Context-aware Adaptive Opportunistic Routing in Mobile Ad-hoc and Sensor Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Zhongliang Zhao

von Hefei, China

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Context-aware Adaptive Opportunistic Routing in Mobile Ad-hoc and Sensor Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Zhongliang Zhao

von Hefei, China

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 26.06.2014

Der Dekan:
Prof. Dr. Silvio Decurtins

Abstract

Mobile ad-hoc networks (MANETs) and wireless sensor networks (WSNs) have been attracting increasing attention for decades due to their broad civilian and military applications. Basically, a MANET or WSN is a network of nodes connected by wireless communication links. Due to the limited transmission range of the radio, many pairs of nodes in MANETs or WSNs may not be able to communicate directly, hence they need other intermediate nodes to forward packets for them. Routing in such types of networks is an important issue and it poses great challenges due to the dynamic nature of MANETs or WSNs. On the one hand, the open-air nature of wireless environments brings many difficulties when an efficient routing solution is required. The wireless channel is unreliable due to fading and interferences, which makes it impossible to maintain a quality path from a source node to a destination node. Additionally, node mobility aggravates network dynamics, which causes frequent topology changes and brings significant overheads for maintaining and recalculating paths. Furthermore, mobile devices and sensors are usually constrained by battery capacity, computing and communication resources, which impose limitations on the functionalities of routing protocols. On the other hand, the wireless medium possesses inherent unique characteristics, which can be exploited to enhance transmission reliability and routing performance. Opportunistic routing (OR) is one promising technique that takes advantage of the spatial diversity and broadcast nature of the wireless medium to improve packet forwarding reliability in multihop wireless communication. OR combats the unreliable wireless links by involving multiple neighboring nodes (forwarding candidates) to choose packet forwarders. In opportunistic routing, a source node does not require an end-to-end path to transmit packets. The packet forwarding decision is made hop-by-hop in a fully distributed fashion.

Motivated by the deficiencies of existing opportunistic routing protocols in dynamic environments such as mobile ad-hoc networks or wireless sensor networks, this thesis proposes a novel context-aware adaptive opportunistic routing scheme. Our proposal selects packet forwarders by simultaneously exploiting multiple types of cross-layer context information of nodes and environments. Our approach significantly outperforms other routing protocols that rely solely on a single metric. The adaptivity feature of our proposal enables network nodes to adjust their behaviors at run-time according to network conditions. To accommodate the strict energy constraints in WSNs, this thesis integrates adaptive duty-cycling mechanism to opportunistic routing for wireless sensor nodes. Our approach dynamically adjusts the sleeping intervals of sensor nodes according to the monitored traffic load and the estimated energy consumption rate. Through the integration of duty cycling of sensor nodes and opportunistic routing, our protocol is able to provide a satisfactory balance between good routing performance and energy efficiency for WSNs.

Contents

1 Introduction **1**
1.1 Overview . . . 1
1.2 Problem Statement . . . 4
1.3 Thesis Contributions . . . 7
1.3.1 Opportunistic Routing Simulation Framework (ORSF) . . 8
1.3.2 Topology and Link-quality aware Geographic Opportunistic Routing (TLG) . . . 9
1.3.3 Context-aware Adaptive Opportunistic Routing Protocol (CAOR) . . . 9
1.3.4 Sensor Context-aware Adaptive Duty-cycled Opportunistic Routing Protocol (SCAD) . . . 9
1.4 Thesis Outline . . . 10

2 Mobile Ad-hoc and Wireless Sensor Networks **11**
2.1 Introduction . . . 11
2.2 Mobile Ad-hoc Networks (MANETs) . . . 12
2.2.1 Characteristics of Mobile Ad-hoc Networks . . . 13
2.2.2 Examples of Mobile Ad-hoc Networks . . . 14
2.3 Wireless Sensor Networks (WSNs) . . . 15
2.3.1 Components of Wireless Sensor Network Nodes . . . 16
2.3.2 Characteristics of Wireless Sensor Networks . . . 17
2.3.3 Requirements of Wireless Sensor Networks . . . 18
2.3.4 Applications of Wireless Sensor Networks . . . 19
2.3.5 Wireless Sensor Network Hardware Platforms and Operating Systems . . . 20
2.3.6 Periodic Duty Cycling . . . 23
2.3.7 Energy Profiling . . . 24
2.4 Wireless Link Variation . . . 25
2.5 Node Mobility . . . 25
2.6 Topology Control . . . 30
2.7 Conclusions . . . 30

3 Related Work **33**
3.1 Introduction 34
3.2 Routing in Mobile Ad-hoc Networks 34
3.2.1 Proactive Routing 35
3.2.2 Reactive Routing 36
3.2.3 Hybrid Routing 38
3.2.4 Geographical Routing 38
3.3 Routing in Wireless Sensor Networks 40
3.3.1 Flat Routing 40
3.3.2 Hierarchical Routing 41
3.4 Opportunistic Routing 42
3.4.1 Challenges of Opportunistic Routing Design 45
3.4.2 Routing Metrics 46
3.4.3 Coordination Mechanisms 46
3.4.4 Candidate list-based Opportunistic Routing Protocols . . . 52
3.4.5 Beacon-less Opportunistic Routing Protocols 55
3.4.6 Drawbacks of Existing Opportunistic Routing Protocols . 59
3.5 Context-aware Routing 62
3.5.1 Definition of Context 62
3.5.2 Context-aware Protocols 64
3.5.3 Consequences of Context-aware Routing 64
3.5.4 Context Combination Rules 65
3.6 Adaptive Communication Mechanism 66
3.7 Decision Making with Analytic Hierarchy Process 67
3.8 Duty Cycling in Wireless Sensor Networks 69
3.9 Software-based Energy Profiling in Wireless Sensor Networks . . 71
3.10 Simulation Framework 72
3.10.1 Types of Wireless Network Simulators 73
3.10.2 Simulation Framework 74
3.11 Conclusions 75

4 Opportunistic Routing Simulation Framework (ORSF) **77**
4.1 Introduction 77
4.2 Framework Description 78
4.2.1 Framework Architecture 78
4.2.2 Opportunistic Routing Module 79
4.2.3 Message Format & Data Structure 83
4.3 Framework Reconfigurability 83
4.4 Framework-based Simulation Study 86
4.4.1 Simulation Description and Evaluation Metrics 86
4.4.2 Simulation Results 88
4.5 Conclusions 95

5 Topology and Link Quality-aware Geographical Opportunistic Routing Protocol (TLG) 97
5.1 Introduction 97
5.2 Motivation 99
5.3 Protocol Description 99
5.3.1 Dynamic Forwarding Delay (DFD) 100
5.3.2 Link Validity Time Estimation 104
5.4 Performance Evaluation 105
5.4.1 Simulation Description and Evaluation Metrics 105
5.4.2 Simulation Results with Scalar Data Transmission 107
5.4.3 Simulation Results with Video Transmission 109
5.5 Conclusions 110

6 Context-aware Adaptive Opportunistic Routing Protocol (CAOR) 111
6.1 Introduction 111
6.2 Motivation 112
6.3 Protocol Description 113
6.3.1 Context Information 113
6.3.2 Dynamic Forwarding Delay (DFD) 113
6.3.3 Run-time Self-adaption of Context Weight by AHP 117
6.3.4 Reducing Packet Duplicates 120
6.3.5 Overhead of Context Awareness and the AHP Operation . 121
6.4 Performance Evaluation 122
6.4.1 Simulation Settings 122
6.4.2 Performance Analysis 122
6.5 Conclusions 127

7 Sensor Context-aware Adaptive Duty-cycled Opportunistic Routing for Wireless Sensor Networks (SCAD) 129
7.1 Introduction 129
7.2 Motivation 130
7.3 Protocol Description 130
7.3.1 Packet Forwarding Scheme in SCAD 131
7.3.2 Dynamic Forwarding Delay (DFD) 132
7.3.3 Adaptive Duty Cycling 133
7.3.4 Duty Cycling *vs.* Opportunistic Routing 135
7.4 Run-time Energy Profiling of Tmote Sky Mote with TinyOS . . . 136
7.5 Performance Evaluation using Simulation 139
7.5.1 Simulation Scenarios and Parameters 139
7.5.2 Simulation Results of the Static Scenario 140
7.5.3 Simulation Results of the Mobile Scenario 146
7.6 Performance Evaluation using Real-World Experiments 150
7.6.1 Technical Challenges in Real-World Implementation . . . 150
7.6.2 Small-scale Desktop Experiment 151

7.6.3 Small-scale Desktop Experiment Evaluation Results . . . 151
7.6.4 Large-scale Testbed Experiment 157
7.6.5 Large-scale Testbed Experiment Evaluation Results 158
7.7 Conclusions 175

8 Conclusions and Outlook **177**
8.1 Main Contributions 177
8.2 Outlook 179

9 Acronyms **181**

List of Publications **199**

Acknowledgment **203**

Curriculum Vitae **204**

List of Figures

1.1 MANETs example 2
1.2 Thesis contribution depicted in network stack 8

2.1 UAVNet as an example of MANETs 16
2.2 WSN architecture and components of sensor nodes 17
2.3 TMote Sky sensor node 20
2.4 BTnodes sensor node 21
2.5 Imote2 sensor node 21
2.6 Irregular radio range and resultant PDR distribution 26
2.7 Travel pattern of a mobile node using the Random Waypoint model 26
2.8 Travel pattern of a mobile node using the Random Walk model . . 27
2.9 Travel pattern of a mobile node using the Random Direction model 28
2.10 Travel pattern of a mobile node using the Reference Point Group Mobility model 28

3.1 GPSR protocol 39
3.2 Single-hop opportunistic routing 44
3.3 Multi-hop opportunistic routing 44
3.4 ACK-based coordination 48
3.5 Token-based coordination 49
3.6 Network coding-based coordination 51
3.7 RTS-CTS coordination 52
3.8 ExOR packet header 53
3.9 BLR working in the greedy mode 57
3.10 BLR working in the backup mode 58
3.11 Area division for the DDFD timer in BOSS protocol 60
3.12 Percentage of invalid priority list under mobility 61
3.13 An example of using AHP to choose a leader from three candidates 69

4.1 Opportunistic Routing Node Structure in OMNeT++ Simulator. . . 79
4.2 Flow Chart of Source Node in ExOR 85
4.3 Flow Chart of Forwarder Node in ExOR 85
4.4 Network topology of framework-based protocol implementation in OMNeT++ 87

4.5 Throughput of MORE, ExOR and OLSR under different channel transmission rate scenarios. 88
4.6 Network topology with different node densities. 89
4.7 Throughput of ExOR and OLSR at different node density. 90
4.8 Throughput of ExOR and OLSR with different number of disjoint routes from source to destination in Figure 4.9. 90
4.9 Network topology of different number of disjoint routes from source to destination. 91
4.10 Average number of collisions encountered at different nodes at Figure 4.9 with route number = 5. 92
4.11 Throughput of ExOR and OLSR under different transmission rate scenarios with different channel quality(Upper: good quality with α = 2 , Middle: medium quality with α = 3, Bottom: bad quality with α = 4.5) 93
4.12 Batch transmission delay of ExOR and OLSR under different channel quality (Upper: good quality with α = 2, Middle: medium quality with α = 3, Bottom: bad quality with α = 4.5) 94

5.1 Ad-hoc network deployment under emergency situation 98
5.2 Definition of candidate progress in the TLG protocol 103
5.3 Link validity estimation calculation of the TLG protocol 104
5.4 PDR and Goodput of different combinations of the TLG protocol . 108
5.5 PDR and goodput of three protocols of TLG, BLR, and GPSR . . 109
5.6 SSIM and VQM of three protocols of TLG, BLR, and GPSR . . . 110

6.1 Inefficient packet relays without mobility relevance of CAOR protocol 115
6.2 AHP hierarchy for next-hop selection in CAOR 117
6.3 CAOR handshake 121
6.4 PDR as a function of (a) Maximum speed; (b) Maximum pause interval; (c) Node density 123
(a) Maximum speed 123
(b) Maximum pause interval 123
(c) Node density 123
6.5 Influence of the node density on (a) Packet duplicates; (b) Average end-to-end delay; (c) Network lifetime 126
(a) Packet duplicates 126
(b) Average end-to-end delay 126
(c) Network lifetime 126

7.1 Packet forwarding scheme of SCAD protocol 132
7.2 State diagram of the duty-cycling adaptation scheme in SCAD protocol 134

7.3 Result of duty-cycle on opportunistic routing: (a) Example of duty cycle interval; (b) Probability of multiple receivers versus the node density and wake-up intervals . . . 137
(a) Duty cycle . . . 137
(b) Probability of multiple receptions . . . 137
7.4 Architecture of SCAD's energy profiling mechanism in TinyOS . 139
7.5 Results of (a) PDR; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the simulation evaluation in the ***static*** scenario 141
(a) PDR . . . 141
(b) End-to-end delay . . . 141
(c) Average hop-count . . . 141
(d) Network lifetime . . . 141
7.6 The effect of ***node density*** on (a) PDR; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***static*** scenario in the ***simulation*** evaluation . . . 144
(a) PDR . . . 144
(b) End-to-end Delay . . . 144
(c) Average hop-count . . . 144
(d) Network lifetime . . . 144
7.7 Simulation results of (a) PDR; (b) End-to-end delay; (c) Hop-count; (d) Network lifetime in the ***simulation*** *evaluation in* ***mobile*** *scenario* . . . 147
(a) PDR . . . 147
(b) End-to-end Delay . . . 147
(c) Average hop-count . . . 147
(d) Network lifetime . . . 147
7.8 The effect of ***node speed*** on (a) PDR; (b) End-to-end Delay; (c) Hop-count; (d) Network lifetime in ***mobile*** scenario in the ***simulation*** evaluation . . . 149
(a) PDR . . . 149
(b) End-to-end Delay . . . 149
(c) Average hop-count . . . 149
(d) Network lifetime . . . 149
7.9 Experiment topologies of SCAD . . . 152
7.10 CDF as function of (a) Throughput; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the ***Linear*** topology in the ***Small-scale Desktop*** experiment . . . 153
(a) Throughput . . . 153
(b) End-to-end delay . . . 153
(c) Average hop-count . . . 153
(d) Network lifetime . . . 153
7.11 CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Merging*** topology in the ***Small-scale Desktop*** experiment . . . 156

(a) Throughput 156
(b) End-to-end Delay 156
(c) Average hop-count 156
(d) Network lifetime 156
7.12 Linear, Merging, and Aggregation Network of Large-scale Testbed 159
7.13 Linear, merging, and aggregation topologies of large-scale testbed 159
7.14 CDF as function of (a) Throughput; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the ***Linear*** topology in the ***Large-scale Testbed*** experiment 160
(a) Throughput 160
(b) End-to-end delay 160
(c) Average hop-count 160
(d) Network lifetime 160
7.15 CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Merging*** topology in the ***Large-scale Testbed*** experiment 164
(a) Throughput 164
(b) End-to-end Delay 164
(c) Average hop-count 164
(d) Network lifetime 164
7.16 CDF as function of (a) Throughput; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the ***Aggregation*** topology in the ***Large-scale Testbed*** experiment 167
(a) Throughput 167
(b) End-to-end delay 167
(c) Average hop-count 167
(d) Network lifetime 167
7.17 Mesh Network of Large-scale Testbed 170
7.18 Mesh Topology of the Large-scale testbed 170
7.19 CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Static Mesh*** topology in the ***Large-scale Testbed*** experiment 171
(a) Throughput 171
(b) End-to-end Delay 171
(c) Average hop-count 171
(d) Network lifetime 171
7.20 CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Dynamic Mesh*** topology in the ***Large-scale Testbed*** experiment 174
(a) Throughput 174
(b) End-to-end Delay 174
(c) Average hop-count 174
(d) Network lifetime 174

List of Tables

2.1 Comparison of TinyOS and Contiki 23

4.1 Basic characteristics of the most representative opportunistic routing protocols 81
4.2 Basic Data Structures 84
4.3 Simulation parameters 86

5.1 Simulation parameters 105
5.2 Combinations of coefficients in equation (5.1) 106

6.1 Pairwise context importance 118
6.2 Threshold definition of context information 119
6.3 Random index (RI) 119
6.4 Baseline simulation parameters 122

7.1 Energy model of CC2420 138
7.2 Simulation parameters 140
7.3 MAC layer parameters of SCAD 140
7.4 CC2420 Transmission Power Level [131] 169

Preface

The following PhD thesis is based on work performed during my employment as a PhD student and Research Assistant at the Institute of Computer Science and Applied Mathematics (IAM) of the University of Bern, Switzerland. The research conducted in this thesis has been partially supported by the Swiss National Science Foundation project "Opportunistic Routing for highly Mobile Ad-hoc Networks (ORMAN)" (grant number: 200021-130211), and project "Mobile Multi-Media Wireless Sensor Networks(M3WSN)", a joint research effort funded by Nano-Terra.ch and Sino-Swiss Science and Technology Cooperation (SSSTC).

I would like to thank everybody who provided me with support, ideas, understanding, discussion, and encouragement during the course of my PhD thesis. First, I want to express my gratitude to Prof. Dr. Torsten Braun, head of the Communication and Distributed Systems group (CDS), who supervised and encouraged my work. He offered me the chance of working in his research group and provided an interesting and challenging work environment and the opportunity to participate in international research and technology projects.

I would also like to thank Prof. Dr. Eduardo Cerqueira for reading this work and providing valuable feedback and Prof. Dr. Oscar Nierstrasz, who was willing to be co-examiner.

Many thanks go to my colleagues at the institute and in our research group for being part of a great team. In particular, I want to thank Marc Brogle, Dragan Milic, Thomas Staub, Philipp Hurni, Markus Anwander, Gerald Wagenknecht, Desislava Dimitrova, Benjamin Nyffenegger, Peppo Brambilla, Carlos Anastasiades, Zan Li, Denis Rosario, Andre Gomes, Jonnahtan Saltarin, Islam Alyafawi, Imad Aad, Dima Mansour, Almerima Jamakovic-Kapic and Eirina Bourtsoulatze. Special thanks go to Ruth Bestgen and Daniela Schroth, the former and current secretary of the CDS research group, for their giant support in the administrative tasks during all these years.

I would also like to thank all the students, who contributed to this thesis in one way or another. In particular, thanks go to Bjoern Mosler and Daniel Weibel, who performed their Bachelor thesis and Master course project under my guidance.

I am deeply grateful to my family, especially my parents, for supporting me in many ways and being always patient during the years of my PhD thesis. This work is also to my beloved Yan, for her understanding and sharing a wonderful time with me. Last, I take this work as my eternal love for my grandparents.

Chapter 1

Introduction

The mobile ad-hoc networks (MANETs) paradigm was born with the idea of extending Internet services to the domain of mobile scenarios. In these types of networks, the network nodes (e.g., the users' laptops or mobile devices) communicate with each other to perform data exchange without the support of any existing network infrastructures. Direct communication only happens among users that are within the transmission range of each other by exploiting the wireless technologies in ad-hoc mode. Packet transmission between two distant node pair is performed in a hop-by-hop fashion with the help of intermediate nodes. Due to the characteristics of the wireless medium, packet routing over wireless radio is one of the most critical challenges in MANETs. Unstable wireless links and unexpected interferences from lossy network environments make data transmission in wireless multihop ad hoc network a challenging task.

This thesis focuses on packet routing problems in MANET, and its instance of wireless sensor network (WSN). We present our proposed opportunistic routing protocols to improve routing performance in lossy wireless mobile scenarios. This chapter briefly introduces the background of the topics that are covered by this work, discusses the fundamental challenges faced in the different parts of this thesis, and summarizes the main contribution of this work. In the end, this chapter outlines the structure of this thesis.

1.1 Overview

Wireless multihop networks, such as mobile ad-hoc networks and wireless sensor networks, have received an increasing amount of attention in the past decade due to their broad range of applications, and the easy deployment at low cost without relying on existing infrastructures. A mobile ad hoc network is a collection of mobile wireless nodes that can dynamically form a communication network without the help of any pre-existing infrastructure. The infrastructure-less and the dynamic feature of these networks demand new networking strategies to provide efficient end-to-end communication. The diverse application scenarios of these networks in many different situations such as battlefield, disaster recovery, emergency response, and industrial monitoring have made MANETs an interesting research topic. As illustrated in Figure 1.1, various types of sensor nodes (blue, green, and orange points in the figure) are deployed in different areas, which constitute multiple communication clusters. The clusters are interconnected via the cluster heads (red points in the figure), which can be either a based station or a command node.

Due to the intrinsic characteristics, such as node mobility, limited resources, interferences from the lossy environment, and the variations of wireless link quality, MANETs face a lot of problems to deliver satisfactory performance. Network protocol design in such networks presents great challenges mainly due to the following reasons. First, an important feature of wireless networks are time-varying channels caused by wireless channel propagation effects. For example, multi-path fading results in major fluctuations of signal quality, and thus, leads to intermittent network connectivity. Second, due to the fact that wireless medium has broadcast characteristics in nature, the transmission on one link may interfere with the transmissions on other neighboring links. Third, wireless embedded devices, such as sensor motes, are typically battery powered with limited energy. Battery lifetime imposes a strict limitation on the operation hours of the networks. Therefore, energy efficiency has been a critical concern in wireless sensor network protocol design. Fourth, wireless devices are usually constrained by limited on-board resources, such as computation power, memory, etc. All together these limitations make the routing protocol design in MANET and WSN a nontrivial task.

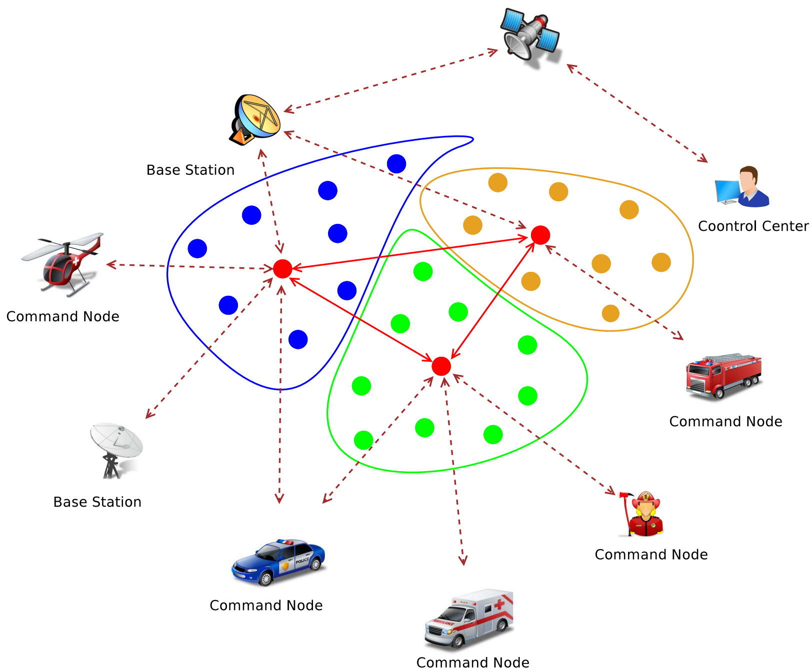


Figure 1.1: Examples of Mobile Ad-hoc Networks (MANETs).

Because routing in MANETs and WSNs is difficult, routing protocols received tremendous attention from research communities [17]. This has led to the development of different types of routing protocols for MANETs. Routing deals with finding appropriate paths between source and destination, possibly via intermediate relaying nodes. In wired networks, this is done by constructing a routing table at each node such that a packet is forwarded following the entries recorded in the table. Early-stage development of MANET routing follows the same idea and treats the wireless link as a wired one. It focused on finding a fixed end-to-end path before data transmission using different types of control messages. However, wireless networks are dynamic, and the selected path may be broken if the environment or network topology changes.

Traditional MANET routing can be classified into four categories: proactive routing, reactive routing, hybrid routing, and geographic routing. The criterion to distinguish the four different routing approaches is the mechanism they use to setup the routing table. In proactive routing, every node keeps an end-to-end path to any other nodes in the network. Routing tables have to be updated periodically independent of data transmission. This means that every node has to update its routing table even if it does not have any data to transmit. The advantage of this approach is that whenever a node wants to send packets, it already has the route information available in the routing table, and it just chooses the node stored in the table as next-hop. Therefore, it does not have to wait for finding a next-hop forwarder, and a packet encounters short delays. However, the weakness of this approach is also remarkable, since nodes have to remain updated with the latest topology by using periodic beaconing messages. The performance of this type of routing approach depends on the frequency to update the routing table. If the routing table is updated not as frequently as the topology changes, then the table will be outdated and the route information is invalid. This leads to transmission failure and degrades routing performance. However, if the routing table is updated frequently, additional control overhead has to be paid, which consumes considerable resources. Therefore, proactive routing can not perform well in case of node mobility.

Protocols based on reactive routing, alternatively called on-demand routing protocols, were designed to reduce the control overhead in proactive routing. They require the routing information only when a route is needed. In reactive routing, whenever a node wants to send packets, it starts a route searching process to build a path to the destination, and then transmits the packet over the calculated path. Compared to proactive routing, reactive routing does not require the periodic updates of routing tables, and thus, it can save resources. However, since a packet sender has to wait until the route information is ready, reactive routing usually has longer delays than proactive routing.

Hybrid routing can be regarded as an integration of both proactive and reactive routing. This approach is designed to increase the scalability by allowing nodes with close proximity to work together to form a backbone network to reduce the route discovery overheads. For example, most hybrid routing protocols are zone-based, which means the network is partitioned into a number of zones according to geographical information.

Independent of the mechanisms and the occasions of routing table updating, conventional MANET routing relies on an end-to-end path to perform data transmission. The data source has to be aware of all the individual hops in the path connecting the destination and itself prior to data transmission. Given the paths, packet transmissions are performed using unicast, which means only the nodes recorded in the routing table are the unicast receivers of the transmission and all other nodes will drop the packets even if they overhear the transmission and opportunistically receive them. However, wireless transmission is broadcast by nature. Therefore, the unicast feature of traditional MANET routing limits the capacity of wireless network and restricts the performance of wireless transmissions.

To overcome the drawbacks of using unicast for conventional MANET routing, opportunistic routing has been proposed to make use of the broadcast nature of wireless transmission. Opportunistic routing takes advantage of the broadcast nature and the spatial diversity of wireless transmissions to improve performance of wireless ad-hoc networks. Instead of using a predetermined path to send a unicast packet, opportunistic routing postpones the choice of the next-hop to the receiver side after packet transmissions at each hop. Multiple packet receivers of a broadcast transmission coordinate with each other and decide which one will be the real forwarder. The decision is carried out for each data packet, so the instantaneous radio conditions are taken into consideration to select the best relaying node.

1.2 Problem Statement

In this thesis, we investigate the feasibility of using opportunistic routing in MANETS and WSNs. We propose to combine context-aware communication with opportunistic routing in MANETs and WSNs, such that packet forwarders are selected based on multiple types of context information of MANETs or WSNs. Our contributions mainly tackle issues related to the following problem domains:

- ***Limited Energy Resources***: Energy efficiency is a challenge in the designs of MANET / WSN and their protocols, since network nodes are usually powered by battery with limited energy capacity. For a WSN and its applications, energy is a more critical issue than other constraints. The limited amount of battery determines the node and network lifetime. To increase the lifetime of a WSN, energy consumption of sensor nodes must be controlled strictly through the design of energy preserving mechanisms. The main energy consumption is due to the operations of different components of a wireless sensor node including onboard sensors, the micro-controller, and the wireless radio transceivers. To reduce their energy consumption, energy preserving mechanisms use sleep and wake-up scheduling technique to put individual component into an energy preserving sleeping mode as long as possible. A duty cycle is defined as the ratio of the wake-up period and the sleep period.

Therefore, reducing the duty cycle of the individual energy consuming component is the most efficient way of conserving energy. The radio transceiver is the most energy-hungry component of a sensor node, and its energy consumption in active mode is much higher than in sleeping mode. Therefore, the sleeping period of a sensor radio transceiver should be adapted to the current traffic load. Long sleeping periods ensure a low energy consumption at low traffic load, while short sleeping periods provide short data transmission delay and offer efficient routing performance at high traffic load. The MAC layer must be able to estimate the incoming traffic rates and make proper decisions about sensor sleep intervals.

- ***Unreliable Wireless Transmissions***: Wireless transmissions are unreliable by nature. This is because packets are broadcast and transmissions are subject to interferences or channel fluctuations. Radio irregularity is a non-negligible phenomenon in wireless communication. Wireless radio transmission ranges are normally irregular and resulting packet delivery ratio (PDR) distribution is nonuniform, which can significantly affect system performance. In WSNs, the low-power wireless channel is prone to a wide range of wireless phenomena, which may ultimately result in packet corruption and packet loss, such as high bit error rate due to multipath propagation, reflection and scattering effects, interferences with nearby nodes, etc. However, most MANET/WSN routing protocols do not consider this and they simply assume that the transmission range is a circle such that nodes within the radio range can always hear each other. Therefore, a realistic routing approach should take into account the instantaneous link quality at the moment of data transmissions when selecting packet forwarders.

- ***Efficient Routing Requirements***: In real world applications, MANETs or WSNs are usually deployed for specific purposes, such as environmental monitoring or fire detection. In all these applications, the deployed networks must be able to deliver the captured information to end users in an efficient and economic way. However, with limited onboard resources and the dynamic nature of wireless environments, efficient packet transmissions in MANETs or WSNs are very difficult. Traditional routing protocols in MANETs or WSNs treat the wireless link as a wired one, and focus on finding a fixed end-to-end path between a source-destination pair. Data transmission is unicast to specific nodes that are recorded in the path. However, the selected path will be invalid if any transmission on the path fails, which is quite often in the presence of channel variation and node mobility. When an error occurs, the source node has to find a new path or has to retransmit the packet, which significantly reduces routing performance. To mitigate this problem, opportunistic routing (OR) has been proposed to cope with the unreliability feature of wireless link and make use of the broadcast nature of wireless transmissions. In OR, a source node does not use fixed nodes as packet forwarders. Instead, it broadcasts the packet and lets the multiple receivers coordinate and choose one node to actually forward the packet. In this way, OR postpones the forwarder selection to the receiver side, and the real forwarder is selected from the nodes that have successfully received the packet.

