. There are good reasons for this skepticism. Without careful system support, cross-layering may result in minimal benefits, may be misused, and may lead to unintended problems in the long run. There are three main reasons that point to the need for a careful design of cross-layering. First, without standard interfaces for information sharing, cross-layering could lead to inefficiencies. Often different modules may collect the same information to adapt their behavior, leading to wastage of computation, memory, and energy resources. For example, neighborhood information is useful for both network level routing and application level role assignment; hence uncoordinated information gathering will result in significant resource wastage (see Table I). Second, piecemeal evolution of cross-layering would lead to a spaghetti design of the protocol stack that is hard to maintain and verify due to the complex interactions among the different modules. Third, without a holistic approach to information sharing and event notification different protocol modules may make sub-optimal decisions leading to poor adaptability. For example, unless the application layer is notified of a sudden change in a link quality by the network layer, its role assignment decisions will be sub-optimal thus affecting application longevity.

The question being addressed in this paper is the following: How can we facilitate holistic adaptability without losing modularity? The main issue boils down to overcoming the inherent tension between adaptability and modularity: adaptability needs cross-layer information that seems difficult to obtain without affecting modularity. In other words, how can we structure cross-layer information sharing that does not compromise the robustness and maintainability of the protocol stack? This problem can be solved by decoupling the adaptability needs (that are cross-layer data oriented) from the modularity needs (that are functionality oriented). We use this intuition of decoupling cross-layer data from functionality to achieve an adaptable and modular protocol stack called *SensorStack*. At the heart of this stack is a novel *Information Exchange Service (IES)* that is available to all the layers. Through a publish/subscribe interface, IES provides a predicate-based event notification service that can be used by the protocol modules for information sharing and for making adaptive decisions. By absorbing the onus of managing the cross-layer data for adaptability, IES allows the protocol modules to focus on the functionalities to preserve modularity.

We have implemented IES in TinyOS [8], and assembled a representative SensorStack using heterogeneous sensor network (HSN) routing layer from shareware [10] and an application level data fusion layer called DFuse [13]. Through the implementation and evaluation we demonstrate the utility of SensorStack with IES both qualitatively and quantitatively. First, there is a qualitative benefit in that the component diagram of SensorStack with IES is simpler, with less interaction among the protocol modules for accessing cross-layer data. From a software engineering perspective, this design lends itself to maintainability and robustness of the protocol stack. Second, we show through micro-measurements that the code-path overhead of using IES to access cross-layer information is minimal. Third, we show that resource wastage (network, memory, and CPU) is minimized by aggregating the collection of neighborhood information that is shared by all the layers via IES.

This paper highlights several contributions:

1. By decoupling cross-layer information gathering and sharing from layer functionality, we facilitate adaptability without sacrificing modularity. The design and evaluation of IES is the primary contribution. There are two main nuggets in the design of IES:
   - *Data management module* provides a declarative publish/subscribe interface for protocols to share information facilitating a modular design. Further, it takes care of efficient use of the available limited node memory for information representation, eviction, and access.
   - *Event management module* provides a condition-based event notification mechanism to alert protocol modules of any changes in the environment thus facilitating adaptability.
2. Representative implementations of SensorStack with IES on TinyOS and Linux showing feasibility of the IES design to promote modularity and adaptability.
3. A simple taxonomy for cross-layer information sharing that provides transparency without affecting modularity.

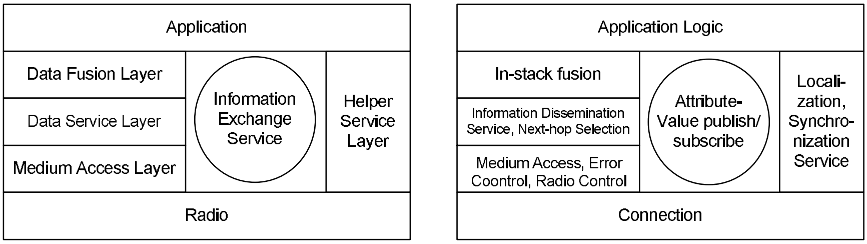
The rest of the paper is organized as follows. Section 2 proposes a taxonomy for sharable information in the SensorStack. IES design is presented in Section 3. The implementation and evaluation of IES are presented in Sections 4 and 5, respectively. Related work is discussed in Section 6. Section 7 concludes the paper with summary and future work.

2 Organization and Information Taxonomy

It is clear from the dynamic nature of WSN environment that decisions in the different layers of the protocol architecture can benefit from cross-layer information sharing. To this end, we first identify the different cross-layer information. Table 1 presents a snapshot of such information commensurate with the functionality provided by a particular layer. For example, the link layer (such as SP [15]) uses the physical condition of the environment as input to produce “link status” information as output that may be useful

Table 1. Cross-layer Information Produced by Different Protocol Layers

Protocol Layer	Sample Implementations	Produced information	Consumed information
Application	DFuse [13], Surge, TAG [14]	Resource requirement, Sensed data, Transmission schedule	Resource availability, Neighborhood, Topology
Routing	Directed diffusion [9], GPSR [11], SPEED [7], TAG tree routing	Routing metric values, topology information	Neighborhood, Application requirement
Medium access control, Duty cycle control	SMAC [19], Z-MAC, T-MAC [17], ASCENT [1], SPAN [2]	Duty cycle, Neighborhood information	Application requirement, Link information
Link layer	SP (sensornetwork protocol) [15]	Link status	Physical condition


Application
Data Fusion Layer
Data Service Layer
Medium Access Layer
Information Exchange Service
Helper Service Layer
Radio
Application Logic
In-stack fusion
Information Dissemination Service, Next-hop Selection
Medium Access, Error Coontrol, Radio Control
Attribute-Value publish/ subscribe
Localization, Synchronization Service
Connection

(A) Layered Architecture (B) Functionalities

Fig. 1. SensorStack: A proposed WSN stack

to other modules. This table is not meant to be exhaustive, but simply serves as a boiler plate for the taxonomy to be presented in this section.

One way to facilitate efficient decision making in each layer is to query the other layers for relevant information. Direct querying of peer modules, however, will result in breaking the modularity of the protocol architecture and lead to an unstructured and hard to maintain code base. The fundamental challenge is in developing a layered software architecture that preserves the modularity while allowing cross-layer information sharing. This raises several important research issues:

1. *Organization:* How do we organize the layered software architecture? One promising approach is to decouple the data needed for such information exchange from the functionality of the layered architecture.
2. *Taxonomy:* How do we develop a useful taxonomy for the kinds of information that will be needed by the different layers?
3. *Information Sharing:* How do we facilitate information sharing across the layers that is efficient and non-intrusive on the functionality provided by each layer?

Organization and Information Sharing. We propose a layered software architecture, called *SensorStack* (see Figure 1). At the heart of the SensorStack is *Information Ex-*

change Service (IES) that serves as an information broker among the different modules of the layered architecture and the application to facilitate cross-layer optimizations. Our approach is to decouple the data needed for such information exchange from the functionality of the stack. To this end, we first identify the different cross-layer data and develop a taxonomy for grouping them. Table 1 presents a snapshot of such data commensurate with the functionality provided by a particular layer. For example, the link layer (such as SP) uses the physical condition of the environment as input to produce "link status" information as output that may be useful to other modules. This table is not meant to be exhaustive, but simply serves as a boiler plate for the taxonomy being presented in this section.

Taxonomy. For the purpose of extensibility and documentation, we represent the attributes in the taxonomy in XML format. Clearly, it will be too inefficient to access information across layers by parsing the XML representation of each attribute. Rather, every attribute in the taxonomy is given a unique identifier known to all the layers, and the identifier is used to refer an attribute, thus avoiding the need of the XML parsing. We discuss the assignment of unique identifiers in Section 3.

The information produced and consumed by each layer to facilitate cross-layering can be grouped into four broad categories: *local resources*, *neighborhood*, *application requirements*, and *wildcard*.

1. *Local resources*: The application layer working in concert with the system monitoring module may produce information about the available node resources. Important resources to identify include details regarding available energy, CPU, memory, radio, and sensors.
2. *Application requirements*: An application may produce information that would be of use in the decision making at the routing and MAC layers.
3. *Neighborhood*: For scalability and load balancing reasons, WSN protocols take many decisions locally, and information about neighboring nodes play a very important role. Link layer protocols can produce link qualities of the neighboring nodes. Routing layer can collect routing metric based information, e.g., energy, location, and availability. Using this information, a link layer protocol can use the *timeOn* and the *timeOff* fields to minimize idle listening. The *listen* attribute can be used to inform the link layer to expect transmission from a neighbor, and it can be used for bi-directional low power communication [15].
4. *Wildcard*: There may be other information produced by a particular protocol layer that may not fall into the categories we have identified so far. Examples include abstract region specification for node cooperation [18], *area* abstraction in SPEED for multicast groups [7], *path* abstraction for energy-aware routing, and *role* abstraction for load balancing [13,6]. We group them as *wildcard* in our taxonomy.

3 Information Exchange Service Design

IES is an information repository for data that helps in cross-layer optimization. Such data may come from one of the modules of the SensorStack or even from the application itself. The taxonomy presented in Section 2 allows grouping the data into different categories irrespective of where it came from and enables easy access by a requesting module. Also, by centralizing all the information in this repository, SensorStack exercises control over access/update rights in a centralized manner.

Design Goals

1. *Efficient use of limited memory*: Memory is a scarce resource in embedded devices, therefore it has to be used prudently. It is quite easy to populate the repository with any and all information that may be useful for cross-layer optimization. However, only a fraction of this information may actually get used by the layers for adaptability. Therefore, IES must filter out unnecessary information and allow the protocol modules to share the memory efficiently.
2. *Simple interface for information sharing*: To ensure that the SensorStack remains modular, cross-module information sharing for adaptability should not lead to coupling of functionalities across layers. Towards this requirement, modules should be able to share data without concerns of synchronization and consistency. Further, the information access should be transparent to producers and consumers (i.e., producers do not know who the consumers are and vice versa). Therefore, IES should provide a simple interface that allows the modules to be implemented independently and efficiently.
3. *Extensibility*: IES should facilitate new information that is outside its repertoire of taxonomy to be added without any change in either the interface or in the underlying architecture. For example, if a new routing protocol is added to the stack, it should be able to publish any new metric into IES that may not be currently in the taxonomy.
4. *Asynchronous access to information*: Producers and consumers of data should not be burdened with unnecessary work. This goal translates to IES providing an asynchronous interface for information from publishers to be *pulled* into the repository, or information to consumers to be *pushed* from the repository, obviating the need for polling on the part of the producers and consumers.
5. *Complex event notification*: To make SensorStack adaptable, protocol modules should be notified of changes reactively. This goal translates to protocol modules being able to register events of interest (which may be a composite of several attributes) with the IES, and receive asynchronous notification when the condition becomes true.

The main objective of IES is to ensure that the SensorStack remains modular (goals 1-3) while supporting adaptability (goals 4 and 5). Access control, security, and protection are also important for IES, but they are outside the scope of this paper.

IES Architecture

As shown in Figure 2, IES comprises of two main components: *Data Management Module (DMM)* and *Event Management Module (EMM)*. DMM is responsible for helping achieve modularity, while EMM is responsible for helping achieve adaptability. DMM is designed as a shared memory abstraction augmented with a fully-associative

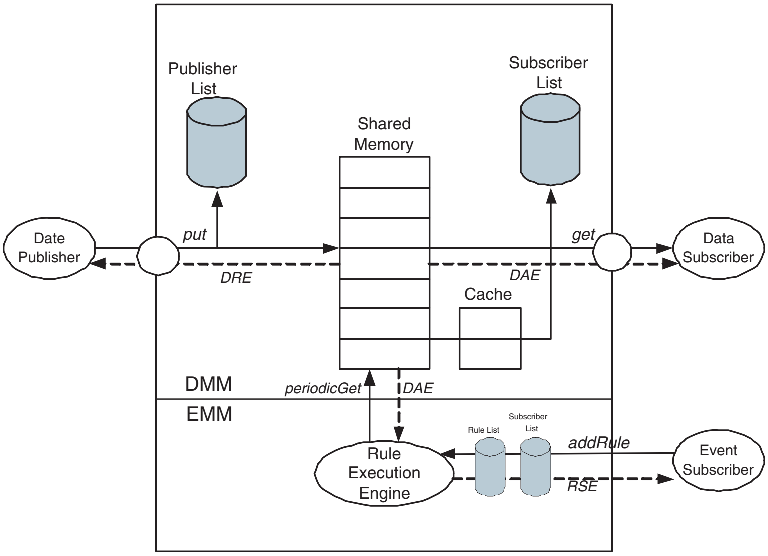


Fig. 2. IES architecture. Top half of the diagram shows the Data Management Module (DMM), while the bottom half shows the Event Management Module (EMM). Note that EMM acts as a subscriber to DMM component.

cache for efficient access; it offers a publish-subscribe interface for sharing information across layers. EMM is designed as a rule-based event notification engine such that protocol modules can be notified as requested, allowing them to adapt to the changes in the environment.

IES API consists of interfaces to publish, subscribe, and notify of any changes. DMM controls access to the data repository, and thus provides the *publisher* and *subscriber* interfaces. DMM maintains the publisher and subscriber list to support asynchronous exchange of information, especially to support the periodic get method. EMM is responsible for rule registration, execution, and notification to the subscribers. It provides a *watchdog* interface. Based upon the periodicity requirements of registered rules, EMM accesses the data published in DMM component using *periodicGet* call.

Below we elaborate on the design elements of IES that match the five goals identified in Section 3.

Efficient Use of Limited Memory. There are two aspects to efficiency in this context: firstly, prudent use of limited memory; secondly, fast access to the stored information. IES uses a block of pre-allocated memory as the information repository. The size of pre-allocation depends on the availability; however, in general it is the case that the amount of information that needs to be stored far exceeds the size. IES uses an LRU eviction policy when information has to be retired from it. There is a possibility that data may be retired from the memory before anyone requests it. For this reason, IES allows the producer to tag the data with a *sticky bit* to over-ride the LRU policy. Alternatively, IES also has the ability to asynchronously "pull" the data from a producer upon a request from a consumer.

For fast access to common data, IES uses a small fully-associative cache to keep the frequently requested data. Motivation behind using the cache is that if some information is requested by one module, it will likely be requested soon by other modules as well. This is especially true in SensorStack because different modules cooperate to achieve some common goal, e.g., energy optimization and hence may be querying some common attribute from IES (such as application's data requirement or the remaining battery level).

Simple Interface for Information Sharing. IES provides a publish/subscribe interface to the shared memory for transparent sharing of information. Publishers can *put* information in standard data format, and subscribers can *get* the same without knowing the publishers. Since information is stored as attribute-value pairs, multiple publishers can publish the same information with different attributes.

Protocol adaptation depends on the information provided by IES. Therefore, it is essential to ensure the freshness of data provided by IES. Producers need to know how frequently they need to update information published by them; consumers need to know if the information they are getting from IES is fresh. Asynchronous access (to be described shortly) deals with the former, while the latter is dealt with by the producers tagging information with an "expiration date".

Extensibility. Extensibility is achieved by using standard interfaces and data formats. IES is accessed using get/put over an *attribute_id*. *get* copies the value (if available) and returns the number of bytes corresponding to the data value; a return value of zero indicates that the data is currently unavailable. *put* writes the value into IES, and returns success/failure of the write operation as a boolean value.

```
int get( int attribute_id, byte[] value );
bool put( int attribute_id, byte[] value, int size);
```

Every *attribute_id* maps to a unique attribute description, an XML-based declarative description of the attribute. The attribute description corresponds to a unique entry in a standard ontology of information pertinent to the WSN.

Given a declaration, the *attribute_id* can be obtained by contacting an attribute name server. The idea of attribute name server is similar to a DNS lookup for an IP address. However, discussion of the name server design is outside the scope of this paper.

Asynchronous Access. With asynchronous access to IES through the publish/subscribe interface, there are four possibilities for information sharing between publishers (P) and subscribers (S) under the arbitration of IES: *push-push*, *push-pull*, *pull-push*, and *pull-pull*.

Push-Push choice yields the best result from the point of freshness of information but it has two downsides: There is a potential for wasted effort if there are no subscribers to published data that is being frequently updated. There is a potential for duplication of effort if multiple modules are publishing the same information. This may be a preferred choice for sharing neighborhood information that is prone to change quite frequently. There are similar pros and cons for the other three choices: *Push-Pull*, *pull-push*, and *pull-pull*.

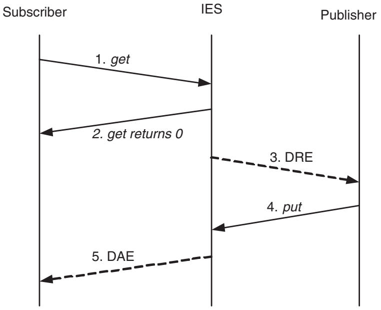


Fig. 3. Use of asynchronous signaling in IES when requested attribute is not available in IES

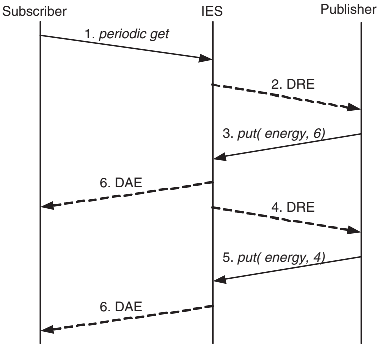


Fig. 4. Use of asynchronous signaling in IES for handling periodic updates

None of the above design choices serves best for exchange of cross-layer information; rather, different attributes may be best shared in different ways. For example, battery information may need to be shared in a reactive manner, while neighborhood information may be shared in a proactive manner. This observation motivated us to explore how to support all of the above design choices with a simple interface. While *proactive* communication can be handled by simple *get* and *put* methods, we added *event based signaling* in IES to support *reactive* communication.

A subscriber can request for reactive access to data either by setting up a periodicity in the *get* call, i.e., the subscriber gets data periodically, or by using complex event notification service, where the subscriber gets notified whenever a specific condition is met. For supporting periodic update, a *get* call expects *periodicity*, and a *put* call expects *expiration* as extra parameters. IES uses two events for signaling an update: *Data Request Event (DRE)* to request a publisher to *put* data when data is either expired or unavailable in IES, and *Data Available Event (DAE)* to notify a subscriber of an available update.

Figure 3 shows the use of asynchronous signaling to handle a failed *get* request because the requested attribute is not available in IES memory. This may happen because either none of the publishers *put* the attribute or the attribute was evicted, possibly expired, from the IES memory. IES selects a publisher (if any) for the requested attribute, and it raises DRE for that publisher. Once the publisher puts the attribute, IES notifies the waiting subscriber using DAE with a data pointer. The subscriber gets the data from IES. However, it may happen that before the subscriber handles the DAE event, the attribute gets evicted from IES, making the DAE void. To avoid an attribute from getting evicted before DAE is handled, IES keeps a time window before which the attribute is not evicted. A subscriber is expected to handle DAE within the time window, or else the subscriber must issue a fresh *get* call.

Figure 4 shows the use of asynchronous signaling to handle periodic update request. IES periodically checks if the requested attribute has expired or is unavailable in the repository; it then signals the publishers with a DRE. IES maintains the periodicity by using multiple timers. Of course, because of the asynchronous nature, the periodicity cannot be guaranteed accurately; it may depend on how fast the publishers are able to handle DREs.

Complex Event Notification. Often a protocol module may need to adapt its behavior when certain conditions are satisfied: changes in the environment, resource availability, and/or application requirements. Such adaptability to dynamic changes is quite common in wireless protocol stacks, and this goal is aimed at helping protocol modules monitor these changes in a fast and efficient manner.

IES uses predicate based rule representation to capture complex conditions. A rule takes the form of 'if *condition* do *notify* module P'. Conditions are well formed formulae over the IES attributes. For example, a simple rule can be 'if *(energy* $<$ *5)* do *notify* routing module'. IES keeps checking if the specified condition is satisfied, and when satisfied, it notifies the respective subscribers with *rule satisfied event (RSE)*. The two important design questions in this context are: how frequently should IES check for rule satisfaction, and how should IES handle the case when the condition attributes are not currently available in IES memory?

There is a trade-off between the promptness of event notification and incurred computation cost. Owing to the resource constrained nature of sensor devices, IES checks for condition satisfaction only periodically. IES uses the frequency of access/updates to the attributes to fine tune this periodicity. In case an attribute is unavailable at the time of checking, IES signals the publishers for the required attribute data.

4 Implementation

This section describes IES implementation in TinyOS. We have implemented all three interfaces, *Publisher, Subscriber,* and *Watchdog*. TinyOS provides support for asynchronous communication among the components, which is very useful in implementing the event notification service. However, the static nature of TinyOS makes memory management restrictive, and event notification inefficient.

TinyOS is a component based operating system designed for concurrent operations and resource constrained embedded devices. Components provide interfaces to be used by others. An application is written as a set of components wired together using the interfaces and events. Though TinyOS itself provides only basic send and receive interface support over CSMA based radio control, the other layers (such as routing and fusion) are implemented as independent modules. The modules are statically wired together through their component interfaces to realize the network protocol stack.

Data Management Module. Since TinyOS is designed for resource constrained devices, e.g., Mica2 with 4 KBytes of RAM, it uses static memory optimization techniques to generate memory efficient codes. Because TinyOS does not support dynamic memory allocation, we allocate statically a chunk of memory to be used by IES, and use priority based eviction to control its usage.

Every IES entry is of fixed length, that helps an easy and efficient implementation of DMM even without any dynamic memory support. However, this restriction limits the flexibility of *get* and *put* methods: the attribute value must be of fixed size, which is 4 Bytes in our case. Figure 5 depicts the DMM implementation. An IES entry is of 9

Bytes length, with 2 Bytes for attribute, 2 Bytes for expiration time, and one Byte for maintaining sticky bits. *Sticky* field value is used to influence memory eviction policy.

DMM is implemented as a two-level cache: first, a direct-mapped cache to keep frequently used attributes, and second, a set associative cache to keep more attributes which we call the data bank. The first-level direct-mapped cache maps an attributeID to a unique index in the second-level cache. The data bank stores a list of attribute-value pairs. So, if the attributeID is available in the direct-mapped cache, then the corresponding index value is used to get the attribute value from the data bank. If the attributeID is not present in the first-level cache, then the data bank needs to be searched. By keeping the data bank set associative, the search space is reduced to the associativity factor. As an example, for a memory mapped cache of 8 entries, and a 16-way set associative data bank of total 256 entries (16 sets), each direct-mapped cache entry is of 24 bits (16 bit attributeID and 8 bit data bank array index). For a hit in the direct-mapped cache (we call a cache hit), an attribute is obtained in 2 accesses (one to the direct-mapped cache, and another to the data bank). For a miss in the direct-mapped cache (we call cache miss), an attribute is obtained in at most 17 accesses (one to the direct-mapped cache, and at most 16 to the data bank as there are 16 entries per set). In case of a miss in the data bank, asynchronous signalling is used to notify a producer (see Section 3).

Event Management Module. EMM implementation supports comparison based conditional rules. A module interested in being notified registers itself with the *Watchdog* interface. EMM, in turn, can register itself as a DMM subscriber for the attribute in the specified rule, and it can then periodically check the rule. Currently, periodic checking of rules is not implemented; rather, the checking is done whenever relevant attributes are updated through a *put* command.

Another source of inefficiency comes from TinyOS limitations. A TinyOS application can be thought of as a set of modules, whose dependency graph needs to be specified statically at compile time. Because of this static nature, event subscription also becomes static. Thus all event subscriptions need to be encoded at compile time itself. In our implementation, we facilitate dynamic rule addition by a simple trade-off: we allow an event notification to be triggered when any one of a *set* of rules are satisfied.

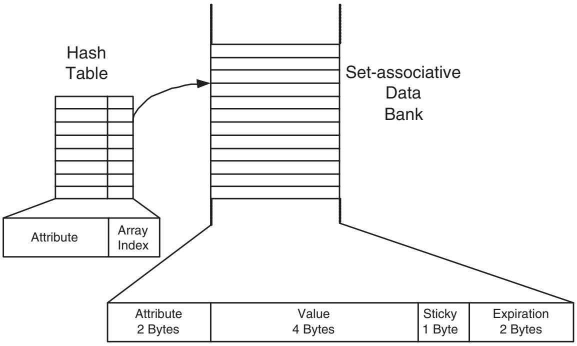


Fig. 5. DMM memory hierarchy. Direct-mapped cache maps an attribute to its location in the data bank.

Thus a rule can be dynamically added to a rule set, but the rule satisfaction is notified to all the modules registered for *any* rule in that set. A rule satisfaction event has a rule identifier field, which can be used by the subscribers to filter the notifications of interest to them.

5 Evaluation

This section evaluates the effectiveness of IES in supporting cross-layering in SensorStack. First, through an extensive set micromeasurements, we investigate the overhead of data access through the IES interface for different scenarios, and compare them with the case where data is accessed directly through protocol modules' interfaces. We also measure the overhead incurred in checking rules in EMM. Finally, we evaluate a complete protocol stack to quantify benefits of using IES, specifically in terms of application longevity and communication savings in data collection from neighboring nodes. Below, because of the lack of space, we present only the micromeasurements.

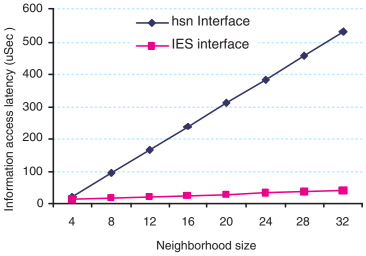


Fig. 6. Comparing information access latency using HSN's neighbors interface and using IES interface. In IES case, the neighborhood data is being accessed directly from the DMM cache.

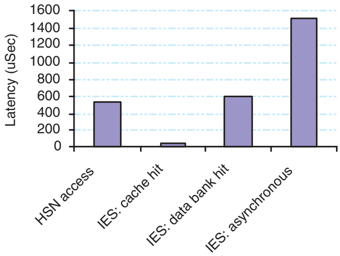


Fig. 7. Memory access overhead comparison for DMM in TinyOS. For IES case, 32-way set associative data bank is used.

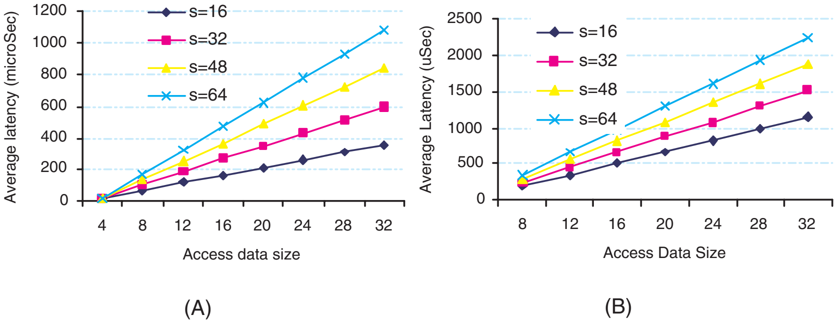


Fig. 8. Memory access overhead for DMM in TinyOS. Figure (A) shows the results when attribute is present in the data bank, and Figure (B) shows the results when the attribute is not in IES.

We use the *SysTime* interface of TinyOS for timing measurements on Mica2 platform. *SysTime* provides timer values at 1.2 micro seconds granularity. The direct-mapped cache size is fixed to 32 entries, and the data bank size is fixed to 256 entries. Set associativity of the data bank is varied from 8 to 64. Each data point is an average over 100 readings.

Figure 6 shows the memory access latency values when all the attributes are present in the direct-mapped cache. It also presents the latency values when the same information is accessed directly by invoking *neighbors* interface using the HSN routing component. As expected for this ideal case, IES memory access is much faster than access using the HSN routing module. IES allows only 4 Byte values for an attribute. As expected, the latency increases linearly with the increase in the number of attributes.

Figure 8(A) shows the case when there is a miss from direct-mapped cache, and Figure 8(B) shows the case when there is miss even from the data bank. As the associativity is increased, the access latency increases linearly because of increase in the number of comparisons DMM has to do to get the attribute. In case of miss from the data bank, the latency results include the cost of signalling DRE, the publisher doing *put*, and finally signalling the DAE. Figure 7 compares the memory access latency of accessing 32 Bytes data for various cases. It confirms the benefit of using the direct-mapped cache. When the data is available in the data bank, the memory access latency for a 32-way set associative data bank is comparable to that of directly accessing data from the HSN interface. For frequently accessed attributes, data access latency using direct-mapped cache is negligible compared to the latency using the HSN interface. If the data is neither in the direct-mapped cache nor in the data bank, the latency incurred is about three times more than the latency using the HSN interface.