- ***Problems of Existing Opportunistic Routing Protocols under Dynamics***: Many OR protocols have been proposed, and most of the existing OR protocols choose the next-hop forwarder based on a predefined candidate list. The candidate list includes a list of nodes that might be the forwarder of a packet (referred to as candidates), and it is calculated prior to data transmission according to certain network metrics, such as expected transmission count (ETX) [48] or expected any-path transmissions (EAX) [159]. Most of the research efforts in OR have been given to the selection and ranking process of candidates. However, existing OR protocols did not fully consider the unreliability of wireless transmissions, and most of them assume that the transmission will succeed when two nodes are within the radio range of each other. In reality, wireless links are unstable, as they often experience quality fluctuation and distortion due to interference. Therefore, the list-based OR features restrict the freedom of opportunism, since only the listed nodes can compete for packet forwarding. Additionally, the list is statically generated based on a single metric prior to data transmission, which is not appropriate for the dynamic feature of MANETs. For example, MANET nodes are mobile, which leads to constantly changing topology. Therefore, the candidate list, which is built based on an outdated network topology, will be invalid.

- ***Context-aware Adaptive Communication***: MANETs and WSNs are composed of large numbers of nodes. Different nodes have their own parameters, which determine nodes' behaviors and further affect network performance. The values of these parameters should be considered in the packet routing process. However, most of the existing MANET/WSN routing protocols choose forwarders based on a single metric, such as hop-count, link quality, or distance progress. The single metric-based routing approach has the drawback of fast resource depletion along the preferred paths. Additionally, most of the existing protocols of MANETs or WSNs are statically configured. Their protocol parameters are of static configuration with fixed values at compile-time and thus are not able to cope with varying network conditions at run-time. Context-aware adaptive communication enables network nodes to consider multiple types of context information simultaneously, and be aware of the updated statues of themselves and the environments. This should enable nodes to make optimal routing decisions.

1.3 Thesis Contributions

The main contributions of this thesis include a context-aware adaptive opportunistic routing protocol for MANETs, and its derivation for WSNs, a sensor context-aware duty-cycled opportunistic routing protocol. We also design a simulation framework for developing opportunistic routing protocols. The contributions of this thesis can be summarized as follows:

Opportunistic Routing Simulation Framework (ORSF) The first contribution is a network simulation framework for opportunistic routing, which abstracts the core steps of a general opportunistic routing scheme [155] [153]. The framework abstracts the main procedures of opportunistic routing, and facilitates the developments of novel protocols.

Topology and Link-quality aware Geographic Opportunistic Routing (TLG) The second contribution of this thesis is an opportunistic routing protocol for mobile ad-hoc networks. TLG [157] is a beaconless opportunistic routing protocol. Different from most of the existing opportunistic routing protocols, TLG does not rely on any candidate list, and it enables all the qualified nodes to participate in packet forwarding. TLG chooses packet forwarders based on three types of context metrics: network topology, link quality, and geographical location of nodes. By considering the mobility information of neighboring nodes, TLG is able to make use of the relative movements of connected nodes to choose packet forwarders.

Context-aware Adaptive Opportunistic Routing protocol (CAOR) The third contribution of this thesis is an extension of the TLG protocol. CAOR is a novel opportunistic routing approach that combines the concept of context awareness with the adaptive communication mechanism for MANETs. On the network layer, we design and implement a Context-aware Adaptive Opportunistic Routing protocol (CAOR) [154] [151]. Our protocol takes into account multiple types of cross-layer network context information. The context information we consider covers a wide range, which includes residual energy, energy drain rate, geographical location, link quality, and estimated traffic load at each node. The adaptation of context weights enables network nodes to change their behaviors according to the latest values of the context metrics. The performance of our design has been evaluated through extensive simulations.

Sensor Context-aware Duty-cycled opportunistic routing protocol (SCAD) Our fourth contribution is the derivation of our opportunistic routing protocol for wireless sensor networks. The proposed Sensor Context-aware Duty-cycled opportunistic routing protocol (SCAD) takes energy as a primary concern, and its MAC layer adaptive duty-cycling mechanism can adjust the duty-cycles of sensor nodes according to traffic loads and energy drain rates [150] [149]. SCAD integrates duty-cycle with opportunistic routing in WSNs. Our goal is to achieve a good balance between performance and energy efficiency in WSNs. We implement our protocol in a real-world WSN testbed environment and evaluate its performance. To provide context awareness of energy, we also implement a run-time energy profiling mechanism to get the real-time energy information.

Figure 1.2 presents the contributions of this thesis, embedded in the wireless multihop communication stack. The individual contributions of this thesis are described as follows:

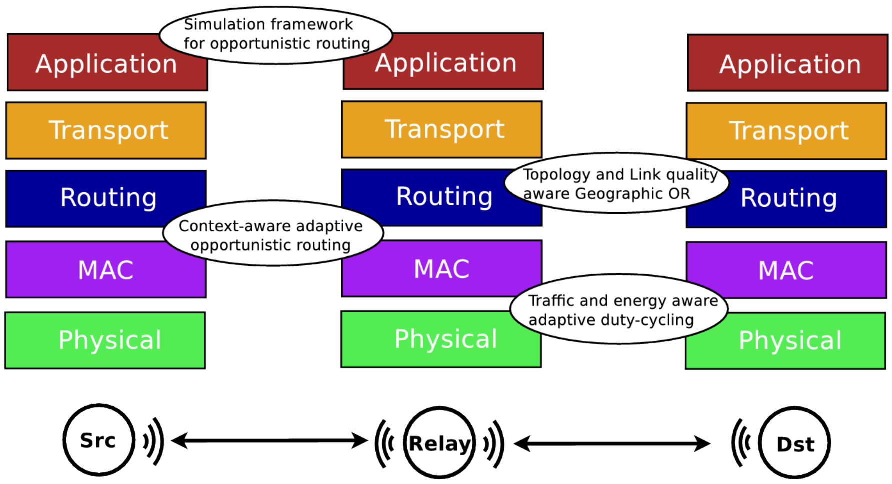


Figure 1.2: Thesis contributions depicted in the multihop wireless communication stack.

1.3.1 Opportunistic Routing Simulation Framework (ORSF)

Opportunistic routing has been proposed to improve performance of wireless multihop ad-hoc networks. Many protocols have been proposed and validated to show their functionalities. However, most of them are implemented and evaluated with specific simulators, and the implementations of different protocols have to be done individually. To simplify this work, we design a framework for simulating opportunistic routing protocols in the INETMANET framework [7] of the OMNeT++ simulator [138]. The proposed modules adopt an abstraction of the generic functions of the most representative opportunistic routing algorithms. The main contribution is an OMNeT++ modeling architecture that could be extended to implement different opportunistic routing schemes. This contribution provides an analysis of the most representative opportunistic routing algorithms. We decouple opportunistic routing into four procedures - *Forwarder Candidate Selection*, *Forwarder Selection*, *Forwarder Role Change Notification* and *Collision Avoidance*. Different protocols will have specific implementations of each procedure. In the framework, these four procedures are defined as virtual functions and act as implementation stubs such that different protocols could be implemented by overriding them in the derived function according to their distributed strategies.

The framework is used to validate the performance of representative candidate list-based opportunistic routing protocols and to identify their drawbacks in the presence of node mobility or network topology variation. Details of the simulation framework can be found in chapter 4.

1.3.2 Topology and Link-quality aware Geographic Opportunistic Routing (TLG)

Our first protocol contribution is called TLG - ***T**opology and **L**ink-quality aware **G**eographic opportunistic routing* for MANETs. TLG is a beaconless geographic opportunistic routing protocol. It does not rely on any candidate list, and it enables all the qualified nodes to participate in the packet forwarding process. TLG chooses packet forwarders based on three types of context metrics: network topology, link quality, and geographical location of nodes. By considering the mobility information of neighboring nodes, TLG is able to make use of the relative movements of connected nodes to choose packet forwarders. The information of neighbor movements are disseminated through embedding the mobility information into the headers of data packets. Therefore, no additional communication overhead is needed to get this information. We compare TLG with well-known existing geographic routing and beaconless routing solutions. Simulation results show that TLG outperforms other protocols in terms of both quality-of-service (QoS) and quality-of-experience (QoE) metrics. Details of the TLG protocol can be found in chapter 5.

1.3.3 Context-aware Adaptive Opportunistic Routing Protocol (CAOR)

We further extend our previous work of TLG into a generalized concept of context-aware opportunistic routing. We propose CAOR - ***C**ontext-aware **A**daptive **O**pportunistic **R**outing* for MANETs. Compared to TLG, CAOR has more flexibility and it takes more context information into account, not only limited to topology, location, and link quality. CAOR is a more generic protocol, which can be configured to include any new context as far as it has impact on the routing decision. Another important improvement is that CAOR supports run-time adjustment of the weights of context information based on their instantaneous values. This is achieved by applying the Analytic Hierarchy Process (AHP) theory [123]. The adaptivity feature enables CAOR to adapt the protocol behavior at run-time. Moreover, CAOR uses an active suppression mechanism to reduce packet duplication, which is an important issue that is not covered by TLG. Simulation results show that CAOR can provide efficient routing performance in highly mobile environments. The adaptivity feature of CAOR is also validated. Details of the CAOR protocol can be found in chapter 6.

1.3.4 Sensor Context-aware Adaptive Duty-cycled Opportunistic Routing Protocol (SCAD)

In this contribution, we move from MANETs to WSNs. We implement our opportunistic routing protocol in wireless sensor nodes, and perform real-world testbed evaluations. To do this, several modifications and adaptations of the previous protocols had to be made due to the specifics of WSNs. Energy is of primary importance in wireless sensor networks. Sensor motes are battery-powered with scarce energy.

Low power transmission makes the wireless links unstable and unreliable, which further leads to variant network topologies. Packet retransmissions aggravate the waste of energy. A beacon-less approach, such as opportunistic routing chooses the forwarder after packet transmissions, and is thus promising for WSNs. In this part, we introduce our protocol of SCAD - ***S**ensor **C**ontext-aware **A**daptive **D**uty-cycled* beaconless opportunistic routing protocol for WSNs. SCAD adapts the concept of beaconless OR into the domain of WSNs and selects forwarders based on multiple cross-layer network context information. To save energy, SCAD adapts duty-cycles of sensor nodes based on real-time traffic loads and energy drain rates. Real-world evaluation on TelosB nodes shows that SCAD outperforms other protocols in terms of both throughput and network lifetime. A real-time energy profiling mechanism has been designed and implemented in TinyOS operating system running on top of TelosB nodes to provide energy awareness. The detailed information about SCAD is presented in chapter 7.

1.4 Thesis Outline

The thesis is structured as follows. In Chapter 2, we describe the background knowledge of this thesis. Concepts of mobile ad-hoc networks (MANETs) and wireless sensor networks (WSNs) will be introduced. Their individual characteristics will also be discussed. In Chapter 3, we investigate the works of others that are correlated with the concepts covered by the thesis, which include the latest developments of MANET/WSN routing protocols, opportunistic routing protocols, context-aware routing, and network simulation frameworks. Chapter 4 presents our contribution of a simulation framework of opportunistic routing. In Chapter 5, we discuss TLG, our proposed topology and link-quality aware geographical opportunistic routing for MANETs. Chapter 6 describes our extension work of TLG, namely a novel concept of opportunistic routing, called CAOR - Context-aware Adaptive Opportunistic Routing for MANETs. Chapter 7 discusses SCAD - Sensor Context-aware Adaptive Duty-cycled opportunistic routing protocol. SCAD is a prototype instantiation of our proposed opportunistic routing protocol on wireless sensor nodes. Finally, Chapter 8 concludes the thesis by summarizing the contributions of this work, and discusses interesting and promising future directions of research.

Chapter 2

Mobile Ad-hoc and Wireless Sensor Networks

2.1 Introduction

This chapter introduces some basics of the network environments where the thesis contributions were conducted: Mobile Ad-hoc Networks (MANETs) and Wireless Sensor Networks (WSNs). Relevant issues are also included in this chapter.

A wireless ad-hoc network is a decentralized type of wireless network. The “ad-hoc” property means that the network does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in infrastructure-based wireless networks. Instead, each node in wireless ad-hoc network participates in the routing process by forwarding data for other nodes, such that the determination of which node will forward the packet is made dynamically on the basis of the real-time network connectivity. Wireless ad-hoc networks have many different instances, and different examples have their characteristics.

Mobile ad-hoc networks are ad-hoc networks consisting of mobile nodes that can dynamically form a network without any pre-existing infrastructure. In a MANET, mobile nodes move either according to pre-defined mobility patterns or randomly. Node mobility makes the network topology constantly changing over time, which brings new challenges to the design of efficient routing protocols. A WSN is a concrete example of wireless ad-hoc networks. It consists of wireless sensor nodes, which can be static or mobile. Wireless sensor nodes have more strict constraints of resources, such as computation power, energy, and wireless radio transceiver. These limitations make routing in WSNs a challenging task.

Wireless sensor networks usually comprise a large set of unattended devices being able to monitor and control the phenomena in remote wide areas. WSNs enable a wide range of applications, in both military and civilian areas. These applications require autonomous operation of the sensor devices after deployment. The sensor devices are featured by the constrained hardware limitations: sensors, wireless transceiver, micro-controller, and battery. All these constraints make the collection and dissemination of the monitored information a hard task.

The description of MANETs and their developments are presented in section 2.2. In section 2.2.1, we describe the features of MANETs. Its examples and applications are presented in section 2.2.2. A WSN is an example of a wireless mobile ad-hoc network, but has many specific characteristics due to its design constraints. Therefore, these features have to be fully analyzed in order to design efficient WSN protocols. We list the WSN characteristics in section 2.3.2. The requirements of a WSN are discussed and presented in section 2.3.3. A various number of application fields of WSNs are introduced in section 2.3.4. The existing WSN hardware platforms and operating systems are described in section 2.3.5. Among the various constraints of WSNs, energy is the most critical one, since WSNs nodes are usually battery-powered and have limited energy capacity. Therefore, periodic duty cycling is the most common approach to save energy by turning off the radio transceiver periodically. The developments of periodic duty cycling are introduced in section 2.3.6. In a wireless environment, the network topology may change frequently due to the variation of wireless links or node mobility. Wireless transmission has the feature of unstable links among nodes, because the environment is subject to multiple types of interferences or collisions. Therefore, in section 2.4 and section 2.5, we briefly discuss the issues of link variation and node mobility in a wireless environment. When the network topology changes, it is important for the network nodes to notice the changes and response properly. In section 2.6, we describe the issue of topology control in mobile ad-hoc and wireless sensor networks.

2.2 Mobile Ad-hoc Networks (MANETs)

Mobile ad-hoc networks were originally derived from the requirements of military applications. For example, in a battlefield application, where there is no fixed networking infrastructure, soldiers and assets are mobile and they have a need for voice, data, and video communications with each other. Since assets are mobile, traditional IP routing using static routing tables is not suitable for creating networks. Dynamic routing that is used in fixed networks will not work well in the highly mobile environment of the battlefield, because network convergence speed is too slow to support real-time communication requirements.

Without any fixed networking infrastructure, a MANET has to be created "on-the-fly". A MANET is a self-configuring, infrastructure-less network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction and therefore will change its links to other devices frequently. Each node must be willing to participate in forwarding traffic even though the packet is not of its interest. The fact that these networks are self-forming and self-healing facilitates the deployment process and minimizes the need for manual configuration and intervention. MANETs support multi-hop networking to extend coverage and provide redundant paths for increased resilience.

2.2.1 Characteristics of Mobile Ad-hoc Networks

A mobile ad-hoc network has many distinct characteristics, which are mainly due to the fact that it lacks any centralized infrastructures. Additional reasons include:

- Time-varying channel caused by wireless channel propagation: nodes located within the radio range of each other can establish a network connection without any pre-configuration or manual intervention.
- Wireless medium is broadcast in nature, the transmission on one link may interfere with transmissions on other neighboring links.
- Wireless embedded devices are usually constrained by limited resources.

All these limitations make protocol design in MANETs a hard task. In the following, we list the main characteristics of MANETs.

- ***Self-forming***: Nodes located within the radio range of each other can establish a network connection without any pre-configuration or manual intervention.
- ***Self-healing***: Nodes can joint or leave rapidly without affecting the operation of remaining nodes. An important requirement of MANET is its fault-tolerance, such that the failure of any node will not significantly degrade performance.
- ***No Infrastructure***: In a mobile ad hoc network, mobile nodes form their own network autonomously. All the nodes within the network have the same role. They are both packet source and forwarder. There is no centralized control of network operation.
- ***Peer-to-peer communication***: Traditional networks typically support end systems operating in client-server mode. In a MANET environment, mobile nodes can communicate and exchange messages without prior arrangement and without reliance on centralized resources.
- ***Highly dynamic network topology***: Mobile nodes are in continuous movement. They move either following certain mobility patterns, or just randomly. The variation of wireless link also changes the connectivity among nodes. The network topology of a MANET is constantly changing over time.
- ***Limited resources***: Some or all of the network nodes are suffering from restricted limitation of energy, computation power, memory, etc.
- ***Limited bandwidth***: Wireless links have significantly lower capacity than infrastructure-based networks. In addition, the realized throughput of wireless communications - after accounting for the effects of multiple access, fading, noise, and interference conditions, etc., is often much less than a radio's maximum transmission rate.

- ***Security threats***: Mobile wireless networks are generally more prone to physical security threats than fixed cable networks. The increased possibility of eavesdropping, spoofing, and minimization of denial-of-service type attacks should be carefully considered in a real system deployment.

2.2.2 Examples of Mobile Ad-hoc Networks

The emergence of low-cost portable devices such as smartphones has led to an increasing research interest in MANETs, where every person, vehicle, or user is able to communicate with neighbors via short-distance wireless radio transceivers. Such a communication paradigm offers multiple advantages: low starting costs, rapid development, resilience to disruption, and high bandwidth.

Although general purpose MANETs may not yet be widespread, specialized networks are already a reality. In the following, a brief description of some typical networks, which are based on the concept of MANETs are presented. These networks are concrete instances of MANETs in certain specific application domains. Therefore, they have different features and also have different application requirements.

- ***Delay Tolerant Networks (DTNs)***: DTNs [56] have been proposed and used to provide connectivity in areas where a fully connected network is not always available. Examples of such networks are those operating in mobile or extreme terrestrial environments.

- ***Vehicular Ad-hoc Networks (VANETs)***: VANETs [146] are used for onboard safety systems, virtual traffic signs, real-time congestion and traffic information, and commercial applications, which require vehicle-to-vehicle or vehicle-to-roadside communication. Vehicle ad-hoc networks have some distinct features compared to other mobile ad-hoc networks such as large computational and infinite power resources. The mobility of the nodes may be quite high, but with mobility patterns constrained to roadways. VANETs applications usually have more strict requirements on packet transmission delay, since a late alarm message is not acceptable for safety.

- ***Wireless Sensor Networks (WSNs)***: WSNs [22] have been proposed to rapidly deploy low-cost, low-power wireless motes in a target area. In practice, WSNs have been used in many industrial applications, such as industrial monitoring, environmental monitoring, or animal monitoring. The data collected and often already partially processed by the sensors is transmitted to the destination node, which is controlled by a gateway node or monitoring center. Wireless sensor networks have very strict limitations on battery level, communication and computation capacities, and memory spaces. Due to the fact that sensors are battery powered, power consumption is the major concern in a wireless sensor network application.

- ***Wireless Multimedia Sensor Networks (WMSNs)***: WMSNs [27] are new types of sensor networks gaining research interest due to the availability of low-cost and mature technologies in camera sensors and scalar sensors. As an extension of traditional scalar wireless sensor networks, WMSNs are composed of wirelessly interconnected sensor nodes equipped with multimedia devices, such as cameras and microphones, and are capable to retrieve video and audio streams, still images, as well as scalar sensor data. WMSNs can visually observe the physical behaviors of the objects in the targeted areas, which significantly enriches the application ranges of wireless sensor networks. The support of multimedia transmission in WMSNs provides additional information to evaluate the network performance from the perspectives of the end users.

- ***Personal Area Networks (PANs)***: PANs [11] are short-range, localized networks where nodes are usually associated with a given person. These nodes could be attached to someone's pulse watch, belt, and so on. In these scenarios, mobility is only a major consideration when interaction among several PANs is necessary, illustrating the case where, for instance, people meet in real life.

- ***Unmanned Aerial Vehicular Networks***: In case of disasters of emergent conditions, existing communication infrastructures may be broken and become unavailable. To facilitate the necessary operation in this kind of scenarios, it is important to deploy quickly a temporary communication network to assist the rescue operation. An unmanned aerial vehicular network can be set up and deployed into the dangerous area to form an unmanned aerial vehicular networks (UAVNet) [99] or Flying Ad-hoc Network (FANET) [120] to perform rescue tasks, such as shown in Figure 2.1. The flying UAVs carrying sensor devices form a flying wireless sensor/ad-hoc network.

2.3 Wireless Sensor Networks (WSNs)

Due to recent technologies advances, the manufacturing of small and low lost sensors became technically and economically feasible. The sensing electronics measure the ambient conditions related to the environment surrounding the sensor and transform them into an electric signal. Processing such a signal reveals some properties about objects located or events happening in the vicinity of the sensor. A larger number of disposable sensors can be networked together in many applications that require unattended operations. A wireless sensor network consists of a certain amount of sensor nodes. These sensors have the ability to communicate either among themselves or directly to an external base station. A larger number of sensors allows for sensing over larger geographical regions with higher accuracy and better coverage.

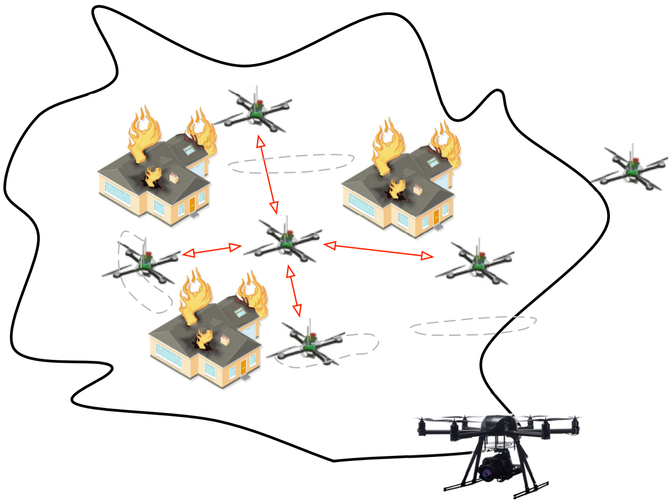


Figure 2.1: An UAVNet deployment for rescue purpose under disaster scenarios.

2.3.1 Components of Wireless Sensor Network Nodes

Figure 2.2 shows the schematic diagram of a wireless sensor node and its components. Basically, each sensor node has typically several components: a wireless radio transceiver with an internal antenna, a micro-controller, an electronic circuit for interfacing with sensors, a certain number of sensors, and an energy source. Size and cost constraints on sensor devices result in corresponding limitations on resources such as energy, memory, computational power, and communication bandwidth. The functionalities and roles of different components in the operation of a sensor node are listed below.

- ***Wireless radio transceiver***: transmits a bit- or byte stream as radio signals. When receiving the signals, the radio transceiver converts it back into a bit/byte stream for further processing. This is the most energy consuming component of a sensor node. Therefore, the operation of packet transmission and reception should be carefully controlled to reduce energy consumption. Many MAC layer protocols have been proposed in this area, the main idea is to switch the radio transceiver completely off whenever possible to put the sensor into sleep mode to save energy. The next section gives more details about this mechanism.

- ***Micro-controller***: is a general purpose processor, and it is optimized for embedded applications with low power consumption. The most popular choice is the MSP 430 from Texas Instruments [8].

- ***Memory***: is for general purpose data storage.

- ***Sensors***: are for sensing the physical phenomena. There are many types

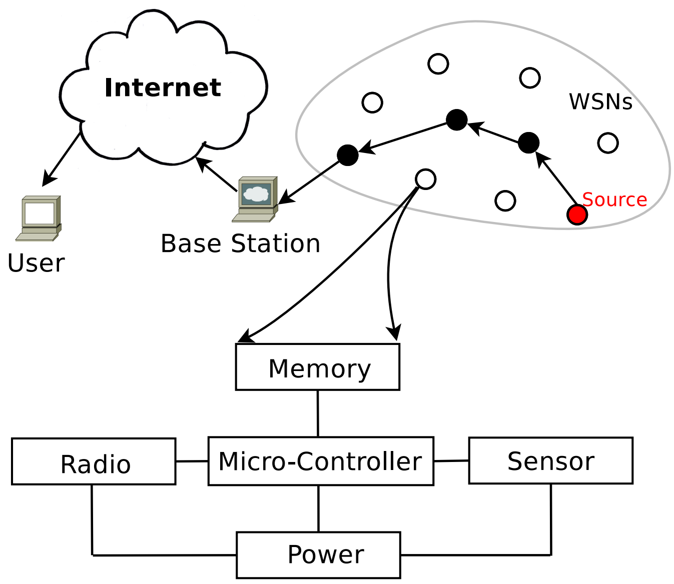


Figure 2.2: WSN architecture and components of a wireless sensor node.

of sensors. Omni-directional sensors include light, thermometer, pressure, etc. Depending on the application requirements, sensors can be added or removed from the board.

- ***Power***: One of the major limitations on performance and lifetime of WSNs is the limited capacity of these finite power sources, which must be manually replaced when they are depleted. Recent works of [82] [79] have explored scenarios in which nodes can harvest energy from their environment (e.g., from the sun) and use it to recharge their batteries. In the absence of such energy (e.g., at night in the case of solar energy), nodes can then subsist on their replenished battery supply.

2.3.2 Characteristics of Wireless Sensor Networks

A wireless sensor network is an instance of a wireless ad-hoc network. Therefore, it has all the features of MANETs presented above. Besides, wireless sensor networks have their own characteristics. In the following, we list some of these characteristics.

- ***Resource constraints***: The wireless sensor nodes are of small size and carry very limited resources. They are usually battery-powered, and the limited battery level constrains the operation of WSNs and poses high requirements for efficient protocol design. The energy consumption of a wireless sensor node is caused by multiple components: wireless radio transceiver, micro-controller computation operation, memory access, etc. An efficient WSN protocol, independent of which network stack layer, should take energy efficiency as the first concern. Other resource limitations also bring difficulties to WSN protocol design, such as very few memory space, low computation power, etc.

- ***Low-power wireless transceiver***: Wireless sensor nodes carry low-power wireless transceivers for communication. The traditional problems associated with wireless channels (e.g., fading, high error rate) may significantly affect the operation of sensor networks.

- ***Heterogeneity of nodes and links***: In a real-world deployment of a wireless sensor network, different types of sensor nodes will be configured to work together. They may have different capacities and play different roles in the system. The existence of heterogeneous sensors raise many technical issues related to data transmission. For example, some applications might require a diverse mixture of sensors for monitoring temperature, pressure and humidity of the surrounding environment. Different types of sensor nodes might work at different traffic rates, subject to diverse quality of service constraints, which brings great challenges to deliver a satisfactory performance.

2.3.3 Requirements of Wireless Sensor Networks

In this section, we summarize the main WSN requirements affecting multihop routing protocols. These requirements often are interrelated, even though sometimes the performance of one requirement is in conflict with another one. For instance, the increment of security produces the reduction of the energy efficiency.

- Energy efficiency is the most critical requirement. Sensor nodes are placed in distant positions, where usually no additional recharging is possible. Thus, nodes using batteries must operate autonomously during months or even years. The battery lifetime is determined by the power consumption of main components (i.e., processing, sensing, and communication). These components are disabled during sleeping states in order to minimize the power consumption. However, wireless communication consumes the majority of the energy, in particular the packet transmission is the most energy-hungry part. To maximize the WSN lifetime, efficient routing protocols must minimize the number of packet transmissions.

- Scalability is a specific property of WSNs where thousands or hundreds of nodes are deployed to sense a target area. For the limited sensing coverage of nodes, the density of neighbor inside the same radio range is from tens to hundreds. This factor requires distributed protocols where nodes take routing decisions using local neighborhood information.

- Topology is a relevant factor for WSNs even in applications where nodes are stationary. In static WSNs, the topology may change by adding or removing nodes. Moreover, topology changes happen because of node failures due to physical damages or power lack. Another important factor affecting the network topology is the fluctuation of link qualities between two connected nodes. These topology changes break multihop paths and damages routing protocols, and thus reduce their performance in terms of latency, efficiency, and reliability.

- Connectivity is of great importance in dense WSNs. The connectivity is defined as the capacity of establishing communication between any two individual nodes. To achieve connectivity among all nodes, WSNs require localized communication protocols avoiding the overhead of route maintenance techniques and flooding discovery mechanisms.

- Security is a key aspect in some WSN applications such as Intrusion Detection System (IDS) where the delivery of warning message is essential. An attack tries to avoid the proper operation of IDS by interfering warning messages. To guarantee multihop communications in the presence of attackers, many cryptographic algorithms need high resources in terms of computations, energy and bandwidth.

2.3.4 Applications of Wireless Sensor Networks

This section describes some applications of wireless sensor networks. The range of applications scales from habitat monitoring at home automation to traffic control and health care, from civil engineering to military war fields. Some emerging application scenarios include home monitoring and water monitoring.

- ***Home, civil and environment engineering applications***: Home control applications provide flexible management of lighting, heating, and cooling systems from anywhere in the home. Civil applications can enable the extension and upgrading of building infrastructure with minimal efforts. Sensor networks can be deployed in remote areas to monitor environmental conditions such as micro-climate changes, volcanic and seismic activities.

- ***Industrial applications***: Specific applications for industrial and commercial spaces include monitoring of warehouses, fleet management, factories, assembly lines, work-flow, inventory, and material processing systems (chemicals, cooling, gas flow).

- ***Medical applications***: One of the most important and rapidly growing application areas is the application in the medical field, e.g., pre-hospital and in-hospital emergency care, disaster response, and stroke patient rehabilitation. Patient and doctor tracking systems allow home monitoring for chronic and elderly patients and provide long-term care facilitation and trend analysis.

2.3.5 Wireless Sensor Network Hardware Platforms and Operating Systems

A wide variety of custom sensors and off-the-shelf sensors have been designed for both academic and industrial applications. In this section, we briefly summarize the latest developments of hardware and software in the domain of wireless sensor networks.

Wireless Sensor Network Hardware Platforms

The following is a selection of the state-of-the-art of the wireless sensor network node platforms currently deployed in research and commercial applications.

- ***TMoteSky***: The Tmote Sky mote [15], as shown in Figure 2.3, is a general purpose wireless sensor network platform with a large market share both in academia and industry. Tmote Sky is one of the few FCC Certified wireless sensor network platforms available on the market. The TelosB node consists of a MSP430 F1611 low-power micro-controller [88] from Texas Instruments with 10kB RAM, 48kB+256B flash ROM and 1024k serial storage, on-board humidity, temperature and light sensors. A CC2420 radio transceiver [91] from Chipcon is used as radio module. Operating elements for diagnostics are a button and three LEDs.

Figure 2.3: TMote Sky sensor node.

- ***BTnode***: The BTnode [2], as shown in Figure 2.4, is a versatile, autonomous wireless communication and computing platform based on Bluetooth radio, a second low-power radio and a micro-controller. The BTnode rev3 is a dual radio device with a CC1000 low-power radio module and a Zeevo ZV4002 Bluetooth radio module. The Bluetooth system supports up to four independent piconets and seven slaves. The BTnode can operate both radios simultaneously or shut them down independently when not in use. Operating elements for diagnostics are four LEDs. The micro-controller is an Atmel Atmega 128L with 64+180 Kbyte RAM, 128 Kbyte FLASH ROM and 4 Kbyte EEPROM.

- ***JN5148***: The JN5148 mote [80], as shown in Figure 2.5, developed by Jennic company, is a new product for high performance sensor node on a single low-cost chip. The chip integrates a powerful 32-bit RISC microcontroller

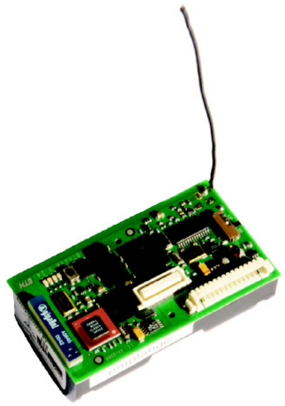


Figure 2.4: BTnodes sensor node.

with large memory 128 KB ROM and 128 KB RAM. Its radio transceiver supports a IEEE 802.15.4 compliant and a 667 kbps data rate at 2.4 GHz band with a ultra-low consumption below 18 mW. The JN5148 chip represents the future generation of wireless sensor nodes that will be manufactured and deployed extensively.

Figure 2.5: JN5148 sensor node.

Wireless Sensor Network Operating Systems

Besides the physical components, a critical step towards achieving the vision behind wireless sensor networks is the design of a software architecture that bridges the gap between raw hardware capabilities and a complete working system. The demands placed on the software of wireless sensor networks are numerous. It must be efficient in terms of memory, processor, and power so that it meets the strict application requirements. It must also be agile enough to allow multiple applications to simultaneously use the system resources such as communication, computation and memory. The extreme constraints of these devices make it impractical to use legacy systems.

In this section, we discuss the operating systems that are specifically designed for wireless sensor networks, tolerating all the device constraints listed before. We focus on two representatives of Contiki and TinyOS, which are the most popular operating systems on today's WSN.

- ***TinyOS***: The TinyOS operating system [89] is an event-based operating system for sensor networks. The system is organized as a collection of components. Each TinyOS configuration is composed of an application, its required operating system services, a scheduler, and a component graph. Each component is composed by commands, event handlers, tasks and an execution frame. Each component declares the commands to which it responds and the events it signals. Commands are non-blocking method calls and are typically used to initiate software and hardware requests and, conditionally, initiate tasks. Event handlers are used to handle hardware interrupts and may call commands or post tasks. The system provides a simplified concurrency model, based on run-to-completion tasks, which may only be preempted by interrupts. This model brings both negative and positive consequences. In a traditional thread-based model, where each thread has its own stack, each thread must reserve space in the node's limited memory for its execution context. Depending on the architecture, context switching may be a lengthy operation. By restricting this model, TinyOS reduces most of this overhead, but also loses most of the characteristics of a traditional multi-thread model. This restriction of concurrency may also hinder the system's ability to deal with real-time metrics. TinyOS does not provide dynamic memory allocation mechanisms. Timing services are provided by a timer interface. The component model of TinyOS, along with its simplified concurrency model, allows the system to run in platforms with less than 1KB of RAM.

- ***Contiki***: The Contiki operating system [51] is an open source operating system designed for networked embedded systems with small amounts of memory, supporting a wide range of target platforms. Contiki to supports various micro-controller chips, ranging from 8-bit over 16-bit to 32-bit architectures. A typical Contiki configuration size is 2 kilobytes of RAM and 40 kilobytes of ROM. Contiki features an event-driven kernel, providing supports for pre-emptive multi-threading using protothreads. Protothreads are lightweight threads that provide a linear, thread-like programming style on top of Contiki event-driven kernel. Another key feature of Contiki is its support of dynamic linking of code at run-time. This facilitates over-the-air programming and integrating new functionalities without the needs of collecting and reprogramming the nodes off-line. A core component of Contiki is its modularized and highly customizable network stack, which has well-defined generic interfaces for various node platforms. In contrast to operating systems that are limited to certain hardware components, most components of Contiki network stack can be run on all ports supported by Contiki.

Table 2.1 shows the comparison between the two operating systems. Both operating systems can generally fulfill all of the discussed requirements. TinyOS is better suited when resources are really scarce. Contiki might be the better option when flexibility is more important, for example when the node software has to be updated often for a larger amount of nodes [117].

Table 2.1: Comparison of TinyOS and Contiki

TinyOS	**Contiki**
Event-driven model, non-preemptive multi-tasking	Event-driven model with optional preemptive multitasking
Programs written as explicit state machines	Programs written in a sequential fashion
Static linked applications	Dynamic linking of binaries
Code written in necC	Code written in C

2.3.6 Periodic Duty Cycling

Energy saving in wireless sensor networks is an issue of paramount priority. This is because in most scenarios, the deployment environment forbids both the use of constant power supply and the frequent replacement of the batteries. Therefore, most of the sensor networks use duty cycling mechanisms to control the awake time of the radio transceiver, which is the most energy consuming component on a sensor board.

In traditional wireless sensor networks, sensors have to stay in the listening state for sensing ongoing data transmissions. If no event happens, nodes are in idle mode for a long time. Idle listening wastes energy when a node is active, even if there is no packet transmission. Traditional MAC protocols, such as IEEE 802.11 [6] are unsuitable for sensor networks' data delivery. Idle listening state in IEEE 802.11 consumes as much energy as the receiving state. In order to reduce energy consumption when nodes are in the idle listening state, duty-cycling based MAC protocols have been proposed to configure nodes to switch to the sleep mode periodically to save energy [143] [137] [112].

The existing duty-cycle patterns can be divided into synchronous and asynchronous. In protocols with synchronous duty cycles, all nodes wake up at the same time, and they use the same predefined duty cycling interval. The synchronous protocols can be divided into two groups. The time division multiple access (TDMA) [14] and the carrier sense multiple access (CSMA) [3] protocols. Both protocols require synchronization messages to keep the network synchronized. In TDMA protocols usually all the time slots are pre-allocated to the given nodes. In CSMA protocols the neighbor nodes use a CSMA mechanism to assign the slot to a node. Real world implementations of synchronous protocols also showed that keeping nodes synchronized is a challenging task. A significant part of the required energy is used to keep nodes synchronized, mainly because of the longer wake-up time used for synchronization messages.

Asynchronous protocols do not make the nodes to wake up at the same time. The sender waits until the receiver node is awake. Therefore, there is no need for any synchronization messages, which are very energy-consuming. There are two different ways to detect that the target node is awake. First, every neighbor node sends a short beacon message after the wake-up. By overhearing this beacon, a sender knows when the receiver is ready for receiving. Second, the sender transmits several beacons in a row until the corresponding receiver wakes up and sends an acknowledgment back to the sender. Another advantage of asynchronous protocols is that the duty cycles of different network nodes can be non-identical. Therefore, the nodes can adapt their duty cycles according to current traffic loads. For instance, nodes located closer to the destination should use a shorter duty cycle interval to be actively involved in packet transmissions, since most of the traffics will go through them to reach the destination. The major challenge for such adaptive protocols is to detect the increasing traffic on time.

By putting sensors into sleep, we can extend the lifetime of a sensor network. However, the price we have to pay is that network communication and sensing capabilities become intermittent, since sensors switch between sleep and wake modes. The reduced sensing capability disrupts the sensing coverage of the network, i.e., certain areas of the network may not be covered by any sensor and events may fail to be detected on time. Similarly, turning off radio transceivers results in loss of connectivity among nodes. In other words, paths among nodes may be unavailable from time to time. Therefore, sensors need to establish new paths to forward data or wait for the nodes in the path to wake up, which leads to additional delays. As a conclusion, there is a trade-off between energy saving and performance degradation. An efficient WSN protocol should carefully design its operation in order to achieve a good balance between energy efficiency and system performance.

One way to mitigate performance degradation caused by low duty cycle is to add redundancy to the network, i.e., to deploy more sensors. However, in real applications, there is no more interaction with testbeds after the deployments. Therefore, a second way to mitigate such performance degradation is more promising. This approach is to design good algorithms and protocols that carefully control the sleep schedules of sensors (i.e., determining when and for how long a sensor's radio transceiver should be switched off).

2.3.7 Energy Profiling

The deployment of energy-efficient applications for wireless sensor networks requires mechanisms and tools for run-time monitoring of energy consumption. A thorough understanding of how energy is spent is the first step to produce energy efficient protocols. The protocol designers have to know which are the fractions of energy dedicated to different application activities, such as communication, sensing, and computation. On the basis of such information, it is possible to reduce the operation of the most energy-hungry activities through protocol design.

Systems for estimating energy consumption can be implemented both in hardware and in software. Hardware-based approaches rely on the presence of additional circuits on the sensor board (to provide run-time measurement of energy consumption) or involve the use of an oscilloscope/multimeter or some external devices, such as sensor node management device (SNMD) [67]. The positive aspect of hardware-based energy estimation is that it can be very accurate. However, hardware-based energy measurements usually suffer from the costs associated with additional hardwares and the analysis of application behavior may be only limited within a research laboratory, where the operating environments can be significantly different from the real deployment scenarios. Software-based approaches [52] do not require any additional hardware and can be easily integrated with run-time energy management strategies. The drawback of software-based energy profiling is that it may introduce overhead (both in term of code size and run-time execution), and measurements cannot be as accurate and detailed as the ones provided by hardware-based techniques.

No matter which approach to follow, energy-efficient WSN protocols must have run-time energy consumption awareness, such that the protocols can adapt their behaviors accordingly to achieve a good balance between energy efficiency and system performance.

2.4 Wireless Link Variation

Mobile ad-hoc networks and wireless sensor networks might frequently suffer the problems of node mobility and wireless link variation, which bring uncertain issues to the design of energy efficient protocols.

Wireless transmission has characteristics of instability and unreliability, due to the wireless radio propagation and varying interferences during the data transmission.

Radio irregularity is a non-negligible phenomenon in wireless communication. As shown in Figure 2.6, wireless radio transmission ranges are normally irregular and resulting packet delivery ratio (PDR) distribution is non-uniform, which can significantly affect system performance [161]. However, in many protocol designs, such as routing protocols, this is not considered. They simply assume that the transmission range is a circle such that nodes within the radio range can always hear each other, which is not true in reality.

2.5 Node Mobility

In mobile ad-hoc networks, nodes are mobile and their movements are subject to certain mobility patterns (either follow some mobility models or follow some mobility traces). Many mobility models have been proposed, and we summarize some of the them.

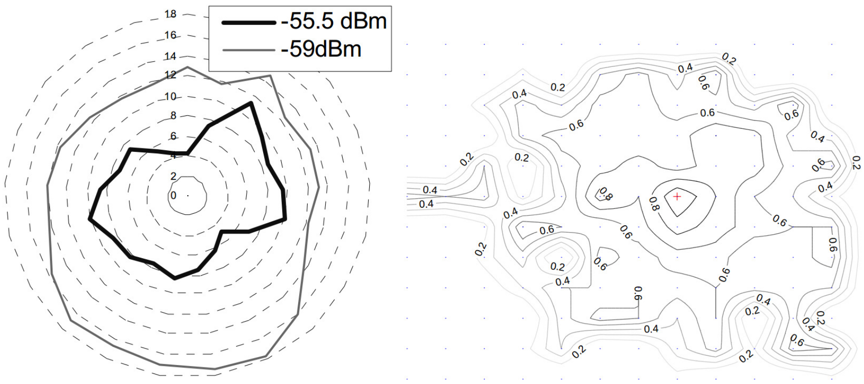


Figure 2.6: Irregular radio range(upper) and resultant PDR distribution(bottom)

The Random Waypoint Mobility model (RWP) The Random Waypoint Mobility model [36] is the most commonly used mobility model in research community. At every instant, a node randomly chooses a destination and moves towards it with a velocity chosen randomly from a uniform distribution [0,V_{max}], where V_{max} is the maximum allowable velocity for every mobile node. After reaching the destination, the node stops for a duration defined by the "pause time" parameter. After this duration, it again chooses a random destination and repeats the whole process until the simulation ends. Figure 2.7 shows an example path of a mobile node moving with the Random Waypoint model.

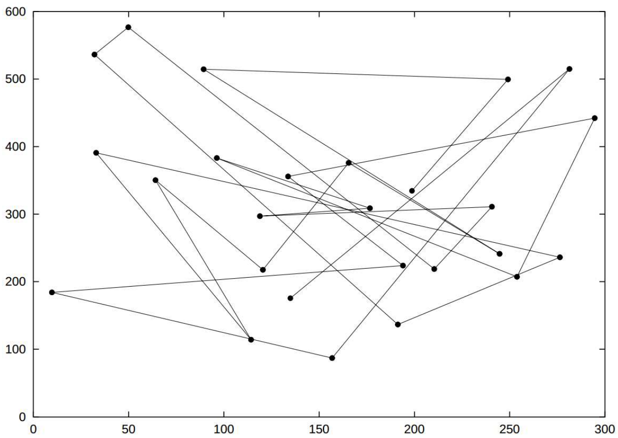


Figure 2.7: Travel pattern of a mobile node using the Random Waypoint model

The Random Walk Mobility model (RW) The Random Walk Mobility model [53] was originally proposed to emulate the unpredictable movements of particles in physics. Because some mobile nodes are believed to move in an unexpected way, Random Walk mobility model is proposed to mimic their movement behavior. The Random Walk mobility model has similarities with the Random Waypoint model because the node movement has strong randomness in both models. In the Random Walk mobility model, the nodes change their speed and direction at each time interval. For every new interval t, each node randomly and uniformly chooses its new direction $\theta(t)$ from (0, 2π]. In a similar way, the new speed $v(t)$ follows a uniform distribution from [0, V_{max}]. An important property of the Random Walk mobility model is the memoryless process where the information about the previous status is not used for the future decision. Compared to the Random Waypoint mobility model, the Random Walk model is the Random Waypoint model with a zero pause time. Figure 2.8 shows an example path of a mobile node moving with the Random Walk model.

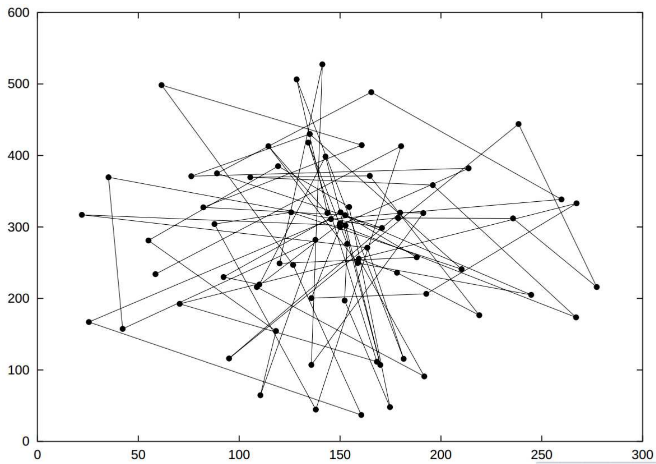


Figure 2.8: Travel pattern of a mobile node using the Random Walk model

The Random Direction Mobility model (RD) The Random Direction Mobility model [121] was proposed to overcome *density waves* in the average number of neighbors produced by the Random Waypoint Mobility model. A density wave is the clustering of nodes in one part of the simulation area. In the case of Random Waypoint model, this clustering occurs at the center of the simulation area. In this model, mobile nodes choose a random direction to travel. The node then travels to the border of the simulation area in that direction. Once the simulation boundary has been reached, the node pauses for a specific time, chooses another direction and continues the process. Figure 2.9 shows an example path of a mobile node moving with the Random Direction model.

The Reference Point Group Mobility model (RPGM) The Reference Point Group Mobility model [69] represents the random motion of a group of mobile nodes as well as the random movement of each individual mobile node within the group. Group movements are based on the path traveled by a logical center (group leader). It is used to calculate group motion via a group motion vector (GM). The motion of the group center completely characterizes the movement of the corresponding group of mobile nodes, including their direction and speed. Individual mobile nodes randomly move around their own pre-defined reference points

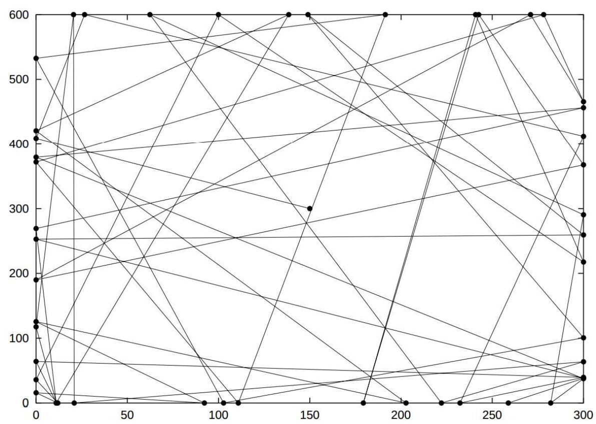


Figure 2.9: Travel pattern of a mobile node using the Random Direction model

whose movements depend on the group movement. As the individual reference point (RP) moves from time t to $t+1$, their locations are updated according to the group logical center. Once the updated reference group points are calculated, they are combined with a random motion vector (RM) to represent the random motion of each mobile node about its individual reference point. The RPGM model can be used in applications where a group of nodes with similar interests move together, such as in social networks. Figure 2.10 shows an example path of three mobile nodes moving with the Reference Point Group Mobility model.

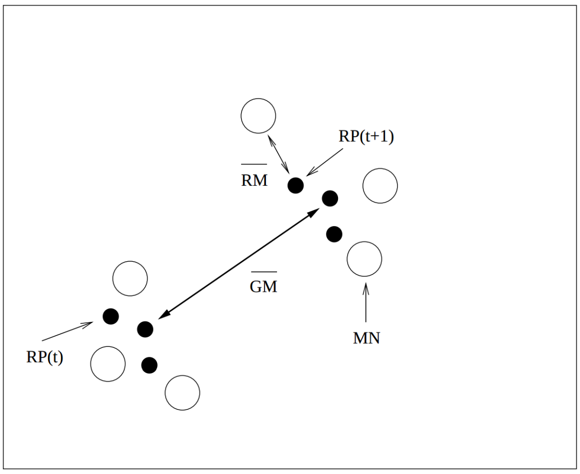


Figure 2.10: Travel pattern of a mobile node using the Reference Point Group Mobility model

Trace-based Mobility Model In recent years, many researchers have tried to redefine existing mobility models to make them more realistic by exploiting the available mobility traces [86], which are stored in the Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD) trace repository [44]. The key underlying idea of these models is the exploitation of available measurements such as connectivity logs to generate synthetic traces that are characterized by the same statistical properties of the real ones. Various studies have been conducted both in infrastructure-based and infrastructure-less environments since the first wireless networks have been deployed. Extensive measurements about the usage of the early deployed Wireless Local Area Networks (WLANs) have been conducted [130]. With respect to mobility models for vehicular networks, a large amount of traces mapping the movements of vehicles in cities and in highways are collected by the traffic authorities but they are not publicly available also due to security reasons. Starting from these traces, several models have been recently presented, such as the Topologically Integrated Geographic Encoding and Referencing (TIGER) traces [134].

Social Network-based Mobility Models are based on a simple observation. In mobile networks, devices are usually carried by humans, so the movement of such devices is necessarily based on human decisions and social behavior. A key characteristic is the presence of clusters that are usually dependent on the relationships among the members of the social group. In order to capture this type of behavior, mobility models dependent on the structure of the relationships among people carrying the devices have been defined. However, existing group mobility model fails to capture this social relationship. In [101], authors proposed a community based mobility model, which allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals. This grouping is only then mapped to a topographical space, with topography biased by the social relationships among them.

The introduction of mobility expands the application spaces of both wireless ad-hoc and sensor networks. However, node mobility modifies the network topology constantly and brings new challenges to the design of network protocols. In the domain of wireless ad-hoc networks, one may think that mobility has only a negative impact on network performance. However, with the capability of moving, nodes could visit places where a static deployment can not visit, and can cover areas that a static deployment can not cover. Moreover, mobility patterns can be configured in a way that nodes will benefit from the encountering with other moving nodes. For example, when the relative moving speed of two nodes is low, nodes will benefit from their movements, since they have chances to meet a better forwarder, which might hep to bring the packet closer to the destination. However, if the moving speed is high, the contact duration between two moving nodes will be short. Then the packet transmission can not be finished within such a short "meeting" interval. In general, the effect of node movements plays a significant role on system performance, and thus, deserves detailed investigation.

2.6 Topology Control

Due to the movement of nodes or the variation of wireless radio link, network connectivity of a wireless mobile ad-hoc sensor network will keep changing over time. It may happen that at some time, one or more nodes lose the connectivity with other nodes. Consequently, the network will be separated into multiple isolated pieces, and performance will degrade sharply. Therefore, to maintain network connectivity, a topology control algorithm is necessary in both mobile ad-hoc networks and wireless sensor networks.

Additionally, topology control is an important technique to reduce energy consumption and interference among multiple data flows. In this context, the goal is to control the network topology with the purpose of maintaining some global graph property, while reducing energy consumption and/or interference [125].

Several approaches have been proposed to control network topology [114] [113], and the most common techniques are listed below:

- *Adapt transmission power of sensor nodes*: Starting from a minimum transmission power, a node adaptively adjusts the transmit powers in response to topology changes and attempts to maintain a connected topology.

- *Turn off the sensor nodes*: Some nodes are deployed to provide coverage and connectivity redundancy. Therefore, they could be temporarily turned off to save energy, given that the network connectivity is not affected.

- *Create a communication backbone*: To keep the network always connected, a communication backbone can be created such that the backbone-nodes are always working properly to provide full-time network connectivity.

- *Adding new nodes into the network to preserve connectivity*: In case of a broken connectivity, new nodes can be added into the broken parts of the network, to re-establish connectivity. Or if the connectivity is detected to be broken soon, new nodes can also be added to preserve connectivity.

In general, topology control has significant impact on energy consumption of wireless ad-hoc and sensor networks. Since it modifies network topology, it also plays an important role in maintaining network connectivity and improving network performance. Therefore, topology control should be considered during the protocol design of MANETs and WSNs.

2.7 Conclusions

In this chapter, a general overview of wireless mobile ad-hoc networks and wireless sensor networks has been presented. Their basic concepts, specific characteristics, and applications scenarios were discussed. We introduced relevant works within these two networks, such as packet routing, energy management, hardware and software platforms, etc. Moreover, we described the design challenges of these two networks.

In the next chapter, we will review the latest developments of the relevant topics that are covered in this thesis.

Chapter 3

Related Work

This chapter introduces the most significant and important related works of this thesis: routing in MANETs and WSNs, opportunistic routing, context-aware adaptive communications, and simulation framework for MANET/WSN study. The latest developments of other relevant topics are also included in this chapter. In section 3.1, we introduce some general background knowledge. The latest developments of traditional routing approaches in MANETs are presented in section 3.2. They can be classified into four groups: proactive routing (section 3.2.1), reactive routing (section 3.2.2), hybrid routing (section 3.2.3), and geographic routing (section 3.2.4). Section 3.3 describes the representative routing protocols in WSNs. A new routing approach, opportunistic routing is introduced and its latest developments are discussed in section 3.4. Two kinds of opportunistic routing: candidate list-based approach and beaconless-based approach are explained individually in section 3.4.4 and section 3.4.5. Section 3.4.3 introduces the different coordination mechanisms of opportunistic routing.

Next, we introduce a new routing concept, called context-aware routing. Context aware routing, in general, takes all the network context information that have impact on the routing performance into account. The background knowledge is depicted accordingly in section 3.5, section 3.5.2 and section 3.5.4, which cover the topics of context definition, context-based routing protocol, and the mechanisms to combine multiple types of context information. An adaptive communication mechanism is promising for wireless ad-hoc networks, since the participants could adapt their behaviors according to the latest network conditions, which vary fast in lossy wireless network. Thanks to the context-awareness, adaptive routing can make the local optimal routing decision. We analyze the latest developments of the adaptive communication mechanisms in section 3.6. The decision making techniques, which are the supporting schemes to provide adaptivity feature, are discussed in section 3.7.

Existing efforts on duty-cycling mechanism and software-based energy profiling mechanism are described in section 3.8 and section 3.9 separately. Section 3.10 describes the efforts on network simulation and simulation frameworks. In section 3.11, we conclude the related works.

3.1 Introduction

Traditional routing protocols for wireless multihop networks have followed the concept of routing in wired networks by abstracting the wireless links as wired ones, and find paths with the shortest lengths or the highest throughput. The protocols use controlling message to set up an end-to-end path from a source to a destination before the data transmission. Once the path is built, the source will stick to that path and packets will be only sent to the nodes on the path. As far as any node on the path is not working properly, the whole path is broken and the source has to search for a new path and retransmit the packet. As a conclusion, since traditional MANET and WSN routing protocols rely on the consistent and stable behavior of individual links, the intermittent behavior of wireless links can result in poor performance such as low packet delivery ratio, high control overhead, and long end-to-end delay. On the other hand, the abstraction of the wireless link as the wired one ignores the unique broadcast nature and spatial diversity of the wireless transmission. In MANETs or WSNs, when a packet is unicast to a specific node, all the neighboring nodes in the effective communication area of the sender might overhear the transmission at the the physical layer. It's possible that some of the neighbors may have received the packet successfully while the designated next-hop node did not.

Based on this observation, a new routing mechanism, known as ***opportunistic routing (OR)*** has been proposed. Opportunistic routing integrates the network and MAC layer. In opportunistic routing, instead of selecting only one node as forwarder and sending unicast packet, the packet sender selects a set of candidate nodes and broadcasts the packet. When receiving this packet, one neighbor of the sender is selected dynamically at the MAC layer as the real forwarder based on the instantaneous wireless channel quality of that transmission and node availability at the time of packet transmission. Opportunistic routing takes advantages of the spatial diversity and broadcast nature of wireless communications and is an efficient mechanism to combat the time-varying features of wireless links. Opportunistic routing improves the network throughput and energy efficiency compared to traditional routing in MANETs and WSNs.

In the following, we start with the introduction of the traditional routing mechanisms in MANETs and WSNs [17]. Their drawbacks in case of node mobilities and link variations are also presented and discussed. This leads to the next section, where we introduce the concept of opportunistic routing. The immunity of opportunistic routing against the node mobility and channel variation is described then. The latest developments of opportunistic routing protocols are presented also.

3.2 Routing in Mobile Ad-hoc Networks

Packet routing in a mobile ad-hoc network is intrinsically different from routing in wired networks with fixed infrastructures. Due to the infrastructure-less feature, routing in MANETs encounters many challenges that do not exist in wired network routing. For example, one challenge is that, a MANET node needs to know the reachability information to its neighbors. However, node mobility and varying of wireless links make the network topology constantly change over time.

Many efforts have been made in the design of routing protocols in mobile ad-hoc networks. Most of them are based on the usage of routing tables. There are many ways to summarize the types of MANET routing algorithms. If the decision is made based on the information used to build the routing tables, then MANET routing can be divided into two types: shortest distance algorithms and link state algorithms. Shortest distance algorithms use distance information to build and maintain routing tables. Link state algorithms use connectivity information to build a topology graph which is then used to build routing tables. If the decision is based on whether and when the routing tables are built, then MANET routing can be divided into four types: proactive routing, reactive routing, hybrid routing, and geographic routing.

In the next sections, we follow the second principle, namely to separate the MANET routing according to whether and when the routing tables are built. In proactive routing, nodes maintain routing information to every other nodes in the network. The routing tables have to be updated periodically independent of data transmissions. This means every node has to update its routing table even if it does not have any data to transmit. The profit of this approach is that whenever a node wants to send packets, it already has the route information available in the routing table, and it just chooses the node recorded in the table as next-hop. Therefore, it does not have to wait for finding a next-hop, and its packet transmission is of small delay. However, the weakness of this approach is also remarkable, since nodes have to keep updated with the latest topology by using periodic beaconing messages, which costs additional energy. The performance of this type of routing approach is up to the frequency to update the routing table. If the routing table is updated not as frequently as the topology changes, then the table will be outdated and the route information is invalid. This leads to transmission failure and degrades the system performance. However, if the routing table is updated frequently, the large controlling overheads will cost tremendous resources. In reactive routing, routes are built on request, therefore it reduces the controlling overhead. The drawback is that there might be some delays before a route is calculated.