6 Related Work

Related work to cross-layering can be broadly divided into two groups: the first considers all the layers together in a holistic way, and the second considers pairs of protocol layers. SensorStack falls in the first group; however, it uses the findings from the specific cross-layering instances between layers.

MobileMan project [4] also has similar goal to SensorStack to support cross-layering in a centralized way by facilitating information sharing. But there are two main differences between IES architecture and MobileMan's architecture [3]. First, instead of providing centralized shared memory, MobileMan provides call-back based approach such that consumers can directly access producer's private data. This approach implies that consumer has to know the publisher, the consumer has to do early binding to the producer, and asynchronous access to data becomes difficult. Second, conditions for asynchronous access are set as black-box functions instead of predicates over shared variables. Using their approach, even when there may not be any change in the shared data, every condition has to be checked periodically, thus leading to inefficiency.

Researches from Berkeley have proposed a sensor network architecture that takes a micro kernel approach. They advocate bringing down the standard interfaces to the applications from the transport layer of the Internet stack to the link layer [5]. The proposed link layer abstraction, SP [15] also aims to share neighborhood information and message pool with all other protocols. While sharing of information is motivated similarly to IES, SP is confined only to link layer information, and they do not provide generic publish/subscribe interface like IES. Also, SP does not allow rule-based event notification as done in IES.

7 Conclusion and Outlook

This paper is a proof of concept that stackability and adaptability can be achieved simultaneously in a network protocol stack. We observe that cross-layering is important to achieve adaptability, but doing so arbitrarily limits the stackability. To solve this problem, we decouple cross-layer data from the functionalities provided by the layers. Based on this idea, we present the design of a novel *Information exchange service (IES)* to facilitate the cross-layering. The publish/subscription based data management module helps achieve stackability by standardizing the cross-layer interaction, and rule-based event management module helps achieve adaptability by supporting reactive notification of changes. We present a simple taxonomy for cross-layer information sharing that provides transparency without affecting stackability. We share our experience in implementing IES on TinyOS and Linux. TinyOS provides support for asynchronous communication among the components, which comes in handy for supporting the event notification service. However, the static nature of TinyOS makes the memory management restrictive, and the event notification inefficient. Linux, on the other hand, does not provide direct support for asynchronous communication among kernel modules, thus needs indirect mechanisms. We have presented results that show that the cross layer information gathering adds little overhead to the basic functionality of the stack. Currently, the IES is limited to information sharing for the modules within a single node. Our future work includes extending this information exchange service across different nodes of the sensor network.

References

1. A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of Infocom*, 2002.
2. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Mobile Computing and Networking*, pages 85–96, 2001.
3. M. Conti, G. Maselli, and G. Turi. Design of a flexible cross-layer interface for ad hoc networks. In *Fourth Annual Mediterranean Ad Hoc Networking Workshop*, June 2005.
4. M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross-layering in mobile ad hoc network design. In *IEEE Computer*, number 2, pages 48–51, Feb 2004.
5. D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, and J. Zhao. Towards a sensor network architecture: Lowering the waistline. In *The Tenth Workshop on Hot Topics in Operating Systems (HotOS X)*, June 2005.
6. C. Frank and K. Romer. Algorithms for generic role assignment in wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 230–242, New York, NY, USA, 2005. ACM Press.

7. T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication. In *Proceedings of ICDCS 2003*.
8. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.
9. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.
10. Intel Research, Berkeley. Heterogeneous Sensor Network: http://www.intel.com/research/exploratory/heterogeneous.htm; Software available in TinyOS 1.1.10 snapshot from Sourceforge.net.
11. B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243–254, 2000.
12. V. Kawadia and P. R. Kumar. A cautionary perspective on cross layer design. In *IEEE Wireless Communication Magazine*, volume 2, pages 3–11, Feb 2005.
13. R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran. Dfuse: a framework for distributed data fusion. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 114–125, New York, NY, USA, 2003. ACM Press.
14. S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *Operating System Design and Implementation(OSDI)*, Boston,MA, Dec 2002.
15. J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica. A unifying link abstraction for wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 76–89, New York, NY, USA, 2005. ACM Press.
16. K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, IETF, Sep. 2001.
17. T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM Press.
18. M. Welsh and G. Mainland. Programming sensor networks using abstract regions. In *First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04)*, March 2004.
19. W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC protocol for Wireless Sensor Networks. In *Proceedings of INFOCOM 2002*, New York, June 2002.
20. H. Yokota, A. Idoue, T. Hasegawa, and T. Kato. Link layer assisted mobile ip fast handoff method over wireless lan networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking (MobiCom '02)*, pages 131–139, New York, NY, USA, 2002. ACM Press.

A Topology Controllable Testing Environment for Mobile Ad Hoc Network Software

Atsushi Kawano, Daisuke Oka, Yasunori Kubo, Shinji Yamashita,
Kaori Maeda, Tomoyuki Ohta, Kenji Ishida, and Yoshiaki Kakuda

Hiroshima City University, Japan
kawano@nets.ce.hiroshima-cu.ac.jp,
{daisuke, yasu, shinji}@pe.ce.hiroshima-cu.ac.jp,
kaori@ipc.hiroshima-cu.ac.jp,
{ohta, ishida, kakuda}@ce.hiroshima-cu.ac.jp

Abstract. A mobile ad hoc network is an autonomous wireless network which consists of mobile nodes without any base stations. When a source node communicates with a destination node outside the transmission range of the source node, communication between the source node and the destination node can be through some other nodes between them. Many schemes such as routings and applications have been proposed for mobile ad hoc networks. However, since these schemes tend to be evaluated only through simulation experiments, it is not known whether they work effectively in real environments or not. Therefore, in order to verify their practical use in mobile ad hoc networks, it is necessary to perform field experiments using actual mobile nodes. If the network size is large, it is difficult to perform field experiments due to problems on limited battery, difficulty of topology control and so on. Realization of rapid topology change of the ad hoc networks topology is especially difficult. In order to solve this problem, this paper proposes a testing environment for mobile ad hoc network software, which emulates the field experiments in wired networks. The existing emulators are scenario-driven. So information of locations and movements of nodes from start of the test to end is given in advance. Unlike the existing emulators, the proposed environment adopts scenario-independent mechanism. The proposed environment can control any network topology for routing in mobile ad hoc networks. The proposed environment consists of a positioning server and multiple testing nodes in wired networks. The positioning server virtually configures mobile ad hoc networks and distributes their information such as the node positions to the testing nodes. And testing nodes themselves deliver their position information to the others. By exchanging messages between the server and testing nodes, any network topology for mobile ad hoc networks can be configured and dynamically changed while testing. It is therefore expected to effectively develop and verify routing protocols and applications for mobile ad hoc networks.

Keywords: Mobile Ad Hoc Networks, Software Testing.

1 Introduction

A mobile ad hoc network [1] is a network consisting of mobile nodes and wireless links without base stations, which facilitates network connectivity without preexisting infrastructure.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 808–819, 2006.

The mobile ad hoc network has the features as follows. When a source node communicates with a destination node outside its transmission range, data packets from the source node are forwarded to the destination node via multi-hop wireless links. In addition, the network topology is in continuous variation because nodes frequently move around the network. Since the ad hoc network can be easily constructed as a temporal network, it is used for various purposes such as disaster recovery and membership management in temporal events.

Recently, many routing algorithms and applications have been proposed for mobile ad hoc networks. However, most of them are only evaluated through simulation experiments and are not verified using actual mobile nodes in real environments, which are dynamic and resource-constrained. However, it is difficult to flexibly realize the ad hoc network in such real environments for testing its software. Realization of rapid topology change of the ad hoc networks topology is especially difficult.

In order to solve the above problem, many network emulators for mobile ad hoc networks have been proposed so far [2] [3] [4] [5] [6]. In these emulators, the node mobility model is not considered as well. Since nodes always move in mobile ad hoc networks, the node mobility model is the most important factor to test the ad hoc network software and affects the test results. Network simulator ns-2 [7], which is commonly used to evaluate routing protocols and applications for mobile ad hoc networks, adopts the random waypoint model [8] as the node mobility model. In the random waypoint model, each node migrates to the specified destination at the specified moving speed within the maximum moving speed. After the node arrives at the destination, it migrates to the next specified destination and these procedures are repeated to some extent. However, application of the random waypoint model is restricted.

The above emulators are $scenario$-driven. The scenario, in which the location and migration information of nodes is predetermined, is obtained from the random waypoint model. However, in fact, as known in the car-to-car mobile ad hoc network applications [9], each node representing a car controls the current moving speed and direction by events such as making speed up or down and turning right or left according to the situation of its neighboring nodes. As a result, the network topology is autonomously changed due to such events even during execution of the emulation.

This paper proposes a topology controllable testing environment for mobile ad hoc network software which adopts $scenario - independent$ mechanism. In this mechanism, the network topology can be changed by events which are not predetermined.

The rest of the paper is as follows. Section 2 explains the related works on emulators for the ad hoc network software. Section 3 describes issues on ad hoc network software development. Sections 4 and 5 propose the topology controllable testing environment and show its implementation. Section 6 describes the case study which utilized the proposed testing environment. Finally, Section 7 concludes this paper with future research.

2 Related Works

This section shows some of the most important related network emulators for mobile ad hoc networks.

MobiNet [2] consists of core nodes and edge nodes. The core nodes are used to emulate topology-specific and hop-by-hop network characteristics. MobiNet can emulate a much larger number of virtual nodes by using multiple “Virtual Edge Nodes”. Thus, it is easy to experiment using a lot of nodes.

NEMAN [3] consists of a “Topology Manager” and multiple application process. NEMAN can emulate the wireless network which consists of hundreds of nodes. Topology Manager manages the topology and the packet delivery.

JEmu [4] consists of an emulation engine and multiple nodes. All packets from a node are delivered to the emulation engine. The emulation engine decides whether the packet are delivered or discarded. The network topology is intensively managed by the emulation engine.

MobiEmu [5] consists of several tested slave nodes and one master node. The master node and the slave nodes have the identical scenario file. The master node delivers a message to the slave nodes to inform the change of the topology information which the master server manages. The slave node that receives the message performs the packet filter based on the topology information. Thus, the network topology is emulated.

MNE [6] consists of several tested slave nodes and one master node similarly to MobiEmu. Each node has two interfaces; which one performs the control channel of emulation and the other is used for the channel of emulated wireless networks. The message about network topology changes is sent by the control channel of the emulation. The slave node that receives this message sets packet filtering rules.

MobiNet, NEMAN and JEmu operate many nodes with one device. It is easy to experiment using a lot of nodes, in which the restriction of the resource is the different from real environments.

MobiEmu and MNE operate one node with one device. The restriction of the resource is the same as real environments. Thus, these emulators can almost realize real environments. However, these emulators are $scenario$-driven. Thus, the scenario of node movement are predefined according to the node mobility model and not changed during execution of the emulation.

Unlike the previous emulators, this paper proposes a topology controllable testing environment for mobile ad hoc network software which adopts $event$-driven mechanism, which enables change of the network topology caused by control of nodes even during execution of emulation.

3 Difference Between Wired Networks and Ad Hoc Networks

In mobile ad hoc networks, a link between each pair of two nodes is determined mainly based on the distance between them. As shown in Figure 1a), node A can directly communicate with node B, while node A cannot directly communicate with nodes C, D, and E. If node A moves as shown in Figure 1b) , A can directly communicate with B, C, and D. As mentioned above, a link between each pair of two nodes is changed according to the physical location of the nodes in mobile ad hoc networks.

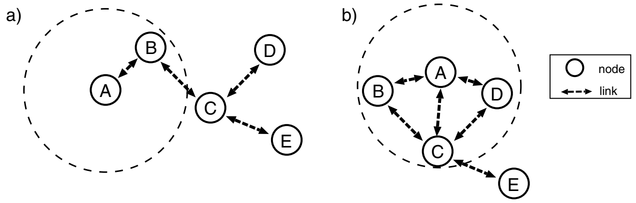


Fig. 1. Ad hoc network topology in real environments

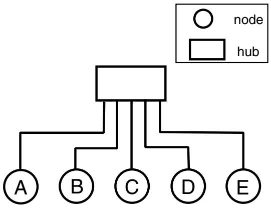


Fig. 2. Emulation environment of a mobile ad hoc network in a wired network

On the other hand, as shown in Figure 2, nodes which are connected to the network hub can directly communicate with each other in a wired network such as Ethernet. A Link between each pair of two nodes is not determined based on the physical location of the nodes. Therefore, using the physical locations of the nodes given in the emulation environment, the topology of an ad hoc network is virtually configured in a wired network.

4 Topology Controllable Testing Environment

This paper proposes a topology controllable testing environment for mobile ad hoc network software. In the proposed testing environment, the topology of a mobile ad hoc network can be virtually configured in a wired network. The proposed testing environment consists of one positioning server and multiple testing nodes as shown in Figure 3. The target application software for testing is assumed to be run on testing nodes. The positioning server manages virtual information of the location and migration (moving speed and destination) of each node and distributes it to testing nodes.

In order to virtually configure the topology of a mobile ad hoc network in the wired network, each testing node determines the neighboring nodes based on the location information which is sent from the positioning server to each testing node. In addition, each testing node can autonomously change its location information by sending the location information from the testing node to the positioning server. The proposed testing environment is thus $event$-driven in the sense that the testing nodes can autonomously alter the location and migration information by its events.

Each testing node can request the positioning server to change the location and migration information. If the positioning server receives the request from the testing node,

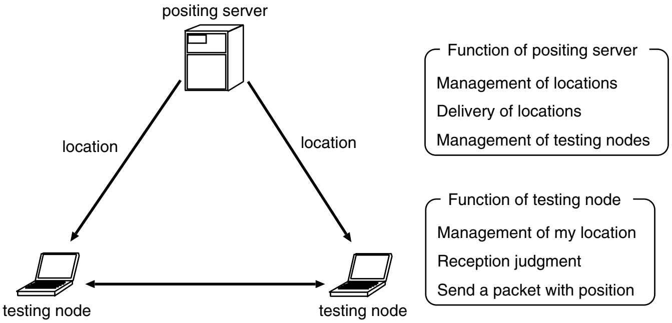


Fig. 3. Each function in topology controllable testing environment

it updates the location and migration information according to the request. This mechanism can therefore realize the emulation of applications such as car-to-car mobile ad hoc networks using wired networks. This is the original and novel issue of the proposed testing environment.

4.1 Positioning Server

The positioning server manages the location and migration information as follows. At first, it virtually configures a field (called a virtual field) and nodes (called virtual nodes) for a mobile ad hoc network. Next, virtual nodes are set in the virtual field by assigning the location information to virtual nodes. Then, moving speed of each node is set randomly as migration information within the given range of the moving speed. After each virtual node migrates to the specified destination, the same procedures are repeated until the emulation stops.

In the proposed testing environment, one testing node is used for one virtual node. As shown in Figure 4, the positioning server periodically distributes the location information to the testing node. As a result, each testing node can get the location information in the virtual field which is uniquely determined by the positioning server.

Control messages which are exchanged between the positioning server and testing nodes are as follows.

- **Join message:** When a node joins the testing environment, it sends this message to the positioning server. The positioning server which received this message creates a virtual node which is assigned to the new testing node in the virtual field.
- **Leave message:** When a testing node leaves from the testing environment, it sends this message to the positioning server. The positioning server which received this message deletes the virtual node which is assigned to the testing node in the virtual field.
- **Configuration message:** The location and migration information for virtual nodes are contained in this message. When the positioning server which received the Join message permits a new node to join, it sends this message back to the new node. The new node which received this message becomes a new testing node in the testing environment.

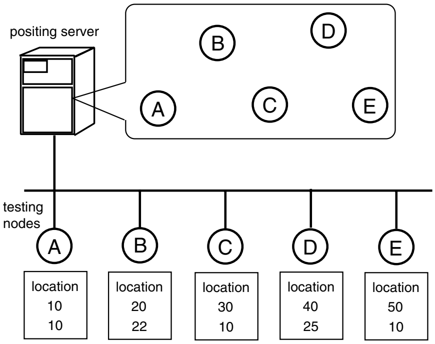


Fig. 4. Example of the communication between the positioning server and testing nodes

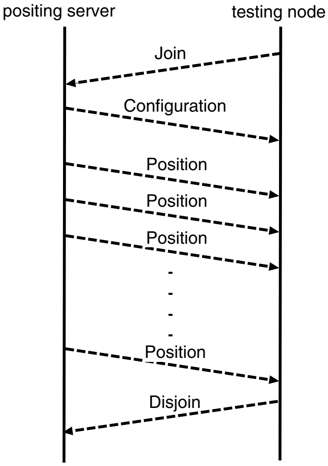


Fig. 5. Message exchange between a positioning server and testing nodes

- **Position message:** The location and migration information for virtual nodes are contained in this message. This message is periodically sent from the positioning server to testing nodes. Each testing node gets its own virtual location information from the positioning server. When a testing node changes the current location and migration information of its own or the other testing node, it sends this message to the positioning server.

Figure 5 shows the message exchange between the positioning server and the testing nodes. As shown in Figure 5, when a node joins the testing environment, it sends the Join message to the positioning server. The positioning server which received the Join message sends the Configuration message back to the node. The node which received the Configuration message becomes the testing node in the testing environment. The positioning server assigns a virtual node to the testing node and periodically sends the

Position message to inform the location information to the testing node. When a testing node leaves from the testing environment, it sends the Leave message to the positioning server.

4.2 Communication Mechanism Between Testing Nodes

Communication mechanism between testing nodes is as follows. In the proposed testing environment, packets for mobile ad hoc network software only are exchanged among testing nodes. Each testing node has the transmission range and the location information by receiving the Configuration and Position messages from the positioning server. A source testing node sends a packet with the current location information to the other testing nodes. Each testing node which received the packet gets the location information of the source testing and compares it with the current location information. If the location of the source node is within the transmission range, each testing node emulates reception of the packet in the mobile ad hoc network. On the other hand, if the location of the source is not within the transmission range, each testing node discards the packet.

5 Implementation

As shown in Figure 6, in the proposed testing environment, the communication between the positioning server and testing nodes to distribute the location and migration information uses TCP, and the communication among testing nodes uses UDP.

5.1 Positioning Server

Figure 7 shows a graphical user interface of the positioning server which is developed by Java. As shown in Figure 7, developers can select the execution mode, set parameters for testing environment and check the current network topology through this interfaces which are indicated by (a), (b), and (c), respectively.

In order to test the ad hoc network software, the following three tests are provided.

1. Test to perform the ad hoc network software in the initial placement and the initial topology both of which developers specified in the virtual field.

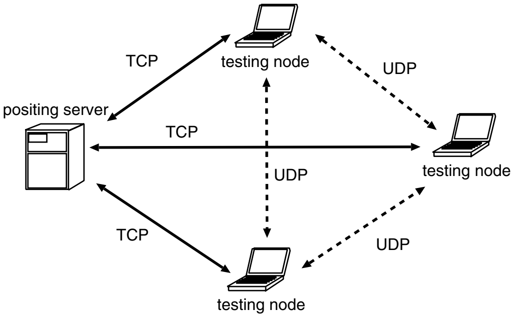


Fig. 6. Protocols used in the topology controllable testing environment

2. Test to perform the ad hoc network software in the initial placement which developers specified and in the dynamic topology gives by testing nodes themselves in the virtual field.
3. Test to perform the ad hoc network software in the initial placement and the dynamic topology both of which are given randomly in the virtual field.

In order to enable these three tests, there are the following three execution modes in the testing environment.

- **Custom mode:** In this mode, developers can test the ad hoc network software in conditions that the initial placement of virtual nodes in the virtual field is given and the network topology is static. Developers create the setting file of the initial placement of virtual nodes described by XML and make a positioning server read it to perform the test.
- **Custom-Random mode:** In this mode, developers can test the ad hoc network software in conditions that the initial placement of virtual nodes in the virtual field is given. A network topology is dynamically changed as time proceeds since virtual nodes are migrated by the positioning server. Developers utilize a setting file for the initial placement of virtual nodes like Custom mode and set the maximum speed of nodes as the node migration information through the GUI (Figure 7(b)).
- **Random mode:** In this mode, developers can test the ad hoc network software in conditions that the initial placement of virtual nodes in the virtual field is randomly placed and the network topology is randomly changed by the positioning server. In addition to the maximum speed of nodes like Custom-Random mode, developers set the number of nodes which can join in the testing environment and the field size of the virtual field (Figure 7(b)).

In each mode, testing nodes can change and control the location and migration information through the interface which is indicated by 7(d).

5.2 Testing Node

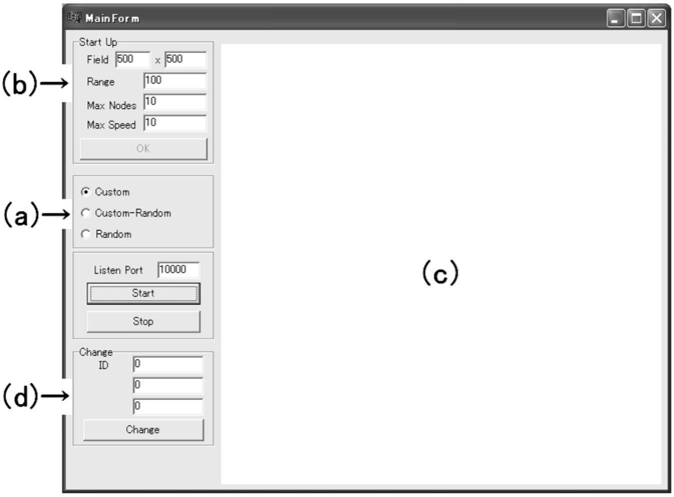
In the proposed testing environment, the function of each testing node is implemented in the AdHocDevice layer of the AdHocEngine which we have developed [10]. AdHocEngine provides multihop wireless communication for applications in mobile ad hoc networks. Owing to AdHocEngine, the source node can communicate with the destination node via some another nodes. Figure 8 shows the architecture including AdHocEngine framework for mobile ad hoc networks, which adopts the cross-layering concept [11].

The AdHocDevice controls the devices such as IEEE802.11 and Bluetooth for communications of nodes in mobile ad hoc networks. Developers can test the mobile ad hoc network software by exchanging packets in the AdHocRouting layer and the application layer.

Each layer of the AdHocEngine is defined as follows.

- **AdHocCtrl:** It manages AdHocTransport and AdHocDevice and provides datagram communication and packet forwarding function. It forwards packets to the next node based on the information from AdHocRouting.

- **AdHocTransport:** It selects a transport protocol for mobile ad hoc networks and provides reliable communication according to the request of the application.
- **AdHocRouting:** It provides routing information to AdHocCtrl and is defined as the abstract class. A developer can implement a routing protocol by inheriting this class.
- **AdHocDevice:** It provides an interface for a wireless device such as IEEE802.11 to AdHocCtrl and implements the function of sending and receiving packets to/from neighboring nodes.



Start Up
Field 500 x 500
Range 100
Max Nodes 10
Max Speed 10

● Custom
○ Custom-Random
○ Random

Listen Port 10000

Change
ID 0
0
0

Fig. 7. GUI of a positioning server

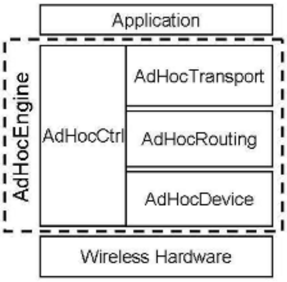


Fig. 8. Structure of AdHocEngine

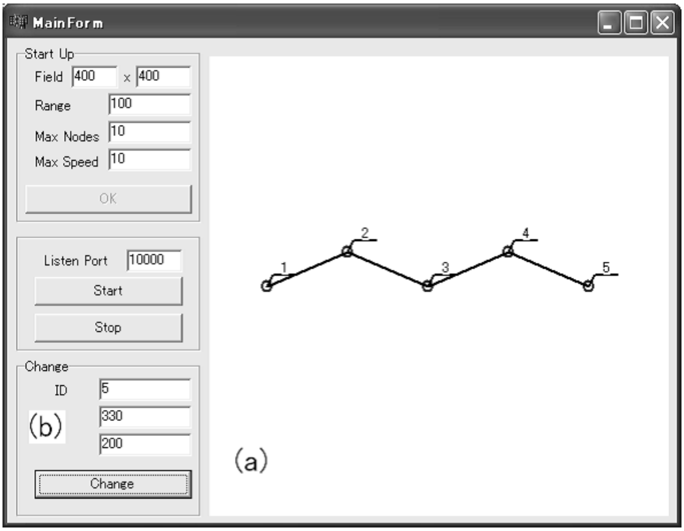
In the AdHocEngine, since AdHocTransport, AdHocRouting and AdHocDevice are implemented as an abstract class, it is easy to exchange algorithms such as routing protocols and transport protocols, which are derived from the abstract classes.

6 Case Study

This section describes the case study to develop the chat application for the ad hoc network environments using the proposed testing environment. In this application, each user can talk with the other users through the ad hoc network which consists of multiple mobile nodes even if the distance between these nodes is very far. In case that the developer tests the application in real environments, the developer deploys multiple mobile computers which have IEEE802.11b or IEEE802.11g in the field, and then tests to be able to communicate between any two nodes or among some nodes. In addition, whenever the developer controls the topology of the ad hoc network to test the application in the various environments, all or some mobile nodes must be moved by hand. Therefore, it costs much manpower and money to test the application in the real environments.

On the other hand, if the developer utilizes the proposed testing environment to test the application for the ad hoc environments, the developer can easily control the topology of an ad hoc network. Since the proposed testing environment provides the topology controllable mechanism, the developer can easily control only the position of a node to create the topology which the developer intends. Therefore, the developer can easily test the application for the ad hoc network environments without moving multiple mobile nodes.

Figures 9 and 10 show the windows of the positioning server and the chat application in the proposed testing environment, respectively. The proposed testing environment consists of one laptop computer for the positioning server and five laptop computers for testing nodes (that is, six laptop computers are totally used). The part which is indicated by Figure 9(a) represents the virtual field which the positioning server manages. Here,

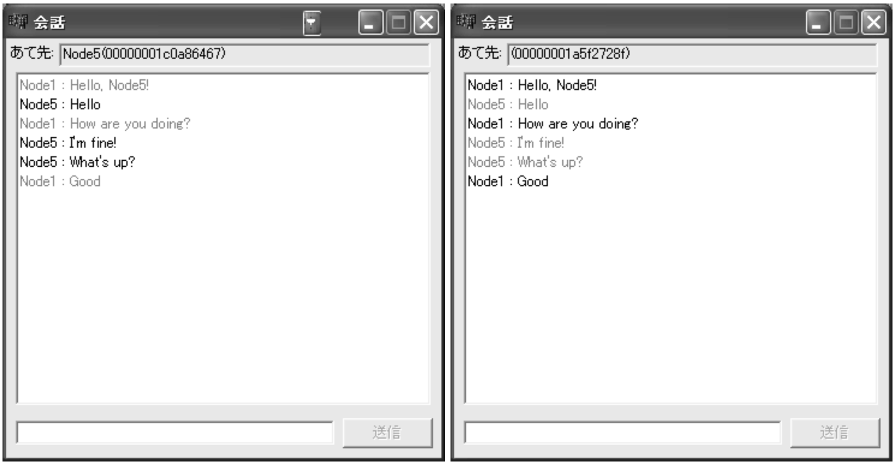


Field 400 x 400
Range 100
Max Nodes 10
Max Speed 10

Listen Port 10000

ID 5
330
200

Fig. 9. Ad hoc network topology working on the positioning server in the proposed testing environment



あて先: Node5(00000001c0a86467)

Node1 : Hello, Node5!
Node5 : Hello
Node1 : How are you doing?
Node5 : I'm fine!
Node5 : What's up?
Node1 : Good

あて先: (00000001a5f2728f)

Node1 : Hello, Node5!
Node5 : Hello
Node1 : How are you doing?
Node5 : I'm fine!
Node5 : What's up?
Node1 : Good

Fig. 10. Chat windows for the application working on node 1 (left window) and node 5 (right window), respectively

node 1 communicates with node 5 in the ad hoc network topology as shown in Figure 9. Figure 10 shows the windows of the chat application working on node 1 and node 5, respectively.

In addition, the developer can change the position of each node in the virtual field through the part which is indicated by Figure 9(b). Therefore, the developer can easily control the ad hoc network topology as shown in Figure 9 and test the application in various ad hoc network topology using the proposed testing environments.

The application developed by the proposed testing environment can work in the real environments by exchanging the AdHocDevice in AdHocEngine which is developed for the proposed testing environment with it which is developed for IEEE802.11.