3.2.1 Proactive Routing

Proactive routing maintains the route from source to destination all the time by exchanging beaconing messages periodically. Therefore, whenever a sender wants to send packets to a destination, the route information are immediately available in the routing table. The drawbacks is the large amount of controlling overhead, which wastes the scarce network resources. In the following, we list some representatives of proactive routing protocols.

Destination-Sequenced Distance-Vector Protocol

The destination-sequenced distance-vector protocol (DSDV) [110] is a proactive hop-by-hop distance vector routing protocol, requiring each node to periodically broadcast routing updates. In DSDV, every node in the network maintains a routing table for all possible destinations within the network and the number of hops to each destination. Each entry in the table is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain consistency in the table.

Global State Routing Protocol

The global state routing (GSR) [41] is based on the traditional link state algorithm. However, GSR has improved the way information is disseminated in link state approach by restricting the update message among intermediate nodes only. In GSR, each node maintains a link state table based on the up-to-date information received from neighboring nodes, and periodically exchanges its link state information with neighboring nodes only. This has significantly reduced the number of control messages transmitted throughout the network.

Optimized Link State Routing Protocol

The optimized link state routing(OLSR) [42] maintains the network topology information by exchanging link-state messages. OLSR minimizes the size of each control message and the number of rebroadcasting nodes during each route update by employing a new concept called multi-point relaying (MPR) node. During a topology update process, each node selects a set of neighbors to retransmit its packets. This set of nodes is defined as the multi-point relay node. Any node that is not in this MRP nodes can only read the packet but can not retransmit it. To select the MPR nodes, each node periodically broadcasts a list of its one hop neighbors using hello messages. From the list of nodes in the hello messages, each node selects a subset of one hop neighbors, which covers all of its two hop neighbors.

3.2.2 Reactive Routing

Reactive routing, or called on-demand routing, protocols were designed to reduce the overheads in proactive routing by maintaining information for active routes only when necessary. This means that routes are determined and maintained for nodes that require to send data to a destination. Route discovery usually occurs by flooding a route request packets through the network. When a node with a route to the destination is reached, a route reply message is sent back to the source node using link reversal if the route request has traveled through bi-directional links or by piggy-backing the route information in a route reply message via flooding.

Dynamic Source Routing Protocol

The dynamic source routing (DSR) [81] protocol uses explicit source routing in which each data packet has in its header a complete list of all intermediate nodes to the destination. DSR is composed of two main mechanisms. In route discovery, a node, which attempts to send a packet to a destination and does not know a route, broadcasts a route request packet. Each node that forwards this packet adds its own address to the header. If the destination received the route request, it sends back a route reply packet containing a copy of the accumulated route along the reverse direction of the path over which the route request packet arrived. Thus, each node forwarding this reply packet is aware of the whole path from the source to the destination. Nodes cache the route information from each packet they overhear. Intermediate nodes may also reply to a route request packet, if it knows a route to the destination (thanks to the cached route information before).

Ad-hoc On-demand Distance Vector Protocol

The ad-hoc on-demand distance vector protocol (AODV) [111] is based on DSDV and DSR protocols. It uses the periodic beaconing messages and sequence numbering procedure of DSDV and a similar route discovery mechanism from DSR. However, there are two major differences between DSR and AODV. The most distinguishing difference is that in DSR, each packet carries a full list of forwarder information, whereas in AODV the packets carry the destination address only. The advantage of AODV is that it is adaptable to dynamic environments. However, nodes may experience large delay during the route setup procedure, and link failures may initiate another route discovery, which introduces extra delays and consumes more bandwidth as the size of network increases.

Location-aided Routing Protocol

The location-aided routing protocol (LAR) [85] is based on flooding algorithm (such as DSR). However, LAR attempts to reduce the routing overheads presented in the traditional flooding algorithm by using location information. This protocol assumes that each node knows its location through a GPS device. Two different LAR schemes were proposed in [85], the first scheme calculates a request zone which defines a boundary where the route request packets can travel to reach the required destination. The second method stores the coordinates of the destination in the route request packets. Both methods limit the control overhead transmitted through the network and hence conserve bandwidth. The disadvantage of this protocol is that it might behave similar to flooding algorithms in highly dynamic environment.

3.2.3 Hybrid Routing

Hybrid routing is a new generation of protocol, which is both proactive and reactive in nature. This routing concept is designed to increase the scalability by allowing nodes with close proximity to work together to form some sort of network backbone to reduce the route discovery overheads. This is mostly achieved by pro-actively maintaining routes to nearby nodes and determining routes to far away nodes using a route discovery strategy. Most hybrid protocols are zone-based, which means that the network is partitioned into a number of zones. Some hybrid protocols separate nodes into trees or clusters. This section describes a number of hybrid protocols.

Zone Routing Protocol

In zone routing protocol (ZRP) [61], each node has an associated routing zone, which defines a range, in terms of hops, that each node is required to maintain network connectivity proactively. Therefore, for nodes within the routing zone, routes are immediately available. For nodes located outside the routing zone, routes are determined on-demand, and it can use any on-demand routing protocols to find out a route to the destination. The advantage of this protocol is that it has significantly reduced controlling overheads when compared to a pure proactive protocol. It also reduces the delays associated with pure reactive protocol, by allowing routes to be discovered faster. This is because to determine a route outside the routing zone, ZRP packet only has to travel to a node lying on the boundaries (edges of the routing zone) of the destination.

Sharp Hybrid Adaptive Routing Protocol

The sharp hybrid adaptive routing protocol (SHARP) [115] currently is a joint routing approach of proactive and reactive routing. SHARP is a hybrid routing protocol that finds the optimal mixture of proactive route dissemination and reactive route discovery. It finds the balance point between proactive and reactive routing by adjusting the degree to which route information is propagated proactively versus the degree to which it needs to be discovered reactively. SHARP enables each node to use a different application-specific performance metric to control the adaptation of the routing layer.

3.2.4 Geographical Routing

Geographical routing is another type of routing scheme, which forwards packets based on the nodes' physical locations. In this routing mechanism, forwarding decisions are made solely based on the position of the current node, the positions of neighboring nodes, and the position of the destination. This type of routing does not include any manipulation of routing tables. In geographical routing, a node that wants to send a packet to a destination chooses one of its neighbors as a next-hop based on some criteria. One big benefit of geographical routing is that it has very few control overhead, since it does not require the establishment of routes and thus removes the overheads for topology updates. For these reasons, geographical routing is regarded as scalable and more robust to changes in the network topology.

Greedy Perimeter Stateless Routing Protocol

The greedy perimeter stateless routing protocol (GPSR) [83] is probably the most cited geographic routing protocol. In GPSR, a packet is routed towards the destination in a greedy manner. Each node selects the node among all its neighbors that is geographically closest to the destination as its next-hop. This process is repeated until the packet reaches the destination. If a node reaches a node where there is no neighbor closer to the destination, then it will switch to face routing, i.e., the packet is routed according to the right-hand rule on the faces of a locally extracted planar subgraph, namely the Gabriel graph, to recover from this local minimum. The planar subgraph extraction is necessary in order to avoid loops. This recovery mode is called perimeter mode routing. As soon as the packet arrives at a node closer to the destination than where it switched to the face routing, the packet switches back greedy routing. It was shown that GPSR guarantees packet delivery for static and connected networks. However, if nodes are mobile, packet may still loop in the network. As an example in Figure 3.1, the packet is first routed in greedy mode to node X, which then has no closer neighbor within its transmission range to destination D. Then the packet switches into the perimeter mode and finds node Z as next-hop. At node Z, the packet switches back to greedy mode.

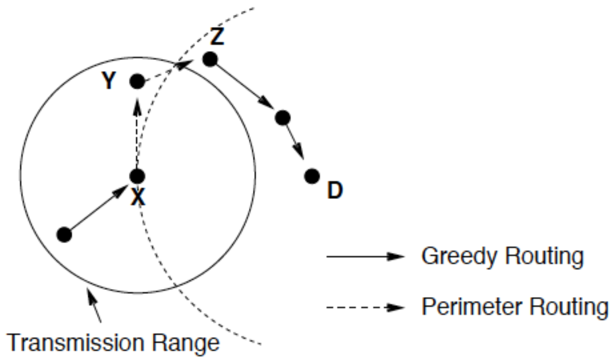


Figure 3.1: Greedy Perimeter Stateless Routing Protocol (GPSR) protocol.

Geographic Adaptive Fidelity Routing Protocol

The geographic adaptive fidelity routing protocol (GAF) [141] is an energy-aware location-based routing algorithm. In GAF, the network area is first divided into fixed zones and forms a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, one node might be elected to stay always awake for a period while others sleep to save energy. This node then is responsible for monitoring on behalf of all nodes within the zone. Each node uses its location to associate itself with a "representative node" in the virtual grid. GAF strives to keep the network connected by keeping a representative node always in active mode for each region on its virtual grid. Although GAF is a location-based protocol, it may also be considered as a hierarchical protocol, where the clusters are based on geographic location. For each particular grid area, a representative node acts as the leader to transmit the data to other nodes. The leader node however, does not do any aggregation or fusion as in the case of other hierarchical protocols discussed earlier in this article.

Geographic and Energy Aware Routing Protocol

The geographic and energy aware routing protocol (GEAR) was proposed in [147]. It discussed the use of geographical information while disseminating queries to appropriate regions since data queries often include geographic attributes. GEAR uses energy aware and geographically-informed neighbor selection heuristics to route a packet towards the destination region. The key idea is to restrict the number of query interests by only considering a certain region rather than sending the query interests to the whole network. By doing this, GEAR can conserve more energy than directed diffusion.

3.3 Routing in Wireless Sensor Networks

Routing in wireless sensor networks is challenging due to the inherent characteristics that distinguish these networks from other examples of wireless ad-hoc networks. Wireless sensor nodes are tightly constrained in terms of energy, processing and storage capacities. Thus they require careful resource management. The task of finding and maintaining routes in WSNs is nontrivial since energy restricts and sudden changes in node status (e.g., failure) cause frequent and unpredictable topological changes. To minimize energy consumption, routing techniques proposed in the literature for WSNs employ the well-known routing tactics, e.g., data aggregation and in-network processing, clustering, and different node role assignment were employed. Almost all of the protocols can be classified according to the network structure as flat routing, and hierarchical routing. In the following, we survey some representatives of each protocol type.

3.3.1 Flat Routing

In flat routing, all nodes in the network are typically assigned equal roles or functionalities. Nodes collaborate with each other to perform the sensing tasks.

Sensor Protocols for Information via Negotiation Protocol

The sensor protocols for information via negotiation protocol (SPIN) [65] disseminates all the information of each node to every node in the network assuming that all nodes in the network are potential base stations. This enables a user to query any node and get the information immediately. This protocol makes use of the property that nodes in close proximity have similar data, and hence there is a need to only distribute the data that other nodes do not posses. SPIN uses data negotiation and resource-adaptive algorithms. SPIN nodes assign a high-level name to completely describe their collected data (called meta-data) and perform meta-data negotiation before any data transmission. This assures that there is no redundant data sent throughout the network. In addition, SPIN has the access to the current energy level of the node and adapts its behavior based on how much energy is remaining.

Rumor Routing Protocol

The rumor routing protocol [34] is a variation of directed diffusion and is mainly intended for applications where geographic routing is not feasible. In general, directed diffusion uses flooding to inject the query into the network when there is no graphical criterion to diffuse tasks. However, in some cases there is only a little amount of data requested from the nodes and thus the use of flooding is unnecessary. An alternative approach is to flood the events if the number of events is small and the number of queries is large. The key idea is to route the queries to the nodes that have observed a particular event rather than flooding the network to retrieve information about the occurring events. Simulation results have shown that rumor routing achieves significant energy saving over event flooding and can also handle node's failure.

3.3.2 Hierarchical Routing

In hierarchical routing, different nodes play different roles in the network. Similar to other communication networks, scalability is one major concern of wireless sensor networks. A single-tier network can cause the gateway to overload with the increasing number of sensors. Such overload might cause latency in many applications. To allow the system to cope with additional load and to be able to cover a large area of interest without degrading the service, network clustering has been employed. The main aim of hierarchical or cluster-based routing is to efficiently maintain the energy consumption of sensor nodes by involving them in multihop communication within a particular cluster and by performing data aggregation and fusion in order to decrease the number of packet transmission to the sink.

Low Energy Adaptive Clustering Hierarchy Protocol

The low energy adaptive clustering hierarchy protocol (LEACH) [64] is one of the most popular hierarchical routing protocol for WSNs. The idea is to form clusters of the sensor nodes based on the received signal strength and use local cluster heads as routes to the sink. This will save energy since the transmission will only be done by such cluster heads rather than all sensor nodes. LEACH randomly selects a few sensor nodes as cluster heads (CHs) and rotate this role to evenly distribute the energy load among the sensors in the network. In LEACH, CH nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the based station in order to reduce the amount of information that must be transmitted to the based station. The operation of LEACH is separated into two phases, the setup phase and the steady state phase. In the setup phase, the clusters are organized and CHs are selected. In the steady state phase, the actual data transfer to the base station takes place. The duration of the steady state phase is longer than the duration of the setup phase in order to minimize overhead.

Multi-hop hierarchical routing protocol for Efficient Video Communication over WMSNs

The multi-hop hierarchical routing protocol for efficient video communication over WMSNs (MEVI) [119] combines a cluster formation scheme with a minimal signaling overhead. MEVI is a cross-layer solution to select routes based on network conditions and energy issues. It provides a smart scheme to trigger multimedia transmission according to sensed data. The cluster approach aims to minimize the energy consumption and is suitable for the distribution of multimedia content in WMSNs.

Virtual Grid Architecture Routing Protocol

The virtual grid architecture routing protocol (VGA) [23] utilizes data aggregation and in-network processing techniques to maximize the network lifetime. Due to the node stationarity and extremely low mobility in many WSNs applications, a reasonable approach is to arrange nodes in a fixed topology. In VGA, square clusters were used to obtain a fixed rectilinear virtual topology. Inside each square zone, a node is optimally selected to act as the cluster head. Data aggregation is performed at two levels: local and global. The set of cluster heads, also called Local Aggregators (LAs), perform the local aggregation, while a subset of these LAs are used to perform global aggregation.

3.4 Opportunistic Routing

Low quality of wireless links leads to perpetual transmission failures. To mitigate this problem, opportunistic routing has been proposed to overcome the deficiencies of conventional MANET routing. Unlike traditional MANET routing, which finds end-to-end paths to send unicast packets, opportunistic routing exploits the broadcast nature of wireless medium to postpone the selection of packet forwarders to the receiver side. Opportunistic routing lets multiple receivers of a transmission coordinate with each other and decide which one will actually forward the packet.

Traditional MANET routing selects one of the multiple intermediate nodes as the packet forwarder prior to data transmissions. The data is then unicast to the selected node, and other nodes will drop the packet even though they opportunistically overhear the transmission. If the unicast transmission is failed, the source node has to retransmit the same packet or even has to find a new path. However, in opportunistic routing, the source node sends the packet without knowing who will be the forwarder. It preselects a set of nodes, called relay candidates, as possible forwarders and broadcasts the packet. The broadcast transmission might be overheard by multiple nodes. As far as one of the candidates receives the transmission, it further forwards the packet. The source node retransmits the same packet only when all the intermediate nodes simultaneously miss the previous transmission, which is of much lower probability than the case of traditional routing. The performance of opportunistic routing depends on several factors, among which candidate selection and forwarder election are the most important.

As an example, a directed graph in Figure 3.2 represents a wireless network in which a link *(x,y)* has a delivery probability *P(x,y)*. Traditional routing mechanisms achieve only 20% end-to-end delivery probability for any possible routing path (via A, B, C, D, or E) from source to destination. However, an opportunistic routing could achieve a delivery probability of $(1-(1-20\%)^5)$ = 67% if all five neighbors of source are selected as relay candidates. As another example, Figure 3.3 illustrates how opportunistic routing can affect an entire routing path. For clarity, the delivery probabilities for some links are not shown in the figure. It should be clear that each of links *(src, B), (B, D), (D, dst)* has a 60% delivery probability, and each of links *(src, C), (C, dst)* has a 40% delivery probability. A packet from a source may follow different paths to reach the destination. Traditional MANET routing would always choose the most reliable link to forward the packet, which results in a path of $src \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow dst$. This fixed end-to-end path has the success packet delivery probability of $((80\%)^5)$ = 26%. With opportunistic routing, if we restrict a node to route packets via paths with at most three hops, there are four paths meeting this requirement: $src \rightarrow C \rightarrow dst$, $src \rightarrow C \rightarrow D \rightarrow dst$, $src \rightarrow B \rightarrow C \rightarrow dst$, and $src \rightarrow B \rightarrow D \rightarrow dst$. The first two paths have a successful delivery probability of P(src,C) $\times$ (1-(1-P(C,dst) $\times$ (1-P(C,D) $\times$ P(D,dst))) = 40% $\times$ (1-(1-40%) $\times$ (1-80% $\times$60%)) $\approx$ 27.5%. Similarly, the last two paths have a successful delivery probability of 60% $\times$ (1-(1-60% $\times$ 60%) $\times$ (1-80% $\times$ 40%) $\approx$ 33.9 %. The overall successful delivery probability by the above four paths is therefore 1-(1-27.5%) $\times$ (1-33.9%) $\approx$ 52.1%, which doubles the value of traditional approaches.

Most of the existing opportunistic routing protocols choose the next-hop forwarder based on a predefined candidate list. A source node, before the data transmission, injects a certain amount of beacon messages into the network to learn network link qualities. Then, it infers and obtains a ranking of nodes as forwarding candidates, according to the estimated quality of each link. This list is then em-

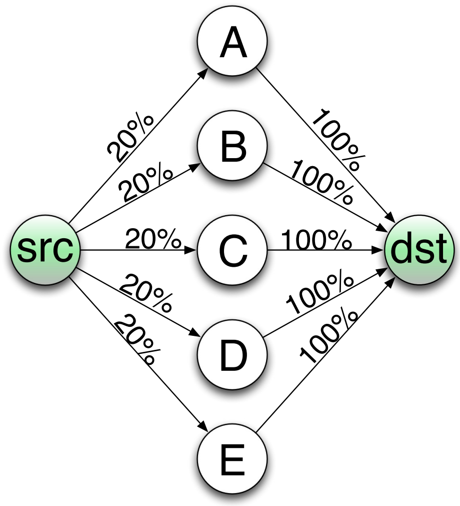


Figure 3.2: An illustration of single-hop multiple candidate-based opportunistic routing.

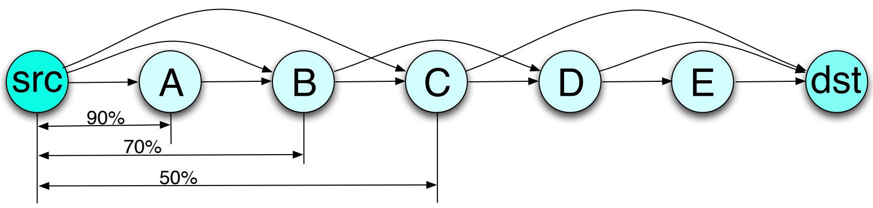


Figure 3.3: An illustration of multihop opportunistic routing.

bedded within the data packet and used as a reference to select a forwarder at each hop. As we can see, the candidate list is calculated prior to data transmission according to certain network metric. However in reality, wireless links are extremely unstable, as they often experience quality fluctuation and distortion due to interference. Therefore, the link quality-based candidate list that is generated before data transmission may not be valid anymore when the data is being transmitted. Moreover, when mobility is introduced, nodes will move according to certain mobility patterns. The network topology will change and thus the estimated candidate list will be invalid. Therefore, the list-based feature of the existing opportunistic routing protocols restricts the freedom of opportunism, and thus, it is not appropriate for the dynamic feature of MANETs. An opportunistic routing protocol without a candidate list is more promising for packet transmissions in dynamic environments, such as MANETs or WSNs.

In the next sections, we first describe the challenges involved in the design of opportunistic routing in section 3.4.1. Section 3.4.2 discusses the metrics that are used in existing opportunistic routing protocols. Different types of coordination mechanisms of OR are discussed in section 3.4.3. Description of different opportunistic routing protocols is presented in section 3.4.4 and section 3.4.5, and their drawbacks are discussed in section 3.4.6.

3.4.1 Challenges of Opportunistic Routing Design

The major challenge in opportunistic routing design is to maximize the routing progress of each data transmission towards the destination without producing duplicate transmissions or incurring significant coordination overheads. In order to achieve the potential benefits of opportunistic routing and avoid the abovementioned problems, an effective protocol should implement the following tasks in a distributed fashion:

- *Candidate selection:* All nodes in the network must run an algorithm for selecting and sorting the set of neighboring nodes (candidate list) that can better help in the forwarding process to a given destination. We refer to this algorithm as *candidate selection*. The aim of the candidate selection algorithm is to guarantee that only the qualified nodes become the candidates and to build the candidate list. In order to accurately build the candidate list, OR protocols require certain metrics to evaluate the network and rank network nodes. In sections 3.4.4 and 3.4.5, we describe some representative candidate selection algorithms.

- *Forwarder selection through candidate coordination:* Forwarder selection provides a scheme to select, among all the candidates that have successfully received the packet, only one node that really forwards the packet. Because there is no central controlling node, the forwarder selection process is done through the coordination of multiple candidates. Coordination requires signaling among candidates, and imperfect coordination may cause duplicate packet transmissions. In section 3.4.3, we describe five types of coordination mechanisms used in literature.

- *Forwarding responsibility transfer:* This function allows the nodes involved in the forwarding process - the actual forwarder plus the candidates - to become aware of the winner of the selection. The responsibility transfer is the distinguishing feature that differentiates opportunistic routing from flooding. In fact, in both opportunistic routing and flooding, multiple nodes will receive the broadcast transmission from a packet sender. However, unlike in the flooding algorithm, opportunistic routing allows only one node at a time to be in charge of packet forwarding.

- *Duplicate transmission avoidance:* This process is required only in case of imperfect responsibility transfer. If the forwarding responsibility is correctly transferred to the winning forwarder, there is only one node in charge of packet forwarding at any time. In contrast, several packet transmissions occur but only one is innovative, i.e., the one made by the winning forwarder.

More effective the duplicate avoidance mechanism is, less network resources will be wasted.

3.4.2 Routing Metrics

The general purpose of opportunistic routing is to minimize the expected number of transmissions required to transmit a packet from the source to the destination. The set of candidates that are used by nodes and their priorities have significant impact on routing performance. Therefore, using a good metric to select and prioritize the candidate is a key factor.

Candidate can be prioritized based on hop count, geographic distance, expected transmission count (ETX), expected any-path transmission (EAX) and so on. Utilization of hop count, ETX or EAX needs an underlying routing protocol (either reactive or proactive) to gather such information. Geo-distance requires the availability of location information of nodes. The accuracy of a metric depends on the proper measurement and timely dissemination of such information. Below, we describe two metrics of ETX and EAX that have been widely used in the literature.

Expected Transmission Count (ETX) [48] is the average number of transmissions required to reliably send a packet across a link or route including retransmissions. The ETX of a single path route is the sum of the ETX for each link in the route. With the assumption of the packet transmission between nodes i and j with delivery probability p_{ij} , the expected transmission count of the link is: $ETX(i,j) = \frac{1}{p_{ij}}$.

In OR, however, it is necessary to consider the fact that there are some candidates, which can receive the packet. Thus, a packet may travel along any of the potential paths. Authors in [49] [92] have shown that using ETX may give suboptimal selection of candidates and in [95] it was shown that OR in combination with ETX could degrade the performance of the network. Therefore, [160] proposed another metric, which has been widely adopted in OR.

Expected Any-path Transmission (EAX) [160] is an extension of ETX and can capture the expected number of transmissions taking into account the multiple paths that can be used in OR. Alternative methods to compute EAX have been proposed by different authors [92] [39].

3.4.3 Coordination Mechanisms

Opportunistic routing differs from traditional MANETs routing in many aspects: multiple relay candidates instead of only one forwarder and dynamic relay selection after the data transmission. Therefore, the problems of candidate selection and coordination are of great importances. Since beaconless opportunistic routing does not include any list of prioritized candidates, all the nodes in the network are of the same priority. Therefore, the coordination among multiple receivers of a broadcast transmission is essential, since an efficient coordination method could make sure that only one node will forward the packet. Coordination requires signaling between nodes, and imperfect coordination may cause duplicate packet transmissions. A good coordination approach should select the best candidate without duplicate transmissions while using the smallest control overhead.

In beaconless opportunistic routing, when receiving a broadcast packet, multiple receivers have to negotiate with each other to agree on one node to exclusively forward the packet. From a packet sender's point of view, the coordination method should choose the best candidate. On the other hand, from the candidate's point of view, coordination is used to consider its current status and decide whether to forward the overheard packet or not. A good coordination method should select the best relay without duplicate transmissions while using the smallest coordination cost (in terms of time and control overhead).

We summarize the existing coordination approaches into five categories based on the mechanism they used: timer-based, acknowledgment-based, token-based, network coding-based, and request-to-send/clear-to-send (RTS-CTS) coordination. In the following, we briefly describe these approaches.

- *Timer-based Coordination*: is the most straightforward approach and easy to implement, but it may lead to duplicate transmissions. It predefines a candidate order before data transmission. The first node that responds is then selected as the next relay. All candidates are ordered based on a predefined metric. The order is generated prior to data transmission and is included in the packet header. After a packet is broadcasted, candidates will respond in order, according to their priorities in the list. A candidate responds only when it does not hear any responses from nodes with higher priorities. Therefore, before a candidate responds, it can confirm that all higher priority candidates failed to receive the packet. Once a candidate responds, the candidate is selected as the next relay and the response will prevent others from responding. The overhead of the timer-based coordination method are candidate waiting time and candidate ordering information included in the packet header. In beaconless opportunistic routing, where the candidate ordering process is completely removed, these two overhead will also be eliminated. Because candidates will not be prioritized, there is no more candidate ordering information in the packet header. When receiving a packet, every node can immediately triggers its responding procedure without waiting for the others. As we can see from the forwarder selection process of the protocols of IGF [128], BLR [97], CBF [57], and BOSS [124], their coordination mechanisms are just a timer, which is based on certain local metrics, such as the distance progress. Therefore, a beaconless opportunistic routing approach using a timer-based coordination mechanism is a promising solution.

- *Acknowledgment-based Coordination*: is one of the first methods that was proposed for candidate coordination. Upon receiving a data packet, candidates send back a short acknowledgment (ACK) in decreasing order of candidate priority. This method was first proposed in [88] as the coordination mechanism for the Selection Diversity Forwarding (SDF) protocol. In SDF, coordination is achieved by means of a four-way handshaking: the candidate receiving the data packet sends back an acknowledgment to the sender. Based on the acknowledgments, the sender sends a forwarding order to the best candidates, which is also acknowledged. A similar approach is used in the famous ExOR protocol. Instead of only indicating that the packet was successfully received, each ACK contains the ID of the highest priority successful recipient known to the ACK sender. All the candidates listen to all ACK slots before deciding whether to forward, in case a low-priority candidates ACK reports a high-priority candidate ID and whose ACK was not correctly received. Including the ID of the sender of the highest-priority ACK heard so far helps to suppress duplicates forwarding. This strategy requires that candidates be neighbors of each other such that the transmission of an ACK can be overheard by all of them.

As an example of the ACK-based coordination, consider a network with source S and destination D. Assume that the candidate set of S is A, B, C (A has the highest priority and C has the lowest). Suppose that all candidates receive a transmission from source. Figure 3.4 shows ACK-based coordination procedure in this example. All candidates transmit acknowledgments in decreasing order of candidate priority: the first ACK slot belongs to A, the second slot belongs to B, and the third slot belongs to C. In Figure 3.4, we suppose that the ACK from A is not received by B, but C overhears the ACK of A. Suppose further that node B overhear C's ACK. If ACK packet does not contains IDs, node B would forward the packet, since to its knowledge it is the highest priority recipient. The fact that node C's ACK contains node A's ID indirectly notifies B that node A did receive the packet. Once node A has successfully determined itself as the responsible node, it forwards the packet.

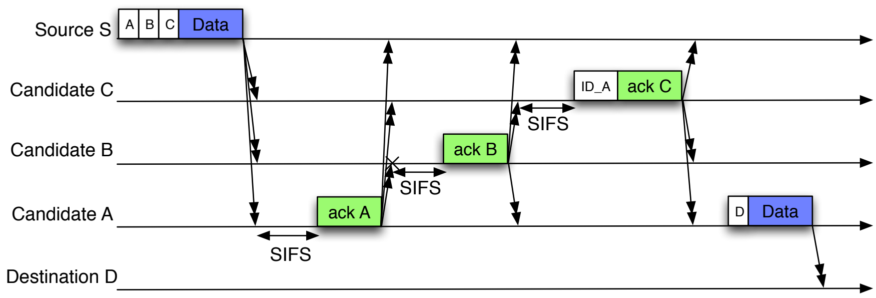


Figure 3.4: ACK-based coordination.

- *Token-based Coordination*: is first proposed in [70]. In token-based coordination, only the token holder can transmit packets and duplicate transmission is totally prevented but at the cost of extra control packets. The process is summarized as follows: In token-based coordination, a relay collects overheard packets and transmits them only when a token arrives. Tokens are passed along "connected candidates", where candidates are ordered in the way such that the i_{th} candidate can hear the $(i+1)_{th}$ candidate. Tokens are generated from the destination and flow from high priority relays (close to the destination) to low priority relays (close to the source). For a source-destination pair, it is possible to have multiple tokens passing among candidates. As acknowledgment information is included in tokens, a relay has a clear view of what packets higher priority relays have received and transmits only unacknowledged packets.