7 Conclusion

This paper has proposed a topology controllable testing environment for mobile ad hoc network software which adopts event-driven mechanism to reconfigure the ad hoc network topology in wired networks. The proposed testing environment realizes a flexible mobility model, in which each node can autonomously change its location and migration information. Therefore, the proposed testing environment can be applied to wide applications in comparison with existing emulators. Consequently, it is expected that the mobile ad hoc network software can be effectively developed by making use of the proposed testing environment.

For future work, we will refine the proposed testing environment and evaluate its usefulness in detail.

Acknowledgments

This work was supported in part by the Ministry of Internal Affairs and Communications of Japan under Grant-in-Aid for SCOPE-C (No.052308002), the Ministry of

Education, Science, Sports and Culture of Japan under Grant-in-Aid for Scientific Research (C) (No.18500062) and Young Scientists (B) (No.18700070), and Hiroshima City University under their research grants.

References

1. J.Wu and I.Stojmenovic.: Ad hoc networks. IEEE Computer Magagine, 37(2):29–31, 2004.
2. P. Mahadevan, A. Rodriguez, D. Becker, and A. Vahdat.: Mobinet: A scalable emulation infrastructure for ad hoc and wireless networks. Proc. UCSD Technical Report, 2004.
3. M. Puzar and T. Plagemann.: Neman: A network emulator for mobile ad-hoc networks. Proc. 8th International Conference on Telecommunications, 2005.
4. J. Flynn, H. Tewari, and D. O'Mahony.: Jemu: A real-time emulation system for mobile ad-hoc networks. Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference, 2002.
5. Y. Zhang and W. Li.: An integrated environment for testing mobile ad-hoc networks. Proc. Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC2002), pages 104–111, 2002.
6. J. P. Macker, W. Chao, and J. W. Weston.: A low-cost, ip-based mobile network emulator (mne). Proc. IEEE Military Communications Conference (MILCOM2003), pages 481–486, 2003.
7. The network simulator - ns-2.: Project web page available at http://www.isi.edu/nsnam/ns/.
8. J.Broch, D.A.Malts, D.B.Johnson, Y.-C. Hu, and J.Jetcheva.: A performance comparison of multi-hop wireless ad hoc network routing protocols. Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98), pages 85–97, 1998.
9. W. Chen and S. Cai.: Ad hoc peer-to-peer network architecture for vehicle safety communications. IEEE Communications Magazine, 43(4):100–107, 2005.
10. S. Yamashita, R. Furukawa, T. Ohta, H. Kojima, and Y. Kakuda.: Development of testbed framework for ad hoc networks. Proc. International Symposium on Wireless Personal Multimedia Communications (WPMC2005), 1:383–387, 2005.
11. M.Contil, G.Maselli, G.Turi, and S.Girdano.: Cross-layering in mobile ad hoc network design. IEEE Computer Magagine, 37(2):48–51, 2004.

Microcosm: A Low Cost 3-D Wireless Sensor Test-Bed Within a Controllable Environment

David Marsh[1], Richard Tynan[1], Stephen Beirne[2], Roderick Shepherd[3],
Gregory O'Hare[1], Dermot Diamond[3], and Brian Corcoran[2]

[1] School of Computer Science and Informatics, UCD Dublin
Belfield, Dublin 4, Ireland
{david.marsh, richard.tynan, gregory.ohare}@ucd.ie
http://www.cs.ucd.ie/csprism/index.html
[2] School of Mechanical and Manufacturing Engineering,
Dublin City University
Glasnevin, Dublin 9, Ireland
{stephen.beirne3, brian.corcoran}@dcu.ie
[3] National Centre for Sensor Research,
Dublin City University
Glasnevin, Dublin 9, Ireland
{roderick.shepherd, dermot.diamond}@dcu.ie

Abstract. This paper describes the creation of Microcosm, a low cost wireless sensor network test-bed within a controlled environment to facilitate WSN experiments in three dimensions, with an emphasis on executing sensing-related experiments. The design of the sensing hardware, software, support tools and the experimental environment itself are given. Issues related to the design of this configuration are discussed, with the potential pitfalls and eventual solutions alike given. Finally, current and future uses for the test-bed are listed.

1 Introduction

Since the potential applications of wireless sensor networks are so diverse, a similar diversity is reflected in test-beds for WSN experimentation. Despite the potential differences, some lessons from the construction of any particular test-bed should be applicable to many other set ups. As yet, not all possible classes of WSN test-beds have been explored, and hence not all problems have been investigated. This paper describes the design of a novel test-bed designed to accommodate approximately 150 individual sensors in a controlled test chamber, along with the necessary support software.

As with many aspects of wireless sensors networks, the design of a test-bed for reliable, recordable and repeatable experimentation is fraught with both expected and unexpected problems, trade-offs and compromises. Additionally, one of the major drawbacks of implementing a large scale sensor network is the cost. Even relatively old technology, for instance the Mica2 mote [1], still costs over $100 for the processing/radio unit alone, with sensing modules priced higher again. Lower cost alternatives exist, but are often far bulkier due to being based on dual inline packaging components. SmartIts [2] are one such example and are generally easy to customise, and so represent a useful

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 820–834, 2006.

alternate avenue for exploration. This paper describes the issues experienced during the construction a low cost test-bed, one which demonstrates that complex, high resolution sensory data can be collected and used in a feedback-based control mechanism, without requiring heavy financial investment. By detailing those experiences, it is hoped that others who may wish to undertake the fabrication of a WSN test-bed will be able to avoid some of the obstacles that were encountered.

Microcosm is tailored towards experiments in the sensing domain. Having a controlled environment rules out the use of open systems, typified by sensor node deployments throughout buildings or outdoors, and thus a carefully constructed test chamber was employed, one in which the conditions could be altered as required, hence the name Microcosm. Since most WSN sensing work has dealt with planar configurations of sensors, it was decided that a high-density 3-D arrangement of nodes would provide new avenues for experimentation. Networking is not the primary concern, and the design choices reflect this relationship. Methods for increasing sensor density without incurring significant additional costs are given. Additionally, a discussion of requirements for closing the control loop, one of the major focuses of research using WSNs, is given.

The remainder of this paper is organised as follows. The next section lists a set of assumptions and design constraints that influenced the design of Microcosm. Section 3 lists the desired features of the test-bed, and describes some of the obstacles that were encountered, and the design choices used to overcome them. Section 4 provides comparative information on other test-beds. Finally, a brief section about the future work that this system will support is given.

2 Design Constraints and Assumptions

In the construction of any sensor network, certain assumptions must be made about the operation of the network. For instance, the sensing tasks it will be required to perform, the specifications for the communications channels, and the method of access to the WSN all contribute to the system requirements at the design stage. For this test-bed, the following constraints governed the design process.

- *Priority is sensory experimentation:* While networking experiments are planned, the primary use for Microcosm is in the sensing domain. Thus resources should be focused on creating a network that can produce good quality sensor data of many different types. As opposed to many wireless sensor networks, where individual sensor nodes are separated by distances on the order of meters, a much higher spatial sampling rate was desired for this WSN so that high-quality data could be collected from a small volume of space. These conditions are akin to those in industrial settings, e.g. factories. To increase flexibility, changing the type of sensor should be a relatively easy task.
- *Environment features:* The dimensions of the environment in which the nodes operate should be large enough to allow complex experiments, but not so large that access to the nodes is problematic. Real-time control over the environment using sensory data is desirable. It should also be possible to modify it beyond the original specifications to facilitate new kinds of initially unforseen experiments.

- *Communication characteristics:* Losing packets is unavoidable when using real-time constrained transmissions. Additionally, wired connections are usually more reliable than wireless communication links, but they need extra infrastructure.
- *Cost:* The test-bed should be optimised to give the best ratio of performance to price. This should include measures such as increasing the work a single node is capable of, using as few extra components (especially costly ones) as possible, and allowing for reuse of existing resources.

3 Desired Features and Their Realisation

There are four main elements in a wireless sensor network test-bed: the sensing hardware, the control software, the support software, and the environment in which the experiments will be performed. There are many choices for sensor node hardware; there exist a range of devices from small 8-bit, memory-poor units (e.g. motes [1]) to 32-bit, WiFi enabled microservers (e.g. Stargates[3]). These devices are often modular, facilitating the connection of different radios or sensor arrays, depending on the task at hand. Control software, that is the software the sensor nodes run, is usually the primary subject of experiment in WSN test-beds, though there is commonly a permanent administration layer which allows for node control, reprogramming, debugging, etc. Support software, which typically runs on servers that monitor the network, should record the state of the network during experiments, for instance which nodes are active, what messages they are transmitting, what data their sensors are generating, and store it for future analysis. Remote access to the network, job scheduling, and visualisation tools are services that are frequently provided by this element of the test-bed. Finally, the environment in which the experiments takes place is critically important to the test-bed. It can be characterised by being open or closed, static or dynamic, and whether it is controllable by the system or not. The following sections deal with the specifics of each of these aspects of Microcosm (see figure 1).

3.1 Sensing Hardware

There are a number of features that are desirable in a sensor network test-bed, not least of which is adequately dense sensor deployment. Many test-beds use hardware that is confined to having a single sensor per modality per node, i.e. they have a 1-to-1 mapping between sensor types and processing/radio platforms. This ratio can be increased so that one sensor node supports multiple sensors of a single type that are spatially separated. This has the effect of producing higher quality data without increasing costs proportionally. Additionally, flexibility in the placement of the sensors provides the facility to investigate the effects of different deployments on the operation of the WSN.

Given the density of the sensors in this set up, a valid question is why use sensor nodes rather than wiring sensors onto a bus attached directly to a server. The reason why they were employed is because a protocol for e.g. determining necessary sensory coverage needs to be distributed, and preferably local, if it is to work in a WSN. Using a centralised system does not fit with this and would not allow for these kinds of protocols to be tested, so we did not use this method in Microcosm.

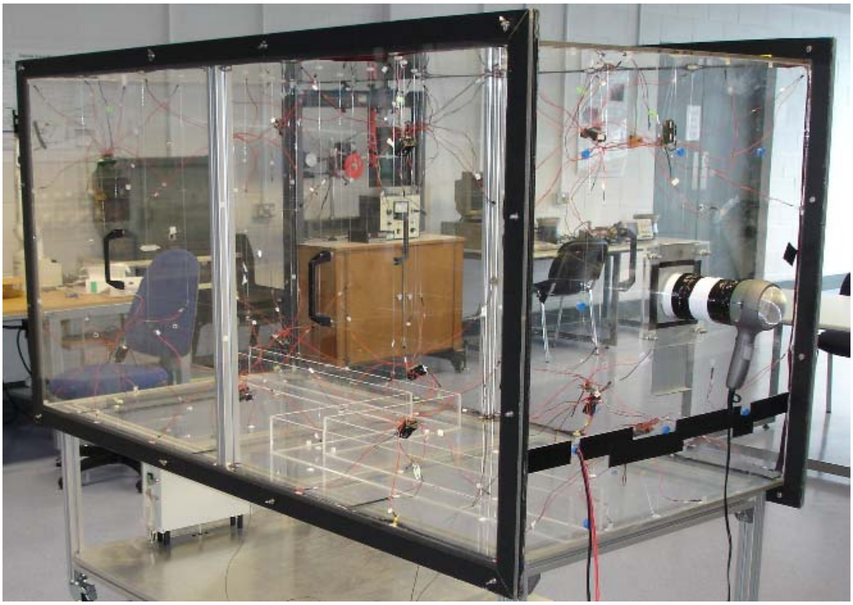


Fig. 1. Microcosm includes a test chamber and a set of sensor nodes deployed within it

Secondly, a test-bed should be able to run experiments that require different kinds of sensors. The ability to easily change the type of sensor that is being used allows for a much greater range of experiments to be carried out. Typically this is achieved by swapping one sensor board for another, however oftentimes much of the hardware is duplicated across these different boards, and so a method to reduce this redundancy would save on costs. What is needed is a plug-and-play sensing capability.

Implementation. Hardware meeting these requirements was devised to work with Mica 2 motes [1] running TinyOS [4]. Briefly, these devices have 7.38 MHz, 8-bit processors, 4 kbytes of RAM, 128 kbytes of program memory, 512 kbytes of flash memory and 19.2 kbit/sec radios. They have 8 ADC channels with 10-bit resolution and an expansion connector to which sensor boards can be attached. While prefabricated sensor boards are available, for the purposes of this set up, they do not satisfy the above constraints.

These boards used a number of features to meet the above requirements. Each one has eight individual sensors, with the sensors divided into four pairs, each pair being wired to a single ADC channel. This was necessary since one of the eight ADC channels is already connected to the radio to measure received signal strength intensity, and so only seven remain available. By ensuring power is only supplied to one sensor in each pair at a time (with an appropriate delay between switching between sensors and sampling the ADC to avoid mixed signals), sensors can share channels without interference. A single board could potentially have far more sensors, however this would require multiplexors, which would increase the complexity and cost of the hardware.

The sensors themselves lie at the end of lengths of wire approximately 30cms long that allow the sensors to be placed at various distances from the node itself. The wires include plug-like terminations (figure 2) to facilitate the replacement of one sensor type with another, for example thermistors (the sensor used in the current incarnation of Microcosm) with light dependent resistors. This fulfills the need for easy alteration of the sensing modality of the test-bed without unnecessary duplication of resources. With eight sensors, it is possible to put each at the corner of a cube around the sensor platform, allowing for a much greater spatial coverage than if the sensors were attached to the sensor board itself. With this arrangement, it is possible for even a single node to gain directional data about a stimulus. Thermistors were used as sensors in this set up, and even with a weak heat source such as a light bulb placed roughly 20cms from the nearest sensor, a 3 K difference was registered between thermistors close to the bulb versus those that were more distantly located. This difference is large enough to be useful given that the combination of these thermistors and the Mica2's 10-bit ADC has a sensitivity of better than 0.1 Kelvin at room temperature.

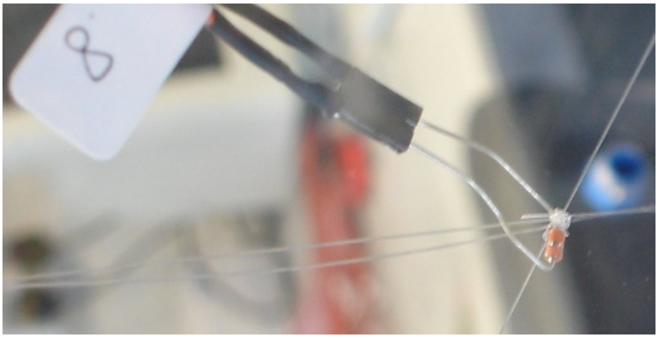


Fig. 2. A thermistor plugged into one of the multi-purpose sensor receptacles on the Microcosm's custom sensor boards. It is held in place by 0.2mm nylon lines.

Problems and Solutions. A number of problems arose during the design of these components.

- *Cost of low run components:* Creating a small number of circuit boards often incurs a high cost. However, when the number is small, it is possible to manually construct the boards. There are a number of methods which could be employed, however the experience gained from the fabrication of the boards for Microcosm indicates that the best method is to use commercially available copper-board etching kits. A single template can be used to produce multiple boards relatively quickly. The only caveat is that, because of the connectors used on the Mica2, the feature size is small enough to require a high level of manual precision during fabrication.

- *Calibration:* The thermistors used in this design had a tolerance of 5%, which translated to roughly ± 1 Kelvin. Calibration was necessary to transform the raw readings into usable data. Since the sensors were deployed in a controlled environment, it was possible to measure the temperature of the interior of the chamber using thermocouples, and using these reference values, adjust the manufacturer-provided resistance-to-temperature equation (1) to correct for the deviation of each sensor.

$$T_{kelvin} = \frac{1}{a + b * ln(R) + c * ln(R)^2 + d * ln(R)^3} \tag{1}$$

  where R = resistance of thermistor, $a = 3.359908 * 10^{-3}$, $b = 2.5788772 * 10^{-4}$, $c = 2.5364809 * 10^{-6}$ and $d = 5.3216393 * 10^{-8}$
- *Unreliable radio communications:* The radio is an unreliable channel. This means data will be lost if sent using it alone. Other test-beds have employed wired channels to improve reliability with positive results (see section 4), and so the same method is being adopted for this WSN. At this point, a small-scale prototype has been built to evaluate the design concepts. The wiring up of the test-bed itself is part of the imminent upgrades for Microcosm (see section 5).
- *Part availability:* Though not encountered during the construction of Microcosm, an obstacle to any future attempt to build a similar system based on hardware that is not very new is the poor availability (for instance, due to RoHS non-compliance) of some components. Any test-bed projects which intend on creating custom parts are strongly advised to choose hardware which will not suffer from this problem.

3.2 Sensor Node Control Software

There are some features of sensor node control software that most WSN test-beds have in common. First is the ability to report back data about their operation. In the context of a test-bed tailored towards sensory experiments, transmitting sensor readings is critical. Having real-time access to this data allows the support software (discussed below) to alter the environmental conditions as part of a closed control loop. Reprogramming the nodes is another useful feature. This can happen either over the radio or via a direct physical connection. Wireless based schemes can often perform quite slowly compared with those set ups that use a wired infrastructure to reprogram the nodes, however they require significantly less hardware to enable, reducing deployment cost and time.

A mechanism for keeping the nodes synchronised also helps with collecting useful sensory data. Even if they start off synchronised, wireless sensor nodes often experience moderate clock skew, and so the nodes can not be relied upon to maintain synchronisation for any extended period. Again, there exist wireless methods of keeping the nodes in step, but it is preferable that the overhead is small so that the wireless channel can be focused on transmitting data of interest. Related to this is the necessity of ensuring maximal efficiency when transmitting data. It is generally accepted that TDMA protocols are effective at increasing efficiency, and that multihop networking protocols reduce overall throughput to the benefit of per node energy consumption.

Implementation. For experiments that are primarily concerned with sensing, the software to control the motes does not need to be particularly complex. Because networking was not the primary concern in the design of Microcosm, currently a simple TDMA MAC layer is used. No multihop network protocol was used because (a) the test chamber is relatively small, (b) overall packet throughput is increased and (c) it would unnecessarily increase the complexity of the software. A few simple commands allow for moderate flexibility in the operation of the network, while employing wireless reprogramming capabilities gives the option of more extensive, though slower, changes to the control software. By identifying the commonly changed parameters of the software, it is possible to construct command packets to change these parameters, thus effecting large changes with a minimum of time and effort. The properties that were found to be most useful to change were:

- The sampling rate of the sensors
- The number of attempts to resend a packet over the radio that should be attempted (for increased reliability)
- The delay between nodes transmitting their packets, and
- The power at which the radio transmits

Since Mica2 motes are subject to clock skew, they cannot be relied upon to remain synchronised for any length of time without a correcting mechanism. The solution used in this test-bed was to sample the sensors when a trigger packet is received from the base station. To reduce overhead, the above mentioned commands were incorporated into this packet. The timing of the transmission of this packet is governed by software running on a desktop computer, and so has far more accuracy than the motes' clocks. Additionally, this allows the rate at which the sensors are sampled to be changed easily.

Once the trigger message is received from the base station, each node samples its sensors. It then waits for a time interval equal to the product of its ID number and the period specified in the trigger packet. This gives each node a unique time slot in which to transmit. By altering the length of this period, the throughput of the network can be configured, either to speed reception of the packets, or to allow more time so that repeat packets can be sent before the next node is due to transmit (as a reliability mechanism). In other words, the TDMA protocol can be optimised for different tasks depending on the particular requirements of the application or experiment. However, the real-time aspect of the system is preserved regardless of where the emphasis is placed.

Because there are multiple sensors connected to one node, it makes sense to collect readings from all sensors and transmit them in a single packet. This cuts down on the number of packets sent through the radio channel, as the overhead associated with the packet headers is now divided over 8 individual readings instead of just one. While a single-sensor node could buffer readings until it had enough to fill a standard 29 byte payload, this would impinge on the real-time aspect of Microcosm. Real-time operation is also the reason why data is not stored locally on nodes to be transmitted later. Transmitting all eight readings in a single packet produces an effective increase in the data extraction rate of the network, and allows the sensors to sample at a higher rate, thus improving the quality of the data collected.

Lastly, the strength at which the radio transmits can be altered. This can be useful when trying to eliminate packet losses. Losses can occur if the signal strength is too low, and the packet fails to register with the receiving mote, or when it is too high, and reflections of the signal interfere with the reception of the packet.

Problems and Solutions. The unreliability of the wireless channel was the principle cause of problems for the sensor node control software.

- *Packet Loss:* One major issue regarding the collection of data from the network is the loss of packets that is unavoidable with wireless links. While mechanisms like acknowledgements, negative acknowledgements, etc. can be used to try to increase the likelihood that a packet will be received, it is impossible to guarantee reception. Because the nodes were expected to send data regularly, there was not much time for attempted retransmissions. It was also found that packets were most often missed because of a physical obstacle, even a person, causing an adverse effect on the signal path. This often meant that no amount of retransmission would work until the physical cause was removed. One solution to this problem is to store all sensor data in the flash memory on the nodes. Any gaps in the data received could be requested once the experiment has finished. Without the time constraints associated with real-time data streams, the node could continuously retransmit until the packet eventually gets through.
- *Wireless Reprogramming:* This can be a slow process. It was found that when using Deluge, the TinyOS network reprogramming tool, even a fairly small program could take many hours to distribute over the network. While this is not a problem if the reprogramming can be set to run overnight, there are times when more immediate action is necessary. In a deployment of a small number of motes, it can be more effective to reprogram each of them directly with a standard programming unit.

3.3 Support Software

Generating copious data is pointless without a sensible means of collecting and reusing it. Different algorithms should be evaluated using the same set of data, which is impossible to guarantee across separate runs of a physical experiment. Logging data generated when the sensor network is exposed to an environmental stimulus, then running a battery of test algorithms on the stored data, is one way of ensuring that there are no discrepancies between runs when performing the comparison. Testing multiple algorithms on live data would potentially require the time consuming process of running multiple instances of the same experiment serially. Instead of this approach, by testing algorithms on recorded data, the requirement of rerunning an experiment is removed, and the different algorithms can potentially now be tested in parallel.

In conjunction with logging sensed data, it is also of paramount importance that the data be monitored as it is logged to ensure its integrity. For example, it would be unwise to run and log a lengthy experiment if a number of the nodes are not capable of transmitting to the base station either due to obstacles, incorrect aerial orientation or other environmental factors. The support software should also provide a mechanism for examining the fidelity of this data while the experiment is running. This can potentially allow an administrator of an experiment to tweak the sensor nodes' control software (discussed earlier) to deliver optimal performance. An example of this control could be increased transmission frequency or alternate routing of packets to avoid obstacles.

Implementation. To deliver this functionality, a number of support tools were implemented for use on one or more of the base stations of Microcosm. When implementing the *logger*, the requirement of data replay requires the time-stamping of event messages received at the base station. Upon receipt of a radio message, the time of arrival of the message and the raw message itself is logged to a file. The packet received is also displayed to the user, this can give a rudimentary indication of faulty nodes or nodes unable to transmit to the base station. When the experiment is completed, the logger is stopped and the file of logged data is submitted through a server script to a web server. This functionality allows sharing of data between multiple team members transparently and without conscious effort on the part of the user. The logs are then available through a web page for download. Another feature of this is the ability to notify one or more team members when a log is uploaded through experiment completion alters, delivered via email. This allows users interested in other team members' experiments to have access to the logs as soon as they become available.

Once the logging process is completed and the logs have been archived, another tool, the *player*, can be used to process the logged files and generate the original packets at the appropriate time intervals. The player parses the file to calculate the time difference between logged packets and can replay the information in one of three modes:

1. Play: it can maintain the absolute time difference between events
2. Accelerated play: it can pre-process the logged file to discover if it is possible to maintain the relative time duration between played events, or
3. Fast forward: it can play the events as fast as possible.

Each of these approaches is arranged in order of increasing speed of execution and reducing temporal fidelity with respect to the original experiment. Some experiments may not be sensitive to the temporal aspect of an experiment and may instead be concerned solely with the contents of the logged packet.

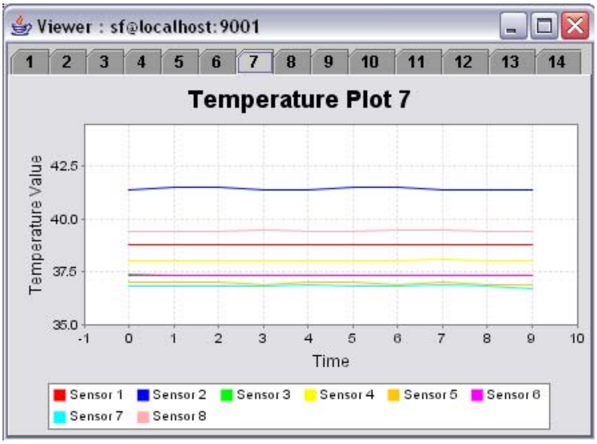


Fig. 3. A screen shot of the viewer tool

The final component of the support software toolset is the *viewer*, use to display the messages in real time as they arrive at the base station. Figure 3 shows an instance where the viewer is displaying the temperature readings of the thermistors appended to the individual nodes. There are 14 nodes in the test-bed, leading to 14 tabs in the viewer which correspond to each node's id. Eight trends are displayed on the graph and these allow the user to visually analyse the data coming from individual nodes. If there is a flaw in a sensor, this can be detected in the trends and can point the user to the node that needs attention.

The support software is a crucial part of a WSN test-bed and it primarily consists of two parts. The first part, such as the logger and player, is relatively general and can be reused for many different applications. On the other hand there are application specific components, such as the viewer, which interpret the data for the user and give an indication of the specific performance of this particular experiment. Most practical test-beds will require both types of support software.

3.4 Experimental Environment

A test chamber within which the operation of various sensors and sensor networks could be analysed has been designed and constructed. A number of critical design requirements were adhered to. In order to allow for video capture equipment and to give multiple viewing angles for demonstration purposes, the unit is constructed from bonded 10mm thick clear Perspex^{TM} sheets and has a dimension of 2x1x1m. The internal volume of the chamber is therefore 2m^3. The applications for sensor networks are not restricted to a purely gaseous environment. To give the test-bed all-round functionality a channel was incorporated to run along the base of the unit allowing for testing in liquid

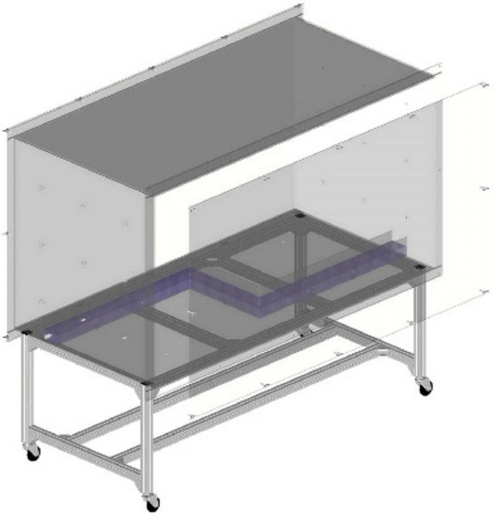


Fig. 4. Rendered view of the test chamber

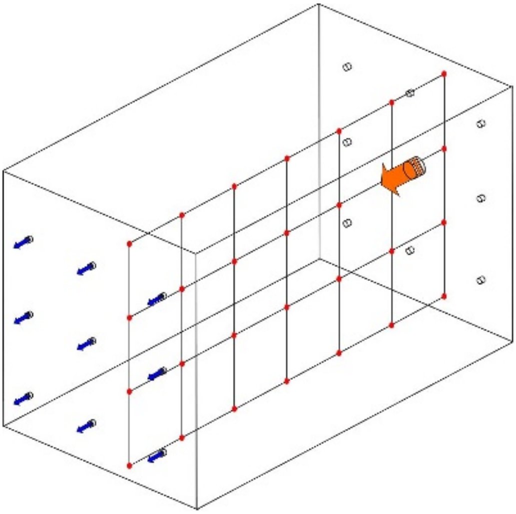


Fig. 5. A cross section through the support lattice for the sensors. Also shown are some of the air vents which allow for gases to be injected into the chamber.

environments. There are multiple inlet/outlet ports to allow seeding of the environment. A rendered representation of the chamber is shown in figure 4.

Implementation. The chamber has been adapted to suit the temperature monitoring requirements of the WSN under discussion. However testing with other sensor platforms and external heat sources was carried out concurrently. An appropriate sensor layout was decided upon to best suit the dimensions of the chamber. The sensors were arranged in a 4 x 4 x 7 matrix (a density of 56 nodes/m^3). One vertical plane of the matrix is shown below in figure 5. There are 14 nodes each consisting of 8 sensors as discussed earlier, resulting in a total of 112 individual sensor points.

Problems and Solutions. The large quantity of sensor devices to be placed within the chamber posed problems, initially due to the accurate positioning of the devices themselves and subsequently due to the reduced accessibility within the chamber with the sensors are in position.

- *Sensor Positioning:* The sensors were located by attaching them to a wire frame that was constructed within the chamber. The frame was made up of a thin gauge (0.2mm) nylon wire (see figure 2) that was tied to anchoring points on the internal faces of the chamber. The anchoring points had been accurately positioned so that the sensors would be evenly spaced with 333mm between each sensor in the X, Y and Z-axes. The anchors used were 10mm x 10mm sticky-back cable tie bases (RS part no. 666-751). Tying together points where two lengths of wire crossed reinforced the lattice. The Mica2 motes positions were also fixed within the chamber using a similar method.

- *Power supply to heat sources:* A control system has been developed which allows for the operation of external heat sources via circuitry operated from the parallel port connection on a laptop PC. The circuit being used to operate a light is shown in figure 6. There are 8 data pins available via the computers parallel port. This circuit layout can thus be replicated up to eight times to control eight individual devices. When the data pin is set high the transistor is activated allowing the 24V DC supply to pass through the coil of relay #1, switching its contacts. 24V is then in turn connected across the coil of the mains relay. The contacts in this relay are closed, connecting the mains power supply to the light. When the data pin is returned to low the transistor is deactivated. No power can therefore flow through both coils, and thus removes the magnetic effect on the switches, thus opening them and breaking the circuit. As this can be run from the base station, data from the sensors can govern the operation of this circuitry, providing a closed control loop.

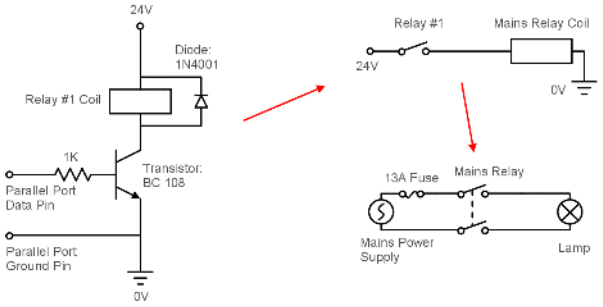


Fig. 6. A diagram illustrating some of the control circuitry used to operate the test-bed remotely

- *Power supply to motes:* For any experiment, it is essential to remove any operational variables which are not under scrutiny. The response of the sensors is dependent on the current state of the battery voltage present in the Mote devices. This problem was overcome by hardwiring a 3V power supply to each of the 14 devices. A parallel supply circuit was employed so that if there were a fault on one of the devices that only that particular device would be removed from the network. This setup also removed the necessity to change discharged batteries that would have proved cumbersome within the constrained space of the chamber. However, batteries may still be employed if it is experimentally necessary.

One of the most important recommendations arising from the experiences in constructing Microcosm is to allow for sufficient flexibility to meet changing requirements during the construction process. An iterative approach was adopted for the design of the separate elements of the test-bed. For example, the mote-based hardware went through several stages, the first incorporating multiple sensors, the second facilitating sensor swap-out, the third adding wired communications ability, and so on. Without modifiable initial designs, much of the early hardware would had to have been replaced in order to allow the later improvements to be made.

4 Related Work

There are several other WSN test-beds described in the literature. Some emphasise the networking aspect, while others have more extensive sensing capabilities. They can be distinguished by a number of features: combined wired and wireless vs. wireless only communications, the presence/absence of feedback-based control, easy reprogrammability, remote access to the test-bed/data, as well as some more unusual abilities such as automated experimentation, capacity for node heterogeneity, mobile elements, hierarchical structuring and 3-D sensor arrangement. All the systems below are situated in open environments and have sensor resolutions far lower than Microcosm.

Motelab [5] is a system offering a set of tools for managing a network of a few dozen motes deployed over 3 floors of the the Electrical Engineering and Computer Science building in Harvard. These tools allow users to create and schedule tasks on the network, record data, reprogram the nodes and access previous results through either a web or a database interface. It is unique in that it includes a power consumption monitor for one of the nodes in the network. Alongside this, it employs a wired infrastructure for node reprogramming and data collection, in order to avoid the unreliability of the wireless channel that has been mentioned previously. One downside to this is that each sensor node requires its own reprogramming board, a significant extra cost. A fairness protocol for time allotment between multiple users has also been implemented.

In contrast to the above static deployment, Mobile Emulab [6] uses a combination of six mobile sensors, in the form of Garcia robots with Mica2 motes attached, and 25 fixed motes in an area of 60m^2 to perform experiments using the Emulab [7] test-bed framework. Emulab provides abstractions to facilitate easy creation and scheduling of experiments, and can log test data and debugging information. Mobile Emulab can track the robots by employing static cameras, mote-based magnetometers and onboard sensors. This multi-sensor modality is unusual among test-beds. The robots can be controlled through the Emulab software, and so represent a method for introducing control of the environment into the experiments in a manner quite different from the method described in this paper. It also contrasts with the majority of set ups, in which the data from the sensors does not influence the environment they are in. Because the robots have motes attached, they can act both as stimulus to the fixed network and data collectors for the system as a whole.

TWIST [8] consists of a tiered network of 90 nodes, with subsets of these nodes connected to supernodes via USB, which in turn are connected to a central server through ethernet. It has the capacity to conduct experiments on heterogenous WSNs that conform to flat, segmented or hierarchical topologies by switching the roles the various components play. The basic sensor node can be any USB-enabled device, such as the Telos mote or the eyesIFX. USB allows communications, power and reprogramming to be integrated into a single connection, reducing costs. These are connected to a Linksys NSLU2, an ethernet connected storage and processing unit, which may function either as diagnostic and management devices or may actively participate in the task of the sensor network, thus introducing a second layer into the WSN. The server acts as a control locus for the entire system. For the cost of additional equipment in the form of the Lynxsys devices, this system gains the benefit of a tiered structure.

SensorScope [9] distinguishes itself from other systems in that it does not use a wired infrastructure. The primary reason for this is to increase the realism of the experiments that are carried out on it. Any test-bed that employs fixed links to retrieve data and management information from the sensor nodes in its network increases the amount of work that the nodes must do, and thus the energy they consume. This is because, on top of the usual messages that the nodes transmit to each other over the radio, they must relay additional messages to the monitoring equipment. SensorScope uses only the wireless channel, with a bare minimum of status information messages, to improve the accuracy of their measurements at the expense of gathering less reliable data. This is similar to the current set up used in Microcosm, and it is an option that will be retained once a wired channel is introduced.

Tutornet [10] is similar to the TWIST, in that it has USB-enabled nodes connected to supernodes, in this instance Stargates [3], but is on a smaller scale. As with TWIST, this is to allow rapid reprogramming while leaving the wireless channel free.

Many other test-beds exist, for instance the sMote, Ω and Trio test-beds at Berkeley [11], and the Re-Mote test-bed at Copenhagen[12], however extensive literature on their design and innovations is not openly available.

5 Future Work

Further automation of Microcosm is the primary improvement planned. Ultimately, it would be useful to be able to reprogram the sensor nodes using a wired connection, however this is not a high priority since radio reprogramming simply takes longer. As was mentioned earlier, there is already a prototype of the wired infrastructure that will be integrated into the existing test-bed. This will bring many benefits. By collecting a complete data set, comparisons with the set produced by the radio can be made, allowing measurements of the impact of packet loss on sensor network performance. Additionally, the complete data gives a better view of what is happening in the environmental chamber, and is more useful when comparing fluid dynamic models of the chamber with real readings, this being another of the uses of the project. Remote access to the network will also be vitally important as the number of users of the test-bed grows. To this end we will make use of the internet to coordinate such access. Fine-grained control of stimuli within the environment, such as location, magnitude and spatial extension, would facilitate assisted or even fully automated experimentation in the future. Naturally, Microcosm will form the basis of many WSN experiments, for instance using interpolation as a coverage calculation mechanism [13] and advanced MAC protocol evaluation [14], however detailed discussion of these experiments is beyond the scope of this paper.

6 Conclusion

This paper has described the trade-offs involved in the construction of a low-cost wireless sensor network test-bed, the focus of which is experiments in the sensing domain and the closing of the sense-process-act loop. A discussion of the required features and the hardware and software designs that implement them was used to illustrate the obstacles that can arise in the construction of a WSN test-bed of this sort. Details of the solutions to various problems that were encountered were related in order to provide future projects looking to construct a test-bed with time- and resource-saving knowledge.

Acknowledgements

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361. We would also like to thank the Irish Research Council for Science, Engineering and Technology (IRCSET) for their support.

References

1. Mica2 motes: (http://www.xbow.com/products/productsdetails.aspx?sid=62)
2. SmartIts Website: (http://www.smart-its.org)
3. Stargate microservers: (http://www.xbow.com/products/productsdetails.aspx?sid=85)
4. Hill, J.: System Architecture for Wireless Sensor Networks. PhD thesis, UC Berkeley (2003)
5. Werner-Allen, G., Swieskowski, P., Welsh, M.: MoteLab: A wireless sensor network testbed. In: In Proceedings 4th Int'l Conf. Information Processing in Sensor Networks (IPSN '05), IEEE (2005) 483–488
6. Johnson, D., Stack, T., Fish, R., Flickinger, D.M., Stoller, L., Ricci, R., Lepreau, J.: Mobile Emulab: A Robotic Wireless and Sensor Network Testbed. In: 25th Conference on Computer Communications (IEEE INFOCOM 2006). (2006)
7. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. In: OSDI02, Boston, MA (2002) 255–270
8. V.Handziski, Köpke, A., Willig, A., A.Wolisz: Twist: A scalable and reconfigurable wireless sensor network testbed for indoor deployments. Technical Report TKN-05-008, Telecommunication Networks Group, Technische Universität Berlin (2005)
9. Schmid, T., Dubois-Ferrire, H., Vetterli, M.: Sensorscope: Experiences with a wireless building monitoring sensor network. In: Workshop on Real-World Wireless Sensor Networks (REALWSN'05). (2005)
10. Tutornet Website: (http://enl.usc.edu/projects/tutornet)
11. Testbeds at the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley: (http://www.millennium.berkeley.edu/sensornets/)
12. Re-Mote Testbed, Department of Computer Science, University of Copenhagen: (http://www.distlab.dk/remote/remote.html)
13. Tynan, R., O'Hare, G.M.P., Marsh, D., O'Kane, D.: Interpolation for wireless sensor network coverage. In: EmNetS-II: The Second IEEE Workshop on Embedded Networked Sensors, IEEE (2005)
14. Ruzzelli, A., Cotan, P., O'Hare, G.M.P., O'Grady, M.J., Tynan, R.: Merlin: A synergistic integration of mac and routing protocol for distributed sensor networks. In: Third Annual IEEE Communication Society Conference on Sensor, Mesh and Ad-Hoc Communications and Networks (SECON 2006). (2006)

System-Level WSN Application Software Test Using Multi-platform Hardware Abstraction Layers

Jochen Koberstein and Norbert Luttenberger

Dept. for Computer Science,
Christian-Albrechts-University in Kiel
{jko, nl}@informatik.uni-kiel.de

Abstract. Software development for Wireless Sensor Networks (WSNs) suffers from the adverse condition that WSN software systems can usually not be tested on a system-level in their final operations environment, as WSN deployment is an expensive and time-consuming process. Several authors therefore propose to interlock application software test tightly with simulation. In this paper, we introduce an XML-based description language that allows the WSN programmer to define a common Hardware Abstraction Layer (HAL) for seamless transfer of WSN application code between WSN node target platforms and simulator-provided platforms. We show how a common network simulator can be enhanced to fully support system-level testing of WSN application code, make some comments on the resulting changes in the software development process, and finally illustrate our approach by an example.

1 Introduction

A Wireless Sensor Network (WSN) consists of possibly up to several hundreds or even thousands of small-foot-print microcomputers being equipped with physical/chemical sensors and sometimes also actuators for interaction with their environment, and a radio interface for interaction with each other. A single such device is called a WSN node. WSNs may be applied, e.g., for monitoring some outdoor area or for overseeing the interior of a building. In many cases, sensor readings are forwarded to a central hub. Other WSN applications avoid a central hub, but instead let the WSN nodes spread out the acquired information to their peers in order to set up a shared information space. Some application areas require mobile WSNs: WSN nodes are either moved passively, e.g. when being part of a vehicle, or they are capable of independent control of their location when being equipped with a drive system of their own.

WSNs are deployed without any kind of network infrastructure. The major challenge for WSN development therefore is to impose a self-organizing behavior on WSN nodes, i.e. WSN nodes must be capable of setting up and maintaining a connected network during the WSN mission time. The quality of self-organization influences all aspects of WSN functionality, above all sensor data

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 835–846, 2006.

acquisition and possibly complex sensor data preprocessing (“sensor fusion”), and the control of the WSN nodes’ motion in the operations area (“motion control”).

The software development for WSNs is an expensive and time consuming process, mostly for two reasons:

- The scarce processor and memory capacities of WSN nodes do not allow to equip WSN nodes with a full-fledged operating system offering many comfortable support functions including a clear and consistent hardware abstraction layer to the programmer.
- The software development process for WSNs is even more constricted by the fact that it is a very complex task to assure on a system-level the fitness of the chosen distributed algorithms for self-organization, sensor fusion and motion control. We call this system-level test. Neither is it a viable approach to test only a single application instance, nor is it feasible to deploy a complete WSN in its intended operations area.

In this contribution we advocate for an extended approach for the definition of a Hardware Abstraction Layer (HAL) for WSN nodes to mitigate these problems. This approach should—above all—be seen as a means to better integrate into the WSN software development process the extra-required system-level test as mentioned under the second bullet above. Thus, our approach does not only provide a system component API to the application programmer, as “conventional” HALs do, but it foresees the combination of two additional features:

1. The system component API—that in our approach is generated from an HAL description—can be “bound” to different platforms, especially to the target system platform and to a network simulator platform. The simulator platform can then be used as “host” for the system-level test. Equipped with a common API, both platforms run exactly the same application code comprising functions for self-organization, sensor fusion and possibly motion control. This multi-platform approach allows the WSN application developer to seamlessly combine component-oriented software test and simulator-based system-level test.
2. To better support the system-level test, it is in most cases very useful to provide “meaningful” input to sensors (i.e., not just random numbers) already at the simulation stage. Comparing what the simulator provides to the WSN “under test” and what the WSN actually achieves to collect, allows the programmer to judge the quality of the installed algorithms for self-organization, sensor fusion and motion control.

The paper is organized as follows: In section 3 we present a programming language independent HAL description language and show section 4 how software development methodology is influenced by using this approach. Section 5 discusses how a network simulator can be enhanced to support WSN system-level testing. In section 6, we illustrate our methodology by an application example. We start with a discussion of related work (following section).

2 Related Work

We discuss related work for two different aspects of our contribution: hardware abstraction and system-level test support by simulators.

Different approaches for the design of an HAL for WSNs have been published. Some of them follow the traditional concept that has borrowed its paradigm from the operating systems world: It is the operating system that exhaustively defines the HAL by providing a ready-to-use API. Lightweight operating systems like e.g. Contiki [11] or MANTIS [12] provide substantial OS support like multithreading, full featured networking stacks, and even graphical user interfaces can be run on WSN nodes—but all this does not leave a reasonable amount of resources to the actual WSN application.

Better suited is a component-based approach as e.g. in TinyOS [1]: Though named an "OS" it avoids the pitfall of monolithic OS designs: During software development, the application programmer selects the TinyOS components that are needed by the application code and links those components with the application code. Furthermore the correct handling of asynchronous code, whose execution is triggered by external events, and synchronous application code is checked at compile time.The resource consumption can be reduced considerably in comparison to fully-fledged operating systems. But unfortunately TinyOS components are coupled to the proprietary programming language NesC.

In [14], the authors present a layered approach for a "flexible hardware abstraction for WSNs". This seems to come close to our approach. But the paper neither discusses multi-platform capabilities, nor cross-layer issues. The paper also lacks a discussion of the underlying software execution model (see below).

In a couple of WSN projects that we discuss below it has been tried out to employ different kinds of discrete event simulators (mostly network simulators) for test support. The programmer tries to get hints on the correctness of the behavior of his/her software from the event traces that the simulator records and possibly visualizes. The border line between simulation and emulation is blurred. For some simulators like e.g. OMNeT++ [6] or ns2 so called "real-time mode extensions" [15] exist, and MobiEmu [4] comes as mobile network emulator right from the beginning. Unfortunately these tools have a serious drawback for the WSN programmer: They lack support for modeling the operations environment of a WSN, i.e. they do not provide data sources and sinks representing the WSN's physical operations environment. Additionally, emulation approaches are non-deterministic and therefore runs are not reproducible, as (host) operating system scheduling suffers from random influences. The SensorSim extension [5] for ns2 allows application components to access operations environment models, but unfortunately it is tied to a specialized task (observation of physical effects with various propagation characteristics), and it is deeply woven into the ns2 simulator.

The EmStar software environment [2] and TOSSIM [3] both claim to be "'real-code' simulation environments", and are centered around the idea to make it "painless to transition among [simulation and emulation] during development and debugging, and eliminating the accidental code differences that can arise when running in simulation requires modifications." [2] But both approaches have severe drawbacks: The EmStar software environment requires a POSIX system call interface that is rarely available on typical WSN nodes, and TOSSIM is completely dedicated to TinyOS, and does not provide a structured approach to WSN operations environment models. The EmStar paper states the necessity for operations environment ("world") models, but does not detail a solution to this problem.

3 The HAL Description Language xHDL

3.1 Design Guidelines

In the following we explain the four guidelines that guided our HAL design approach.

(1) The HAL offered by usual workstation operating systems is built on top of a file abstraction layer, i.e. all system resources are modeled as files. The related API is comprised of read/write/ioctl functions to be called by application programs. In the absence of a multitasking operating system, an API of this kind leads the programmer to a software execution model that is based upon a large while-loop, inside which the application software actively polls sensors and network interfaces. For the Embedded Sensor Boards (ESB, [13]) e.g., this kind of execution model has been proposed and applied. The approach is rather simple and in widespread use because of its simplicity, but it is neither energy-efficient nor appropriate for typical tasks of sensor networks.

For WSN nodes, sensors/actuators must be readable and writable, i.e., read/ write functions of the above mentioned kind apparently have to be provided. But to avoid the active scanning for events, it is useful to let the application software additionally provide callback routines that are capable to react to sensor and communication events. We thus enhance the while-loop execution model by adding capabilities of the run-to-completion model as proposed in [1]: "The run-to-completion model precludes blocking calls: all system services, such as sampling a sensor or sending a packet, are split-phase operations, where a command to start the operation returns immediately and a callback event indicates when the operation completes."

With regard to API design we denote the resulting software execution model as call/callback model, i.e. an API has to provide some operations (call), but the related device vice versa relies on application components to also provide some operations (callback) that are invoked at the occasion of events. In our HAL description, we group all operations related to a single device—be they of call or callback type—together into what we call a contract. This kind of "contract" is closed between the platform on the one side (call type functions) and the application software on the other side (callback type functions).

(2) Most programming environments allow to export interface declarations of implemented components. We have chosen a different approach: In our concept, an API interface is not derived from an implementation, instead it is generated from a programming language independent, XML-based HAL description.

This makes the HAL design as independent from any influences of programming languages, operating systems or simulators as possible and allows for additional “degrees of freedom” that we discuss below. We designed a dedicated HAL description language that is called XML-based HAL Description Language (xHDL). xHDL is described in a formal grammar written in the W3C XML Schema Definition language [8].

From an xHDL description, a generator produces the programming interfaces of those API functions that have been specified in the HAL description. At the time of writing this paper, an xHDL generator for the C++ language is available. The API interfaces clearly separate application-related and platform-related code and thus help to impose a minimum structure upon the WSN code.

(3) The HAL as described so far makes no distinction between platforms, for which the given APIs have to be implemented, i.e., the target system platform and the simulator platform. We call the part of the HAL description that is inpendent of platform the abstract part. In the concrete part of an HAL description, the HAL is bound to a platform. “Binding to a platform” implies to specify additional information that is required for implementing the respective platform. It is in this part of the HAL description that the programmer can specify e.g. an Operations Area Model (OAM), from which sensors read during simulation. Such an OAM could be e.g. a temperature distribution which is common to all WSN nodes and provides time- and location-dependant input values for sensors.

(4) Network interfaces and other devices like e.g. GPS receivers deserve a special discussion. These devices may need some kind of protocol stack for their full functionality, i.e. some kind of functionality beyond simple input and output. When designing an HAL description language, a decision must be taken on how to deal with this additional functionality.

We decided to leave it to the WSN programmer which protocol functionality to include or exclude in the HAL description. If the WSN programmer needs—as outlined above—interfaces at different hierarchical layers he/she needs to specify respective operations in the HAL description.

3.2 xHDL Language Description

As outlined before, an HAL description is divided into an abstract and a concrete part. The abstract part first defines the data types that are exchanged between some application software component and the device. The specification relies on the W3C XML Schema Definition language.