As an example shown in Figure 3.5, the arrows show how a token flows from destination D to source S through candidates $R1$, $R2$, $R3$, and $R4$. $R5$ is excluded since it can not receive tokens from $R4$. The box next to each node shows its packet collection statues: black for acknowledged, gray for collected, and white for empty. $R2$ has collected all four packets. As a token flows from D to $R1$ and arrives at $R2$, $R2$ learns that packet no.2 and packet no.3 have been received by higher priority candidates. Then, $R2$ removes the acknowledged packets (packet no.2 and no.3) and transmits unacknowledged packets (packet no.1 and no.4). Finally, $R2$ passes the token to $R3$ and waits for the next token from $R1$.

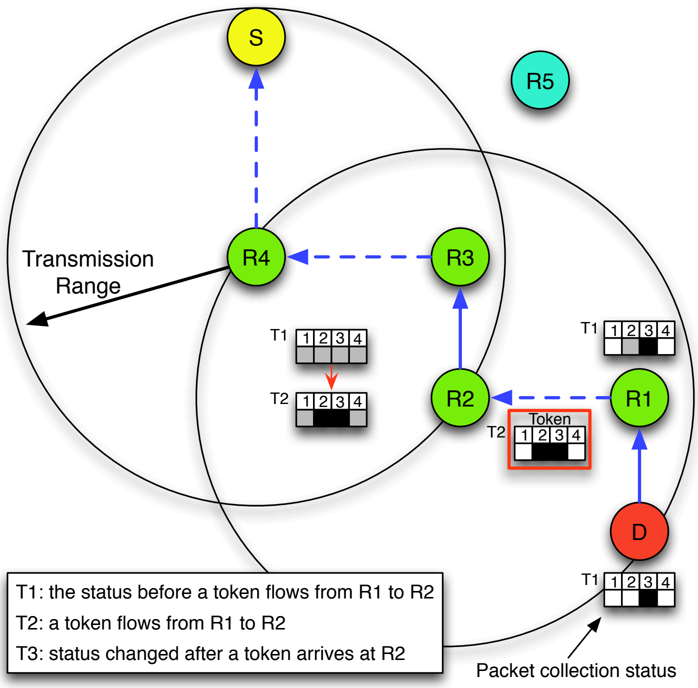


Figure 3.5: Token-based coordination.

The main advantage of this solution is the absence of duplicate transmission. However, the cost of extra control packets makes token-based coordination unsuitable for scenarios where the source-destination pair is too close (such as one or two hops). Though this solution does not have the waiting time overhead like the timer-based solution, candidates may be idle if no token arrives. Token generation speed at the destination is the key to keep the pipe full. In addition, candidates need to be "connected"; in other words, some relays are filtered out and this reduces the number of relays that can participate in the forwarding process. The impact of this filtering and token generation speed have not been analyzed.

- *Network coding-based Coordination*: Another approach to prevent duplicate transmission is to combine opportunistic routing with network coding. In [20], network coding was firstly introduced into packet routing. When referring to opportunistic routing, network coding can avoid duplicate transmissions without explicit coordination. A general concept of how network coding is applied to opportunistic routing is as follows. When transmitting packets from a source to a destination, a flow between them is divided into batches to code and decode. A batch contains several native packets, which are original packets without coding. Then, the source broadcasts random linear combinations of native packets, and relays forward the linear combination of received coded packets to the destination. Coded packets are decoded only when the destination has collected enough linearly independent coded packets. Eventually, native packets can be recovered by Gaussian elimination. This approach of coordination was first proposed by MORE [126]. The main advantage of using network coding is that there is no coordination overhead. However, the operations of encoding and decoding cost additional overheads at each node.

  In order to better clarify the advantage of combining network coding with OR, consider the example illustrated in Figure 3.6. Assume that source S transmits two packets a and b using candidate list $\{C_1, C_2\}$. Assume that C_2 receives both packets but C_1 receives only one of them. Node $C1$ transmits first because it is closer than C_2 to the destination. Node C_2 has the following three choices: forwarding a, b, or both a and b. In the network coding, node C_2 can forward a coded packet $a \oplus b$. When D receives transmitted packets from C_1 and C_2, it can decode and restore the original packets. It performs an XOR operation on the two received packets: $a \oplus b \oplus a = b$. Thus, no duplicate transmission occurs at D.

  However, using network coding with OR may lead to a high number of potential forwarders sending coded packets, and thus, resulting in redundant transmissions. There exists a trade-off between transmitting a sufficient number of coded packets to guarantee that the destination has enough coded packets to reconstruct the original packets, and avoiding in inject in the net-

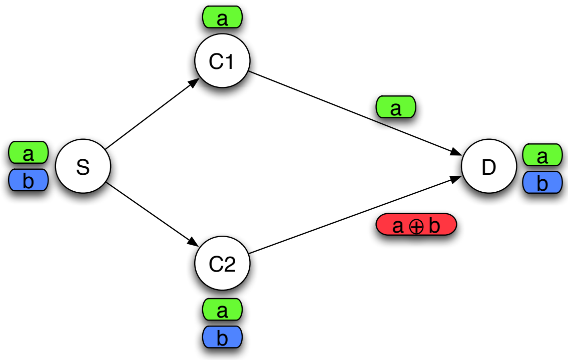


Figure 3.6: Network coding-based coordination.

work unnecessary beacon messages.

- *RTS-CTS Coordination*: Other mechanisms [77] [163] use explicit control packets exchanged immediately before sending a data packet. In this approach, the sender multicasts the Request-to-Send (RTS) packet to its candidate list (it is actually a broadcast control packet). The RTS contains all the candidate addresses that are ordered according to a metric. When an intended candidate receives the RTS packet, it responds a Clear-to-Send (CTS) packet. These CTS transmissions are sent in decreasing order of candidate priorities: the first candidate in priority transmits the CTS after a Short Interframe Space (SIFS), the second one after 2 $\times$ SIFS, and so on. When the sender receives a CTS, it transmits the DATA packet to the sender of this CTS (which would be the highest priority candidate that responded) after a SIFS interval. This ensures that other low priority candidates hear the DATA before they send CTS and suppress any further CTS transmission. All such receivers then set their Network Allocation Vector (NAV) until the end of ACK period. This mechanism is guaranteed to have a single winner and it can avoid duplicate transmission, with the cost of additional control overhead.

  Figure 3.7 shows an example of RTS-CTS coordination. Assume that there are three candidates A, B and C (A is with the highest priority and C is the lowest one). After receiving a RTS from source node S, candidates send CTS in order of their priorities. Here we assume that the first CTS, which belongs to A was not received by S, but the second one was received. When the sender S receives the first CTS from B, who has the second priority, S sends the data packet to it. Therefore, the highest priority candidate whose CTS is received by the source will forward the data packet.

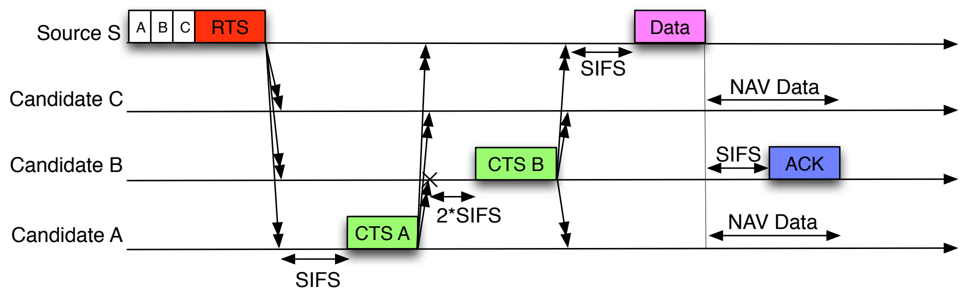


Figure 3.7: RTS-CTS coordination.

3.4.4 Candidate list-based Opportunistic Routing Protocols

Most of the first opportunistic routing protocols were proposed to use a candidate list. As described before, a candidate list is like a routing table, but contains multiple potential forwarders associated with a destination. Prior to data transmissions, the source node has to use some beacon messages to learn network conditions and build up this list. Then the list is embedded in the headers of data packets, such that when receiving a data packet, each node can infer its priority and wait for its turn to forward the packet. Many candidate list-based opportunistic routing protocols have been proposed, which use different mechanisms to build candidate lists. In the next sections, we briefly describe some representative approaches.

Extremely Opportunistic Routing Protocol

The extremely opportunistic routing protocol (ExOR) [31] is the most popular opportunistic routing protocol and one of the first protocols proposed to exploit the broadcast nature of wireless communications for increasing resilience and throughput. ExOR assumes that estimations of the path loss ratios for each pair of nodes are available at each node. Such loss ratios are evaluated by means of a metric similar to that of Expected Transmission Count (ETX) [48]. Although the authors suggest using a link-state flooding technique to distribute loss ratio estimations across the networks, in the performance evaluation they do not account for it by resorting to a simple centralized mechanism for loss rate distribution. To reduce the overhead relevant to the forwarding responsibility transfer mechanism, ExOR operates on batches of packets, that is, the receiving nodes buffer the packets until the end of the batch. Clearly, this increases the end-to-end delay and makes ExOR unsuitable for real-time applications. Moreover, the authors point out that the batches could badly interact with the TCP congestion avoidance mechanism, since in the presence of low loss rates, the window's size would limit the batch sizes.

The loss ratios are used for both candidate selection and the forwarder election. According to ExOR, the sender must include in the header of each packet the list of candidates, namely, the forwarder list (shown in Figure 3.8), prioritized by closeness to the destination according to the ETX-like metric. For a given batch, the forwarder list never changes. Thus, both the candidate set and the forward election are predetermined at the sender side during the transmission of the first packet of the batch. Clearly, this could potentially reduce the opportunism of the protocol.

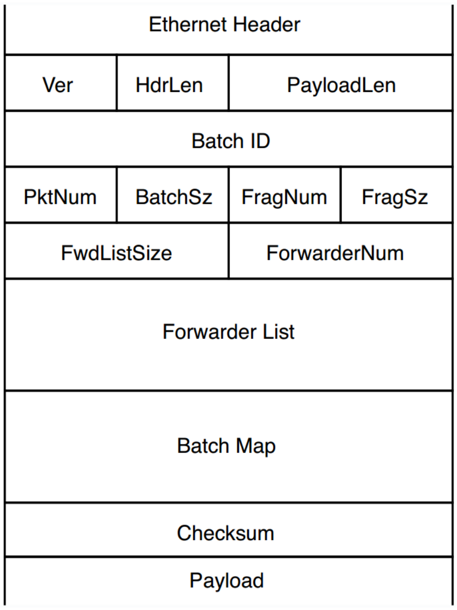


Figure 3.8: ExOR packet header.

The forwarder role change notification mechanism is implemented as an implicit strategy based on the batch map field in the packet header. This field lists, for each packet in the batch, the sender's best guess of the highest priority node that has received such a packet. From an operational point of view, when a node receives a packet, it first checks whether itself is included in the forwarder list. If so, it first buffers the packet and then updates its local batch map by replacing an entry if the packet's header indicates a higher-priority node. If the header does not include a higher-priority node, it simply discards the packet. The batch map acts like a gossip mechanism, carrying reception information from higher-priority nodes to lower priority nodes. When the batch is complete, each candidate forwards the packets that are not yet acknowledged by the highest priority candidates. Clearly, each forwarded packet also acknowledges the packets already received by means of the batch map stored in its header.

ExOR's duplicate transmission avoidance is a passive distributed procedure based on the gossiping mechanism implemented by the batch lists. Since there is no explicit cancellation of redundant transmissions, a candidate may need several responsibility transfer phases to become aware that its buffered packets are not innovative.

Simple Opportunistic Adaptive Routing Protocol

The simple opportunistic adaptive routing protocol (SOAR) [122] tries to solve one of the issues of ExOR, the lack of support for multiple simultaneous flows due to batch processing, by introducing an explicit forwarding responsibility transfer. Like ExOR, SOAR implements a predetermined candidate selection process based on the estimates of the path loss rates for each pair of nodes according to the ETX metric. The candidate set, namely the forwarder list, is prioritized by the closeness to the destination, and is included in the packet header. When a candidate receives a packet, it stores the packet in a buffer and sets a timer based on its priority (i.e., its position in the forwarder list). The higher the candidate priority is, the earlier the timer will expire. Since the node rebroadcasts the received packet when its timer expires, the other candidates will be aware of this transmission and can infer that a node has already taken the role of forwarding, and it will discard the packet.

Thus, like ExOR, both the candidate selection and the forwarder election processes are predetermined at the sender side on the basis of the loss rates. However, unlike ExOR, the forwarder role change modification mechanism of SOAR implements an explicit acknowledgment strategy based on the packet reception. Moreover, while ExOR implements a batch-level acknowledgment, SOAR adopts a packet-level acknowledgment, and each candidate is aware of the selected forwarder of each packet. Clearly, the priority-based timers require that all the candidates can hear each other. To ensure this condition, SOAR selects the allowed candidates at the sender side in order to avoid diverging routes. The candidate selection consists of two phases: (1) shortest-path candidate selection, that is, the selection of the nodes belonging to the shortest-path, and (2) near shortest-path candidate selection, that is, the selection of additional nodes that allow an increase in opportunities, but at the same time do not produce diverging routes.

Besides the implicit duplicate transmission avoidance based on the diverging route prevention, SOAR also implements an explicit mechanism based on selective and piggybacked acknowledgments (ACKs). The ACKs are selective since the same ACK can acknowledge multiple data packets, and they are piggybacked because if there is a data packet in the queue, the acknowledgment is stored in the data packet header, limiting the throughput related to the duplicate transmission avoidance.

The MORE Protocol

The MORE [126] protocol has been proposed to overcome the issues related to ExOR forwarding responsibility transfer, mainly the lack of spatial reuse. The key feature of MORE is the adoption of network coding into opportunistic routing at the packet level (intra-flow). MORE shares several features with ExOR. Both protocols implement a predetermined candidate selection process based on the estimates of the path loss rate for each pair of nodes, and they both adopt the ETX metric to estimate such loss rates. They both include the *forwarder list* in the packet header, prioritized by the closeness to the destination, and both operate on batches of packets. Finally, they both limit the candidate set size to reduce the overhead.

However, unlike ExOR, the forwarder election process of MORE allows multiple nodes to forward the packets. In fact, when a node receives a packet, it first checks whether it is in the packet's forwarder list. If so, the node then checks if the packet is an innovative one, that is, whether it is linearly independent of the packets of the same batch it has previously received. If both conditions are satisfied, the node stores the packet in the buffer and broadcasts a linear combination of the received packets.

Another difference between MORE and ExOR is that each packet sent by MORE is a coded packet, i.e., a linear combination of all the packets in the batch. Therefore, a duplicate transmission occurs every time a packet is linear dependent from the packets previously received. MORE does not use any explicit strategy to avoid duplicate transmissions, since there is no explicit cancelation of redundant transmissions. Instead, it resorts to the path loss rates to estimate the number of transmissions needed to forward a packet to a node closest to the destination, and such estimates implicitly limit duplicate transmission events. Each time that a packet has been received from the most distant node, a credit counter is incremented by such an estimate, and each time that the node forwards a packet, its credit counter is decremented by one.

An explicit acknowledgment strategy is used to notify the source that a batch is correctly received by the destination, and the ACK is routed using traditional unicast routing. Clearly, the batch size affects the MORE overhead because the smaller the batch sizes, the more frequent the ACKs. Moreover, the batch size also affects the duplicate transmission occurrence because the smaller the batch, the more likely the duplicate transmission event.

3.4.5 Beacon-less Opportunistic Routing Protocols

Based on the analysis above, we can observe that existing opportunistic routing protocols build candidate lists prior to data transmissions based on source node's knowledge of the network link conditions. This knowledge is based on the transmission of short control messages, called beacons. Each source node has to broadcast beacons to learn the network conditions and build the candidate lists. However, the candidate lists will be invalid if nodes move or link quality varies. If the source node wants to keep always updated with the latest network conditions, it has to frequently broadcast the beacons, which will significantly waste a lot amount of energy. To overcome such issues, various beaconless routing protocols have been proposed. In the following, we introduce the concept of *beaconless opportunistic routing*, which is a variant of typical opportunistic routing without using beacons. Beaconless opportunistic routing algorithms employ a reactive scheme to discover 1-hop neighbors and select packet forwarders. In particular the current forwarder broadcasts the packets to discover its neighbors. Neighbors receiving the packet can participate as next-hop candidates. They wait for a delay timer, which is calculated based on one metric (i.e., the distance to the final destination). Therefore, the candidate closest to the destination has the shortest timeout. When the timer expires, the respective candidate transmits first the packet and becomes the next hop. In the following, we present some representatives of beaconless opportunistic routing.

Implicit Geographic Forwarding

The implicit geographic forwarding (IGF) protocol [128] is one of the first beaconless routing protocols proposed in the literature. IGF combines MAC and network layers. The selection of the next hop is carried out at the MAC layer, and the actual delivery is done at the network layer. In IGF, the node holding the packet broadcasts a Request to Send (RTS) frame and waits for the first Clear to Send (CTS) response. Each neighbor receiving the RTS frame evaluates its own suitability as next hop. The neighbor providing the largest advance towards the destination is preferred and should answer first. Finally, at the Network layer, the forwarding node transmits the data message and the selected neighbor confirms the reception by answering with an Acknowledgment message (ACK).

IGF includes two optimizations to reduce the number of responses and collisions. The first mechanism avoids simultaneous responses from neighbors based on timers. The second scheme cancels unnecessary responses when other neighbors' responses are overheard. Upon receiving a RTS message, each neighbor sets a timer to wait before answering with a CTS message. The timer value depends on the reduction in distance towards the destination provided by the node plus a random component. Thus, neighbors located closer to the destination answer first. Besides, neighbors overhearing an earlier CTS from another neighbor cancel their own timers.

IGF defines a forwarding area so that all nodes within that area are separated by a distance lower than the theoretical radio range. That is, in theory, all nodes inside it can hear one another. Only those nodes located inside the forwarding area can take part in the selection process. However, in practice, radio propagation can make nodes within the forwarding area not to overhear some answers. Also, as a side effect, the use of a forwarding area may neglect some neighbors providing a higher advance because of being outside that area.

Beacon-Less Routing Protocol

The beaconless routing (BLR) [97] relies on a distributed contention process as the only way of determining the next-hop. BLR selects a forwarder in a fully distributed manner among all its neighboring nodes without having any information about their position or even about their existences. In the normal working mode (greedy mode), data packets are just broadcasted by the source node, and the protocol takes care that just one of the receiving neighboring nodes will actually forward the packet. This is accomplished by computing a timer called *Dynamic Forwarding Delay (DFD)* at each neighbors. The calculation of DFD is dependent on the position of the node relative to the current forwarder and the destination.

Among all neighbors that provide geographic progress, the one in the best position forwards the data packet first. The other neighbors cancel their scheduled transmissions, when they overhear the data packet. To ensure that all nodes detect the forwarding, only nodes within a certain forwarding area take part in the forwarding contention. Furthermore, passive acknowledgments are used. That is, by detecting the transmission of the packet, the previous forwarder could conclude that the packet it sent out was successfully received by its next hop.

Additionally, BLR includes a face strategy (backup mode) to deal with local maximum. The current forwarder broadcasts a short request, and all neighbors reply with a packet indicating their positions. If there is a neighbor located closer to the destination than the current forwarder, the neighbor is then chosen as the next-hop. Otherwise, the actual forwarder extracts a planar subgraph (e.g., Gabriel Graph) for its neighbors and forwards the packet according to the right-hand rule, as in the GPSR protocol. Figure 3.9 shows the greedy mode of BLR, where node P chooses node A as its next-hop forwarder, since A provides the biggest progress among all the neighbors of node P. Figure 3.10 depicts the procedure of backup mode of BLR. As we can see, when node S does not have any neighbor providing distance progress, then it will start the extraction of a planar subgraph for its neighbors and forwards the packet according to the right-hand rule. Therefore, it chooses node A as next-hop.

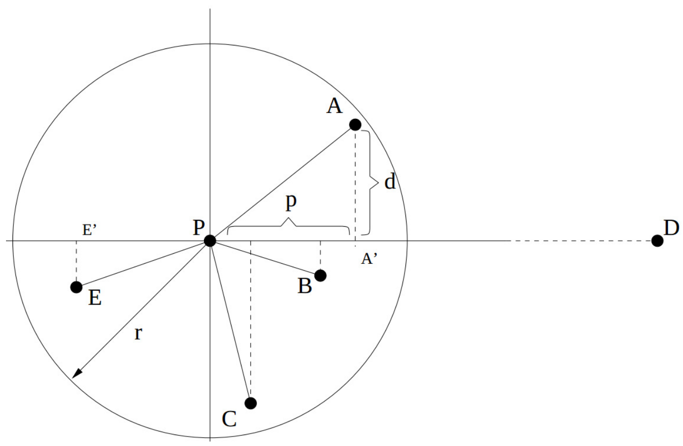


Figure 3.9: BLR working in the greedy mode.

Contention-Based Forwarding Protocol

In Contention-based Forwarding [57], the routing procedure basically has two phases: contention process and suppression phase. In the contention process, the

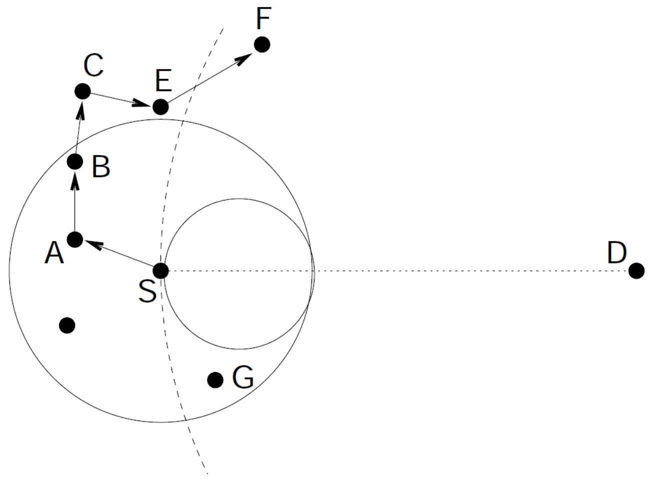


Figure 3.10: BLR working in the backup mode.

current packet holder broadcasts the data packet and waits for its neighbors to determine which one will be the next forwarder in a distributed contention process by themselves. During the contention process, candidate neighbors compete for being the packet forwarder by setting timers related to their actual positions. The neighbor providing the most advance towards the destination waits for the shortest time before forwarding the data packet. The remaining candidates cancel their timers when they hear the transmission from the winning node.

The second phase is the suppression of redundant messages. The suppression phase is used to reduce the chance of accidentally selecting more than one node as the next hop as well as to reduce the overhead of duplicated packets. Three different suppression schemes are proposed. The basic scheme consists of canceling timers after hearing a transmission from another neighbor. The area based scheme defines a forwarding area as in IGF. Authors of CBF propose three different areas: Sector, Reuleaux triangle and Circle. Their results show that a Reuleaux triangle is the forwarding area with better performance than Sector and Circle in terms of packet duplications and average advance in each hop. Finally, a third suppression mechanism is defined, called active suppression. The active suppression is equal to the RTS/CTS approach proposed in IGF that allows the forwarding node to determine which neighbor is selected as the next hop among the neighbors whose CTS frames were received. The active scheme selects explicitly a unique next hop preventing packet duplications. Multiple nodes may send a CTS control packet, but only one is selected because the forwarding node acts as a central authority. Obviously, this requires the additional overhead of RTS/CTS control packets.

Beaconless On-demand Strategy for Sensor Networks Protocol

The beaconless on-demand strategy for sensor networks protocol (BOSS) [124] was designed considering that packet loss and duplicates are common in realistic wireless communication. To avoid duplicates, BOSS employs a three way handshake to forward packets in a similar way to the RTS/CTS scheme used in IEEE 802.11 [5]. BOSS provides a retransmission mechanism based on both active and passive acknowledgments to improve reliability without increasing the controlling overhead. A contention timer function (called *Discrete Dynamic Forwarding Delay (DDFD)*, shown in Figure 3.11) is included at each node to decrease the collisions and the number of answers during the neighborhood discovery. DDFD divides the neighbors area into various sub-areas according to the distance advance towards the destination. Thus, the neighbors located in a high-advance sub-area answer before the remaining neighbors placed in low-advance sub-areas. Moreover, DDFD also prevents collisions among neighbors in the same sub-areas.

The major contribution of BOSS is the use of a DATA message including the data payload to discover neighbors. The reason is that bigger messages are often more error-prone than short ones. For this reason, short RTS and CTS messages can be sent over a link that a big DATA message cannot. By sending first the big DATA messages, BOSS performs the next-hop selection only among those neighbors that successfully receive the data payload.

Unlike other beaconless protocols, the DDFD timer in BOSS combines a uniformly distributed value that is dependent on the geographic advance of each node together with a random value. With this definition, the total delay does not mix the responses from neighbors in different sub-areas. Thus, the function reduces the number of responses and the probability of generating simultaneously responses from neighbors, which are in the same sub-area. Other proposals calculate the waiting time solely based on the distance advance value. In those cases, the forwarding node could have several neighbors providing similar distance advances, and therefore increases the probability of collisions.

3.4.6 Drawbacks of Existing Opportunistic Routing Protocols

The existing opportunistic routing protocols have their drawbacks and thus are limited to certain application scenarios. In the following, we briefly discuss the problems of both the candidate list-based opportunistic routing and beaconless opportunistic routing. These limitations trigger the design of a more efficient and flexible opportunistic routing protocol that is able to be compatible with dynamic features of wireless mobile ad-hoc and sensor networks.

Problems of Candidate list-based Opportunistic Routing Under Dynamics

Candidate list-based opportunistic routing protocols share the same feature that a candidate priority list has to be generated using beacon messages prior to data

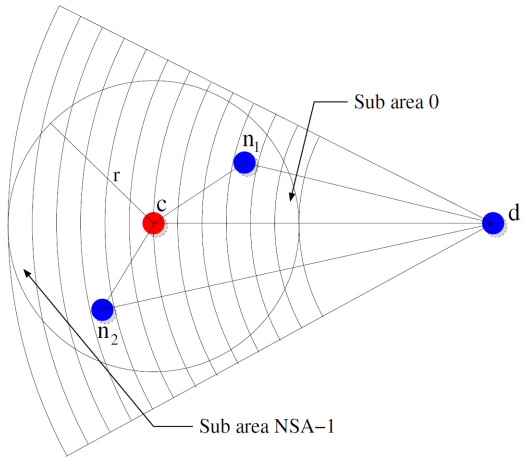


Figure 3.11: Area division for the DDFD timer in BOSS.

transmission. The criterion to rank the candidates is mostly the link quality of different nodes. The list is then included in the header of every data packet, such that when receiving a data packet, a node can check the list and infer whether it has been "defined" as a forwarder candidate of this packet and what is its priority. However, this list is usually generated only once prior to data transmission in a static way. When the network topology changes, due to either node mobility or wireless channel variation, the link quality of different nodes will certainly change. This means the priority list that was generated based on an old network topology is no-longer valid. In MANET environments, a fast candidate selection and prioritizing process has been proven to be preferable, where the candidate list can be updated frequently such that the generated list can reflect the latest situation of the network [46]. However, the frequency of updating the candidate list will be high for a dynamic network where network topology constantly changes over time. This leads to significant amount of additional overhead, which might even eliminate the benefits of opportunistic routing. Therefore, the candidate list-based opportunistic routing protocol will encounter severe problems in dynamic environments.

To show the problem of candidate list-based opportunistic routing under dynamic environments, we present the results of invalid priority lists under different mobility situations (using the simulation framework presented in Chapter 4). We take the most representative candidate list-based opportunistic routing protocols of ExOR, SOAR, and MORE as examples, since the other candidate list based opportunistic routing protocols share the principle of ranking the candidates. In Figure 3.12, the Y index indicates the percentage of the pre-generated candidate priority list that is not valid anymore, and the X index is the average moving speed of nodes that are moving with Random Waypoint mobility model [13]. We can clearly see that, for all of the these three protocols, the percentages of invalid candidate list increase significantly as node speeds increase. This means, when the nodes become mobile, most node-rankings on the priority list will be invalid, and thus, the ranking of the candidates' priorities becomes useless. As a conclusion, candidate list-based opportunistic routing selects packet forwarders based on a priority list, which is generated before data transmissions. However, the network topologies at the moment of generating the priority list and the moment of data transmission are usually different due to node movements or channel variations. Therefore, candidate list-based opportunistic routing is not suitable for dynamic situations where network topology changes constantly.

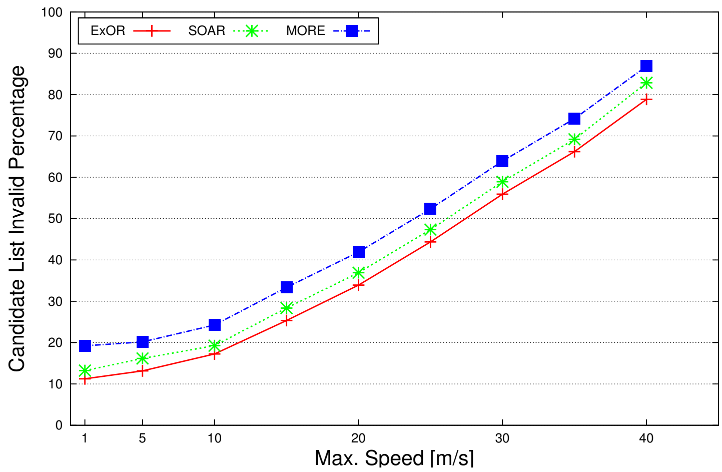


Figure 3.12: Percentage of invalid priority list under mobility.

Problems of Beaconless Opportunistic Routing Considering A Single Context

Beaconless opportunistic routing protocols are promising for wireless environments, since network nodes do not have to transmit beacon messages to keep updated with varying network conditions. However, most of the existing beaconless opportunistic routing protocols select packet forwarders based on solely one network metric, such as link quality, geographic distance, or hop count. This is still similar to conventional MANET routing, where a least cost path-based approach is favored. The cost metric reflects the criterion to be optimized. However, the single metric-based approach has the drawback of fast resource depletion along the preferred paths. Therefore, a beaconless opportunistic routing approach considering multiple types of metric is promising for MANETs or WSNs.