The operations that allow interaction between an application layer component and a device are defined in the following section of the abstract part. Operations are grouped together in contracts. Fig. 1 e.g. shows contracts for a GPS device, a radio network interface, a temperature sensor, and a simple drive system, capable of turtle movements. An operation has a pattern, either call or callback. A call-patterned operation is provided by the device related API, a callback-patterned operation must be provided by the application component that interacts with the device. Call-patterned operations are described by their name and required parameters. Always required is a fault element that describes the data structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xhdl>
<!-- ******* TYPES ******* -->
 <types>
  <schema
    targetNamespace="http://www.swarms.de/wisebee"
    xmlns="http://www.w3.org/2001/XMLSchema">
   <complexType name="tErrorMsg"><sequence>
    <element name="FreeFormMessage" type="string"/>
    <element name="ErrorID" type="integer"/>
   </sequence></complexType>
   <complexType name="tPosition"><sequence>
    <element name="latitude" type="float"/>
    <element name="longitude" type="float"/>
   </sequence></complexType>
   <complexType name="tGpsData"><sequence>
    <element name="position" type="tPosition"/>
    <element name="speed" type="float"/>
    <element name="course" type="float"/>
   </sequence></complexType>
   <complexType name="tGpsConfigData"><sequence>
    <!--information on when a GPS event is triggered-->
    <element name="timeInterval" type="integer"/>
    <element name="distanceInterval" type="float"/>
   </sequence></complexType>
   <complexType name="tTempReading"><sequence>
    <element name="position" type="tPosition"/>
    <element name="value" type="float"/>
   </sequence></complexType>
   <complexType name="tMessage"><sequence>
    <element name="srcID" type="integer"/>
    <element name="msg" type="string"/>
   </sequence></complexType>
   <element name="ErrorMsg" type="tErrorMsg"/>
   <element name="Void" type="void"/>
   <element name="ReturnValue" type="integer"/>
   <element name="GpsData" type="tGpsData"/>
   <element name="RadioMessage" type="tMessage"/>
   <element name="TempReading" type="tTempReading"/>
   <element name="RotValue" type="float"/>
   <element name="SpeedValue" type="float"/>
  </schema>
 </types>
 <!-- ******* CONTRACTS ******* -->
 <contract device="GPS">
  <operation name="PositionUpdate">
   <pattern>callback</pattern>
   <out element="GpsData"/>
  </operation>
  <operation name="GpsConfig">
   <pattern>call</pattern>
   <in element="GpsConfigData"/>
   <out element="ReturnValue"/>
   <fault element="ErrorMsg"/>
  </operation>
 </contract>
 <contract device="Radio">
  <operation name="ReceiveMessage">
   <pattern>callback</pattern>
   <out element="RadioMessage"/>
  </operation>
  <operation name="SendMessage">
   <pattern>call</pattern>
   <in element="RadioMessage"/>
   <out element="ReturnValue"/>
   <fault element="ErrorMsg"/>
  </operation>
 </contract>
 <contract device="TempSensor">
  <operation name="ReadTemp">
   <pattern>call</pattern>
   <in element="Void"/>
   <out element="TempReading"/>
   <fault element="ErrorMsg"/>
  </operation>
 </contract>
 <contract device="Motion">
  <operation name="Rotate">
   <pattern>call</pattern>
   <in element="RotValue"/>
   <out element="ReturnValue"/>
   <fault element="ErrorMsg"/>
  </operation>
  <operation name="SetSpeed">
   <pattern>call</pattern>
   <in element="SpeedValue"/>
   <out element="ReturnValue"/>
   <fault element="ErrorMsg"/>
  </operation>
 </contract>
 <!-- ******* BINDINGS ******* -->
 <binding platform="targetSystem">
  <generator
    name="http://www.swarms.de/xhdl/cpp/sensorNode"/>
 </binding>
 <binding platform="omnetSimulator">
  <generator
    name="http://www.swarms.de/xhdl/cpp/omnet"/>
  <operation name="ReadTemp"
    class="external" scope="global">
   <source>
    <file name="example0.gif"/>
    <startTime>0</startTime>
    <overlay>add</overlay>
    <scaleValueRange min="-20" max="100">
     true
    </scaleValueRange>
   </source><source>
    <file name="exampleMap.gif"/>
    <startTime>5.2</startTime>
    <overlay>multiply</overlay>
   </source></operation></binding>
</xhdl>
```

Fig. 1. xHDL Sample Document

that is returned by the operation in case a fault occurs. The in and out elements describe input parameters and return values, respectively. A callback-patterned operation provides to the called application component a data structure that is described by an out element.

The concrete part of an HAL description consists of a number of bindings. In Fig. 1, we show bindings for the target platform and for the simulator platform. Both bindings name the generator instance that should process the HAL description and generate API interfaces. The simulator binding has additional elements. These elements specify how the simulator feeds different device operations with sensory input. Operations with input of `class="internal"` are fed with data computed by routines running inside the simulator. An example for an internal device is the real time clock. A simulator internal routine computes an inaccurate time as compared to the exact simulation time, thus simulating a clock drift. Operations with input of `class="external"` get their input from external information sources.

Operations with `scope="global"` access a common source of information. During simulation, sensors e.g. read from a common operations area model (OAM) that describes the anticipated operations area of the WSN. Such an OAM could be e.g. a temperature distribution which is the same for all WSN nodes and provides time- and location-dependant input values for sensors. During simulation, temperature sensors e.g. read from a temperature map modeled, e.g., by the color values of one or more images. Operations with `scope="local"` do not share common information. An example for a local device is a battery power meter.

Since it is possible to define multiple external sources for a single operation of `class="external"` it has to be specified how the overlay of these sources is achieved. Using e.g. `add` with the overlay element adds the actual values to previous ones. Input values can either be scaled or cropped setting the `<scaleValueRange>` element. It is applicable for external devices only. Upper and lower bounds are given by the `<minValue>` and `<maxValue>` elements. Several optional xHDL elements are available like e.g. the `<startTime>` and `<stopTime>` elements for the definition of start and stop times for device input.

4 Software Development Using xHDL

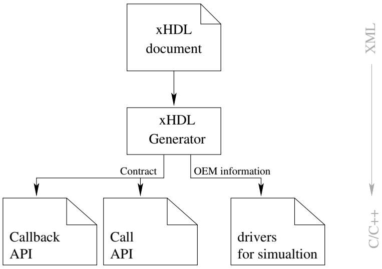


Fig. 2. xHDL Generator instance for simulation

The first step of a xHDL-based WSN software development process is to write an abstract HAL description that describes the HAL for the projected WSN application software. This might be done using a generic XML editor—preferably an editor using the provided xHDL schema for contextual support. Writing an HAL description is also supported by the xHDL generator GUI which is implemented as Eclipse plugin. The second step is to enrich the HAL description by bindings describing simulation-specific operation area model (OAM) information providing e.g. filenames of data sources or overlay information. An xHDL generator produces (cf. Fig. 2):

- the call-patternd API interfaces to be used by application components (furtheron called the “call API”), and
- a callback-patterned API interfaces for events to be sent to the application (furtheron called the “callback API”), and possibly
- an implementation of the call API for the simulator (shown as “drivers for simulation” in fig. 3).

Thus the APIs are not predefined but are adapted to the actual platform configuration. Since the WSN programmer specifies the contract he/she is free to adapt all interfaces to his/her needs. In the case of the radio interface this may e.g. include passing parameters for crosslayer issues.

Using the call API the application is able to configure or poll devices actively and to write data to devices like the radio or other data sinks. Typical examples for such function interfaces are `tTempReading* readTemp()`, `int setSpeed(float value)` or `int sendMessage(tMessage* msg)`. The xHDL generator instance for the simulator binding generates implementation for the call API. The implementation of the call API is complete for operations with `class= "external"` since all necessary information like value ranges, file names etc. are provided by the OAM information contained in the xHDL binding. In contrast, the code provided for internal operations comprises appropriate interfaces only. This enables adding arbitrary functionality to these devices by implementing these interfaces with crafted code.

Application components have to provide implementations for the callback API. Whenever an interrupt or event occurs the corresponding callback method provided by the application is called.

5 Simulation Building Blocks

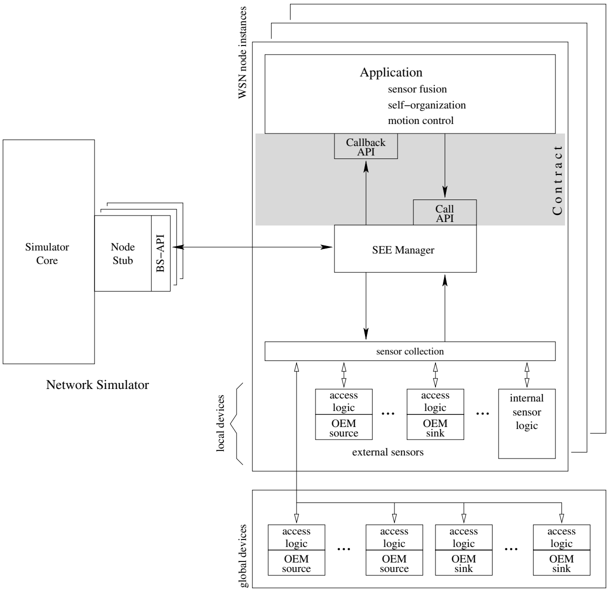


Fig. 3. SEE architecture

To be suited as platform for system-level testing of WSN application code, a simulation system must offer interfaces and data sources and sinks for the WSN nodes' sensors and actuators, radio interfaces, and possibly drive systems. Similar as e.g. [5], we decided not to build a completely new simulation system, but to rely on an existing network simulator and enhance it appropriately. We concentrated on the OMNeT++ network simulator extended by the Mobility Framework [10]. The OMNeT++ extension SEE ("Simulator Extension for Operations Environment Models") that we introduce in this chapter consists of the following components (fig. 3):

1. the Basic Simulator API (BS API),
2. data sources and sinks for devices, and
3. the SEE manager that ties together WSN node application, network simulator, and devices with their sources and sinks (not discussed in depth here).

5.1 Basic Simulator API

The BS API provides a general abstraction layer for the network simulator. It converts API calls from their simulator-specific form into more general calls—independent of the used simulator. The BS API is used by the SEE Manager that acts as a “mediator” between WSN node application software and BS API. The BS API functions can be grouped into functions for:

1. simulation time and WSN node location,
2. drive control,
3. message handling, and
4. visualization and housekeeping.

For scenarios, where WSN nodes are capable to control their motion by themselves, the BS API provides access to a turtle drive controller accepting calls for turtle drive operations, namely `setSpeed` and `rotate`. For the application software to access these operations, they must by bound to the simulator platform with the according xHDL class attribute set to internal. For adapting various high-level motion control operations to the turtle drive model, the WSN programmer must implement according application components that convert between both sets of drive control primitives.

For message handling, the BS API provides message re-formatting functionality. Messages sent by a WSN node are encoded in a generic byte array format, and are transformed by the BS API to the simulator specific format. The same applies vice versa for messages to be received by a WSN node.

5.2 Data Sources and Sinks for Devices

When using SEE, the sensors and actuators of WSN nodes interact with an OAM that provides the required data sources and sinks.

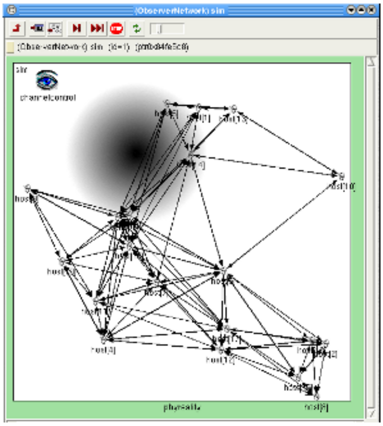


Fig. 4. Simulator visualization of a mobile WSN operating over its operations area

The sensors of WSN nodes are fed during simulation not from random sources, but from coherent OAMs that provide time and location dependant values for sensor readings. Such an OAM data source may, e.g., be a temperature map of a region which is to be explored by the WSN. Fig. 4 shows a sample WSN operating over an area linked to a temperature field, which is represented by a greyscale image where dark tonal values represent higher temperatures than the default temperature, which is depicted in white. By comparing the OAM data with the fusioned data acquired by the WSN, the programmer can assess the fidelity of the applied distributed algorithms for sensor fusion, self-organization and possibly motion control, and their related parameters.

SEE collects the output of actuators in appropriate OAM sinks. This mechanism may also be used for logging purposes. As an example for logging, figs. 5 and 6 show the effects of different motion control algorithms on the number of messages sent in a WSN. Fig. 7 shows the related local view of a single WSN onto its OAM.

6 Application Example

To illustrate the usefulness of our approach for system-level testing of WSN application software, we present a simple example: Suppose autonomous firefighter robots had to monitor the temperature in their operations area in order to control their locations such that each robot can most effectively contribute to the common goal. All robots would surely profit from a near-complete overview over their operations area (or a certain region thereof) and therefore exchange temperature sensor readings with each other.

Our application example uses the in [9] described paradigm of the distributed virtual Shared Information Space (dvSIS) to disseminate information uniformly over the network. Each WSN node (i.e. firefighter robot) creates its own local view of the environment, there is no central hub. Information is disseminated using a flooding protocol with content-based flooding control. By this protocol, each node forwards information, only if it is new in reference to its local view. To overcome temporary network partitions, WSN nodes emit old information in random periodical manner.

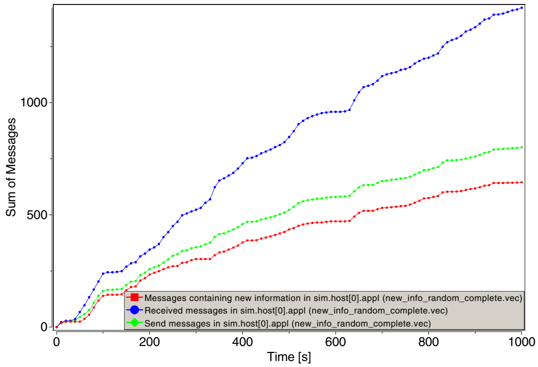


Fig. 5. Random Waypoint

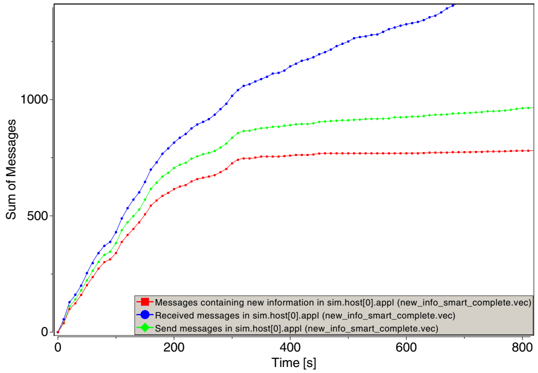


Fig. 6. Smart Random Waypoint

Before deploying the WSN in its real operations environment, the WSN programmer possibly wants to weigh up a simple motion control algorithm with random waypoint behavior against a "smart random waypoint" motion control algorithm. The latter places the next waypoint in the area the WSN node received the last new information from. If the WSN node arrives there, but does not receive any new information, the motion controller component falls back to the random waypoint model.

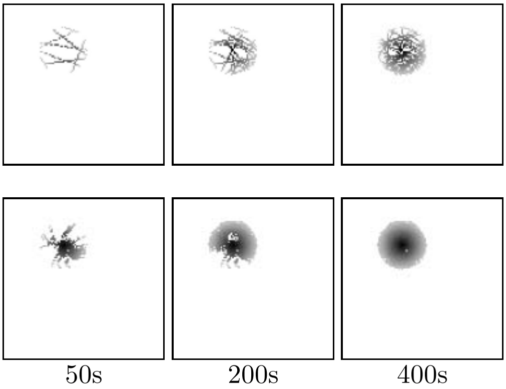


Fig. 7. Local views of node #0 for random waypoint (upper row) and smart random waypoint (lower row)

Common network simulators deliver sophisticated statistical evaluations as presented in fig. 5 and 6. Theses figures show evaluations for the random waypoint motion control algorithm and its smart counterpart. Both graphs analyze message sending and receiving by node #0. The uppermost curve shows the total number of received messages, the middle curve the total number of messages sent, and the lowermost curve the total number of received messages containing new information. The difference between the uppermost and lowermost curve is the number of messages received containing already known information. In the smart controller case node #0 does not receive any new information after about 350s of simulation time. This can be seen from the fact that the gradient of the lowermost curve is almost zero. Obviously at this point in time, most of the available information is already disseminated across the network. Additionally it may be noticed that the number of received messages (uppermost curve) is growing quite fast using the smart algorithm. This is a result of the higher density of the network—leading to better connectivity—since all nodes aim at the area of new information.

Besides these statistical evaluations, SEE also provides a more intuitive assessment to the WSN programmer. To monitor how close the local view of a WSN node comes to a correct and complete view on the operations environment, we let WSN node #0 periodically dump its actual local view to a simulator provided sink. Fig. 7 shows the evolution of local views over time (from left to right) using the same graphical elements as used for the OAM representation. By comparing the local views with the complete OAM, the WSN programmer quickly gains an impression of the networks' performance while running the simulation. The lower row of images in fig. 7 clearly shows that the smart motion controller helps the WSN node to establish a more complete view on its operations environment. From these observations the WSN programmer may conclude that the additional expenses for a more complex motion control algorithm are very effective.

7 Outlook

In the ongoing SWARMS research project[1], we are going to use this simulation system for the comparison of different approaches for sensor fusion, self-organization, and motion control. From these simulations, we want to derive

[1] Funded by the Deutsche Forschungsgemeinschaft (DFG).

suitable metrics that allow us to describe the fidelity that can be obtained from a Wireless Sensor Network.

References

1. Gay, D., Levis, P., and Culler, D. ”Software Design Patterns for TinyOS” In: Proceedings ACM SIGPLAN/SIGBED 2005 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2005), Illinois.
2. Girod, L., Elson, J., Cerpa, A., Stathopoulos, T., Ramanathan, N., and Estrin, D. ”EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks” In: Proceedings of USENIX General Track 2004, Boston.
3. Levis, P., Lee, N., Welsh, M., and Culler, D. ”TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications” In: Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles.
4. Zhang, Y., Li, W. ”An Integrated Environment for Testing Mobile Ad-Hoc Networks” In: Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), Lausanne.
5. Park, S., Savvides, A., Srivastava, M.B. ”SensorSim: A Simulation Framework for Sensor Networks” In: Proceedings of the Third ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000)
6. Varga, A. ”The OMNeT++ Discrete Event Simulation System” In: Proceedings of the European Simulation Multiconference (ESM 2001).
7. Baldwin, P., Kohli, S., Lee, E.A., Liu, X., Zhao, Y. ”VisualSense: Visual Modeling for Wireless and Sensor Network Systems” In: Technical Memorandum UCB/ERL M04/08, University of California, Berkeley, CA 94720, USA (2004).
8. Biron, P.V., Malhotra, A. ”XML Schema Part 2: Datatypes” http://www.w3.org/TR/xmlschema-2/ (2001)
9. Koberstein, J., Reuter, F., Luttenberger, N. ”The XCast Approach for Content-based Flooding Control in Distributed Virtual Shared Information Spaces - Design and Evaluation” In: 1st European Workshop on Wireless Sensor Networks (EWSN 2004)
10. Drytkiewicz, W., Sroka, S., Handziski, V., Koepke, A., Karl., H. ”A Mobility Framework for OMNeT++” In: 3rd International OMNeT++ Workshop (2003).
11. Dunkels, A., Grnvall, B., and Voigt, T. ”Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors” In: Proceedings of the First IEEE Workshop on Embedded Networked Sensors (IEEE EmNetS-I 2004), Tampa
12. Bhatti, S., Carlson, J., Dai, H., Deng, J., Rose, J., Sheth, A., Shucker, B., Gruenwald, C., Torgerson, A., Han, R. ”MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms” In: ACM/Kluwer Mobile Networks & Applications (MONET), Special Issue on Wireless Sensor Networks, vol. 10, no. 4, August 2005
13. Schiller, J., Liers, A., Ritter, H., Winter, R., Voigt, T. ”ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing” Hawaii International Conference On System Sciences (HICSS 2005), Hawaii
14. Handziski, V., Polastre, J., Hauer, J.-H., Sharp, C., Wolisz, A., Culler, D. ”Flexible Hardware Abstraction for Wireless Sensor Networks” In: Proceedings of the 2nd European Workshop on WirelessSensor Networks (EWSN 2005)
15. Mahrenholz, D., Ivanov, S. ”Real-Time Network Emulation with ns-2” In: Proceedings of the 8-th IEEE International Symposium on Distributed Simulation and Real Time Applications, Budapest Hungary, 2004.

An Integrated Self-deployment and Coverage Maintenance Scheme for Mobile Sensor Networks

Xu Li and Nicola Santoro

School of Computer Science, Carleton University
1125 Colonel By Drv., Ottawa, ON Canada, K1S 5B6
{xlii, santoro}@scs.carleton.ca

Abstract. In mobile sensor networks, the *coverage improvement* problem, i.e., maximizing and/or maintaining overall sensing coverage, is a fundamental research issue attracting many researchers. Existing coverage improvement algorithms such as sensor self-deployment algorithms and sensor relocation protocols enhance coverage with limitations due to their specialized design purposes. In this paper, we propose an integrated self-deployment and coverage maintenance scheme, which solves the coverage improvement problem in a complete sense. The proposed scheme is an integration of four algorithms: a node redundancy determination algorithm, a sensor self-deployment algorithm, a sensor relocation protocol, and a sensor replenishment protocol. By this scheme, redundant sensors are placed together with non-redundant ones in the target field at random; non-redundant sensors autonomously scatter to form a network with maximal coverage after initial placement; all the sensors collaborate to compensate coverage loss throughout network lifetime. Mentionably, we notice that no existing scheme besides ours take into account the impact on coverage from nodal sensing range diminishment. At the end, we summarize the paper and discuss our future work.

1 Introduction

Mobile sensor networks (MSNs), as a new paradigm of wireless sensor networks (WSNs), emerged approximately five or six years ago. They inherit all the properties such as the severe resource constraint and the infrastructureless nature from WSNs, and meanwhile, they are featured with their own particularity, i.e., node mobility. This feature allows sensors to act in a more intelligent way and make MSNs more flexible and adaptive to unknown/hazardous environment compared to their static counterparts. An increasing number of research activities are currently being carried out for MSNs. One of the fundamental and attractive issues is *coverage improvement.* In a sensor field, a point is said to be covered iff it falls into at least one sensor's sensing range. The overall sensing coverage of a sensor network is just the aggregation of the areas covered by all the network nodes. A MSN with maximal coverage can timely capture the interesting events happening in the sensor field; a MSN with constant coverage is able to offer

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 847–860, 2006.

sensing service without quality degradation. Hence, the coverage improvement problem aims to find optimal solutions to maximizing and/or maintaining the overall sensing coverage of a sensor network.

There are two main streams of algorithms, i.e., *sensor self-deployment*[1–7] and *sensor relocation*[8–10], for coverage improvement in MSNs. Other streams include, for example, robot-assisted approaches[11, 12]. Since uniform sensor distribution may yield optimal coverage, sensor self-deployment focuses on the way of converting a randomized sensor distribution to a uniform one without human assistance. As for sensor relocation, it concentrates on how to strategically move sensors to maintain existing coverage in the presence of node failure. Due to their specialized design purposes, the two types of approaches supplement each other and may combine to solve the coverage improvement problem on a complete basis. However, to our knowledge, no such an integrative solution has been presented in literature. In this paper, we propose an integrated sensor self-deployment and coverage maintenance scheme to fill the blank.

The proposed scheme is designed to empower MSNs to maximize their overall sensing coverage and operate without coverage degradation in the scenarios (e.g., Mars exploration) where human assistance is infeasible or too costly. It involves the utilization of redundant sensors and requires the original network size and the expected network operating period to be known as a priori. The proposed scheme is composed of four algorithms: a node redundancy calculation algorithm (NRC), a virtual-force-based self-deployment algorithm (VFSD), a zone-based sensor relocation protocol (ZONER)[10], and a sensor replenishment protocol (SRP). The execution of the scheme spans the entire networking process from pre-deployment to post-deployment. First of all, the NRC is run to determine the number of redundant nodes (or, R-nodes for short) to be dropped together with the initial set of network nodes, i.e., non-redundant nodes (or, NR-nodes for short). After node dropping, NR-nodes autonomously spread out by executing the VFSD to form a network covering the target field as much as possible. During the operating period of the network, some R-nodes are activated by the ZONER to replace failed NR-nodes; the other R-nodes are gradually injected into the network by the SRP to compensate the coverage loss due to sensing range diminishment. On a periodical basis, the network is geographically reorganized through the VFSD to eliminate the gaps and overlapping between the sensing ranges of nodes. The novelty of the proposed scheme exists in the following four aspects:

1. the introduction to the effect on coverage from sensing range diminishment;
2. the development of the NRC that determines node redundancy in advance;
3. the design of the VFSD that is adaptive to nodal sensing radius difference;
4. the design of the SRP capable of activating a specified number of R-nodes.

The remainder of this paper is organized as follows: Section 2 reviews some existing work on sensor self-deployment and sensor relocation; Section 3 introduces the two main reasons for coverage loss; Section 4 presents the proposed scheme in detail; Section 5 summarizes the paper and discuss our future work.

2 Related Work

In this section, we will briefly review some existing sensor self-deployment algorithms and sensor relocation protocols.

2.1 Sensor Self-deployment

Howard, Mataric, and Sukhatme[1] proposed an incremental deployment algorithm for mobile sensor networks. Based on previously deployed nodes, this algorithm deploys nodes one-at-a-time and maintains a line of sight relationship between nodes. Howard, Mataric and Sukhatme [2] introduced a potential field based approach to sensor self-deployment problem. In their approach, nodes receive virtual repulsive force from potential fields generated by other nodes and seeable obstacles. Driven by the virtual force, nodes keep moving until a static equilibrium status is reached. Similar algorithms include the VEC[4], the one proposed in [3] and the DSSA/IDCA[5]. Heo and Varshney[5] proposed a deployment algorithm VDDA based on Voronoi diagram. In their approach, the effective area of a node is defined as the intersection of the node's sensing range and its Voronoi polygon, and coverage is improved by increasing each node's effective area with minimal energy consumption. Similar algorithms include the VOR presented in [4]. Wu and Yang[6] proposed a scan-based sensor deployment scheme (SMART). By this algorithm, the target field is partitioned into a 2-D mesh, and the nodes in a cell of the 2-D mesh is treated as load. The goal is to balance the load in each cell of the mesh.

2.2 Sensor Relocation

Wang, Cao and Porta[8] presented a proxy-based sensor relocation protocol for the sensor networks containing both statics and mobiles. By the protocol, mobile nodes always intend to move to large holes from small ones until no larger holes can be detected. To save energy, mobiles perform logical move for transient locations, and they conduct actual movement is conducted only when final location is found. Wang, Cao, Porta and Zhang[9] proposed a grid-quorum based sensor relocation protocol. In this protocol, the network field is geographically partitioned into grids, in each of which, a node is elected as grid head. Each grid head publishes redundant node information inside its grid row (demand quorum). When a grid head finds a sensing hole, it broadcasts a request in its grid column (demand quorum). Because every demand quorum intersects with all the supply quorums, redundant nodes are then discovered. The closest redundant node is then relocated in a cascaded way along a carefully selected path to fill the sensing hole. Li and Santoro[10] proposed a zone-based sensor relocation protocol (ZONER). This protocol shares similar idea with the grid-quorum based protocol[9] in node registration and node discovery, but it outperforms the grid-quorum based protocol in that it requires zero knowledge about the network field and has the immunity to the void-areas caused by obstacles or unbalanced node distribution.

3 Inevitable Coverage Loss

Coverage loss is an inevitable phenomenon in real world scenario. There are two main coverage impairment factors, i.e., *node failure* and *sensing range diminishment*. In this section, we will discuss them in detail.

3.1 Node Failure

A node is said to be a failed node if it is no longer able to deliver sensing service. Failed nodes may possibly generate sensing holes in a network since the coverage provided by these nodes is completely lost. The reasons why a node fails could be multifold: hardware defects, harsh environmental condition, and so on.

Consider a network composed of n number of identical nodes. Suppose that the network start operating at time 0, and that all the nodes are initially operational. Define *node reliability* $R(t)$ as the probability that a node functions correctly throughout interval $(0, t]$. Let $n_{oper}(t)$ (resp., $n_{fail}(t)$) represent the number of functioning (resp., malfunctioning) nodes at time t. By definition, $R(t) = \frac{n_{oper}(t)}{n} = 1 - \frac{n_{fail}(t)}{n}$. Taking differential on both sides, we get $\frac{dR(t)}{dt} = -\frac{1}{n}\frac{dn_{fail}(t)}{dt}$, where $\frac{dn_{fail}(t)}{dt}$ is the instantaneous rate at which nodes fail. Let us define *failure rate function* (or, simply *failure rate*) as

$$Z(t) = \frac{1}{n_{oper}(t)}\frac{dn_{fail}(t)}{dt} = -\frac{n}{n_{oper}(t)}\frac{dR(t)}{dt} = -\frac{1}{R(t)}\frac{dR(t)}{dt} \ .$$

For an electronic component like sensors, experimental data shows that its failure rate function $Z(t)$ obeys a bathtub curve. The bottom part of the bathtub curve is a horizontal line, i.e., $Z(t)$ is equal to a constant value, which corresponds to the useful life of the component. Assume nodal failure rate function $Z(t) = \lambda$ ($\lambda > 0$) for any t during entire network operating period. We have $\frac{dR(t)}{dt} = -\lambda R(t)$ and thus $R(t) = e^{-\lambda t}$, implying that node failure actually follows exponential distribution. Hence, $n_{oper}(t)$ is expected to be

$$n_{oper}(t) = ne^{-\lambda t} \ , \tag{1}$$

and the number of nodes that fails q time units later at $t + q$ is expected to be

$$n_{fail}(t + q) = n_{oper}(t)(1 - e^{-\lambda q}) \ . \tag{2}$$

3.2 Sensing Range Diminishment

There exist two sensor models. One is the most commonly used binary sensor model [1, 2, 4–10]. In this model, a sensor detects with probability 1 (resp., 0) the target events happening inside (resp., outside) its sensing range, a disc centered at itself. The other is so-called stochastic sensor model[3], where the target detection probability however follows a decaying function of the distance between a target and a sensor. In this paper, we use the binary sensor model.

After a sensor is placed in the target field, it starts sensing its surroundings and participating in network operations. As the sensor operates, its battery power decreases, and its hardware wears out, therefore resulting in the performance decline of its sensing module: an originally-detectable target becomes undetectable. We model this sensibility degradation phenomena as nodal sensing range diminishment. For an arbitrary wireless sensor having been operating for q time units, its sensing range can be computed by a monotonically decreasing *sensing range function* $f(E,q)$, where E denotes the sensor's remaining energy level. The sensing range function is heavily affected by the material and the hardware technology that the sensor uses. Under this circumstance, sensing range function is very likely to be different for different types of sensors and should be determined on an empirical basis rather than theoretical analysis. Nodal sensing range diminishment can be easily computed once sensing range function is defined. For instance, after q time unit period of operation from startup, a node's sensing range diminishment is $f\left(E - q\Delta E, q\right) - f\left(E, 0\right)$ where E is initial energy level and ΔE is per-time-unit energy consumption.

4 The Proposed Scheme

In this section, we will present an integrated self-deployment and coverage maintenance scheme. We first state our assumptions, give an overview on the scheme, and then elaborate on scheme detail.

4.1 Assumptions

1. Nodes are homogeneous. They initially have the same amount E of energy, and their communication radii are at least twice their sensing radii.
2. Each node is associated with a unique ID and aware of its global coordinate as well as its remaining energy level.
3. Nodes fail following exponential distribution at failure rate λ.
4. Nodal sensing range decreases over time, while nodal communication range keeps constant.
5. Every node executes an effective routing protocol and a sleeping/wakeup protocol enabling R-nodes (i.e., redundant nodes) to receives messages from NR-nodes (i.e., non-redundant nodes).
6. The number n of NR-nodes and the expected network operating period T are known as a priori.
7. The sensing range function $f(.,.)$, the average per-time-unit energy consumption ΔE of a NR-node and that $\Delta E'$ of a R-node are empirically determined beforehand. And, $\Delta E \geq \Delta E'$.

4.2 Overview

The proposed scheme is a framework constructed on top of four algorithms including a node redundancy calculation algorithm (NRC), a virtual-force-based sensor self-deployment algorithm (VFSD), a zone-based sensor relocation protocol (ZONER)[10], and a sensor replenishment protocol (SRP). Its objective is to enable a mobile sensor network (MSN) to achieve maximal sensing coverage after initial node placement and maintain the achieved coverage in the presence of coverage loss. The execution of the proposed scheme is composed of two stages, a *node redundancy determination stage* and an iterative *self-configuration stage.* The node redundancy determination stage involves human interference and takes place foremost. During this stage, the network administrator run the NRC to estimates the number n' of R-nodes needed for coverage maintenance during the expected network operating period T; afterward, he/she drops n number of NR-nodes together with n' number of R-nodes in the target field at random.

What follows is the iterative self-configuration stage. Throughout this stage, each NR-node maintains a neighboring map by listening to a periodical HELLO message carrying sender's coordinate and sensing range from its every neighboring NR-node; R-nodes stay "sleeping" most of time by executing a sleeping/wakeup protocol. All the iterations of this stage have equal length and together constitute the whole network operating period. In an arbitrary iteration, three algorithms, the VFSD, the ZONER[10], and the SRP, are executed. The VFSD is run only by NR-nodes at the beginning of the iteration. Through the VFSD, NR-nodes moves around to close the gap and open the overlapping between their sensing ranges, therefore maximizing the network overall coverage. After the VFSD terminates, both the ZONER and the SRP starts. By the ZONER, failed NR-nodes are timely replaced with R-nodes in a one-to-one fashion; by the SRP, boundary nodes collect R-node information and activate R-nodes to make up the coverage loss caused by sensing range diminishment.

4.3 Scheme Detail

The four algorithms, the NRC, the VFSD, the ZONER[10] and the SRP, constitute the core of the proposed scheme. We shall go through their details below.

Node Redundancy Calculation. This algorithm, denoted by NRC, is designed for estimating coverage loss and determining node redundancy in advance of actual node dropping. It is composed of a group of formulas derived completely from probability and approximation. Under the assumptions stated in Sect. 4.1, the NRC outputs an expectation instead of an exact predication on the number of R-nodes needed for coverage maintenance. Before going into the detail of the algorithm, we need to understand the following important definitions:

- *Target coverage* ($\mathcal{C}$) is the coverage that a mobile sensor network (MSN) achieves by the VFSD during the very first iteration of the self-configuration stage of the scheme.
- *Potential coverage* ($\mathcal{P}$) is the maximal coverage that a MSN could possibly obtain through geographical reorganization.
- *Coverage gain* ($\mathcal{G}$) is the difference between the target coverage and the potential coverage of a MSN.

The NRC splits the operating period T of the network evenly into k consecutive time slots, each of which contains q time units and matches an iteration of the self-configuration stage of the scheme. Consider the j-th time slot (i.e., the j-th iteration) TS_j, in which a set S_j^j of R-nodes are injected into the network by the ZONER[10] and the SRP. To simplify analysis, we assume that no node fail in its injection time slot. Let S_j^i represent the subset of nodes in S_j^j that are still functioning at the end of TS_i $(i \geq j)$. Define $s_j^i = |S_j^i|$. By (1), we have $s_j^i = s_j^j e^{-(i-j)\lambda q}$ for $(i \geq j)$. To be consistent with above notations, let S_0^0 represent the initial set of NR-nodes. By assumption, $s_0^0 = n$. Taking into account R-node failure and according to (1), the total number n' of R-nodes needed for maintaining the target coverage $\mathcal{C}$ during T should satisfy the inequality $(\cdots((n'e^{-\lambda q} - s_1^1)e^{-\lambda q} - s_2^2)\cdots)e^{-\lambda q} - s_k^k \geq 0$. Solving this inequality, we get

$$n' \geq \sum_{j=1}^{k} s_j^j e^{jq\lambda} \quad . \tag{3}$$

Therefore, in order to compute n', we need to determine s_i^i for $(1 \leq i \leq k)$.

Because all the failed NR-nodes are replaced with R-nodes in a one-to-one fashion, the size s_i^i of the set S_i^i of R-nodes added in the network during TS_i will be at least the number n_{fail}^i of failed NR-nodes during TS_i. Namely,

$$s_i^i = n_{fail}^i + X^i \quad , \tag{4}$$

where X^i is a non-negative number whose value, as explained later, depends solely on if network potential coverage after node replacing is smaller than target coverage $\mathcal{C}$. Recall that S_j^{i-1} $(j < i)$ is the set of nodes activated in TS_j and still functioning at the end of TS_{i-1}. The set of nodes constituting the network at the beginning of TS_i is the union of all the S_j^{i-1}'s. According to (2),

$$n_{fail}^i = \sum_{j=0}^{i-1} \left(s_j^{i-1}(1 - e^{-\lambda q})\right) \quad . \tag{5}$$

Let $\mathcal{G}^{i-1}$ represent the coverage gain in time slot TS_{i-1}. For TS_i, denote by L^i the total coverage loss, i.e., the aggregation of the coverage loss caused by node failure and the coverage loss due to sensing range diminishment; by C_f^i the compensating coverage from the replacements of failure NR-nodes; and by $\hat{A}^i$ the average sensing range of a R-node[1]. Then, the X_i in (4) is given by

$$X^i = \begin{cases} 0 \ , & \text{if } L^i \leq (\mathcal{G}^{i-1} + C_f^i) \ ; \\ \left\lceil \frac{(L^i - \mathcal{G}^{i-1} - C_f^i)}{\hat{A}^i} \right\rceil \ , & \text{otherwise.} \end{cases} \tag{6}$$

[1] For simplicity, we consider the average sensing range of a node during a time slot equal to the sensing range of the node at the beginning of the time slot, which can be readily computed by sensing range function.

For time slot TS_i, denote by A^i_j the average sensing range of a NR-node[1] in S^i_j $(j \leq i)$ and by $\varDelta A^i_j = A^i_j - A^{i+1}_j$ its average sensing range diminishment. The coverage loss due to node failure and that due to nodal sensing range diminishment are respectively $\sum_{j=0}^{i-1}\left(A^i_j s^{i-1}_j(1-e^{-\lambda q})\right)$ and $\sum_{j=0}^{i-1}\left(\varDelta A^i_j\ s^{i-1}_j e^{-\lambda q}\right)$. Then, the total coverage loss L^i will be

$$L^i = \sum_{j=0}^{i-1}\left(s^{i-1}_j(A^i_j\ (1-e^{-\lambda q}) + \varDelta A^i_j\ e^{-\lambda q})\right) \quad . \tag{7}$$

Assume that the VFSD yields a node distribution with no sensing range overlapping. By definition, the target coverage is just the aggregation of the initial sensing ranges of all the NR-nodes in S^0_0, namely, $\mathcal{C} = nf(E,0)$. Hence, the coverage gain $\mathcal{G}^{i-1}$ during time slot TS_{i-1} is

$$\mathcal{G}^{i-1} = \begin{cases} 0 \ , & \text{if } i = 1 \ ; \\ \sum_{j=0}^{i-1}\left(s^{i-1}_j A^{i-1}_j\right) - nf(E,0) \ , & \text{otherwise.} \end{cases} \tag{8}$$

The compensating coverage C^i_f from failure node replacements in TS_i is

$$C^i_f = A'^i \sum_{j=0}^{i-1}\left(s^{i-1}_j(1-e^{-\lambda t})\right) \quad . \tag{9}$$

The NRC estimates each $s^i_i(1 \leq i \leq k)$ in the increasing order of i by (4) – (9), and then finds the minimum n' by (3). Besides, a *redundancy table* as side-product is created and stored at every single node during the execution of the NRC. This table records the mapping between time slot TS_i and its corresponding X_i for every possible i, and it is going to be used by the SRP to determine how many extra R-nodes need to be activated in each time slot.

Virtual-Force-Based Self-deployment Algorithm. All the existing distributed sensor self-deployment algorithms (e.g., [1–6]) assume equal and constant nodal sensing range and thus is not suitable for our scheme where nodal sensing radii decrease over time. We develop a Virtual-Force-based Self-Deployment algorithm, denoted by VFSD, without such an assumption.

The VFSD is executed only by NR-nodes. It makes NR-nodes able to autonomously spread out to form a network, and in order for the network to have as-large-as-possible coverage, it attempts to keep the distance between any two neighboring NR-nodes equal to the summation of their sensing radii. Because the virtual-force-based type of self-deployment algorithms are so sensitive to node failure as to cause frequent topology change and thus large amount of energy loss, in our scheme, the VFSD does not stay active all the time but run only at the beginning of each iteration of the self-configuration stage.

By the VFSD, a NR-node receives virtual force only from its neighboring NR-nodes. Consider an arbitrary pair of neighboring NR-nodes N_i and N_j. Let r_i

and r_j respectively denote the sensing radii of N_i and N_j, and let XY_i and XY_j respectively represent the coordinates of N_i and N_j. Furthermore, define

$$\delta_{i,j} = \mid \vec{d_{i,j}} \mid - r_i - r_j \ ,$$

where $\vec{d_{i,j}}$ stands for the distance from N_i to N_j and is given by $\vec{d_{i,j}} = XY_i - XY_j$. In the case of $\delta_{i,j} < 0$, we model the two NR-nodes as electriferous particles that exert repulsive force on each other, while in the case of $\delta_{i,j} > 0$, we model them as massive matters that exert gravitational force on each other. In either of the two cases, the magnitude of virtual force is computed following Newton's Law of Gravitation. We consider that there is no virtual force between N_i and N_j if $\delta_{i,j} = 0$, because their total coverage is maximized when their sensing ranges adjoin without overlapping.

Since Newton's Law of Gravitation is a function of mass, we treat a node as a massive sphere of its sensing radius. Suppose that the density of a node is ρ. The virtual mass M_i of N_i is $M_i = 4\pi r_i^2 \rho$. If we define the *virtual force constant* K as $K = G(4\pi\rho)^2$ where G is Newton's constant, for any two neighboring NR-nodes N_i and N_j, the force $\vec{F_i^j}$ that N_j exerts on N_i will be

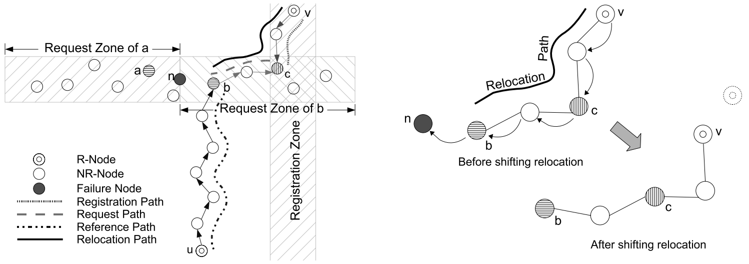
$$\vec{F_i^j} = \begin{cases} K(\frac{r_i r_j}{\delta_{i,j}})^2 \frac{\vec{d_{j,i}}}{\mid \vec{d_{j,i}} \mid} \ , & \text{if } \delta_{i,j} > 0 \ ; \\ \vec{0} \ , & \text{if } \delta_{i,j} = 0 \ ; \\ -K(\frac{r_i r_j}{\delta_{i,j}})^2 \frac{\vec{d_{j,i}}}{\mid \vec{d_{j,i}} \mid} \ , & \text{if } \delta_{i,j} < 0 \ . \end{cases} \tag{10}$$

The total virtual force $\vec{F_i}$ exerted on node N_i is the vector summation of the virtual force that N_i receives from all its neighboring nodes. Let NS_i denote N_i's neighbor set. Then, $\vec{F_i}$ is given by

$$\vec{F_i} = \sum_{N_j \in NS_i} \vec{F_i^j} \ . \tag{11}$$

To compute $\vec{F_i}$ using (10) and (11), node N_i must know both the r_j and the XY_j of every N_j, which are in fact available in its neighborhood map. Driven by $\vec{F_i}$, N_i moves toward the direction of $\vec{F_i}$. The movement of N_i in turn causes the change in $\vec{F_i}$. This mutual effect leads to N_i's unpredictable migration itinerary. Node N_i stops moving when it reaches either a static or a dynamic equilibrium status. The former is the situation that $\vec{F_i} = \vec{0}$; the latter is the situation that N_i fluctuates between several positions, and in this case, N_i stops at the centroid of those positions. Once N_i stops moving, it notifies all its NR-node neighbors. When N_i finds that its neighborhood is stabilized, it becomes fixed and starts the relocation protocol ZONER[10].

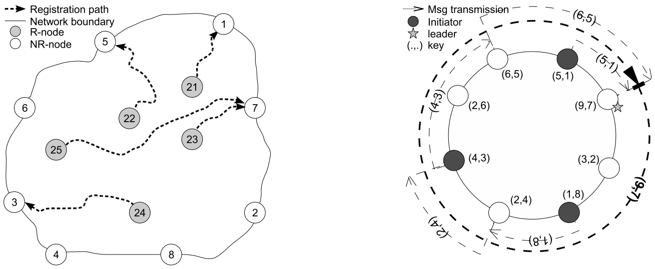
ZONE-Based Sensor Relocation Protocol. The ZONER protocol is our early work proposed in [10] for sensing hole healing. In this integrative scheme, it starts after the termination of the VFSD and stops after the termination of the SRP, during each iteration of the self-configuration stage.



(a) Node discovery (b) Node relocation

Fig. 1. An illustration of how the ZONER works

The execution of the ZONER consists of three core processes, i.e., node registration, node discovery, and node relocation. These processes are performed using a restricted flooding technique, ZFlooding, to save energy and messages. The node registration process is executed first. During this process, a R-node floods its unbounded vertical *registration zone* with a registration message to register with all the NR-nodes inside the zone. After a NR-node failed, its westmost neighbor and eastmost neighbor respectively initiates a node discovery process by flooding their bounded horizontal *request zones* with a request message to find a replacement for it. The westmost neighbor and the eastmost neighbor are called *discovery partner* of each other, and their request zones are adjacent by an imaginary line vertically across the failed node. During a node discovery process, the process initiator first searches its local memory space for the registered R-node with shortest relocation path, and then takes this R-node as reference to inquires all the NR-nodes inside its request zone for a R-node with yet shorter relocation path. For message-saving purpose, the length of the request zone is made subject to the reference node's relocation path length. Because the request zone intersects with a number of registration zones, the NR-nodes in the intersection areas may be able to reply the initiator's request as *recommender*. Finally, the initiator chooses the one with shortest relocation path among all the discovered available R-nodes as the failure node's replacement candidate. Having found the replacement candidate, the initiator communicates with its discovery partner to determine the official replacement node. Figure 1(a) is a big picture about a discovery process. Sequentially, the replacement discoverer triggers a relocation process by a relocation message. In this process, the nodes along the replacement node's relocation path relocate in a shifting manner to replace the failed node. That is, every node in the path simultaneously moves to the location of its path neighbor toward the replacement node discoverer, while the replacement discoverer moves to the location of the failed node as illustrated in Figure 1(b). After such a relocation process, the failed node is in fact replaced by the replacement node discoverer rather than by the replacement node itself. Once a R-node actually involves in a relocation process, it becomes active and automatically transforms to a NR-node.



(a) Registration

(b) Election

Fig. 2. An illustration of how the SRP works

Sensor Replenishment Protocol. The sensing holes caused by failure NR-nodes are filled with R-nodes by the relocation protocol ZONER[10], while the other factor of coverage loss, i.e., nodal sensing range diminishment, still remains untreated. To compensate the coverage loss due to sensing range diminishment, extra R-nodes may have to be released into the network. However, with the absence of centralized controller, where to look for R-nodes and how to release R-nodes become an issue. Under this circumstance, we devise a sensor replenishment protocol, denoted by SRP. The execution of the SRP consists of two phases, i.e., the *node registration phase* and the *node activation phase*, respectively answering the "where" and the "how" question.

Node Registration Phase starts at the beginning of an iteration of the self-configuration stage. In this phase, the SRP, through a Greedy-Face-Greedy (GFG) routing mechanism[13, 14], distributes R-node information onto the outer face perimeter of a Gabriel graph (GG) constructed over the network.

A gabriel graph (GG) is a planar graph, where the closed diametral disc of each edge contains no other vertices than the two edge ends. A GG-construction algorithm, which takes a connected graph G as input and outputs a GG G' spanning G, can be the following: remove non-GG edges from G by testing every edge using the GG definition; an edge e remains in G iff it passes the GG test; finally, G becomes G'. Hence, a GG can be easily built over a connected network in a localized and distributed fashion without message transmission, as long as each network node knows about the position (coordinate) of its every neighboring node. This is just the case in our proposed scheme since each NR-node maintains its neighborhood map. In a GG network, the outer face perimeter is called network boundary. Without losing generality, the network boundary can be modeled as a ring, denoted by R. The network boundary has a special property, that is, *it contains all the global directional optima.* What it is trying to say is that the globally foremost node in certain direction, e.g., the northmost node, must be on the network boundary. This property is referred to as *network boundary property* by us. Its correctness follows the fact that all the nodes but boundary nodes reside in the area surrounded by the network boundary. To avoid ambiguity, directional foremostness must be explicitly defined beforehand, and tie must be broken according to some policy.

When a R-node RN_r wishes to register on the network boundary, it randomly picks a direction as its registration direction RD_r, and sends a *registration message* carrying its ID, coordinate, energy remaining level and its registration direction RD_r to its foremost NR-node neighbor in RD_r. The randomization here is for the purpose of load balancing among boundary nodes. This registration message is routed in a GFG manner[13, 14]. Specifically, after a NR-node N_i receives the registration message of RN_r, it first obtains RD_r from the message and then greedily forwards the message to its own foremost NR-node neighbor in RD_r. In the case that N_i itself is the foremost in RD_r among its neighborhood, it attaches its ID and coordinate to the registration message and retransmits the message in face routing mode, and thereafter, the message keeps being processed in face routing mode until it reaches a yet-foremost NR-node N_j, which will resume the greedy message transmission. When the globally foremost NR-node N_k, which is a boundary node according to the *network boundary property*, in direction RD_r receives the registration message, there are two cases to be explored. One is that N_k knows about the fact that it itself is a boundary node, while the other is that it does not. In the former case, N_k just records the information about RN_r retrieved from the message. In the latter case, N_k tries to find a node yet foremost in RD_r by retransmitting the message along R in face routing mode. Since N_k is actually the global directional optimum, the message will traverse all the way R and get back to N_k at the end. After N_k receives the message back, it becomes aware of its role of boundary node, and then stores RN_r's information as well as notifies all the other boundary nodes of their role through message relay along R. Figure 2(a) shows an example of the node registration phase.

Note that, if the VFSD algorithm (refer to Sect. 4.3) does not yet globally terminate, the constructed GG will not be stable, resulting in the failure of the node registration process introduced above. Hence, the SRP requires that NR-nodes ignore any registration message before they become fixed, and that boundary nodes reply R-nodes' registration request to confirm their successful registration. Under this circumstance, if a R-node does not receive any response after sending a registration message, it "sleeps" for a while and then tries to register once again. When many registration retrials happens, the time interval between two successive ones has incremental length. Once a R-node finds that it succeeds in registration, it turns off to save energy.

Node Activation Phase starts at the end of each iteration of the self-configuration stage. In this phase, the SRP elects a boundary node as leader, which then activates a specified number k of R-nodes. The number k is determined by the leader using the index of current iteration and its locally stored *redundancy table* (see Sect. 4.3). Considering the possible insufficiency in the R-nodes that a single boundary node (i.e., the leader) can activate, the node activation phase is executed recursively until k number of R-nodes are injected into the network.

Denote by N_i an arbitrary boundary node, by id_i the ID of N_i, and by v_i the number of R-nodes currently registering with N_i. Furthermore, define the key

K_i of N_i as the value pair (v_i, id_i). For two keys K_i and K_j, we define $K_i < K_j$ for $(v_i < v_j) \vee (v_i = v_j \wedge id_i < id_j)$. When t time units elapse since the start of current iteration, N_i spontaneously initiates the node activation phase. Taking into account the inaccuracy of local lock, N_i first polls among a collection of NR-nodes. This collection of NR-nodes can be randomly selected or predefined (e.g., one-hop neighbors). An extreme case is that it includes all the NR-nodes. N_i initiates the node activation phase iff majority of polled NR-nodes agree.

Node N_i starts the node activation phase by sending a *start message* carrying its key K_i and the k along R. After a NR-node N_j receives a start message, it compares its own key K_j with the key K embedded in the message. If $K_j < K$, N_j simply forwards the message to its next hop; otherwise, N_j updates the message with K_j and retransmits the message along R iff it is not an initiator. The start message with largest key will traverse entire R and get back to its generator, which is then becomes the leader. Figure 2(b) shows an example of the leader election process. We would like to indicate that this lead election method is by no means the optimal one. We use it only because of its simple description. Leader election is a classic and well-studied problem of distributed computing. Reference [15] provides a systematical study on existing leader election algorithms.

The elected leader picks k closest registered R-nodes, sends them an *activation message*, and waits for their replies. If the number of replying R-node is less than k, the leader will try to activate other locally registered R-nodes in the same way. Both replying R-nodes and unreplying R-nodes are removed by the leader from future consideration. The leader's activation attempt stops when the total number of replies is equal to k, or when no more registered R-nodes are available. In the latter case, the leader updates k with $k - v$ where v represents the total number of replies it receives and restarts the leader election process.

The last elected leader in above recursive process notifies all the NR-node and R-node of the termination of current iteration via a flooding process. Thereafter, the SRP terminates, and the self-configuration stage enters its next iteration.

5 Conclusion and Future Work

In this paper, we discussed the two main reasons, *node failure* and *sensing range diminishment*, for coverage loss in sensor networks, and proposed an integrated self-deployment and coverage maintenance scheme for mobile sensor networks (MSNs). The proposed scheme is a combination of four algorithms, i.e., NRC, the VFSD, ZONER[10], and the SRP. It provides a guidance to systematical coverage loss analysis and node redundancy estimation in advance of actual node dropping, and enables a MSN to autonomously achieve maximal coverage and maintain the achieved coverage using redundant sensors for a given period of time. We noticed that our scheme is the first one that considers the impact from nodal sensing range diminishment when analyzing coverage loss.

The proposed scheme is an ongoing project. It currently has the following incompleteness: 1) the ZONER[10] and the SRP functionally overlap each other to some extent in their node registration processes; 2) the shifting relocation strategy of the ZONER may impair network coverage because of the sensing range difference among the nodes along a relocation path; 3) the SRP is vulnerable to boundary node failure. Solving these problems will be part of our future work. We also plan to evaluate the scheme's performance through experiments.

Acknowledgments

The authors would like to thank Dr. Ivan Stojmenovic for his valuable discussions on improving this work, and the anonymous reviewers for their useful comments.

References

1. A. Howard, M. J. Mataric, and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks." *Autonomous Robots*, 13(2):113-126, 2002.
2. A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem." In *Proc. of DARS*, pp. 299-308, 2002.
3. Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces." In *Proc. of IEEE INFOCOM*, vol. 2, pp 1293-1303, 2003.
4. G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment." In *Proc. of IEEE INFOCOM*, vol. 4, pp. 2469-2479, 2004.
5. N. Heo and P. K. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks." *IEEE Tran. on Systems, Man, and CyberNetics - Part A: Systems and Humans*, 35(1):78-92, 2005.
6. J. Wu and S. Yang, "SMART: A Scan-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks." In *Proc. of IEEE INFOCOM*, vol. 4, pp. 2313- 2324, 2005.
7. S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor Networks Deployment Using Flip-based Sensors." In *Proc. of IEEE MASS*, 2005.
8. G. Wang, G. Cao, and T. L. Porta, "Proxy-Based Sensor Deployment for Mobile Sensor Networks." In *Proc. of IEEE MASS*, pp. 493-502, 2004.
9. G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks." In *Proc. of IEEE INFOCOM*, pp. 2302-2312, 2005.
10. X. Li and N. Santoro, "ZONER: A ZONE-based Sensor Relocation Protocol for Mobile Sensor Networks." In *Proc. of IEEE LCN/WLN*, 2006. To appear.
11. L. E. Parker, B. Kannan, X. Fu, and Y. Tang, "Heterogeneous Mobile Sensor Net Deployment Using Robot Herding and Line-of-Sight Formations." In *Proc. of IEEE IROS*, vol. 3, pp. 2488- 2493, 2003.
12. Y. Mei, C. Xian, S. Das, Y. C. Hu, and Y. Lu, "Replacing Failed Sensor Nodes by Mobile Robots." In *Proc. of ICDCS/WWASN*, 2006. To appear.
13. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks." In *Proc. of ACM DIALM*, pp. 48-55, 1999.
14. H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks." In *Proc. of ACM MobiCom*, 2006. To appear.
15. N. Santoro, "Election." *Design and Analysis of Distributed Algorithms (N. Santoro, ed.)*, 2006. In Publication.

A Power-Aware Peer-to-Peer System for Ad-Hoc Networks*

Hyun-Duk Choi[1] and Miae Woo[2]

[1] MMC Technology, Seoul, Korea
[2] Sejong University, Seoul, Korea
choihd@cnet.sejong.ac.kr, mawoo@sejong.ac.kr

Abstract. Recently, many peer-to-peer (P2P) systems have been introduced to implement large scale resource sharing systems. Such P2P systems exhibit interesting features like self configuration, self-healing and complete decentralization, which make the systems appealing for deployment in ad hoc environments as well. This paper proposes a power-aware peer-to-peer system specially designed for the ad-hoc networks based on Gnutella with hierarchical structure. The objectives of this paper is to enhance performance as well as to prolong the lifespan of the participating P2P nodes. To pursue the objectives, the proposed system chooses ultrapeers which can serve the overlay network better, by considering battery power, connectivity with other peers and commitment level of each node. It also introduces proactive approach for distributing ultrapeer information to reduce P2P overheads. According to the simulation results, the proposed system provides higher query success rate, shorter query response time, less overhead and extended lifespan of peers.

1 Introduction