In the next section, we describe context aware communication and context-aware routing, which are new concepts of building communication protocols based on multiple types of context information at run-time that have impact on system performance.

3.5 Context-aware Routing

In complex systems, such as MANETs or WSNs, various context information have impact on system performance. Different types of context information should be considered in protocol design. Context is any information that can be used to characterize the situation of an entity and the environment the entity is currently in. Context-awareness enriches entities with knowledge of the status of themselves and the environments, which enables the automatic adaptation to the changing environments. In the scope of this thesis, context means any information with impact on routing decisions, e.g. geographical location, energy level, connectivity, etc. There are some works on context-based routing in MANETs or WSNs. However, most of these works focused only on a single network metric, such as energy (energy-aware routing) or geographic location (geographical routing). An efficient routing solution should simultaneously take into account multiple contextual metrics that have impact on routing performance.

One of the main contribution of this thesis is the proposal of a context-aware opportunistic routing protocol. This section first presents a formal description and definition of the terms “context” and “context-aware routing”, as they are frequently mentioned in the scope of this thesis. Then a requirement analysis for the protocol design is done and the protocols that have been designed based upon the requirements are presented.

3.5.1 Definition of Context

For a formalization of context-aware routing, a definition of context and context-awareness is required at the first place. A definition of context is given by Dey and Abowd [18]:

“Context is any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.”

Within the scope of this thesis, the use of contextual information is not restricted to the interaction between users and applications, but the interaction among the devices within a mobile ad-hoc network or a wireless sensor network. Take wireless sensor network as an example, the term “context” refers to the situation and the environment of the sensor nodes, which are objects in the terminology of the given general definition. The concrete context metrics of the sensor node can be, for example:

- *location*
- *energy level*
- *connectivity*

- *sensed data*
- *individual preferences*
- *mobility*
- *traffic rates*
- *link quality*

In the scope of this thesis, context can be any information that can have impact on the routing process. Furthermore, the handling of contextual information is not only dependent on context values (such as the geographic position), but also relevant with semantics, which means the correct interpretation of the given values. Therefore, the semantics can be regarded as an integral part of the context information. A full description of a context criterion, including the semantics, comprises the following:

- *information on what kind of context is described, e.g., delay*
- *information about the scale, e.g., seconds or minutes*
- *possible value ranges, e.g., [-1,1]*
- *value interpretation rules, e.g., target values.*

The description of a current context then at least consists of the description of relevant criteria as defined above, as well as the current context values for all these criteria. Additionally, it can also contain rules for correct interpretation of the combined context. In this thesis, we classify the context into three groups: local, link, and global context.

- *Local context*: local context includes local attributes of network nodes, such as location, mobility and residual energy.
- *Global context*: global context includes diverse attributes of the network, such as network topology and traffic conditions.
- *Link context*: link context includes various properties associated with wireless links, such as link quality and bandwidth.

Due to the dynamic nature of mobile ad-hoc networks and wireless sensor networks, it is expensive to obtain and maintain global contexts. Therefore, local and link context should be exploited efficiently to improve system performance.

Context-aware means that an entity performs an action while taking into account its own current context and the context of those it is interacting with. In the scope of routing in wireless ad-hoc networks and wireless sensor networks, context-aware routing refers to routing methods that use the context information that are mentioned above to determine routes. The concrete decision on which context metrics to use should depend on the specific requirements of the application (e.g., required speed or limited delay).

3.5.2 Context-aware Protocols

Context-aware systems are able to adapt their operations to the current context values without explicit user intervention and thus aim at increasing usability and effectiveness by taking the latest environment context into account. Context-awareness characterizes a system that consults context information when performing certain tasks. Different types of context-aware system, middleware, management platforms for wireless networks have been proposed [162] [58] [29] [139].

In the scope of this work, context means any information with impact on the routing process. When referring to packet routing in dynamic networks, context-aware routing means routing methods make use of the context information to select next hops. Some efforts have been made to efficiently make use of the network context information to improve routing performance. In [38], authors proposed to collect context information, such as location, distances, interest profile, and aggregate traffic flows, to help the content distribution in a vehicle ad-hoc network. [50] also focused on the application of context utilization into the domain of vehicle network. It collected the context information of road conditions, vehicle density, average vehicle speed and roadside facilities to help the task of efficient route planning, traffic load balance, etc. In [140], Xiao et al. analyzed the context of intra-community centrality, inter-community closeness on the performance of social network. In [144], You et al. exploited the context information like the size of network holes, or the remaining energy of nodes, to improve the performance of geographic routing in wireless sensor network environment. In the area of wireless mesh networks, [71] proposed adopting a reconfigurable context management system to simplify the task of accessing a variety of information required by adaptive routing protocols and to hide the low-level complexities of information sources management. In [45], authors proposed a context and content-based routing protocol (CCBR) for mobile sensor networks. CCBR adopts content-based addressing to effectively support the data-centric communication paradigm usually adopted by WSN applications. It also takes into account the characteristics of the sensors to filter data. In [129], a context-aware and adaptive security scheme for wireless networks was proposed. The key contribution is the implementation and evaluation of a context-aware and adaptive manager (CASM) that selects appropriate security protocols for specific wireless network applications in real-time.

3.5.3 Consequences of Context-aware Routing

The more relevant context parameters are involved in context-aware routing, the more accurately the protocols can understand the network. Then, the protocols can make better decisions to choose the packet forwarder. However, for proactive MANET routing, this either causes a lot of signaling messages relevant to the updates of routing tables, as the network members should be aware of the context changes, or some outdated context information have to be accepted.

For reactive routing without any route caching, the frequency of context changes is of minor importance. The routes are determined by route discovery as they are needed, independent of whether the context has changed or not. When there is a frequent need for routes, route caching can be used to reduce the frequency of route discoveries, and also the signaling overheads. If the cache time is too long and the context changes are too frequent, the cached routes can of course be outdated if the context has changed in the meantime. Therefore, reactive routing is not totally independent of the context change frequency any more.

Reactive routing has another advantage over proactive routing when it comes to context-aware routing: by specifying a parameterized evaluation function in the route requests, it provides greater flexibility. Route discovery with different context choices, different evaluation functions, etc., can be performed simultaneously in a network. To achieve the same flexibility in a proactive routing scheme, all potentially relevant context information would constantly have to be kept up to date, which requires not only many signaling overheads but also more local memory, which will be a severe problem for resource-constrained devices like wireless sensor nodes.

Therefore, to answer a question like whether proactive or reactive routing is more appropriate for context-aware routing, the answer really depends on several constraints, of which the frequency of context changes, the interval between the requests for a new route, and the flexibility are the most important ones. In case of slow context changes and frequent needs for routes, proactive routing is the preferred choice, especially if there are low requirements on flexibility. In case of fast context changes, high flexibility demand and/or infrequent need for routes, a reactive scheme should be chosen.

However, all the problems relevant to proactive routing or reactive routing will be completely eliminated if the beaconless opportunistic routing approach is employed. Because beaconless opportunistic routing does not maintain any routing tables, and thus, no signaling overhead will be paid. The forwarder is selected after the packet transmission at the receiver side, therefore there is no need to store any context information. The decision is made solely based on the instantaneous context values at the moment of packet reception. As a conclusion, beaconless opportunistic routing does not create any additional overhead when integrated with context-aware routing, and thus, is a promising solution for packet transmission in dynamic environments.

3.5.4 Context Combination Rules

In the design of wireless routing protocols, packet transmission is one part of great importance. Another part, which is of similar importance, is the decision-making procedure at each node, i.e., algorithms to choose one route from multiple route alternatives. Most routing algorithms in communication networks only use a single criterion to choose forwarders or routes. When routing is context aware, however, multiple types of context criteria have to be considered simultaneously. As there are multiple criteria that can influence the routing decision, these criteria have to be combined together and the final decision is to be made with the help of a multi-criteria decision algorithm.

There are some routing algorithms that use more than one criterion to decide routes. A multi-criteria routing approach for wireless sensor networks was presented in [91], where a Multi-Criteria Routing (MCR) algorithm based on three criteria, namely *remaining energy per node*, *power consumption model*, and *group membership* is applied. In [96], authors focused on three parameters of *energy*, *latency*, and *bit error rate*. A *Normalized Weighted Utility Function (NWAUF)* was used to find the best route with respect to the set of criteria. This NWAUF is based on normalized criteria values and weight is between 0 and 1 with a cumulative sum of 1. The most significant difference between these two multi-criteria routing methods of MCR and NWAUF is the point of aggregation within the process of route evaluation. In MCR, there is a separate evaluation and ranking for each of the criteria before the multi-criteria ranking is determined based on the separate rankings. The drawback of this method is that by only using the ranking positions in the aggregation, some information is lost.

[93] proposes EM-GMR for wireless sensor network routing, which combines three context attributes of relative distance to sink, remaining energy, and node mobility. [102] utilizes the node movements and resource predictions for the selection of data forwarding direction within a wireless sensor networks. In [102], each node evaluates the change rates of its connectivity, it collocation with sinks and remaining energy. Based on the history of these parameters, a prediction is made using timer series forecasting and the forecasted values are combined into a delivery probability for data delivery to a sink. This combination is computed locally on the node by a weighted summation of utility functions related to each of the context criteria. Information about the current delivery probability and the available buffer space is periodically exchanged with the neighboring nodes. However, most of the existing context-aware routing approaches are either proactive or reactive. Therefore, they can not fully avoid the problems presented in section 3.5.3.

3.6 Adaptive Communication Mechanism

When multiple context information are combined together, one problem is how to assign weights of different context information. The weights of context information reflect the importance of different context information and should depend on their values. This is because fixed weights only reflect the relative importance of different context information and fail to consider instantaneous situations. Therefore, an adaptive mechanism, which is able to adjust context weights at run-time, is promising for dynamic features of MANETs or WSNs.

Mobile ad-hoc and sensor networks have the feature of inherent uncertainty, which makes packet routes unstable. Therefore, adaptive communication, which enables network nodes to adjust their behaviors at run-time according to the latest network situations, is a promising approach to provide satisfactory performance in dynamic environments.

There are several routing protocols that adapt their behaviors based on the network characteristics to enhance routing performance. FSR [109], Fisheye State Routing, is a link-state protocol that exchanges periodic link-state information. The period of link state propagation is determined by the distance to the destination. ADV [32] (Adaptive Distance Vector algorithm) exhibits on-demand characteristics by varying the frequency and the size of routing updates. Some research has been made to replace reactive protocols with timer-directed route discoveries to produce backup routes prior to losing the primary link [104]. Their protocols used a fixed timer across all nodes, and the value of the timer is determined off-line from the history of link-failure statistics.

Adaptive and low-power routing in wireless sensor networks proposes dynamic change of parents in routing. DSF [59] selects the next hop of a packet based on the sleeping schedules of neighboring nodes and other metrics such as delay, reliability, and energy consumption. However, DSF focuses on synchronized networks. Furthermore, it requires iterative message exchange to stabilize the forwarding schedules of all nodes, leading to additional control overheads in the presence of dynamic links. The Backpressure Routing Protocol, BRP [98], forwards packets to the neighbor with the lowest queue level. This improves throughput when compared to traditional unicast routing. However, BRP can only be applied when the overall system is saturated, i.e., nodes always have packets to forward. BRE [24] reduces hop counts by exploiting link dynamics: when a far-ranging link of intermediate quality becomes temporary available, BRE uses it as a shortcut in the routing tree. In duty-cycled environments, BRE shows two key limitations: (1) its short-cuts are only stable for a couple of milliseconds, making it difficult to exploit them in low traffic scenarios. (2) In BRE, nodes overhear data traffic to determine possible short-cuts. This is not practical when nodes are asleep most of the time.

3.7 Decision Making with Analytic Hierarchy Process

Adaptive communications may require selections or decision making among multiple alternatives. The method used in this thesis for the decision making in parametrical and structural adaptation is Analytic Hierarchy Process (AHP) [123]. AHP is a flexible decision making process that helps to set priorities and make the best decision when both qualitative and quantitative aspects of a decision need to be considered. The method reduces a complex decision to a series of one-to-one comparisons and provides rationale for the results.

AHP is one of the most widely used multi-criteria analysis approaches. It allows users to assess the relative importance of multiple criteria or multiple options against the given criteria in an intuitive manner. If quantitative ratings are not available, policy makers or assessors can still recognize whether one criterion is more important than another. Therefore, pairwise comparisons are appealing to users. The AHP algorithm, as a compensatory method, assumes complete aggregation among criteria and develops a linear additive model. The weights and scores are achieved basically by pairwise comparisons among all options with each other.

The basic procedure to carry out AHP consists of the following steps:

- *Structuring a decision problem and selection of criteria:* The first step is to decompose a decision making problem into its constituent parts. In its simplest form, this structure comprises a goal or focuses at the topmost level, criteria at the intermediate levels, while the lowest level contains the options. Arranging all the components in a hierarchy provides an overall view of the complex relationships and helps the decision maker to assess whether the elements at each level are of the same magnitude so that they can be compared accurately.

- *Priority setting of the criteria by pairwise comparison (weighting):* For each pair of criteria, the decision maker is required to respond a question such as "How important is criterion A relative to criterion B?" Rating the relative "priority" of the criteria is done by assigning a weight between 1 (equal importance) and 5 (extreme importance) to the more important criterion, whereas the inverse of this value is assigned to the other criterion of the pair. The weights are then normalized and averaged in order to obtain an average weight for each criterion.

- *Pairwise comparison of options of each criterion (scoring):* For each pairing within each criterion the better option is awarded a score, again, on a scale between 1 (equally good) and 9 (absolutely better), while the other option in the pairing is assigned a rating equal to the reciprocal of this value. Each score records how well option x meets criterion y. Afterwards, the ratings are normalized and averaged.

- *Obtain an overall relative score for each option:* In a final step, the option score are combined with the criterion weights to produce an overall score for each option. The extend to which the options satisfy the criteria is weighted according to the relative importance of the criteria.

In general, AHP can effectively support decision making with regard to complex sustainability issues and can help to recognize and define a problem in detail. It is widely used to decompose a decision making problem into its constituent parts, which are then structured hierarchically. Multiple and even conflicting goals can be taken into consideration.

As a simple example, Figure 3.13 shows how the AHP theory can be used to select the most suitable leader from a field of three candidates. The evaluation factors to be considered are experience, education background, charisma, and age. According to the judgement of the decision makers, Dick is the strongest candidate, followed by Tom, then Harry. The figure shows the AHP hierarchy at the end of the decision making process. Dick is the preferred alternative, with a priority of

0.493. He is preferred about a third more strongly than Tom, whose priority is 0.358. Experience is the most important criterion with respect to reaching the goal, followed by Charisma, Education background, and Age. Details of the procedure can be found in [21].

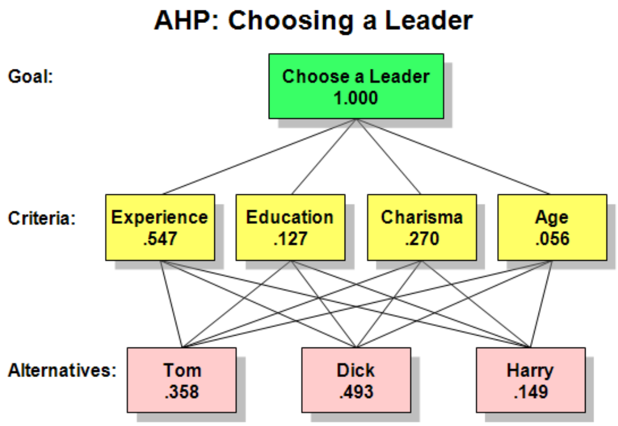


Figure 3.13: An example of using AHP to choose a leader from three candidates.

3.8 Duty Cycling in Wireless Sensor Networks

Wireless sensor networks are characterized by multi-hop lossy wireless links and severely resource-constrained sensor nodes. Among the resource constraints, energy is the most critical one since sensors are usually powered by batteries with limited energy capacity, while the deployment of WSN has the lifetime requirement of couple of years. To close the gap between limited energy and long-term deployment requirement, a reliable and energy-efficient routing solution is an essential task for WSNs.

It has been observed that low power, low range sensors consume significant amount of energy while idling compared to the energy consumed during transmission and reception. Consequently, it has been widely considered a principle method of energy conservation to turn off sensors that are actively involved in sensing or communication. By operating at a low duty cycle, i.e., reducing the fraction of time that a sensor is active, sensors can conserve energy and consequently increase their lifetimes. This is especially applicable in scenarios where sensors are naturally idle for most of the time (e.g., detection of infrequent events such as fire, fault, etc., and transmission of very short messages). In some cases we may also be forced to put sensors in a power-saving (or sleep) mode for a large fraction of the time in order to meet a certain lifetime requirement.

By putting sensors into sleep mode, the lifetime of sensor network can be significantly prolonged. However, the price we pay is that the network communication and sensing capabilities become intermittent. The intermittent sensing capability disrupts the sensing coverage of the network, i.e., certain areas of the network may not be covered by any sensor and consequently the events within that area fail to be detected on time. Similarly, turning off radio transceivers results in loss of connectivity among nodes, which reduces the possibility that nodes concurrently overhear a packet transmission. As a result, it prevents the spatial reuse in the forwarding process. There are many challenges in designing low duty-cycled wireless sensor networks. The temporary unavailability of nodes can adversely affect both the coverage and connectivity of the network. Therefore, there is a trade-off between energy saving and performance degradation.

Many duty-cycled based MAC protocols have been proposed to let the nodes go into sleep mode periodically or aperiodically. Synchronous protocols, such as S-MAC [143], D-MAC [94] and FPS [68], determine the perfect duty-cycle of all sensor nodes. Synchronous approaches do not completely eliminate the idle listening time, but help to reduce it. Most synchronous approaches are based on establishing communication schedules, so that nodes communicate on known time-slots. S-MAC schedules all nodes in the network to wake up, listen, and then sleep at the same time. D-MAC and FPS schedule the node wake-up time along a dissemination tree. The tree is constructed during configuration phase and then nodes are assumed to be static thereafter. Asynchronous approaches have been developed and tested in static wireless sensor networks. These approaches are robust: they do not require fixed topology or precise time synchronization and therefore are more suitable for mobile scenarios. X-MAC [37] protocol uses a shorter preamble such that it retains the advantages of low power listening. Therefore, it has the feature of low power consumption, simplicity, and a decoupled sleep schedule of a transmitter and a receiver. WiseMAC [54] optimizes low power listening by making senders learn the wake-up schedules of their neighbors. The nodes thus can use very short preambles to selectively wake up a specific neighbor. The main drawback of asynchronous protocols is that they have to deal with high discovery cost. Discovering neighboring nodes usually requires staying awake for a longer time either continuously transmitting beacons or overhearing messages from potential neighbors.

MaxMAC [73] [74] is an energy efficient MAC protocol that targets at achieving maximal adaptivity with respect to throughput and latency. By adaptively tuning essential parameters at run-time, the protocols reaches the throughput and latency of energy-unconstrained CSMA in high-traffic phases, while still exhibiting a high energy-efficiency in periods of sparse traffic. MaxMAC operates similarly as existing energy-efficient MAC protocols in low traffic load situations, it is able to maximally adapt to changes in the network traffic load at run-time. By taking advantages of design principles for energy-efficient MAC protocols, MaxMAC introduces novel run-time adaptation techniques to effectively allocate the energy-costly radio transceiver truly in an on demand manner.

Challen et al. proposed IDEA [40], Integrated Distributed Energy Awareness, a sensor network service enabling effective network-wide energy decision making. IDEA integrates into the sensor network application by providing an API allowing components to evaluate their impact on other nodes. IDEA distributes information about each node's load rate, charging rate, and battery level to other nodes whose decisions affect it. Moreover, IDEA enables awareness of the connection between the behavior of each node and the application's energy goals, guiding the network toward states that improve performance. One key feature of IDEA is the adaptivity of sensor duty cycles based on the awareness of energy level, which enables a sensor node to sleep longer if its energy depletion rate is high.

One of the contributions of this thesis is that it adopts the idea of opportunistic routing to wireless sensor networks, with the consideration of the specific demands of wireless sensor node. When adopting opportunistic routing to wireless sensor networks, we have one more source of dynamics: the uncertainty due to the sleep scheduling, in addition to the uncertainties caused by wireless transmission and node mobility. Opportunistic routing was originally proposed to improve the throughput of multihop wireless mesh networks, where wireless transceivers are always on and energy consumption is not an important concern. In contrast, wireless sensor nodes are commonly duty-cycled to ensure long lifetime, limiting the use of overhearing for opportunistic routing. Therefore, a detailed investigation about how to integrating opportunistic routing to wireless sensor networks is needed, in order to fully exploit the benefits of opportunistic routing in wireless sensor networks.

3.9 Software-based Energy Profiling in Wireless Sensor Networks

Energy is one of the most critical constraints for the design and implementation of wireless sensor networks and their protocols. By being able to estimate the energy consumption of the sensor nodes, applications and routing protocols can make reasonable decisions that increase the lifetime of the networks. According to the experiment results in [108], the network lifetime can be improved by around 52% if the information about energy consumption is available.

While commonly used network metrics such as packet delivery ratio, end-to-end delay, or throughput can be easily obtained in real-world WSN testbeds, measuring the power consumption of sensor nodes is much harder: costly high-resolution digital multi-meters need to be hooked to the nodes in order to sample the varying low currents and voltages. For years, experimental research in the field of energy-aware and energy-conserving protocols has required long period of measurements. Existing simulation tools provide different degrees of analysis in communication, application, and energy domains. However, none of them provides enough flexibility to estimate the consumed energy for a wide range of wireless sensor hardware platforms. This, in turn, does not allow researchers to use them to study wireless sensor networks from the perspective of energy consumption. On the other side, current wireless sensor platforms, such as Tmote Sky sensor node, do not provide hardware mechanisms for measuring the energy consumption of the sensor node. Furthermore, the unique characteristics of sensor network applications make the hardware-based energy measurement difficult [78].

Therefore, a software-based on-line energy profiling mechanism is promising for small sensor nodes. The mechanism runs directly on the sensor nodes and provides real-time estimates of the current energy consumption. Some efforts have been made in this domain and most of them proposed simple state-based energy estimation models. Younis and Fahmy suggested the use of a simple linear model for estimating energy consumption [145]. However, their model requires extensive changes to all applications and protocols that use it.

In [52], Dunkels et al. proposes that an intentionally simple mechanism for on-line node-level energy estimation can provide a good estimation of energy consumption. Their solution motivated the need for software-based on-line energy estimation, because only on-line estimation mechanisms running on the node itself enable the node to take energy-aware decisions about routing, clustering or transmission power scheduling. The authors derive the state-based models (3.1) and experimentally correlate the estimated energy with the sensor nodes lifetime.

$$E = (I_m t_m + I_l t_l + I_r t_r + \sum_i I_{c_i} t_{c_i}) \times V \tag{3.1}$$

In this energy model, V is the supply voltage, and I_m, t_m are the current draw of the node's microchip and the time it has been fully active. The variable I_l and t_l correspond to the current draw and time of the microchip in the low power mode. Variable I_t and t_t correspond to the current draw and time of the radio transceiver in the transmit mode, and I_r and t_r in receive mode. Furthermore, I_{c_i} and t_{c_i} denote current and time of operation of further onboard components.

Hurni et al. examines the accuracy of different software-based on-line energy estimation techniques [76]. They evaluate today's most widespread energy estimation models in order to investigate whether the current methodology of pure software-based energy estimation running on a sensor node itself can indeed reliably and accurately determine energy consumption.

In [63], authors proposed PowerBench, a system elaborates on the difference between their software-based energy estimations and the physically measured energy consumption of sensor nodes running different energy-efficient MAC protocols. The model (3.2) applied is the same as on S-MAC [143] and B-MAC [112]. The consumed energy E is calculated as the sum of the total time spent in the receive state multiplied by the respective power level $T_{rcv}P_{rcv}$, and the respective terms for the transmit and sleep states ($T_{slp}P_{slp}$ and $T_{tx}P_{tx}$).

$$E = P_{rcv}T_{rcv} + P_{tx}T_{tx} + P_{slp}T_{slp} = (I_{rcv}T_{rcv} + I_{tx}T_{tx} + I_{slp}T_{slp}) \times V \tag{3.2}$$

3.10 Simulation Framework

During developments of applications, systems, and protocols for MANETs and WSNs, a large part of the time will be spent on compiling, testing, debugging, and evaluating. Either a network of real MANET/WSN nodes or a MANET/WSN network simulator is used during the testing, debugging and evaluation phases.

When using network simulators for research experiments, the evaluation of the experiment can be much less time consuming and information about nodes and their communication protocols can be measured at a high level of detail. It is also possible to repeat exactly the same experiment several times, which is almost impossible when evaluating an experiment on a real MANET/WSN testbed. In a simulation environment, it is also possible to control most aspects of the network environment such as number of nodes, mobility, wireless channel propagation models, etc. Therefore, simulation-based study is a good option to analyze performance of MANET/WSN routing protocols before deploying the protocols in a real testbed.

To perform simulation, a network simulator is required. In general, network simulators can be used for a wide range of tasks that are involved in MANET/WSN communications. In the following, we briefly introduce some of these tasks.

- *Application and System Development:* When developing applications or systems, simulators can be used as a tool for testing the complete behavior of the system. By executing the application or system in a simulator with supports for debugging, it is possible to find out the system design flaws before deploying the application on real nodes. Installing, executing, and debugging using a simulator can save a substantial amount of time compared to using real nodes since it takes much less time to install and execute in a simulator. It is also easier to get detailed information about internal states and other debugging related information of the simulated nodes than on real nodes.

- *Evaluation of New Communication Protocols:* When developing new communication protocols for wireless sensor networks, it is necessary to evaluate some aspects of the protocol such as energy consumption, throughput, reliability in varying conditions. Simulators provide detailed evaluations as well as control and variation of the conditions in the simulated environment. Evaluation in a simulator is typically both easier and faster than on a real world deployment or testbed.

- *Power Profiling of Applications:* In WSNs, many applications have high network lifetime requirements. Using an simulator, it is possible to get an expectation of how long batteries in the nodes will last. It is, however, important that the simulator has a fine-grained energy model of the nodes such that it can provide accurate power consumption predictions.

3.10.1 Types of Wireless Network Simulators

Simulations can be performed at several different abstraction levels, from generic simulation where only the most important aspects are simulated to high detailed simulations where many details are simulated. We classify the available simulators into three categories: generic network simulator, code level simulator, and firmware level simulator.

- *Generic Network Simulators:* Generic network simulator simulates systems with a focus on networking aspects. The simulator user typically writes the simulation application in a high level language different from the one used for the real sensor network. Since the focus of the simulation is on networking, the simulator typically provides detailed simulation of the radio medium, but less detailed simulation of the nodes. The application or protocol code is usually written in the same language as the simulator itself. Most network simulators provide implementations of network stacks, MAC protocols, radio medium simulation, etc. Examples of generic network simulator include NS-2 [105], NS-3 [106], OMNeT++ [138], and GloMoSim [148].

- *Code Level Simulators:* Code level simulators use the same code in simulation as in real nodes. They provide implementation of the network stacks that are available for the specific operating system since the code is the same as on real nodes. Code level simulators can be used for interoperability testing, but since they are operating system specific, the tests will be limited to communication stacks within the same operating system. Example of code level simulator include TOSSIM [90] and COOJA [55].

- *Firmware Level Simulators:* These simulators are based on emulation of the sensor nodes and the software that runs in the simulator is the actual firmware that can be deployed in the real nodes. This approach gives the highest level of details in the simulation and enables accurate execution statistics.

3.10.2 Simulation Framework

Many simulators provide specific frameworks for the simulations of certain types of network. For example, OMNeT++ provides different frameworks for simulating MANETs, WSNs, or Body Sensor Networks [142]. Different frameworks have different focuses on the specific applications, which enrich the simulator to simulate the application detailedly. As an example, the INETMANET framework [7] includes multiple radio wave propagation models, simple battery models and supports multi-radio communications, which are the key characteristics for the simulation of opportunistic routing protocols in MANETs.

OPPONET [107] [87] provides basic mechanisms for simulating opportunistic and delay-tolerant networks in OMNeT++. OPPONET allows simulating *open* systems of wireless mobile nodes where synthetic or real mobility traces are used to drive the simulations. However, OPPONET is too much limited to the mobility modeling (scripted mobility), and object creation, while it does not provide any routing functionalities.

ONE [84] is probably the most successful simulator specifically designed for evaluating DTN and opportunistic routing protocols. It allows users to create scenarios based upon different synthetic movement models or real-world traces to offer a framework for implementing routing and application protocols. However, ONE focuses on the modeling of the behavior of store-carry-forward networking, and hence refrains from detailed modeling of the low layer mechanisms such as signal attenuation and congestion of the physical medium. Instead, the radio link is modeled as a communication range and a link with a certain bit-rate, which are assumed to be constant over the simulation. All these limitations make ONE imperfect for simulating opportunistic routing protocols, which heavily make use of various links with channel fluctuation.

There have been earlier works in the MANET community to develop frameworks to implement ad hoc routing protocols. ASL [136] and FRAd-Hoc [133] present such routing frameworks in MANETs. [16] provides a MANET routing protocol framework for the OMNeT++ community. [103] designs a framework for opportunistic routing protocols in ad-hoc networks, but it targets to emphasize that the throughput gain achieved by opportunistic routing is not clearly attributed to the opportunistic selection of forwarder but also partly due to its acknowledgment and scheduling features which may also be implemented by traditional MANET routing protocols. It does not focus on the compositional architecture of generic opportunistic routing protocols. Our framework, consisting of abstract components and common functionalities, builds an architecture for designing and implementing opportunistic routing protocols.