Traditional Internet-based service paradigm based on the client-server environment is starting to shift to ubiquitous computing environment. An ubiquitous environment has two main features: peer-to-peer environment and nomadic environment. One of nomadic environment which is expected to be dominant in the future is the ad-hoc networks. The concept of the ad-hoc network was first developed from DARPA packet radio network in 1970s. Peer-to-peer systems were initiated in the middle of 1990s. They are widely used as resource sharing systems nowadays, generating significant traffics in the Internet backbone [1]. Gnutella [2] is one of the most widely used peer-to-peer systems.

Ad-hoc networks and peer-to-peer networks share several common characteristics [3]. First, each entity in both networks can organize a network by itself. Network topology of both networks is changing dynamically. Also, operations are performed by issuing routing query in a distributed environment. Such common

* This research is supported by the Ubiquitous Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 861–872, 2006.

characteristics raise a basic issue that is how to communicate each other without a specific management entity. Because of the distributed, unstructured nature of both systems, they face a difficult task of delivering messages. Unlike wired Internet, an ad-hoc network is not reliable and has limited resources such as memory, processing power, bandwidth, and battery. Such characteristics of an ad-hoc network reduce the query success rate and connectivity in the peer-to-peer system. It has been shown that the performance of Gnutella is not satisfactory when it is implemented straight-forward in the ad-hoc network under the point of view of the produced overhead and the average overlay connectivity [4].

Gnutella-based P2P systems on top of ad hoc networks have been studied in [5],[6] and [4]. [5] is one of the researches whose goal was more focused on the performance of ad-hoc routing protocols such as Destination-Sequenced Distance-Vector routing (DSDV), Dynamic Source Routing (DSR), and Ad hoc On-Demand Vector Routing (AODV) when Gnutella is operated over an ad-hoc network. [6] identified that Gnutella produced better performance when proactive ad-hoc routing protocol was used and hierarchical structure consisting of ultrapeers and leaves was introduced in Gnutella. [4] applied cross-layer interaction between a P2P platform and the routing agent at the network layer, producing simplified overlay management and improved the quality of the resulting overlay.

In this paper, we propose a power-aware peer-to-peer system for the ad-hoc networks based on Gnutella with the peer-node hierarchy. We set two objectives for our proposal. The first one is to increase the performance of the P2P system by reducing overall overheads incurred by the P2P system and by evaluating goodness of the peer nodes. The other one is to increase the lifespan of the peer nodes by taking battery power into account. Ultrapeers spend more battery power than leaves because they are the major message forwarders in the P2P overlay network. In order to achieve the objectives, new metric is introduced to judge peer's suitability as an ultrapeer. Also, we introduce a proactive approach to provide up-to-date information on the ultrapeers. According to the simulation results, the proposed method gives better performance than Gnutella in terms of query success rate, query response time, overhead and remained battery power.

The remainder of the paper is organized as follows. Section 2 overviews Gnutella protocol. The proposed system is given in Section 3. Section 4 describes the simulation environment used. The performance of the proposed system is analyzed in Section 5. Finally, Section 6 concludes this paper.

2 Overview of Gnutella Protocol

Gnutella is a fully distributed peer-to-peer resource locating protocol. With such characteristic, Gnutella network potentially has very good reliability and fault-tolerance properties, but the search process is complex and costly.

Originally Gnutella network consists of a number of equal nodes, called peers or servents. These peers are connected by an application level overlay network [7] that provides routing and forwarding of Gnutella messages. A newly participating servent can connect to Gnutella network by handshaking with the already connected node whose address is learned somehow out-of-band [8]. Once a servent has connected successfully to the network, it communicates with the other servents by sending and receiving Gnutella protocol messages.

Ping, pong, query, and query hit are the crucial messages for Gnutella operation. Ping is used to discover servents on the network. A peer receiving a ping message sends one or more pong messages. A pong message contains information on a peer. When a peer receives a pong message, it stores the obtained peer information in its pong cache and tries to make connection to the peer. Each entry in the pong cache corresponds to one pong message. The number of pong messages generated in response to a ping message is the number of entries in the pong cache of the responding peer. Query is used as a primary mechanism for searching the distributed network. When a servent receives a query message, it searches its local files for matches to the query and returns a query hit message containing all the matches it finds [9]. The actual download of files is executed via the HTTP protocol and bypasses the Gnutella network. Ping and query messages are broadcasted over the network. Pong and query hit messages are routed back to the originator of the ping and query messages.

Having random connections with the other servents results in routing inefficiency. To address this problem, the ultrapeer system has been introduced by organizing nodes into hierarchical fashion with ultrapeers and leaves. A leaf keeps only a small number of connections with ultrapeers. On the other hand, an ultrapeer maintains many leaf connections as well as a small number of connections to the other ultrapeers [10,11]. It acts as a proxy to the Gnutella network for the leaves connected to it and shields leaves from the majority of message traffic. An ultrapeer forwards a query to a leaf only if it believes the leaf can answer it. Leaves never relay queries to ultrapeers.

In Gnutella, the ping and pong messages between a leaf and an ultrapeer as well as between ultrapeers are used to find out a new peer for maintaining connectivity. Although the message lengths of ping and pong messages are only 23 bytes and 37 bytes respectively, the traffics generated by these messages are fairly huge because ping messages are flooded up to hop count defined in the time to live (TTL) field and pong messages are generated as many as the number of entries in each node's pong cache. If Gnutella is applied to an ad-hoc network, bandwidth occupied by such messages would not be negligible.

Another concern in Gnutella is that a peer may make a connection with another peer which would not provide good service to it. Information contained in a pong message is the address and port number of a peer collected by other peers in the system. So, there is no measure to decide which node is good to connect or which pong message contains better information. Since it is plausible to make inefficient connections, a new measure is necessary to differentiate the goodness of peers.

3 The Proposed System

To provide a solution to the issues in Gnutella discussed in Section 2, we propose the following mechanisms:

- Use a metric value to judge goodness of peers
- Change the role of a peer between an ultrapeer and a leaf based on the metric value
- Advertise ultrapeer information proactively

In the following subsections, we describe each mechanism in detail.

3.1 Metric

All the peers calculate their metric values based on the battery power, the number of connected ultrapeers, and whether it is freeloader or not. The metric value of node i, W_i, is defined as follows:

$$W_i = f \cdot (\beta E_i + (1-\beta)C_i), \quad W_{\min} < \beta \leq 1 \tag{1}$$

W_i calculated using Eq. 1 can take values from zero to one. It gives some measures to differentiate nodes which are more suitable for ultrapeers. A peer with higher metric value is regarded as a better candidate for an ultrapeer.

In Eq. 1, E_i is the remained energy ratio and C_i is the connectivity of node i. β is used to give weights to two terms. f is an indicator for a freeloader. If node i is a freeloader, then f takes value zero, resulting to set the metric value zero. Otherwise, f is set to 1. Since freeloaders deteriorate the overall performance of the P2P system, our scheme is intended to rule out freeloaders becoming ultrapeers.

The ratio of the remained energy of node i, E_i, is defined as

$$E_i = \frac{E_{\mathrm{cur}_i}}{E_{\mathrm{max}_i}}, \tag{2}$$

where E_{cur_i} is the amount of remained energy of node i, and E_{max_i} is the amount of maximum energy.

Let U_i be the number of ultrapeers to which node i is connected. Also, let $U_{\max}$ be the maximum number of ultrapeers that node i can make connections. Then the connectivity of node i, C_i, is calculated as follows:

$$C_i = \frac{U_i}{U_{\max}} \tag{3}$$

Eq. 3 represents the connectivity of node i with ultrapeers. As the connectivity gets higher, it is possible to deliver ping or query messages to more nodes by flooding. Therefore, if a node connects to an ultrapeer with higher value of C_i, it is easier to have better connectivity and is probable to get more query hits.

In Eq. 1, β is a constant whose value can take from $W_{\min}$ to 1. $W_{\min}$ is used as a minimum metric value for any ultrapeer to have as explained in Section 3.3.

The connectivity of a peer which participates in the peer-to-peer system for the first time is zero. So, if β is set to less than or equal to $W_{\min}$, then the node should act as a leaf although it has maximum battery power. Consequently, all the nodes participate the P2P system as leaves and there would be no ultrapeer in the system. The result of such phenomenon leads to difficulty in forming node hierarchy in the P2P system. To avoid such a situation, the minimum value of β should be greater than $W_{\min}$.

3.2 Delivery of Metric Values

Each node maintains its own metric value. It also needs to know the up-to-date metric values of other nodes in order to operate properly. To inform the metric values among ultrapeers, we introduce a new message, namely Ultrapeer Advertisement (UADV) message. UADV message is a modified version of a pong message. It carries information on the advertising ultrapeer. Between an ultrapeer and a leaf, the metric value is basically informed using pong messages. For that purpose, we extend the format of a pong message to include the metric value. In addition to UADV and pong messages, bootstrapping, bootcache updating, and handshaking are used for the delivery of metric values.

A servent node which participates the P2P system for the first time informs its address and metric value to the bootstrap server during bootstrapping operation. If the node wishes to act as a leaf, it sets its metric value to zero. When a bootstrap server receives servent information, it saves the received information in its cache except for the information that contains metric value zero. When the bootstrap server sends the information of currently active servent to a newly joined node, it only provides the node address with higher metric values. Consequently, the newly joined node can make connections to the nodes with higher metric values preferentially.

3.3 Ultrapeer Selection and Operation

For the ultrapeer selection, we set a minimum metric value, $W_{\min}$, for which an ultrapeer should have. When the metric value of an ultrapeer becomes less than $W_{\min}$, then the ultrapeer changes its status to a leaf. Also, any leaf whose metric value becomes relatively higher than others can become an ultrapeer.

An ultrapeer whose metric value is less than $W_{\min}$ is called as a WeekUltrapeer in this paper. Every ultrapeer determines whether it is a WeekUltrapeer or not before it sends an UADV message. If it is a WeekUltrapeer, then it broadcasts UADV messages to all the ultrapeers and leaves connected. TTL value in such UADV messages is set to 1 to prevent flooding in the system. After sending the UADV messages, WeekUltrapeer changes its status to a leaf. When an ultrapeer receives an UADV message from a WeekUltraPeer, it sets the corresponding WeekUltraPeer as a leaf, and removes the entry of the WeekUltraPeer from its pong cache. It then disconnects the connection with the WeekUltraPeer. If there is no connection left, then it executes necessary steps to make connections using the information in its pong cache. If it cannot make any connection after all, it tries bootstrapping.

A leaf node can inform its metric value to the other nodes by executing bootcache update operation to bootstrap server or by handshaking with other ultrapeers. If the metric value of a leaf node becomes relatively higher than those of other nodes, other peers may request connections to it. In this case, the leaf node accepts the connection requests and performs as an ultrapeer. It generates UADV messages immediately to the connected ultrapeers to notify its status change.

3.4 Ultrapeer Advertisement

As discussed in Section 2, ping-pong operation in Gnutella generates quite bit of traffics. To address this problem, we introduce ultrapeer advertisement operation. Ultrapeer advertisement is used by an ultrapeer to inform other ultrapeers its presence proactively using UADV messages. In UADV operation, an ultrapeer which needs to make connections informs its information to other ultrapeers using UADV messages, rather than request information of other ultrapeers by broadcasting ping messages. Upon receiving an UADV message, an ultrapeer behaves as if it receives a pong message in Gnutella. In other words, if a node receives an UADV message, it tries to make a connection with the node which sent the UADV message.

Since UADV operation requires to deliver only a node's information, it introduces much less overhead than ping-pong operation. UADV operation eliminates flooding of ping messages among ultrapeers. For the operation between a leaf and an ultrapeer, ping-pong operation is used.

4 Simulation Environment

In this section, we describe the simulation environment on which simulations are executed. The Network Simulator (ns-2 version 2.26) [12] is chosen as a simulation tool.

Our evaluations are based on the simulation of 50 wireless nodes forming an ad-hoc network, moving over a 1500 m × 300 m rectangular flat space. A rectangular space was chosen to force the use of longer routes between nodes than those would occur in a square space with equal node density [13]. Total simulation time was set to 300 seconds. The link layer used in the simulation is IEEE 802.11 standard. The bandwidth and transmission range are 2 Mbps and 250 m respectively. These values are default values of ns simulator.

For the routing protocol of the ad-hoc network, we used DSDV based on the observation made in [6]. For DSDV, routing update interval was set to 15 seconds and the minimum time interval for the triggered update was set to 1 second.

Nodes in the simulation moved according to random waypoint model [14], which defines mobility pattern of nodes by pause time and the maximum node speed. Each node began the simulation by remaining stationary for the specified pause time. It then selected a random destination in the given space and moved to that destination at a speed distributed uniformly between 0 and some maximum node speed. Upon reaching the destination, the node paused again for the pause time, selected another destination, and proceeded from there as previously described. Each node repeated this behavior for the simulation time. Each run of the simulator accepted a scenario file as an input that describes the initial location and mobility pattern of each node in the network. We ran our simulations with movement patterns generated for 5 different pause times; 0, 30, 60, 120 and 300 seconds. A pause time of 0 second corresponds to continuous motion. On the other hand, a pause time of 300 seconds corresponds to no motion since the length of the simulation was set to 300 seconds. We experimented with two different maximum speeds of node movement, 1 m/sec and 20 m/sec.

For the energy model, initial energy for all nodes was set to 100 J. The amount of energy consumption for packet transmission were set to 0.66 J. For packet reception, it was assumed that 0.395 J was consumed. A mobile node was set to consume 0.035 J during idle state.

For P2P systems, we set the number of P2P nodes to 30 among 50 mobile nodes. Half of the P2P nodes were set to freeloaders. Non-P2P nodes were used just to form an ad-hoc network. We set the ratio of ultrapeers to leaf nodes to 1/4 among the P2P nodes. For each P2P node, PING_TIMEOUT was set to 30 seconds. GnutellaSim [15] was used for Gnutella. For each Gnutella messages, initial TTL value was set to 7 as recommended in [9]. For the proposed system, the minimum interval to generate UADV messages was set to 30 seconds. $U_{\max}$ was set to 4. Considering minimum value for the remained energy amount for ultrapeers after conducting simulations for Gnutella, $W_{\min}$ was set to 0.15. The value of β in Eq. 1 was set to be 0.7 after investigating query success rates with various values of β.

5 Performance Evaluation

In this section, we present the results of simulations that were conducted accordingly as described in Section 4, and evaluate the obtained results to see the performance of the proposed system over Gnutella. As measures for the performance, we investigated query success rate, query response time, overhead generated by the P2P systems, and the remained energy ratio in the peers.

The query success rate represents the ratio of queries that are replied by one or more query hits over the total initiated queries. The query success rate obtained from the simulation is shown in Fig. 1 for the various pause times and the maximum node speeds. Overall, the query success rate became lower as the mobility of nodes decreased. When the maximum node speed was 1 m/sec, query success rates were almost same regardless of pause time for the proposed system and Gnutella. However, both systems gave better query success rates when the maximum node speed was 20 m/sec. As it can be seen in Fig. 1, the proposed system gave better query success rates consistently over Gnutella regardless of maximum node speeds. On average, the proposed system gave about 19% higher query success rate than Gnutella.

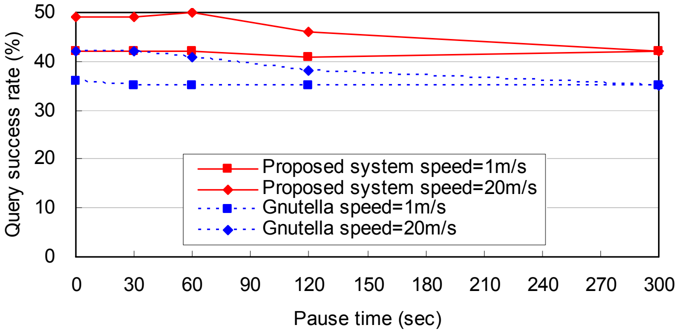


Fig. 1. Query success rate

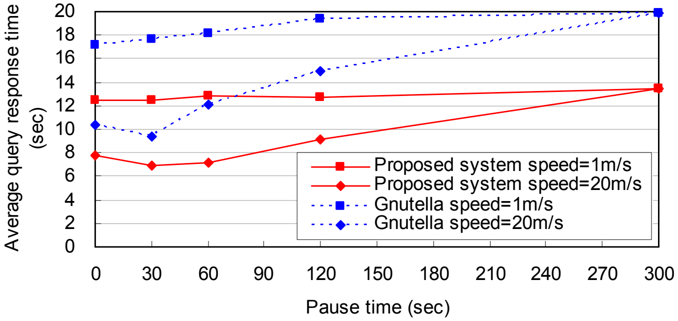
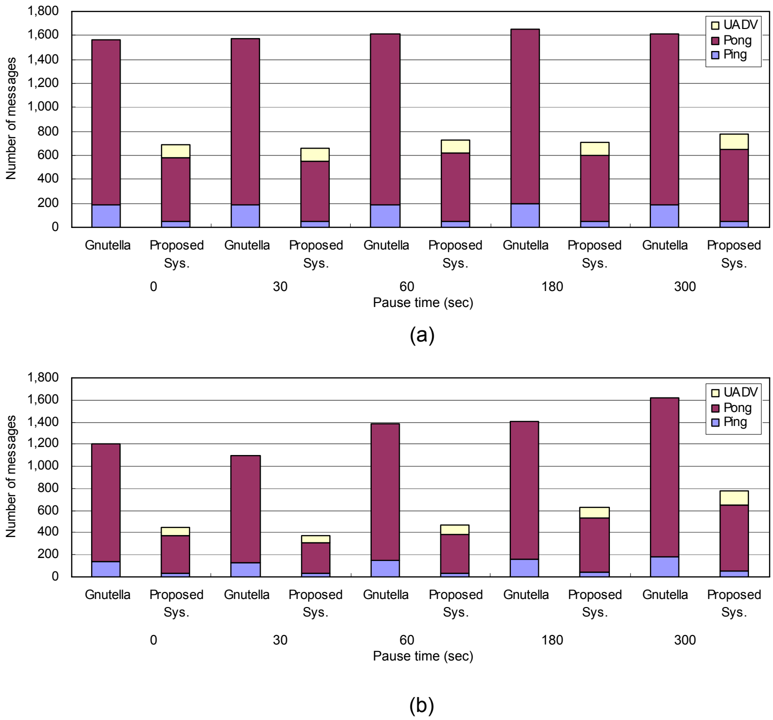


Fig. 2. Average query response time

Next, we investigated the query response time. The query response time is the time duration from the time when query is sent to the time when the corresponding query hit is received at the query initiator. Fig. 2 shows the average query response time. The average improvement of the query response time of the proposed system over Gnutella were 31% for the maximum speed of 1 m/sec and 33% for the maximum speed of 20 m/sec. As the node mobility got higher, the query response time became shorter. For the proposed system, the query response time for the stationary nodes was 71% slower than the query response time when the maximum node speed was 20 m/sec and pause time was 0 second. Such a trend was very similar for Gnutella too.

In order to calculate the overhead generated by the P2P systems, messages that are used to maintain the connectivity and to report status of the system were considered as overhead messages. For Gnutella, ping and pong messages were counted for the overhead. For the proposed system, ping, pong, and UADV messages were counted. Fig. 3 shows the average number of the P2P overhead messages generated, and Fig. 4 shows the consumed bandwidth by the P2P overhead messages. On average, the proposed system generated 75% less ping messages than Gnutellla. Also, the average number of pong and UADV mes-



(a)

(b)

Fig. 3. Overhead generated by the P2P systems in message counts (a) Maximum node speed=1 m/sec (b) Maximum node speed=20 m/sec

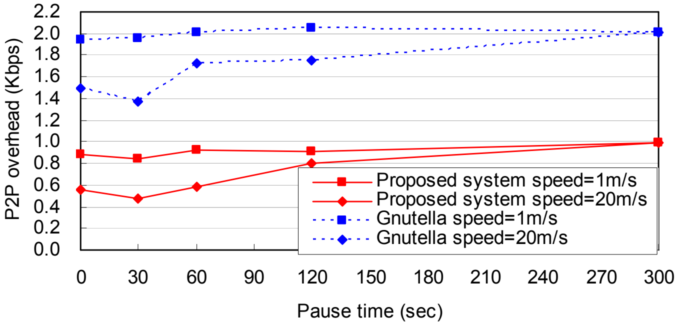


Fig. 4. Overhead generated by the P2P systems in bandwidth

sages generated by the proposed system was 66% less than the number of pong messages generated by Gnutella. When the maximum node speed was 1 m/sec, the overall overhead messages generated by the proposed system were only 44% of those by Gnutella. When the maximum node speed was 20 m/sec, the overall overhead messages generated by the proposed system were 39% of the overhead

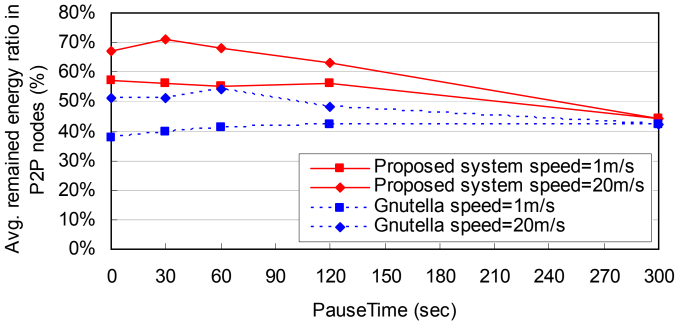


Fig. 5. Average remained energy ratio in P2P nodes

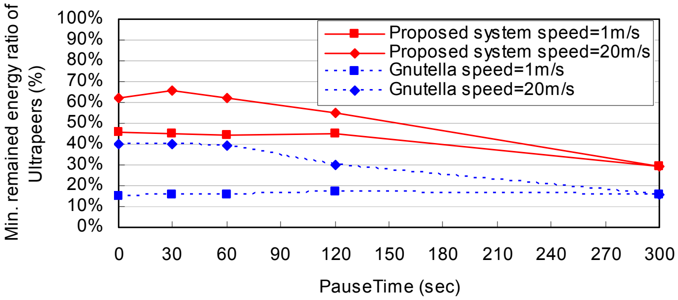


Fig. 6. Minimum remained energy ratio in Ultrapeers

messages generated by Gnutella. For the bandwidth consumed by the overhead messages, the proposed system consumed 57% less bandwidth than Gnutella.

For the remained energy ratio, we investigated two terms. The first one is the average remained energy ratio with respect to the full battery power for all the P2P nodes in the system. The second one is the minimum remained energy ratio among ultrapeers in the system. The result of the average remained energy ratio investigated for all the P2P nodes is shown in Fig. 5. The proposed system provided better energy efficiency as the pause time decreased. When the nodes did not move, the difference in the remained energy ratio between the proposed system and Gnutella was only 5%. However, as the pause time decreased, the remained energy ratio in the proposed system became 1.5 times of that in Gnutella. Fig. 6 shows the minimum remained energy ratio among ultrapeers after the simulations. The proposed system provided the minimum remained energy ratio which was three times of the minimum remained energy ratio by Gnutella when maximum node speed was 1 m/sec and the pause time was 0 second. For the maximum node speed of 20 m/sec, the proposed system provided 50% more minimum remained energy ratio than Gnutella for ultrapeers. Since a specific peer acted as an ultrapeer without considering any power condition in Gnutella, the average remained energy ratio became lower and the overall system lifespan became shorter. On the other hand, the proposed system used an efficient ultrapeer election scheme in order to incorporate the remained battery power as a consideration for the role of an ultrapeer.

6 Conclusion

In this paper, we proposed an enhanced version of Gnutella system that can efficiently operate in the ad-hoc networks. The objectives of our proposal is to increase the performance of the P2P system by reducing overall overhead incurred by the P2P system and by choosing appropriate ultrapeers with consideration of battery power. For that, we introduce a metric for servents in the P2P system. The metric takes connectivity, energy level, and the commitment level of the participating servents into account, and is used for the selection of ultrapeers. Proactive approach is used to provide up-to-date information on the ultrapeers by delivering ultrapeer advertisements periodically. Based on the analysis of the results obtained through extensive simulations, the proposed P2P system outperformed Gnutella remarkably.

References

1. C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely and C. Diot, “Packet-level traffic measurements from the sprint IP backbone,” IEEE Network, vol. 17, no. 6, pp. 6–16, Nov. 2003.
2. Gnutella, http://www.gnutella.com/
3. R. Schollmeier, I. Gruber and M. Finkenzeller, “Routing in Mobile Ad Hoc and Peer-to-Peer Networks. A Comparison,” in Workshop on Peer-to-Peer Computing, held in conjunction with IFIP Networking 2002, May 2002.
4. M. Conti, E. Gregori and G. Turi, “A Cross-Layer Optimization of Gnutella for Mobile Ad hoc Networks,” in Proc. MobiHoc ’05, May 2005.
5. L. B. Oliveira, I. G. Siqueira and A. A. F. Loureiro, “Evaluation of Ad-hoc Routing Protocols under a Peer-to-Peer Application,” in Proc. WCNC 2003, pp. 1143–1148, 2003.
6. H. Choi, H. Park and M. Woo, “Performance Analysis of Peer-to-Peer Application in Ad-Hoc Networks,” in Proc. ITST 2005, pp. 49–52, June 2005.
7. D. Diego and O’Mahony Donal, “Overlay Networks - A Scalable Alternative for P2P,” IEEE Internet Computing, Vol. 7, No. 3, pp 2–5, June/Junly 2003.
8. M. Portmann, P. Sookavatana, S. Ardon and A. Seneviratne, “The Cost of Peer Discovery and Searching in the Gnutella Peer-to-peer File Sharing Protocol,” in Proc. IEEE ICON 2001, Sep. 2001.
9. T. Klingberg and R. Manfredi, “Gnutella 0.6,” available from http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html, Jun. 2002.
10. A. Singla, C. Rohrs and Lime Wire LLC, “Ultrapeers: Another Step Towards Gnutella Scalability,” available from http://rfc-gnutella.sourceforge.net/src/Ultrapeers_1.0.html, Nov. 2002.
11. Limwire: http://www.limewire.com/

12. K. Fall, K. Varadhan, editors. ns notes and documentation. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, available from http://www.isi.edu/nsnam/ns/, Nov. 1997.
13. J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu and J. Jetcheva, "A Performance Comparison of Multi-Hop Ad Hoc Network Routing Protocols," in Proc. MobiCom'98, Oct. 1998.
14. D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers, pp. 153–181, 1996.
15. Packet-level Peer-to-Peer Simulation Framework and GnutellaSim version 1.1, http://www.cc.gatech.edu/computing/compass/gnutella, Sep. 2003.

Load Balancing by Distributed Optimisation in Ad Hoc Networks*

André Schumacher, Harri Haanpää, Satu Elisa Schaeffer, and Pekka Orponen

Laboratory for Theoretical Computer Science, Helsinki University of Technology,
P.O. Box 5400, FI-02015 TKK, Finland
Andre.Schumacher@tkk.fi, Harri.Haanpaa@tkk.fi,
Elisa.Schaeffer@tkk.fi, Pekka.Orponen@tkk.fi

Abstract. We approach the problem of load balancing for wireless multi-hop networks by distributed optimisation. We implement an approximation algorithm for minimising the maximum network congestion as a modification to the DSR routing protocol. The algorithm is based on shortest-path computations that are integrated into the DSR route discovery and maintenance process. The resulting Balanced Multipath Source Routing (BMSR) protocol does not need to disseminate global information throughout the network. Our simulations with the ns2 simulator show a gain of 14% to 69% in the throughput, depending on the setup, compared to DSR for a high network load.

1 Introduction

Ad hoc networks are communication networks formed by a number of nodes, which are small radio devices with limited computational capacity [1]. Perhaps the most significant advantage of ad hoc networks – and simultaneously an important design goal – is the ease of deployment. Ideally, it should be possible to deploy the nodes in the area of operation and have them self-organise to route traffic as necessary. Such a setup would be useful in a variety of environments ranging from military operations and disaster relief to commercial applications.

Ad hoc networks also present challenges. Nodes are usually battery-operated, as they should not depend on an external energy supply, and battery life is often a limiting factor. The radio transmission channel is limited in bandwidth and shared among nearby nodes. Determining and maintaining the network topology in a distributed fashion is a challenging problem, particularly if the network topology changes during operation due to addition, removal, or mobility of nodes.

Two properties of algorithms are particularly desirable in an ad hoc context. First, an algorithm should be mathematically justified. Analysing an algorithm mathematically gives insight into when it can be expected to work and when not. Linear and integer programming formulations can typically be applied in this approach to gain optimal solutions for small problem instances or good

* This research was supported by the Academy of Finland under grants 209300 (ACSENT) and 206235 (ANNE).

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 873–884, 2006.

approximate solutions for larger instances. Such methods have been applied to optimisation of sensor-node coverage [2] and lifetime in energy-constrained networks [3], but these approaches typically require collecting state information to a central location to perform the optimisation, adding undesired hierarchy and a point of failure.

Second, an algorithm should be distributed and non-hierarchical. Each node should follow a simple set of rules to cooperate in computing the optimum. Neither the size nor the number of messages should grow rapidly with the size of the network. Such approaches have been used for bandwidth optimisation [4]. Certain energy-aware modifications of routing protocols such as AODV or DSR also fall into this category. However, formal analysis of heuristic optimisation methods is difficult and usually only simulation-based analysis is applicable.

In mathematically justifiable distributed algorithms, the nodes typically compute graph-based properties, such as shortest paths or spanning trees, in a distributed and iterative manner. This enables theoretic analysis of the expected quality of the solution and the convergence of the algorithm towards the optimum. Such methods have been applied to adjusting transmission power levels based on lowest-cost energy paths [5] and routing around congested nodes based on node potentials and the steepest gradient method [6].

In this paper, we present the Balanced Multipath Source Routing (BMSR) protocol that extends the Dynamic Source Routing Protocol (DSR) [7] to use *multipath routing* for balancing data traffic. Multipath extensions to DSR have been previously studied: Nasipuri, Castañeda, and Das [8] introduce alternate routes to the route discovery process, whereas Wu and Harms [9] propose a heuristic redirection of `RREP` messages to gain alternative routes. The focus has been primarily on the computation of node or link-disjoint paths, as they provide a higher fault tolerance in the presence of failures. Ganjali and Keshavarzian [10] state that multipath routing *alone* can not improve load balancing: as node density increases, the choice of shortest paths connecting any pair of nodes leads to congestion in the centre of the network. They conclude that additional incentive is needed to push traffic away from the centre.

Multipath-based network optimisation has been studied extensively for wired networks. Vutukury and Garcia-Luna-Aceves [11] propose an algorithm to minimise delay by heuristic redirection of flow over multiple paths. Basu, Lin and Ramanathan [6] present a potential-based routing method that forwards packets using steepest gradient search and propose a traffic-aware routing algorithm. This approach relies on a link-state routing algorithm for the dissemination of link information throughout the entire network.