3.11 Conclusions

In this chapter, we gave a review of related work in the area of routing in wireless mobile ad-hoc networks and wireless sensor networks. We started with the introduction of traditional routing approaches in mobile ad-hoc networks and wireless sensor networks, which leads to the introduction of opportunistic routing. Then, we presented the latest developments of routing protocols in MANETs. Four types of routing approaches, namely proactive routing, reactive routing, hybrid routing, and geographic routing are discussed separately. Next, we moved to routing protocols in wireless sensor networks. Two types of routing principles, namely flat routing and hierarchical routing, are introduced and their examples are also indicated in detail. After that, we introduced opportunistic routing, and explained why opportunistic routing is promising for mobile ad-hoc networks and low-power wireless sensor networks. The main benefits and design challenges of opportunistic routing are discussed. Next, we gave a detailed description of the existing candidate list-based opportunistic routing protocols. Their drawbacks in the presence of link variation and node mobility are highlighted, since this motivates our contribution of a beaconless opportunistic routing. Some other works correlated with the thesis are also introduced, such as context-aware communication and routing approaches. Adaptive communication, which enables nodes to adjust their behaviors according to the latest network situation, is also explained. Duty-cycling mechanism, which makes the sensor node to switch among different states, is explained. Last, the software-based on-line energy profiling mechanisms are discussed.

In the next chapter, we start the description of the first contribution of this thesis: a simulation framework for opportunistic routing protocols.

Chapter 4

Opportunistic Routing Simulation Framework (ORSF)

4.1 Introduction

Opportunistic routing has been proposed to improve the performance of wireless multihop ad-hoc networks. Many protocols have been proposed and validated to show their functionalities. However, to analyze the performance of different protocols, implementations of different protocols have to be done individually. This chapter describes a framework for simulating opportunistic routing protocols in the INETMANET framework [7] of OMNeT++ [138]. Our goal is to facilitate the implementation and comparison of new opportunistic routing protocols. The proposed modules adopt an abstraction of the generic functions of the most representative opportunistic routing algorithms. The main contribution is a modeling architecture in the OMNeT++ simulator, which could be extended to implement different opportunistic routing schemes [155]. Our work provides an analysis of the most representative opportunistic routing algorithms. We decouple opportunistic routing into four procedures - *Forwarder Candidate Selection*, *Forwarder Selection*, *Forwarder Role Change Notification* and *Collision Avoidance*. Different protocols should have specific implementation mechanisms of each procedure. In the framework, these four procedures are defined as virtual functions and act as the implementation stubs such that different protocols could be implemented by overriding them in the derived function according to their distributed strategies.

OMNeT++ [138] is an open-source modular simulation platform that has primarily been used for simulating wired and wireless communication networks. It includes, and is continuously complemented by, multiple modeling frameworks like INET [9], INETMANET [7], MiXiM [10], etc. The INETMANET framework includes multiple radio wave propagation models, simple battery models and supports multi-radio communications, which are the key characteristics for the simulation of opportunistic routing protocols in MANETs.

The idea of this work is based on the fact that OMNeT++ (including the INETMANET framework) lacks the supports for some key features of simulating opportunistic routing protocols. Opportunistic routing tries to take advantage of the time-varying nature of wireless environment to provide hop-by-hop packet forwarding in scenarios where traditional MANET routing may not perform well. The goal is to implement a simulation framework, which facilitates the simulation of any opportunistic routing protocols with the INETMANET framework in the OMNeT++ simulator.

The design and implementation details of our framework are presented in section 4.2, which includes the framework architecture (section 4.2.1), the abstracted component modules of opportunistic routing protocols (section 4.2.2), and message and data structures that are used in the framework (section 4.2.3). To evaluate the effectiveness of the framework, we implemented some opportunistic routing protocols using our framework and tested their performance in section 4.3. Section 4.5 concludes this chapter.

4.2 Framework Description

In this section, we describe the general architecture of our framework in section 4.2.1. Among the different framework components, the OppRoutingProtocol module, which is the core of the architecture is described in section 4.2.2. The data structure, message format, and other related issues that are used in the framework are described in section 4.2.3.

4.2.1 Framework Architecture

Our implementation is based on the OPPONET [107] [87] project. OPPONET allows simulating *open* systems of wireless mobile nodes where synthetic or real mobility traces are used to drive the simulations. However, OPPONET is too limited to mobility modeling (imported from pre-generated mobility traces) and the creation or deletion of moving objects. It does not include any routing functionalities. Our framework is based on the mobility simulation features provided by OPPNET, and to extend it by adding routing modules.

The architecture of our framework is presented in Figure 4.1, which includes different modules that coordinate with each other to provide the routing functionality. The ***Navigator*** module is responsible for node movements. It is designed as a module interface, which should be implemented as specific mobility model like Random Walk [12] or Random Waypoint [13]. During the simulation, the configuration file *omnet.ini* could be set to use prepared xml-formatted mobility trace file, which was beforehand produced by the mobility trace generation tool BonnMotion [28] to control the movement and subsistence of mobile nodes.

The ***Controller*** module is simply in charge of the initialization of the node. It mainly includes channel utilization, packet storage management and other functionalities.

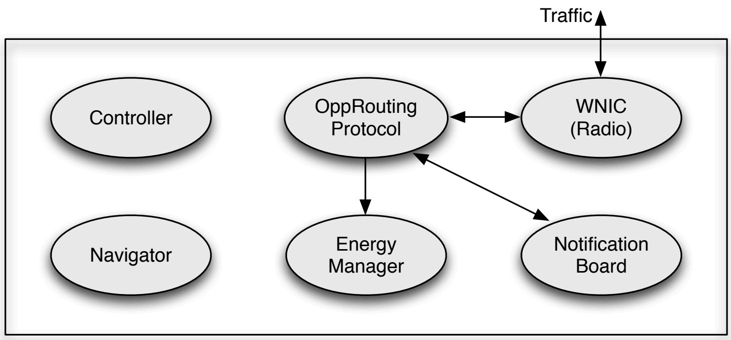


Figure 4.1: Opportunistic Routing Node Structure in OMNeT++ Simulator.

The ***WNIC*** module is an implementation of a wireless network interface controller, composed of physical and MAC layer. We choose IEEE 802.11 from INETMANET as our WNIC implementation, which includes the IEEE 802.11a/e/g implementations. A node could have multiple WNIC modules working at different radio frequencies to support multi-radio communication. The *ChannelControlExtend* module from INETMANET is adopted to implement the multichannel related functions.

The ***EnergyManager*** module is derived from the INETMANET *InetSimpleBattery* module and is a simple energy related implementation. The *InetSimpleBattery* module provides a linear model of battery usage with fairly coarse estimate of battery consumption, together with little computational overhead.

The ***NotificationBoard*** is employed for modules to notify each other about the "event" of state changes, such as interface status changes (up/down), mobile node position updates, etc. The NotificationBoard acts as an intermediary between modules, where state changes can occur, and modules that are interested in learning about those changes. Modules should "subscribe" to the notification categories they are interested in. The *NotificationBoard* module from INETMANET is adopted in our framework.

The ***OppRoutingProtocol*** module is the core component of the framework. It is implemented as a simple module such that it could be easily extended. It abstracts the general functions of the most representative opportunistic routing algorithms and modularizes them such that a specific protocol could be implemented by extending the module. For example, OppRoutingProtocolExOR is the ExOR implementation module by extending the OppRoutingProtocol. In next section we give more details of the OppRoutingProtocol module.

4.2.2 Opportunistic Routing Module

As our focus is mainly on the design of opportunistic routing protocols, it makes sense to analyze the kernels of the most representative protocols and perceive some fundamental structures. Through the analysis of the distinguished opportunistic routing protocols, we find out that they share some common processes. Therefore, we describe a general procedure of opportunistic routing and decompose it into four phases. We regard these four steps as the key features of an efficient opportunistic routing algorithm. The four phases are:

- *Forwarder Candidates Selection*
- *Forwarder Selection*
- *Forwarder Role Change Notification*
- *Collision Avoidance*

Forwarder Candidates Selection is the first procedure of the opportunistic routing. The sending node utilizes the peer-discovery service provided by the WNIC module. It periodically polls the node factory to check the nodes inside its range. Certain attributes (e.g., geographic region or nodes movement tendency) are adopted additionally to build the set of potential next-hop nodes. The design of these attributes should take into account that only the nodes that are closer to the destination or that have the movement towards it, should be the candidates. The frequency of the polling operation should be correlated with the nodes' speed and the rapid change of the network topology.

Forwarder Selection defines rules how the actual forwarding node is picked from the candidates set. Each node inside the candidates set will be added into a peer collection and marked as unreachable once the WNIC reports its un-reachability. Unreachable peers will remain in the collection for a period, which enables the re-acquisition of nodes that are temporarily unreachable in an intermittent environment. One design proposal is that the sending node periodically broadcasts a message containing its current available channels, transmission bit-rate, and movement statistical information. Candidates that successfully receive these packets will consider the status of these information, its remaining battery lifespan, and the pre-calculated Expected Transmission Count(ETX)/Expected Any-path Transmission(EAX) metrics to the destination. A comprehensive utility function will be executed, based on the combination of the ETX/EAX value and the relationships between the received and local data. Each candidate will return an utility value and all the successfully received candidates will share its value with others. The candidate with the highest value will be the one winning the election process.

Forwarder Role Change Notification enables the winning forwarder to announce its new role and responsibility to surrounding nodes, to make them aware of the selection winner and stop the competition. This procedure is important, because if it is well-designed, the duplicated transmission could be avoided. Otherwise duplicated transmission leads to retransmissions, which means additional

Table 4.1: Basic characteristics of the most representative opportunistic routing protocols

Protocol	***Candidate Selection***	***Forwarder Selection***	***Role Notify***	***Collision Avoidance***
ExOR	Sender predefines ETX-based	Sender determines ETX-based	TDMA	Implicit Gossip mechanism
MORE	Sender predefines ETX-based	None: multiple forwarders allowed	Packets RX based	Implicit TX No. based
MIXIT	Sender predefines ETX-based	None: multiple forwarders allowed	Packets RX based	Implicit TX No. based
SOAR	Sender predefines ETX-based	Sender determines ETX-based	TDMA	Explicit Acks based
MCExOR	Sender predefines ETX-based	Sender determines ETX-based	TDMA	Explicit Acks based

overhead. A possible implementation could be that the selected forwarder broadcasts a "StartToSend(STS)" packet to indicate the start of data transmission, including the adopted channel usage and bit-rate. The data transmission will start if no more messages are received within an interval after the STS. In the framework, we implement this module as a broadcast function. The data transmission to a network node will be aborted whenever the node is detected as lost by the WNIC module.

Collision Avoidance concerns how the nodes that wish to access the wireless medium at the same time and contend for the channel. A subsequent contention resolution mechanism must be defined. Contention could happen in two cases: the first case is imperfect design of the Forwarder Role Change Notification process, which leads to duplicated transmission; the second case is when two or more nodes want to send packets at the same time, which could result in packet collisions. To avoid this, multiple channel access mechanisms could be applied, such as CSMA-CA.

To show the applicability of our design, Table 4.1 presents how the representative opportunistic routing protocols can be decomposed into the four steps. In the implementation, these four procedures are defined as ***virtual functions***, which just have the general interfaces with necessary data structures. Concrete routing modules need to be created for respective protocol by extending the *OppRouting-Protocol* module.

The four virtual functions are defined as following:

- ***candidateSelection(Src, Dst)***: This function returns a vector of nodes by selecting the candidate nodes as the potential relays to a given destination(Dst), from the neighbors of a given node(Src) based on specific rules.

- ***forwarderSelection(HostVector)***: This function returns a forwarder from the candidates set(HostVector).
- ***roleChangeNotification(Host)***: This function broadcasts a message notifying the Host's surrounding nodes about its new role. The receiving nodes will stop competing for the channel access.
- ***collisionAvoidancne()***: This function avoids that two nodes attend to access the medium at the same time.

Besides the core virtual functions, there are some other common functionalities, which are fundamentals for most opportunistic routing protocols. The framework also includes the implementation of these shared functions. Although there might be differences for each protocol(some protocols may not explicitly include all the four procedures), we believe that most of the protocols could be easily adapted to use the common mechanisms provided by the framework. These common utility tasks include:

- *Neighbor Discovery & Management*
- *Packet Broadcasting*
- *Packet Buffer Management*
- *Transmission Reliability Control*
- *Time Scheduling*
- *Node Interface Management*
- *ETX/EAX Calculation*

Neighbor Discovery & Management: Nodes need to detect neighbors that are physically reachable in one hop. Neighbor detection is essential for opportunistic routing because a well-designed neighbor detection mechanism acts as a basis for forwarder selection. Neighbor management service of INETMANET is adapted to control neighbors via periodic beacons to build the neighbor information.

Packet Broadcasting: In almost all routing protocols, nodes have to distribute information throughout the network. An implementation from OMNeT++/INETMANET framework is used to provide this function.

Packet Buffer Management is another compulsory operation for nodes. Nodes need to store received packets and do other manipulations. Potential data structures and corresponding operations are defined inside the framework to fulfill this task.

Transmission Reliability Control: In the simulation, the delivery of a packet from one node to another has a pre-determined probability. OMNeT++ assigns three parameters to each link: propagation delay, bit error rate and data rate. INETMANET includes numbers of channel propagation models, which provides a detailed simulation basis for transmission control. The INETMANET link layer implementation is adapted to control the packet transmission.

Time Scheduling plays a vital role in opportunistic routing, because nodes need to schedule their transmission based on the information they observe from the Transmission Reliability Control. An accurate time scheduling mechanism could avoid collision.

Node Interface Management: Nodes inside the network may be equipped with more than one physical antennas to increase the network throughput. The management for multiple interfaces is necessary for benefiting from more antennas, i.e., to support multichannel communication. This function is to be implemented in the future.

ETX/EAX Calculation: Most of the "Candidate Selection" processes of the opportunistic routing protocols are based on the same principle that the source node pre-determines a forwarder priority list based on the estimates of the path loss rates according to ETX/EAX value. This function is implemented to calculate the ETX/EAX of each node pair.

4.2.3 Message Format & Data Structure

The message format is also an important issue of the OMNeT++ simulation, because it triggers the basic event handlers. In our framework, implementation of the message format follows the C++ language structures *struct*, such that all the fields could be easily manipulated. A general message structure of the opportunistic routing protocols is defined in the *OppRoutingProtocol* simple module, which includes message id, source/destination node id, etc. One important composition would be a set consisting of ranked nodes, which are selected as the candidate forwarders based on certain metrics. We implement this as a *vector* container from C++ Standard Template Library(STL) libraries using C++ generic programming. Because this will facilitate the operation of a user-defined data type according to interested metrics, e.g., prioritize nodes according to the ETX value. Some additional data structures are listed in the Table 4.2.

4.3 Framework Reconfigurability

To illustrate the usefulness of the framework, we first show how different opportunistic routing protocols can be decomposed into the combination of our framework components in section 4.3. We take the representative opportunistic routing protocol ExOR as an example, and examine the difficulties of integrating framework components with the key data flows of ExOR.

Table 4.2: Basic Data Structures

Data Structure	*Implementation*	*Function*
Packet Message	*typedef struct packet* Packet	Packet Format
Packet Buffer	std:list $\langle Packet \rangle$ PacketQueue	Stores received packets
Host Node Entry	*typedef* struct HostEntryExtendedExOR HostEntryExtended	Node structure type definition from ChannelControlExtended
Host Node Entry Vector	std::vector $\langle HostRefExtended \rangle$ HostRefExtendedVector	Vector of pointers to the HostEntryExtended, stores the pointers to nodes

We first demonstrate the usage of our framework in building the well-known ExOR routing protocol, and show how to use the defined modules to accommodate the main procedures of ExOR. Figure 4.2 and Figure 4.3 are the flow charts of a source/forwarder node of ExOR protocol. The defined virtual procedures and implemented common functionalities could be found embedded to show their roles (presented as dashed frame).

Figure 4.2 depicts the flow chart of a source node in the ExOR routing protocol. Dashed frames of “ETX Calculation()”, “Neighbor Discovery()”, and “Broadcast Pkt()” are functionalities that are implemented by the framework. “ETX Calculation()” provides the function of calculating ETX values of network links, “Neighbor Discovery()” gives the information of neighboring nodes, and “Broadcast Pkt()” supports the data broadcast transmission with certain transmission models and delivery probability as input parameters. The black boxes of “Candidate Selection” and “Forwarder Selection” are two procedures of ExOR, which define how ExOR selects forwarder candidates, and how the real forwarder is selected from the multiple candidates. Figure 4.3 depicts the flow chart of a forwarding node in the ExOR routing protocol. Dashed frames of “Buffer Pkt()”, “Batch Map Update/Buffer/Delete”, “Broadcast Pkt()”, and “Role Change Notification()” are functionalities provided from the framework. Black boxes of “What to forward()” and “When to forward()” are two specific operations, which have to be implemented by the forwarding node of ExOR. As we can see, the proposed framework components (dashed frames) can be smoothly integrated with the core operation procedures of ExOR (candidate selection, forwarder selection, what and when to forward a packet). This evaluates our framework from the perspectives of both reconfigurability and expressibility.

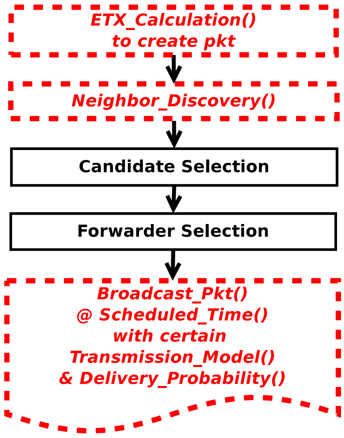


Figure 4.2: Flow Chart of Source Node in ExOR.

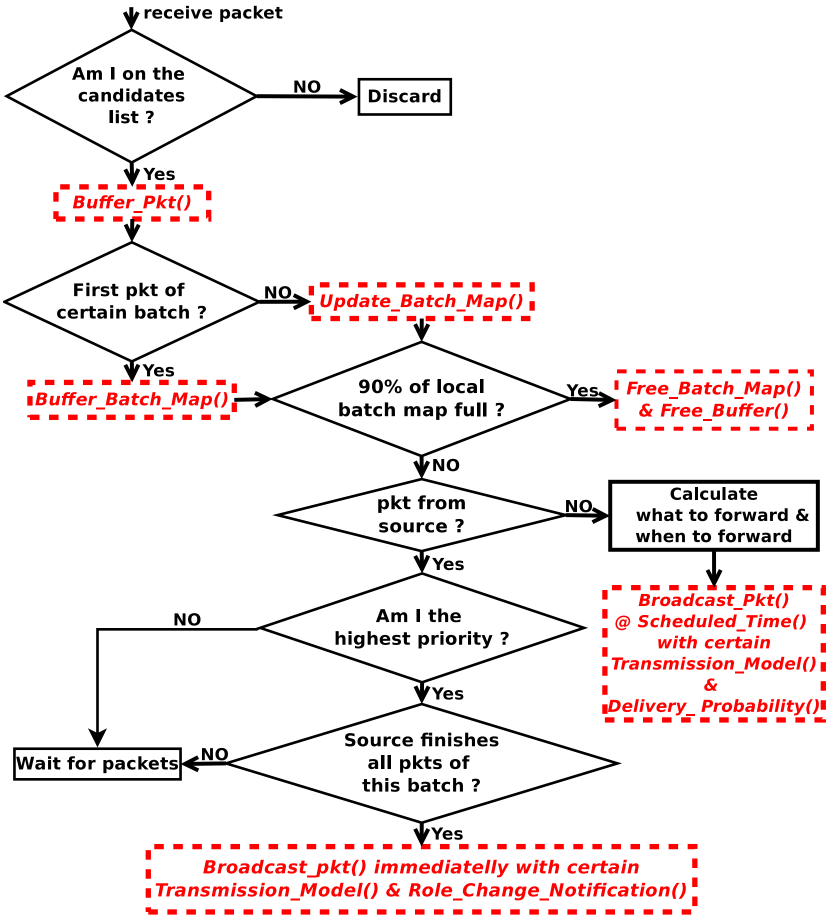


Figure 4.3: Flow Chart of Forwarder Node in ExOR.

4.4 Framework-based Simulation Study

To show the effectiveness of the framework-based implementation, we implement multiple opportunistic routing protocols using the framework [153]. In this section, we present the evaluation results of the ExOR and MORE protocols using the proposed simulation framework and one traditional MANETs routing protocol OLSR. The simulation results justify scenarios where opportunistic routing may perform better than traditional MANETs routing.

4.4.1 Simulation Description and Evaluation Metrics

All protocols are evaluated with the same stationary topology that was used in [126], see Figure 4.4. The network consists of 12 static nodes, in which node 11 acted as the source and node 5 as the destination for all transmissions. Simulations were run for each batch of 32 packets, and every packet is of size of 1024 bytes. The inter-packet time equals to zero for the packets within the same batch. For each simulation run, the warm-up phase is 200s, during which some initialization work will be done, i.e. the calculation and distribution of ETX values. For each combination of those parameters, 20 simulation runs are repeated to collect statistics. We vary the channel transmission rate to 6 Mbps, 9 Mbps and 11 Mbps, to see how the channel transmission rate affects performance. Detailed simulation parameters are listed in Table 4.3.

Transmission Power	100mW
Propagation Model	Path-loss reception
Number of Nodes	12
Radio Sensitivity	-90dBm
Simulation Time	900s
Warm-up Phase	200s
Radio Range	25m
Network Size	120x80m
MAC Protocol	inet.linklayer.ieee80211.mac
Channel Transmission Rate	{6, 9, 11} Mbps
Node Density	{5, 10, 17, 32, 44}

Table 4.3: Simulation parameters

The following evaluation metrics are collected to analyze the performance of different routing protocols:

- **Transmission delay** was the time interval between the arrival time of the first packet and the arrival time of the last packet within a batch.
- **Throughput** was measured at the destination as the ratio of the number of packets received and the time it takes to receive these packets.

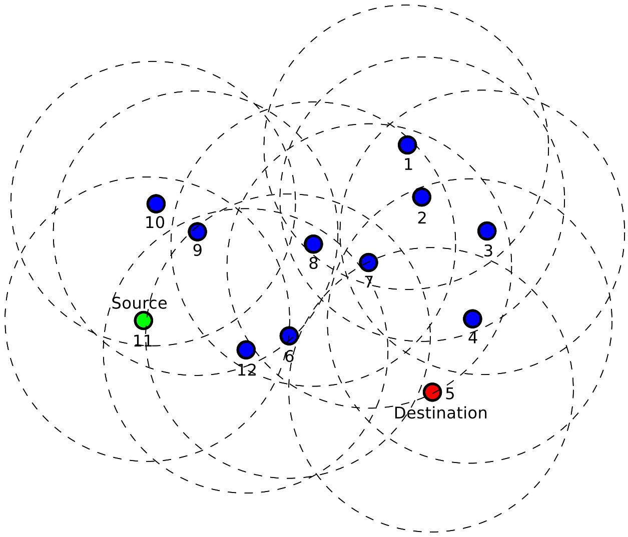


Figure 4.4: Simulation network topology: the circles mark the transmission ranges.

- **Collision** was an interesting value to see how the coordination mechanism between multiple potential forwarders performs. Also it is an indicator of the bottleneck nodes that suffer the worst collision situation in the network. It is calculated at the IEEE 802.11 MAC layer whenever an invalid packet is detected.

To investigate the influences of different system parameters on the simulation results, we elaborate the following metrics to see how they affect the protocols' performance:

- **Transmission rate** to see how opportunistic routing and traditional routing behave under different physical channel transmission rates.

- **Node density** to see how large the candidate forwarder set should be to make opportunistic routing perform optimally.

- **Route number** to see whether the number of available paths to a destination has an effect on protocol performance.

- **Channel quality** to see in which channel condition opportunistic routing should be applied to show its benefit. We take *Path Loss Alpha* as the indicator of the channel condition in a wireless environment. *Path Loss Alpha* (α) is an indicator used to approximate signal attenuation in a wireless environment. Its value is normally in the range of 2 to 6 (indoor), where 2 is for a good channel, 6 is for lossy environment.

4.4.2 Simulation Results

Channel transmission rate

The physical channel transmission rate could have significant effect on the performance of routing protocols. To see how opportunistic and traditional routing mechanisms behave under different transmission rates, we vary the channel transmission rate (megabit per second, or in short Mbps) to 6 Mbps, 9 Mbps, and 11 Mbps. The channel is configured as medium quality (path loss alpha = 3). Figure 4.5 gives the throughput evaluation of the three protocols OLSR, ExOR and MORE. There is no surprise that MORE and ExOR outperform OLSR at different transmission rate scenarios. This is an expected result since we know that opportunistic routing achieves better throughput than traditional routing. The performance gain mainly is twofold: First, opportunistic routing is able to exploit more packet forwarders and inherently utilizes the broadcast nature of wireless communication. Second, the ACK (acknowledge) redundancy introduced by the batch map of ExOR and MORE is removed, since both protocols do not send per-packet acknowledgments. MORE performs better than ExOR due to its support of spatial reuse, which is prevented by ExOR. ExOR forbids multiple nodes from accessing the medium simultaneously, even if two transmission flows are outside the interference range of each other.

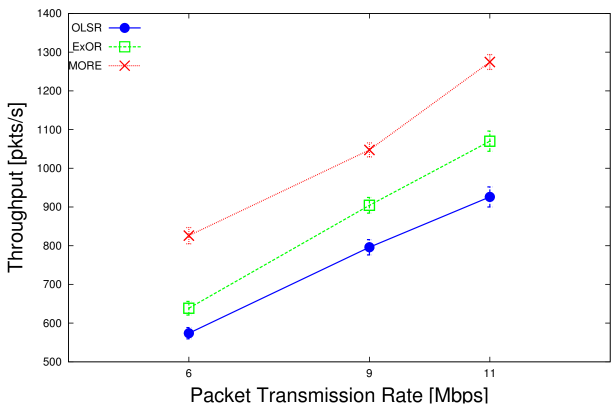


Figure 4.5: Throughput of MORE, ExOR and OLSR under different channel transmission rate scenarios.

Node density

Due to the integrated network coding operation of MORE, it is difficult and unfair to compare it with OLSR. From this section on, we will mainly focus on the performance comparison of ExOR and OLSR. Opportunistic routing owes its performance superiority to the adoption of multiple forwarders. It is therefore intuitive to analyze how opportunistic routing depends on the node density, basically the average number of neighbors per node. Node density should have significant influence on routing protocol performance. In general, low density may cause the network to be disconnected and high density will increase contention, resulting in low throughput. To see the performance of ExOR and OLSR for different node densities, we modify the network node number from 5, 10, 17 to 32 and 44, corresponding to 1, 3, 6, 8 and 12 neighbors per node on average, as shown in Figure 4.6. The channel is fixed with path loss alpha = 3, which means a fair (medium) channel condition. The result is shown in Figure 4.7, with default transmission rate equal to 6 Mbps. As we can expect, ExOR performs better than OLSR in all cases.

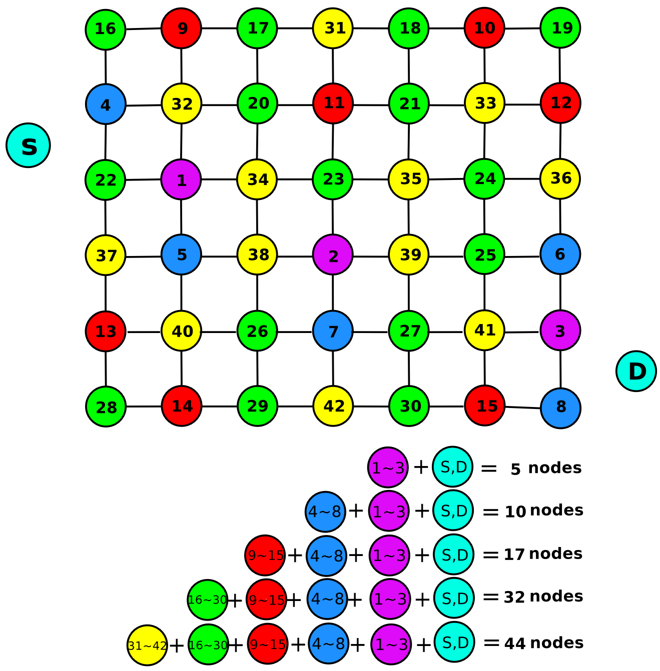


Figure 4.6: Network topology with different node densities.

When the node density is low, the network is far from being saturated. Therefore, OLSR throughput increases when more nodes are added. After a certain point the network gets saturated and OLSR starts to degrade when the node number keeps increasing. This is probably due to the overhead of periodic control traffic, i.e., Hello and Topology Control messages in OLSR. When more nodes contend for channel access, the collision occurrence probability and interference increase. Besides, we can find out that the increased number of forwarding nodes has little influence on the throughput of ExOR, which means ExOR does not benefit from additional relay candidates. A possible reason is that adding more nodes in this topology will build joint paths between source and destination. To further discuss the reason of this phenomenon, we next evaluate the throughput over different numbers of disjoint routes from source to destination to check whether opportunistic routing depends on the number of available disjoint routes.

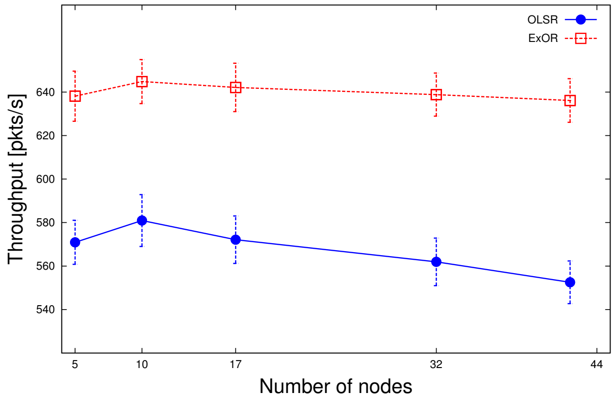


Figure 4.7: Throughput of ExOR and OLSR at different node density.