However, most proposals are not directly applicable to ad hoc networks due to the aforementioned limitations. Our proposed BMSR protocol constitutes modifications of the DSR protocol and obtains multiple source-destination routes by a linear programming approximation algorithm that minimises flow congestion [12]. The algorithm relies on the computation of shortest paths determined by an adaptive cost metric using link weights. Using distributed weight updates we avoid dissemination of global information. In our simulations we achieve a gain of 14% to 69% in the throughput, depending on the setup, compared to DSR.

The paper is organised as follows. In the next section, we describe the linear programming approximation algorithm, followed by a brief overview of the basic operation of DSR. Thereafter, we describe extensions that were made to DSR in order to implement the approximation algorithm, which constitute the proposed BMSR protocol. Section 3 presents simulation results with the `ns2` [13] network simulator. Finally, Section 4 concludes the paper and outlines future research directions.

2 Distributed Load Balancing

In this section we describe the BMSR protocol, which is an extension to DSR for load balancing by multi-path routing. DSR normally uses one route from the source node to the destination node. However, extending DSR to use more routes is relatively easy and may improve reliability, throughput, and load balancing. We model choosing a set of source routes as a min-max congestion multi-commodity flow problem and describe the implementation of the approximation algorithm BMSR is based on.

2.1 Approximation Algorithm

We model the ad hoc network as a directed graph $G = (V, E)$ with vertices representing the radio nodes of the network and edges representing links between the radio nodes. For two vertices $i, j \in V$, we have a directed edge $(i, j) \in E$ if there exists a link from radio node i to j.

We consider routing as a multicommodity flow problem: each commodity c represents one data stream of traffic of volume v^c from the source s^c to the destination d^c. When $t^c(i)$ represents the supply of commodity c at vertex i, we have $v^c = t^c(s^c) = -t^c(d^c)$ and $t^c(i) = 0$ for all other nodes i. The task is to find flows x^c_{ij} of commodity c along each edge (i, j) that satisfy the flow requirements

$$t^c(i) + \textstyle\sum_{(j,i)\in E} x^c_{ji} - \sum_{(i,j)\in E} x^c_{ij} = 0 \qquad (1)$$

for each commodity c and vertex i. Within these constraints we choose x^c_{ij} to *minimise the maximum congestion*:

$$\min \max\nolimits_{(i,j)\in E} f_{ij}/u_{ij} \ , \qquad (2)$$

where $f_{ij} = \sum_c x^c_{ij}$ is the total flow along edge (i, j) and u_{ij} is its capacity.

Many algorithms exist for solving such linear optimisation problems when the whole state of the network is known. In contrast, we need an optimisation algorithm that can be implemented so that the individual nodes cooperate to determine the optimum by passing only a reasonable number of messages of reasonable size. The approximation algorithm that we use for min-max congestion multi-commodity flow is from [12]; it computes a flow x over a set of paths, taking as input a graph $G = (V, E)$ and a list of flows of volume v^c from source s^c to destination d^c, with parameters I and ϵ.

1. Initialise $w_{ij} = 1$ for each edge $(i, j) \in E$. For each edge (i, j) and every commodity c (with source node s^c and destination node $d^c \in V$), set the flow $x^c_{ij} = 0$.
2. For each of the I iterations, do the following computation:
   (a) For each source-destination pair of nodes s^c and d^c, compute the shortest path $p(s^c, d^c)$ with respect to the edge weights defined by w.
   (b) Let y^c be the flow vector resulting from routing v^c units of flow on the shortest path $p(s^c, d^c)$. For each edge $(i, j) \in E$, assign $x^c_{ij} := x^c_{ij} + y^c_{ij}$ and
$$w_{ij} := \big(1 + \epsilon \sum_c y^c_{ij}\big) w_{ij} \quad . \tag{3}$$
3. Scale the total flow by letting $x := x/I$.

In this formulation, each edge has the same capacity u. To obtain flows for which the maximum congestion is at most $(1 + \epsilon)$ times the optimal value, it suffices to run the algorithm for

$$I \geq \left\lceil 4m \log m/\epsilon^2 \right\rceil \tag{4}$$

iterations, where m is the number of edges.

2.2 DSR Operations

DSR [7] is an on-demand *source routing* protocol: the source includes the whole route in every packet sent. This property eliminates the need for actively maintaining routing information at intermediate nodes and enables an easy integration of multipath routing. Nodes keep routing information in their *route cache*, which can also contain routing information that was overheard from neighbouring nodes.

The basic DSR protocol consists of two operations: *route discovery* and *route maintenance*. If a source node wishes to send a packet to a destination to which it does not have a route in its route cache, it initiates the route discovery process by broadcasting a *route-request* (RREQ) message to its neighbours. Upon receiving the RREQ, nodes consult their route cache and can decide to send a *route-reply* (RREP) message back to the source. If they do not know a route to the destination, they append their own address to the list of nodes in the RREQ and forward the request further, until it eventually reaches the destination. The destination obtains a route from the source to itself by consulting the list of nodes that forwarded the RREQ. In the presence of bidirectional links, it can simply reverse this route and use it for sending a RREP message along this route to the source.

A sequence number mechanism ensures limited forwarding of RREQ's by intermediate nodes. In route discovery, a node only forwards each RREQ at most once. Since shorter routes require fewer hops, the first RREQ to reach the destination is likely to have taken a route that is (close to) minimal in terms of the hop count. Therefore, DSR chooses routes not much longer than the shortest route between source and destination. Although in principle multiple routes to the same destination may be contained in the route cache, e.g. by overhearing other routes, the nodes always pick the shortest route from the cache.

The basic route maintenance includes reliable packet transmissions from one hop to the next, e.g. utilising link-layer acknowledgements. Additionally, there are other operations initiated on-demand. If a source route breaks, the source is notified by an intermediate node detecting the break. The source can then choose to select an alternative route to the destination by consulting its route cache, or initiate a new route discovery. In the case that the intermediate node has a different route to the destination in its own cache, it can initiate *packet salvaging* and forward the packet using this alternative route.

2.3 The BMSR Protocol

The shortest-path methodology of the approximation algorithm enables a simple extension to DSR; the computation of shortest paths is similar to that of the original protocol. Our approach differs from DSR in that DSR initiates route discovery when necessary, while BMSR uses an initial setup phase to proceed through the iterations of the balancing algorithm. Each source obtains one *balanced route* to the destination per iteration. Some routes may occur more than once. After the setup phase, every packet sent by the source follows a randomly chosen cached route. Unlike in DSR, the routes are not removed from the cache when link failures occur, as the failure may be due to temporary link congestion.

We implement BMSR by modifying DSR's route discovery and route maintenance operations. The DSR route control messages `RREQ` and `RREP` are extended to include *iteration-index*, *cost* and *flow-value* fields. These fields correspond to the variables needed for the algorithm of Section 2.1. For clarity, we refer to these modified messages by `BREQ` and `BREP`. Instead of computing shortest routes based on hop-counts, the nodes compute the minimum-cost route for each iteration of the balancing algorithm and each source and destination pair. The cost of a route is the sum of the link costs w on that route. Each node keeps track of the weight of and the flow on each incoming link (i.e., those links that it may use to receive a `BREQ` message from a neighbour). `BREQ` messages carry, in addition to the list of addresses of nodes that re-broadcasted the message, the accumulated route cost from the source. An intermediate node adds the cost of the incoming link on which it received the `BREQ` to the accumulated route cost of the `BREQ` upon re-broadcasting it. Later, however, an intermediate node may receive another `BREQ` packet with the *same* iteration index. If the new `BREQ` has a lower-cost route from the source than the previous one, the intermediate node re-broadcasts it.

When the destination receives a `BREQ` packet, it must wait a short period for possible lower-cost `BREQ` packets. The destination only replies with a `BREP` to the `BREQ` with lowest cost. The flows and weights are updated along the route used when the destination sends the `BREP` packet back to the source. As the link weights and therefore the least-cost routes are subject to change at each iteration, the balanced routing algorithm can not rely on DSR's caching mechanism to narrow down the dissemination of `BREQ` messages in the network. Therefore, `BREQ`'s have to spread by flooding through the network. Since the parameters ϵ and I can be used for a trade-off between route-control overhead and quality of the solution, this effect can be adjusted to the network setup. Additionally, the setup phase is only performed once even for long data streams.

As mentioned above, routes that are broken due to temporarily congested links *stay* in the cache and do not get invalidated. For a larger number of iterations the effect of a single link failure diminishes, as the source randomly selects balanced routes from the cache.

3 Experiments

We consider a stationary grid network with source and destination pairs. The chosen traffic pattern resembles a mesh-network scenario, where a large amount of constant bit rate (CBR) data is transfered through an already congested network. When a sudden demand arises for transmitting a large amount of data between a dedicated pair of nodes, e.g. between a control centre and rescue teams, one aims to deliver as much of the critical data as possible. For this purpose one must balance the traffic among the nodes and utilise the network capacity to maximise throughput over source-destination pairs.

We compare BMSR to DSR by using `ns2` to simulate it on a 10 by 10 square grid with two CBR flows, from s_1 to d_1 and from s_2 to d_2; see Fig. 1 for the network setup. Both CBR sources are transmitting with a previously determined rate and packet size. See Table 1 for the particular parameter values. Prior to initiating the CBR traffic, we run the balancing algorithm of Section 2.1 for a chosen value of ϵ and a chosen number of rounds I to select routes that give an approximately balanced flow in the sense of minimising the maximum congestion. We run a series of long simulations to obtain estimates of the throughput of the network, defined as the average rate of CBR data that was received by the destinations. In the following we will refer to this metric as the performance for the particular choice of parameters. We use the same source-destination setup to transmit data using the DSR implementation provided in `ns2`.

Table 1. The parameters used in `ns2` simulations

CBR packet size (B)	256, 512,1024, 2048	MAC bandwidth	1 Mbit
CBR data rate	160 Kbit/s	MAC protocol	802.11 with `RTS/CTS`
Antenna type	OmniAntenna	Propagation model	TwoRayGround
Max. IFQ length	50	Max. route length	22
Network size	2.4 km × 2.4 km	Node count	100
Simulation time	1500 s	Balancing setup	500 s

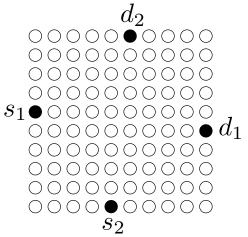


Fig. 1. The simulation setup: two source-destination pairs (s_1, d_1) and (s_2, d_2) are placed "off-by-one" on the opposite sides of the grid. The source nodes s_1 and s_2 send data packets at constant rate to their respective destination nodes d_1 and d_2. Each node may communicate with the nodes beside, above or below it.

In addition to throughput, we study the distribution of routes over the nodes by calculating the number of forwarded CBR packets at each node. We expect most packets to be forwarded by nodes located near the centre of the network, as these routes are shortest and the algorithm initially prefers shorter routes over longer ones. However, the central nodes should not be loaded much more heavily than those on slightly longer paths.

A balanced network load should also reduce collisions and *interface queue* (IFQ) overflows in the network. The IFQ contains packets that are scheduled to be transmitted over the network interface. Hou and Tipper [14] observed that one of the main reasons for the decline in throughput for congested networks running DSR is the overflow of the IFQ of congested nodes. Besides queue overflows, collisions of the *media access control* (MAC) layer control messages and CBR packets are expected to degrade the performance. Although we do not expect the number of collisions to be significantly lower compared to the DSR route selection, we would expect a more even distribution over the nodes, preventing bottleneck formation. Figure 2 shows simulation results for two CBR packet sizes.

We use the following measures: CBR *packet load*; the number of CBR packets sent by the MAC layer of the node. Note that there are in total 20000 and 10000 packets per source for packet sizes of 1024 and 2048 bytes respectively. This value does not correspond to the actual number of successfully forwarded packets, as drops and collisions have to be subtracted. Sources were excluded from Fig. 2 for clarity. CBR *packet collisions*; the number of CBR MAC layer collisions caused by interference that occurred at each node, excluding the sources. These numbers do not necessarily coincide with the number of dropped packets, as the MAC layer uses a retransmission scheme. IFQ *overflows caused by* CBR *packets*; the number of IFQ overflow events that occurred at each node.

One might expect DSR to favour shorter routes, yielding an increased network load within the centre of the grid that results in interference and a low network throughput. BMSR should recognise areas of higher congestion and after

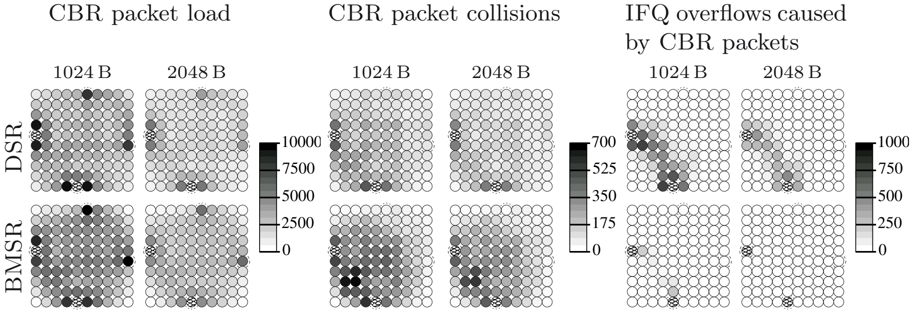


Fig. 2. Averages over five runs for the performance measures of DSR and BMSR for $I = 160$ iterations and $\epsilon = 0.05$ for two CBR packet sizes; variations were negligible. Source and destination nodes are indicated by dashed circles.

initially selecting shorter routes, select routes that avoid the potentially congested areas. In Fig. 2, we only observe minor differences for BMSR and DSR. Depending on the averaging of packet load over the rather long simulation run, the load for DSR appears to be well balanced. The reason is that within the congested network, rediscovered routes will typically be different from recently broken routes. There is a slightly higher utilisation of boundary nodes by DSR, but the overall network load for BMSR is higher than for DSR, which can be explained by the higher throughput, discussed later in this section.

Due to higher load, BMSR encounters more collisions compared to DSR. A remarkable effect is the concentration in the quadrant of the network formed by the square with the sources on its diagonal. The effect is apparent for both algorithms and packet sizes, but emphasised for BMSR and 1024-byte packets. Nodes within this part of the network may be relaying packets from both sources in roughly opposite directions. Hence they have to transmit packets in more diverse directions than nodes within the vicinity of the destinations.

As the MAC layer transmission of a CBR packet includes a *request to send* (`RTS`)/*clear to send* (`CTS`) handshake, collisions are more likely to occur when nodes are transmitting in different directions than when the packets travel roughly in the same direction. DSR always uses the shortest known route to the destination. Therefore, subsequent packets for the same destination are less likely to interfere with each other. The distribution of IFQ overflows follows basically the same principle. We, however, observe a major difference between BMSR and DSR: the single-path routing of DSR leads to the formation of bottleneck nodes due to congestion in the bottom left quadrant of the network. As DSR prefers shorter paths, such overloading of nodes is restricted to the band of nodes between the sources. The effect is stronger for smaller packet sizes, explained by the increased MAC layer overhead. BMSR shows hardly any IFQ overflows at all, except within the vicinity of the sources.

Figure 3 shows the performance of both routing protocols over time. Comparing throughput for BMSR and DSR, one observes larger fluctuations for

DSR. A major reason for the throughput stability of BMSR is that broken links do not cause route invalidation. Therefore, its performance is determined during the initial setup phase of the algorithm. To compensate the fluctuations of DSR, we consider the throughput over 1000 s from the time when CBR transmissions have been initiated to compare both algorithms in the following. For both packet-sizes BMSR clearly outperforms DSR.

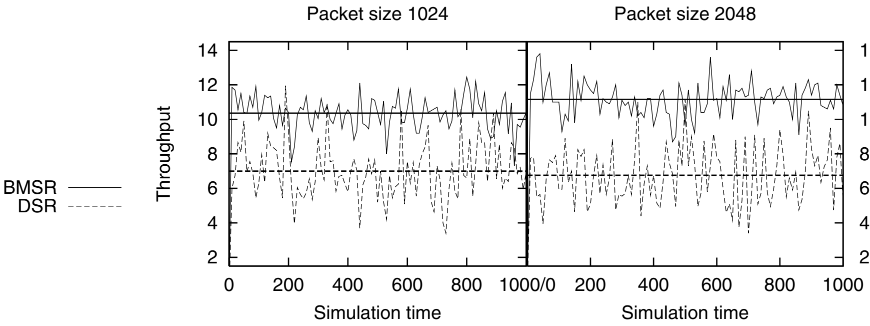


Fig. 3. Average throughput of both source-destination pairs in KB/s versus simulation time for a single run of BMSR and DSR. Note that the setup stage for BMSR is omitted from the plot. The parameter values for the balancing algorithm were $I = 160$ and $\epsilon = 0.05$. The horizontal lines are averages over the entire simulation.

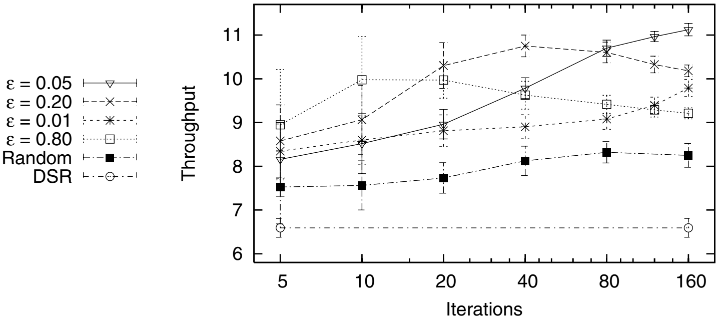


Fig. 4. Performance in KB/s as a function of I and ϵ for CBR packet size 2048: BMSR, DSR, and random route selections of I routes between source-destination pairs. All values are averages over at least 15 repetitions (standard deviations shown). The legend ordering corresponds to the throughput value at $I = 160$.

We also studied the effect of the I and ϵ parameters on the performance. The results are summarised in Fig. 4 and are mostly as expected; already for a modest number of iterations we obtain throughput superior to DSR. There is a dependency of the throughput on ϵ and I: for larger values of ϵ, fewer iterations are needed to obtain a good throughput, but running a large number of iterations with a small value of ϵ yields a slightly better throughput.

A curious phenomenon in the results is that for a given value of ϵ, the throughput first increases rapidly as I increases but after reaching a maximum, the throughput starts to decline gradually. We can only offer a heuristic explanation of this phenomenon. As the optimisation algorithm progresses, the weights of the most congested edges will come to completely dominate the search for the least cost route from the source to the destination. With a large enough iterations count, the algorithm only seeks to balance the flow on those edges without any regard for the traffic situation in the rest of the network. We also ran the tests for other values of ϵ, but omitted some from the figure for clarity; for $\epsilon \leq 0.05$, the peak performance had not yet been reached for $I = 160$.

However, in our experiments we ran considerably fewer iterations than recommended by (4). For small values of ϵ, in the first iterations the weight of each edge remains at approximately 1, and thus the paths found by BMSR will be essentially fewest hop paths. It seems plausible that instead of only optimising the hop count, or only balancing the flow along the most congested edges, good results could be obtained by taking both factors into consideration – and we hypothesise that this is what happens when the number of iterations I is less than recommended by (4).

The results shown in Fig. 2 indicate that there is a qualitative difference in the performance of BMSR and DSR for different packet sizes. Figure 5 shows throughput and packet delay for various packet sizes. The throughput performance of DSR seems to increase until a critical packet size, after which increasing the packet size further decreases DSR's performance. We assume this to be caused by the interdependence of the two main reasons of packet loss: collisions of CBR packets due to interference and IFQ overflows.

BMSR aims at decreasing link congestion; it reduces the number of IFQ overflows, as shown in Fig. 2. We conclude that increasing the packet size reduces the negative effect of collisions on throughput for BMSR: increasing packet size

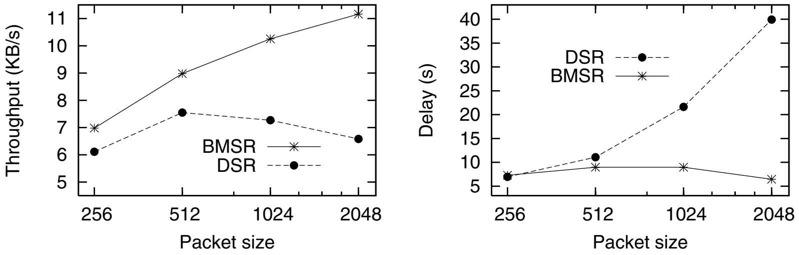


Fig. 5. On the left, throughput in KB/s versus packet size, and on the right, delay in seconds versus packet size, for parameter values $I = 160$ and $\epsilon = 0.05$. All values are averages over at least 15 repetitions. Note the logarithmic scale for the packet size and that the CBR rate is 160 Kbit/s for all runs.

for a constant CBR rate reduces the number of packets and therefore the total MAC layer overhead. However, the time frame required for the transmission of a single packet increases correspondingly and retransmissions become more costly. Still the effect of losing larger packets due to IFQ overflows seems to outweigh the impact of collisions.

As Fig. 5 indicates, DSR packet delay grows nearly linearly with packet size, whereas BMSR shows a clearly lower, approximately constant delay. This is most likely due to the fact that after the initial setup phase BMSR uses a static routing scheme.

4 Conclusions and Future Work

We studied the application of a linear programming approximation algorithm to distributively optimise network bandwidth in a wireless multi-hop network. The algorithm aims at minimising the maximum flow over any edge in the input graph. We integrated it into the DSR route-discovery process in a distributive manner and obtained significant increase in throughput for the studied topology. The topology considered was static and regular. As future work we are interested to consider more general network topologies, non-uniform spatial node distributions, and to incorporate mobility.

We believe that optimising link congestion proves successful also for other topologies with a uniform distribution of nodes and a relatively regular graph structure. For non-uniform topologies we expect the optimisation for node-based metrics to work better. We plan to study the effect of node-based metrics on the balancing algorithm, such as optimising for node congestion. We would expect edge congestion to serve well in uniform topologies, but a node-based approach to give better results in non-uniform topologies, where the load on single hubs may get heavy due to a high number of neighbouring nodes.

The static-network and the uniform-node-distribution assumptions are essential in the current formulation of the algorithm. Besides considering node-based optimisation metrics, we want to consider a steady-state formulation of the algorithm, e.g. by enabling a calculation of the edge weights depending on the present edge flow. Further applications of the BMSR protocol, such as energy-efficient routing, are to be considered as well.

The results presented in this paper show the potential of using mathematically justified distributed optimisation techniques for ad hoc networks. By utilising shortest-path computations integrated into the DSR route discovery, we obtain an improvement in throughput of 14% to 69% compared to DSR for a network with high load. The assumption of a static network with a uniform spatial distribution of nodes does not seem too restrictive. We are convinced that it can serve as a starting point for further investigating the potential of distributed optimisation for ad hoc networks.

References

1. Perkins, C., ed.: Ad Hoc Networking. Addison Wesley, Reading, MA, USA (2001)
2. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.: Coverage problems in wireless ad-hoc sensor networks. In: Proc. 20th Annual Joint Conf. of the IEEE Computer and Communications Societies. (2001) 1380–1387
3. Floréen, P., Kaski, P., Kohonen, J., Orponen, P.: Lifetime maximization for multicasting in energy-constrained wireless networks. IEEE J. Selected Areas in Communications **23**(1) (2005) 117–126
4. Aggelou, G., Tafazolli, R.: RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In: Proc. 2nd ACM Int'l Workshop on Wireless Mobile Multimedia. (1999) 26–33
5. Kawadia, V., Kumar, P.: Power control and clustering in ad hoc networks. In: Proc. 22nd Annual Joint Conf. of the IEEE Computer and Communications Societies. (2003)
6. Basu, A., Lin, A., Ramanathan, S.: Routing using potentials: A dynamic traffic-aware routing algorithm. In: Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Communication, (2003) 37–48
7. Johnson, D., Maltz, D., Hu, Y.: The dynamic source routing protocol for mobile ad hoc networks (DSR). Tech. report, IETF (2003) IETF Draft, July 2004.
8. Nasipuri, A., Castañeda, R., Das, S.: Performance of multipath routing for on-demand protocols in mobile ad hoc networks. Mobile Networks and Applications **6**(4) (2001) 339–349
9. Wu, K., Harms, J.: Performance study of a multipath routing method for wireless mobile ad hoc networks. In: Proc. 9th Int'l Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Washington, DC, USA, IEEE Computer Society (2001) 99–107
10. Ganjali, Y., Keshavarzian, A.: Load balancing in ad hoc networks: Single-path routing vs. multi-path routing. In: Proc. 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies. (2004)
11. Vutukury, S., Garcia-Luna-Aceves, J.: A simple approximation to minimum-delay routing. In: Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Communication, New York, ACM Press (1999) 227–238
12. Bienstock, D.: Potential Function Methods for Approximately Solving Linear Programming Problems: Theory and Practice. Volume 53 of International Series in Operations Research & Management Science. Kluwer, Norwell, MA, USA (2002)
13. McCanne, S., Floyd, S., Fall, K., Varadhan, K.: The network simulator `ns2` (1995) The VINT project, available for download at `http://www.isi.edu/nsnam/ns/`.
14. Hou, X., Tipper, D.: Impact of failures on routing in mobile ad hoc networks using DSR. In: Proc. Communication Networks and Distributed Systems Modeling and Simulation Conf. (2003)

Author Index

Amini, Fereshteh 497
Assi, Chadi 436

Badache, Nadjib 685
Baggio, Aline 317
Bahi, Jacques M. 555
Beirne, Stephen 820
Benslimane, Abderrahim 424, 436
Bertino, Elisa 461
Bradford, Phillip G. 543
Burri, Nicolas 1

Cai, Kan 13
Cha, Hojung 748
Chen, Hekang 600
Chen, Kai 281
Chen, Ling-Jyh 724
Chen, Rong 698
Chen, Yingwen 60
Chen, Yu 711
Chen, Yung-Chih 724
Cheng, Wenqing 209
Cho, Yookun 760
Choi, Hyun-Duk 861
Choi, Seung-Hyuk 294
Choo, Hyunseung 341
Chuah, Mooi Choo 377
Chung, Min Young 294, 341
Corcoran, Brian 820
Cui, Jianwen 220, 232

Diamond, Dermot 820
Du, Junzhao 304

Eom, Doo-Seop 329

Fan, Guangbin 84
Feeley, Michael J. 13
Fleury, Eric 711
Foo, Mao Ching 770
Forné, Jordi 485
Fu, Fen 377
Fukazawa, Y. 353

Gajic, Zoran 197
Gao, Shan 268
Gao, Zhenguo 220, 232
Garoche, P.-L. 353
Goncalves, Betânia Steffen Abdallah 25
Gong, Haigang 638
Gu, Boncheol 760
Gu, Naijie 131
Guan, Jihong 600
Guang, Lei 436
Guérin-Lassous, Isabelle 25
Guerroumi, Mohamed 685

Haanpää, Harri 873
Hardy, Gary 244
He, Tian 365
Heo, Junyoung 760
Hernández-Serrano, Juan 485
Hong, Jiman 760
Hong, Xiaoyan 461, 543
Honiden, S. 353
Huang, Bo 600
Huang, Jiaqing 209
Huang, Liusheng 650
Huang, Xinyi 473
Hutchinson, Norman C. 13
Hwang, Kwang-il 329

In, Jeongsik 329
Ingelrest, François 143
Ishida, Kenji 808

Jeon, Jun-Cheol 416
Jeong, Jongpil 341
Jia, Weijia 131
Jia, Xiaohua 622
Jiang, Fan 281
Jiang, Shengming 736
Jin, Hai 698

Kakuda, Yoshiaki 808
Kang, Byung-Heon 416
Kang, Chul-Hee 185
Kawano, Atsushi 808

Khan, Moazzam 497
Ki, Hyung Joo 294
Kim, Bonam 171
Kim, Jae-Wan 329
Kim, Kiseon 782
Kim, Se-Min 416
Koberstein, Jochen 835
Koo, Insoo 782
Kotz, David 155
Kramarev, Dmitry 782
Kubo, Yasunori 808
Kumar, Rajnish 155, 793

Lam, Vinh The 770
Langendoen, Koen 317
Lau, Francis C.M. 38
Lee, Byung-Jin 107
Lee, HyunJun 449
Lee, Jihoon 329
Lee, KyungOh 449
Lee, Tae-Jin 294
Lee, Wan-Soo 416
Lee, YoungHan 449
Li, Deying 622
Li, Huifang 736
Li, Jing 131
Li, Minglu 119, 674
Li, Xiang 220, 232
Li, Xiangfang 197
Li, Xiang-Yang 392
Li, Xu 847
Li, Yingshu 268
Lian, Jie 47
Lim, Ee-Peng 662
Lim, Hock Beng 770
Limnios, Nikolaos 244
Liu, Chi 95
Liu, Chungui 392
Liu, Hai 622
Liu, Jun 461
Liu, Liang 256
Liu, Ming 638
Liu, Wei 209
Liu, Yi 404
Liu, Zhong 612
Lu, Jianheng 38
Lucet, Corinne 244
Luttenberger, Norbert 835
Lv, Shaohe 95

Ma, Huadong 256
Maeda, Kaori 808
Marsh, David 820
Min, Hong 760
Mišić, Jelena 497
Mitton, Nathalie 25
Mostefaoui, Ahmed 555
Moussaoui, Samira 685
Mu, Yi 473, 521

Naik, Kshirasagar 47

Oh, Hyukjun 760
O'Hare, Gregory 820
Ohta, Tomoyuki 808
Oka, Daisuke 808
Okada, Hiromi 171
Orponen, Pekka 873

PalChaudhuri, Santashil 793
Pan, Qunhua 674
Park, YoonYoung 449
Paul, Arnab 155
Pegueroles, Josep 485

Qian, Lijun 197

Rachedi, Abderrezak 424
Ramachandran, Umakishore 155, 793
Ramamurthy, Byrav 509
Reiss, Charles 793
Rickenbach, Pascal von 1
Ruiz, Pedro M. 567

Sakai, Kazuya 171
Salomon, Michel 555
Sanchez, Juan A. 567
Santoro, Nicola 847
Satizábal, Cristina 485
Schaeffer, Satu Elisa 873
Schumacher, André 873
Selvakennedy, S. 588
Shen, Changxiang 404
Shepherd, Roderick 820
Shi, Ke 698
Shi, Weisong 304
Shin, Hyojeong 748
Shu, Wei 119
Shu, Yantai 392
Simplot-Ryl, David 143

Sobelman, Gerald 365
Sommer, Ch. 353
Song, Byunghun 760
Sun, Min-Te 171
Sun, Tony 724
Susilo, Willy 473, 521

Tang, Lei 543
Tang, Xueyan 662
Tang, Zongyao 281
Tao, Dan 256
Tei, K. 353
Thukral, Amandeep 509
Toiruul, Batbold 449
Tynan, Richard 820

Vaman, Dhadesugoor R. 197
Vu, Chinh T. 268

Wang, Dajin 579
Wang Jichun 650
Wang, Ling 220
Wang, Xiaodong 95
Wang, Xiaomin 638
Wang, Xu 650
Wang, Yang 650
Wang, Yuanyuan 72
Wattenhofer, Roger 1
Weber, Yves 1
Wei, Gang 736
Woo, Miae 861
Wu, Junmin 650
Wu, Min-You 119, 674
Wu, Shuhua 533
Wu, Wei 473
Wu, Xiaoxin 461
Wu, Yi 60

Xi, Feng 612
Xiao, Bin 600
Xie, Li 638
Xiong, Zhiqiang 209
Xu, Fang 404
Xu, Hongli 650
Xu, Ming 60

Yamashita, Shinji 808
Yang, Huabing 72
Yang, Junmo 171
Yang, Nan 84
Yang, Yongtian 220, 232
Yang, Zongkai 209
Yao, Yuxia 662
Yi, Sangho 760
Yoo, Kee-Young 416
Yoo, Sang-Jo 107
Youn, Joo-Sang 185
Yu, Wanrong 60

Zeng, Yulian 770
Zhang, Lianfang 392
Zhang, Qingquan 365
Zhang, Wenzhe 119
Zhang, Xiaofei 404
Zhang, Xing 404
Zhang, Xingyuan 72
Zhao, Feng 612
Zhao, Jing 232
Zhao, Zenghua 392
Zhou, Jipeng 38
Zhou, Lan 521
Zhou, Shuigeng 600
Zhou, Xingming 95
Zhu, Yuefei 533
Zou, Xukai 509