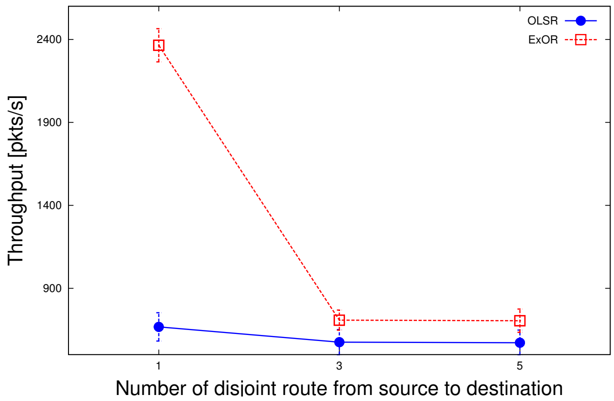


Figure 4.8: Throughput of ExOR and OLSR with different number of disjoint routes from source to destination in Figure 4.9.

Number of routes

To see the effect of different numbers of disjoint routes from source to destination, we define the network in Figure 4.9. This means that we deploy three network topologies using different numbers of intermediate nodes to make the source-destination route numbers equal to 1, 3 and 5. These three topologies include only the intermediate nodes of route 1, routes 1 & 2 & 3, and routes 1 & 2 & 3 & 4 & 5 respectively, as shown in Figure 4.9. Figure 4.8 plots the throughput of ExOR and OLSR for different source-destination route numbers, using a default channel transmission rate of 6 Mbps. From the plots we can see that for ExOR the throughput is maximized when only one route is available between source and destination. To see the possible reasons of this phenomenon, we plot the average number of packet collisions that are encountered at different nodes within the 5-routes topology. The results are shown in Figure 4.10, with a confidence interval of 95%. As we can see, the 5-routes topology is a symmetric deployment of nodes. The average number of collisions encountered at the different nodes are distributed symmetrically in most cases, i.e., node-pairs 8-13, 10-11, etc. Most collisions happen at the nodes that are suffering the severest medium access contention, i.e., nodes 13, 8, 11 and 10, which are concurrently within the radio range of three different neighbors. The high number of collisions encountered at nodes 13, 8, 11 and 10 might be the reason for the phenomenon that ExOR performs best when only one route is available. Nodes 13-11 and 8-10 are the bottlenecks of routes 2/4 and routes 3/5 respectively. The same explanation could be applied to the very high collision number at node 1, which is the bottleneck of route 1.

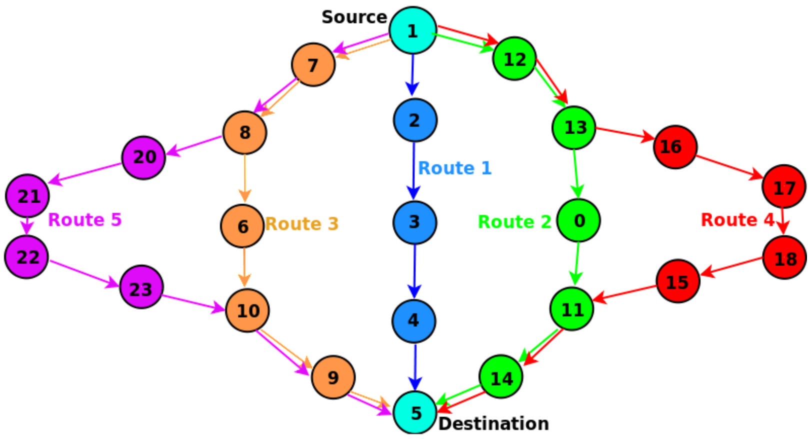


Figure 4.9: Network topology of different number of disjoint routes from source to destination.

As we can observe, results in Figure 4.7 show that the throughput of ExOR almost does not change with additional candidate nodes. However, Figure 4.8 reveals that network throughput drastically degrades by adding more nodes and by providing more routes from source to destination. The explanation for this observation might be as follows: when adding more nodes in Figure 4.6 more joint routes from source to destination are set up and there are no bottleneck nodes within the network. However, when introducing more nodes in Figure 4.9 to set up more disjoint routes from source to destination, bottleneck nodes will appear, i.e., node 8 and 13, which will restrict the performance of the network. For example, when the bottleneck nodes in Figure 4.9 suffer from high contention, the network throughput degrades significantly. Therefore, more routes may increase or decrease the throughput depending on the topology and where the bottleneck nodes reside.

Channel quality

The high loss rates in wireless networks (e.g., 20-40% as observed in several deployments [19]) make traditional routing inefficient. To achieve better performance, opportunistic routing exploits communication opportunities that arise by chance due to the broadcast nature of the wireless medium. When a sender broad-

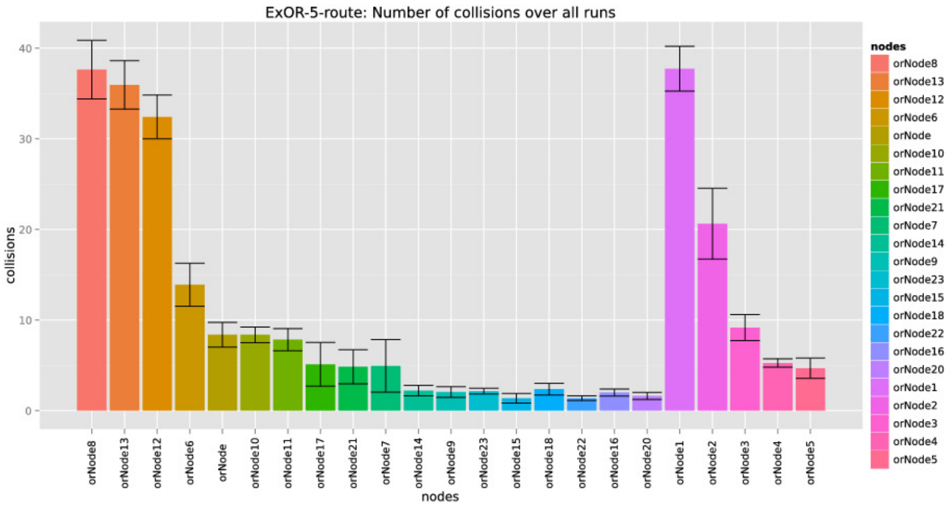


Figure 4.10: Average number of collisions encountered at different nodes at Figure 4.9 with route number = 5.

casts its data, any node that hears the transmission may forward the data toward the destination. Although individual nodes may experience high loss rates, as long as there exists one forwarder that is closer to the destination and receives the transmission successfully, the date could move forward. In this way, opportunistic routing can efficiently combine multiple weak links into a strong link. To see how opportunistic routing makes use of poor channels, we need to know first what is the effect of channel quality on the performance of opportunistic and traditional routing mechanisms. To see this, the parameter of path loss alpha is varied between 2, 3, and 4.5, to represent the channels of good, medium and bad quality respectively.

We use the network topology of 17 nodes in Figure 4.6. We vary the channel transmission rate at the source between 6 Mbps, 9 Mbps, and 11 Mbps to see the superior performance of ExOR with different channel qualities. The results are shown in Figure 4.11 and Figure 4.12. As expected, ExOR behaves better than OLSR in all situations of 6 Mbps, 9 Mbps and 11 Mbps. An interesting observation is that the performance of OLSR significantly degrades when channel quality gets worse, while ExOR shows a stable performance under different channel conditions. The superiority of ExOR increases as the channels become worse. When the channel quality degrades, OLSR nodes suffer from an increased number of packet retransmissions and performance degrades, while the diversity of multiple neighbors in ExOR will alleviate this problem to some degree. Therefore, the performance gain of opportunistic routing will increase in a lossy wireless environment.

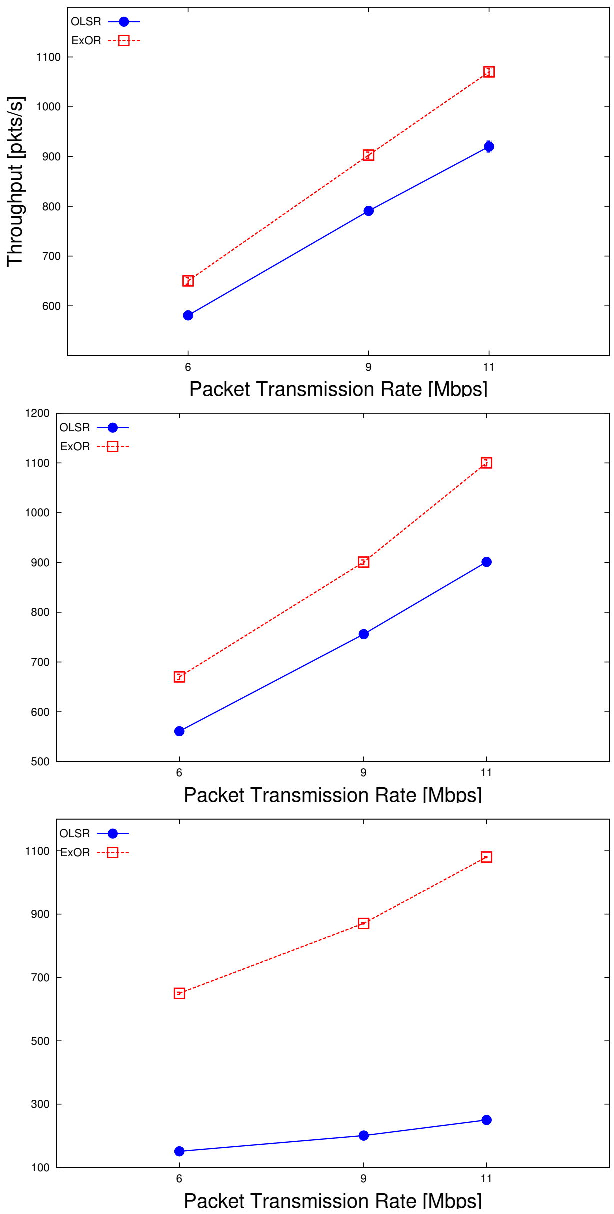


Figure 4.11: Throughput of ExOR and OLSR under different transmission rate scenarios with different channel quality(Upper: good quality with α = 2 , Middle: medium quality with α = 3, Bottom: bad quality with α = 4.5)

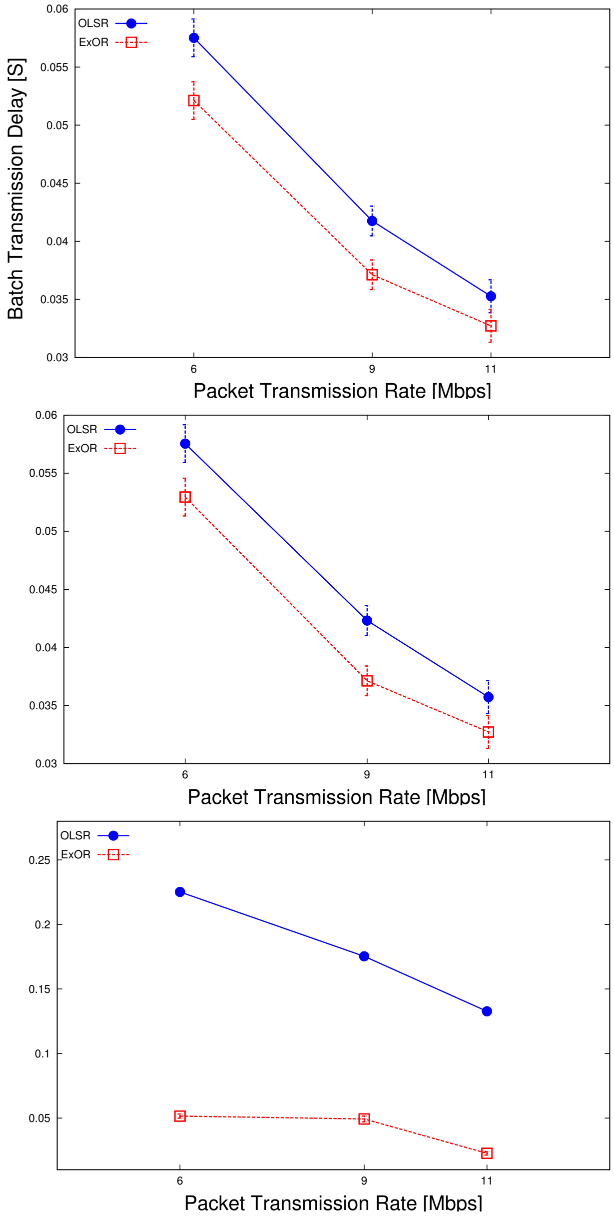


Figure 4.12: Batch transmission delay of ExOR and OLSR under different channel quality (Upper: good quality with $\alpha = 2$, Middle: medium quality with $\alpha = 3$, Bottom: bad quality with $\alpha = 4.5$)

4.5 Conclusions

In this chapter, we present our contribution of a simulation framework in the OMNeT++ simulator for opportunistic routing protocols. The framework can be used to facilitate the implementation of opportunistic routing protocols. The motivation of this work is that many opportunistic routing protocols have been proposed, however, to analyze the performance of different protocols, implementations of different protocols have to be achieved individually. To facilitate the implementation and evaluation of multiple protocols, we abstract the common features of the representative opportunistic routing protocols and implement them as virtual functions in the framework. The framework decouples opportunistic routing into four general components and abstracts them as virtual functions, which are *Forwarder Candidate Selection*, *Forwarder Selection*, *Forwarder Role Change Notification* and *Collision Avoidance*. Different protocols should have specific implementations of each phase. These four functions act as implementation stubs such that different protocols just override them in the derived function according to their distributed strategies.

With the proposed framework, we implemented multiple opportunistic routing protocols to validate the effectiveness of our framework. Throughout the simulation, we also justify in which situations opportunistic routing performs better than traditional MANET routing. Thanks to the intrinsic support of node mobility, our framework can easily be configured to test the performance of different opportunistic routing protocol under dynamic environments (section 3.4.6). With the help of the framework, the implementation of candidate list based opportunistic routing can be done in an easy way. This significantly facilitates the performance comparison between existing opportunistic routing protocols and our proposed beaconless opportunistic routing protocols, which are described in the following chapters.

Chapter 5

Topology and Link Quality-aware Geographical Opportunistic Routing Protocol (TLG)

5.1 Introduction

In this chapter, we propose TLG - ***T**opology and **L**ink quality-aware **G**eographical* opportunistic routing protocol for mobile ad-hoc networks [157]. Opportunistic routing (OR) has been proposed to increase the performance of multihop wireless communication. Many opportunistic routing protocols have been proposed. However, existing opportunistic routing protocols choose the next-hop forwarder based on a predefined candidate list, which is generated using a single network metric and the list is calculated before data transmissions. The idea of using a candidate list is not appropriate in dynamic environments. When the network topology changes, the priority list will be invalid. Therefore, TLG completely abandons the idea of candidate list and allows all qualified nodes to participate in the packet forwarding process. Additionally, unlike existing opportunistic routing, where the forwarder is selected solely based on a single metric (link quality or distance progress), TLG simultaneously uses multiple network metrics such as network topology, link quality, and geographic location to implement the coordination mechanism.

As examples of wireless ad-hoc networks, wireless sensor networks (WSNs) [47] and wireless multimedia sensor networks (WMSNs) [26], received great attentions from academic and industry communities in the past decade. Their broad applicability and fast deployment at low cost without relying on existing network infrastructures make them suitable solutions for a variety of applications. For example, mobile robots or Unmanned Aerial Vehicles (UAVs) equipped with scalar or multimedia sensors could be used to set up a multi-hop UAV ad-hoc network (UAVNet [100]) to explore the hazardous area that rescuers cannot reach easily. A swarm of UAVs can be sent to monitor a certain area to transmit scalar/multimedia content to the control center, as shown in Figure 5.1. In such applications, multimedia data provides civil authorities (e.g., rescuers or polices) more precise information to help them to make suitable decisions. Therefore, in these applications, besides Quality of Service (QoS) metrics that measure the system performance from the network's view, Quality of Experience (QoE) metrics have to be collected to reflect the user's perception.

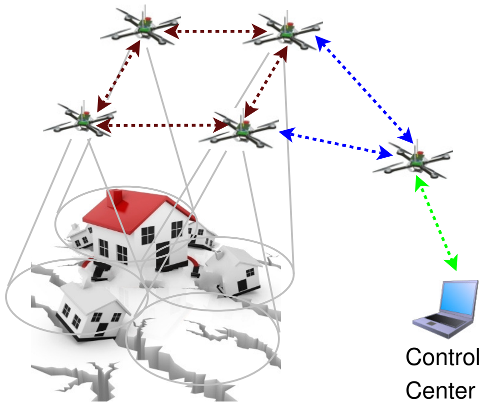


Figure 5.1: Ad-hoc network deployment under emergency situation

Most of the efforts in opportunistic routing have been made on the issues of candidate selection and relay priority assignment. However, existing OR protocols did not fully consider the unreliability of wireless transmission, and most of them assume the connection between nodes will remain constant after the connection has been set up. In reality, wireless links are extremely unreliable, as they often experience significant quality fluctuations or distortions. Moreover, some opportunistic routing protocols use geographic data to select relay nodes. For example, Dynamic Forwarding Delay (DFD)-based approaches include a dynamic delay at each candidate before they forward the packet [35]. This delay function is inversely proportional to the progress of each node such that the node closer to the destination has higher priority. However, due to the unreliability of wireless transmission, the most distant node within the radio range of a sender might suffer from a bad connection, which might lead to high packet loss in lossy environments. Many other OR protocols have been proposed, which use different network metrics to select packet forwarders. However, most of them solely rely on a single metric, either link quality, or geographic location.

To address the above issues, we propose the Topology and Link quality-aware Geographical opportunistic routing protocol (TLG). TLG takes into account different network metrics simultaneously to make a joint routing decision. TLG uses the idea of DFD, and it considers link quality, progress, and remaining energy when calculating DFD. Simulations were carried out to show the benefits of considering multiple metrics during the routing process. This chapter includes both QoS and QoE evaluations for the proposed protocol. The simulation results show that TLG could improve QoS metrics by nearly 40% and QoE metrics by nearly 30% compared to existing protocols that consider single metrics.

5.2 Motivation

The research of OR mainly focuses on two issues: candidate set selection and priority assignment of candidates. The candidates have to coordinate to avoid duplicated transmission. This is usually achieved by ordering the candidates according to some criteria, such as Expected Transmission Count (ETX) [31]. In location-aware protocols, progress is the most used metric. This leads to the fact that the node that is closer to the destination will have a higher priority. However, the concept of prioritizing a fixed list of candidates reduces the freedom of opportunism. Additionally, the predefined candidate priority list may not hold anymore if the wireless environment or network topology changes.

The fluctuation of wireless channels makes it difficult to route packets in a lossy wireless environment, and the quality of the wireless channel might be affected by many unknown factors, such as interference, fading, etc. Additionally, when nodes become mobile, the network topology will change over time, which increases the difficulty to transmit packets. In this situation, a topology control process is usually needed for each node to keep their connectivity with neighbors. Most of the works about topology control are limited to tuning the transmission radius of nodes, and few of them analyze how the protocol should manipulate the mobility information to improve network performance.

From the analysis presented above, we find that it is beneficial to consider multiple network metrics to make a joint routing decision in a wireless environment. In this chapter, we present a new opportunistic routing protocol called TLG, which selects packet forwarders according to multiple types of network context information, such as network topology, link quality, geographical location, and energy.

5.3 Protocol Description

This section describes the Topology and Link quality-aware Geographical (TLG) opportunistic routing algorithm. TLG borrows the concept of dynamic forwarding delay (DFD) from BLR as the coordination mechanism among multiple packet receivers.

However, BLR calculate the DFD timer only based on the geographic location, such that the forwarder candidate with the largest geographic progress has its timer expired first and retransmits the packet first. However, this timer calculation has a drawback of choosing a node with poor link quality. Since the node with the largest progress is more likely to be affected by wireless channel fluctuations, which leads to transmission failures. To mitigate this problem, TLG simultaneously considers multiple types of network parameters of link quality, geographic progress, and remaining energy of node to compute the DFD delay function. In the following, we describe the two key components of TLG: the calculation of DFD, and the utilization of node mobility information to help the packet forwarding.

5.3.1 Dynamic Forwarding Delay (DFD)

In TLG, when a source node has data to transmit, it includes the geographical information of itself and also of the destination into the packet header and broadcasts it. This is under the assumption that each node is carrying a GPS-like device such that it has the location awareness. The neighboring nodes that receive the packet, first check whether they are closer to the final destination than the last-hop. If not, they drop the packet. Otherwise, they are considered as qualified relay candidates, and apply a Dynamic Forwarding Delay (DFD) function to start a local timer. The idea of using DFD as the coordination mechanism was first introduced in BLR to give a delay before a node rebroadcasts the received packet. The node that generates the smallest delay will retransmit the packet first. By overhearing this transmission, other candidates stop their timer calculation or cancel the scheduled transmission and drop the packet. Because of the transmission overheard, they could infer that another candidate has retransmitted the same packet, and thus, they should drop the packet to avoid duplicate transmissions. In the meantime, the re-broadcasted packet is used as a passive acknowledgment to inform the original sender about which neighbor is selected as the real forwarder.

After this, the sender is aware of its next-hop forwarder. Therefore, the sender transmits subsequent packets using unicast to reduce the drawbacks introduced by broadcasting [35]. The problem then is to get the information about how long this unicast transmission should last. In TLG, the duration of this unicast is calculated according to the validity time of the link between the sender and the selected relay node. We will discuss this issue in section 5.3.2.

In TLG, we propose a new definition of the **DFD** calculation, which is based on multiple metrics. We consider: *progress*, *remaining energy*, and *link quality* of nodes to increase the reliability and energy-efficiency of the protocol. The new calculation of DFD is defined according to equation (5.1):

$$\begin{aligned} \text{DFD} = (&\alpha \times \text{Remaining Energy} + \beta \times \text{Link Quality} \\ &+ \gamma \times \text{Progress}) \times DFD_{Max} \end{aligned} \tag{5.1}$$

in which α, β, and γ are the weights of each metric and $\alpha+\beta+\gamma = 1$. Depending on the application requirements, TLG may assign different weights for different metrics. DFD_{Max} is the predefined maximum delay allowed at each node, which means a node will wait maximally DFD_{Max} amount of time before it forwards the received packet. Link Quality, Progress, and Remaining Energy are computed according to equations (5.2), (5.3), and (5.4), respectively.

Link Quality

Existing OR protocols do not consider the instantaneous link quality for the routing decision. These works assume that the transmission will be successful as long as two nodes are within the transmission range of each other. They also ignore the time-varying characteristics of wireless channels. They assume that the channel quality at the moment of selecting and ranking the candidates is identical with the moment when the packet is to be transmitted. Therefore, TLG considers the instantaneous link quality at the moment of packet transmission to calculate the DFD function. The calculation of the "Link Quality" part of equation (5.1) is shown in equation (5.2).

Link quality is usually measured by means of physical layer information. For example in wireless sensor networks, the CC2420 radio chip [131], a widely used off-the-shelf low power radio chip, provides the Received Signal Indicator (RSSI) and Link Quality Indicator (LQI) for each received packet. These parameters directly reflect the immediate link quality.

Link Quality Indicator (LQI) is a metric of the current quality of the received signal. LQI gives an estimate of how easily a received signal can be demodulated by accumulating the magnitude of the error between the ideal signal and the received signal over the 64 symbols immediately following the sync word [30]. LQI is best used as a relative measurement of the link quality (a low value indicates a better link than what a high value does), since the value is dependent on the modulation format. To simplify, if the received modulation is FSK [135] or GFSK [116], the receiver will measure the frequency of each "bit" and compare it with the expected frequency based on the channel frequency and the deviation and the measured frequency offset. If other modulations are used, the error of the modulated parameter will be measured against the expected ideal value.

The received signal strength indicator (RSSI) has largely been perceived by the wireless sensor networks community as an inadequate estimator and metric for determining the link quality between connected nodes. This problem is exacerbated by the known fact that, in indoor radio communication, the transmitted signal suffers from multi-path fading due to reflection, refraction, and scattering of radio waves by structures insides the buildings. Performance of communication is seriously degraded in indoor scenarios, but not much can be done to eliminate this problem, even if the multi-path medium is well characterized and nodes model the channel to reduce the effects of different disturbances. Earlier experimental results have confirmed the finding by showing that while detecting good links is possible with RSSI, estimates with imperfect links are difficult.

There are four to five "extreme cases" that can be used to illustrate how RSSI and LQI work [132]:

- A weak signal in the presence of noise may give low RSSI and high LQI.
- A weak signal in "total" absence of noise may give low RSSI and low LQI.
- Strong noise (usually coming from an interferer) may give high RSSI and high LQI.
- A strong signal without much noise may give high RSSI and low LQI.

- A very strong signal that causes the receiver to saturate may give high RSSI and high LQI.

In general, RSSI is a received signal strength indication. It does not care about the "correctness" or the actual signal strength of the received signal, but the signal quality often is linked to signal strength. This is because a strong signal is likely to be less affected by noise and thus will be seen as "cleaner" or more "correct" by the receiver. Therefore, LQI is a better option to represent the quality of a wireless channel, and in this thesis we choose LQI as the channel quality indicator.

$$\text{Link Quality} = \begin{cases} \frac{LQI_{Max}-LQI_t}{LQI_{Max}} & \text{if } LQI_{Bad} < LQI_t < LQI_{Good} \\ 1 & \text{if } LQI_t < LQI_{Bad} \\ 0 & \text{if } LQI_t > LQI_{Good} \end{cases} \tag{5.2}$$

Equation (5.2) shows how the "Link Quality" is calculated, in which LQI_t is the LQI value of the link between two nodes and LQI_{Max} is the predefined maximum value of LQI_t. The candidate node must ensure that a minimal link quality is achieved to guarantee successful packet transmission. Therefore, based on the parameters configured in the experiment, we classify LQI_t into three ranges, namely bad links (if $LQI_t < LQI_{Bad}$, with LQI_{Bad} = 10), good links (if $LQI_t > LQI_{Good}$, with LQI_{Good} = 20), and average links (if $LQI_{Bad} < LQI_t < LQI_{Good}$). When a node receives a packet, it will derive the LQI_t for the incoming link (the link over which the packet is received). Depending on LQI_t, equation (5.2) returns a value for "Link Quality" as the input for equation(5.1). For example, a node with a good link ($LQI_t > LQI_{Good}$) will return 0 to "Link Quality", which means a node with a good link will produce no input to the delay function. A node with a bad link ($LQI_t < LQI_{Bad}$) will produce a significant impact on DFD.

Progress

Progress is one of the most popular metrics used in conventional MANET routing and opportunistic routing. In this work, we also take it as an important routing metric. However, we have a new definition of node progress, which could significantly reduce the collision caused by concurrent transmissions from multiple candidates. Equation (5.3) shows the definition of the progress of each node. The node with a higher progress generates a shorter "Progress" value, which means a small contribution to its DFD.

$$\text{Progress} = \begin{cases} \frac{2R-P_i}{2R} & \text{if } Dist_{Relay-Dest} > R \\ 0 & \text{if } Dist_{Relay-Dest} < R \end{cases} \tag{5.3}$$

in which P_i is the progress of a node i, R is the radio range, and $Dist_{Relay-Dest}$ is the distance between the relay node and the destination node.

We define progress as the sum of two segments, as shown in Figure 5.2. S is the source, D is the destination. A and B are two possible relay nodes for S within its transmission range. A' and B' are the intersection points of the circles that are centralized at the candidate nodes A & B and line S-D. In Figure 5.2, the progress of candidate A is composed of two parts. One part is the projection of line S-A on line S-D, p_1. Another part is the projection of line A-A' on line S-D, p_2. Therefore, the progress of node A is P_A = p_1 + p_2 and the progress of node B is P_B = p_3 + p_4. With this definition, we solve the possible collision that is caused by two nodes of the same projection progress. For example in Figure 5.2, candidates A and B have the same projection progress on line S-D (p_1 = p_3). With the progress definition in BLR, A and B will generate the same forwarding delay, and this will introduce collisions since they will rebroadcast packet at the same time. However, with the new definition of progress, even if p_1 = p_3, B is closer to line S-D, and it has a larger progress than A ($P_B = p_3 + p_4 > P_A = p_1$ + p_2). Therefore, in this case, S can reach D via B with only one hop, and this can not be achieved if S chooses A as next hop.

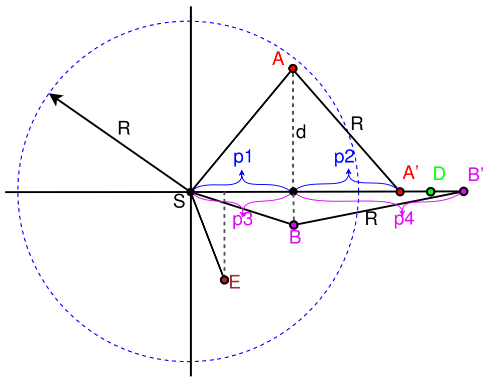


Figure 5.2: Candidate progress

Energy

Energy is another important issue in wireless ad-hoc networks due to the fact that wireless nodes are usually battery-powered and energy resources are scarce. In mobile ad-hoc networks, mobile nodes, such as UAVs, have very limited energy resources and they spend most energy for moving and hovering. Thus, energy should be considered for routing decision to provide energy-efficiency. In mobile sensor networks, energy is a more critical issue, since the limited size makes strict constraints on on-board battery. Equation (5.4) defines the energy part of the DFD function. A node with high remaining energy (E_r) generates small "Remaining Energy" value, which means a small contribution to the DFD. Therefore, a node with more remaining energy is more preferable.

$$\text{Remaining Energy} = \begin{cases} \frac{E_0 - E_r}{E_0} & \text{if } E_r > E_{Min} \\ 1 & \text{if } E_R < E_{Min} \end{cases} \tag{5.4}$$

in which E_0 and E_r are initial and remaining energy of each node, respectively. In a mobile scenario, such as an UAV ad-hoc network (UAVNet [100]), a UAV can only be selected as a forwarder if: (i) it has enough energy (E_{Min_1}) to transmit packets during the validity time of a link with a sender; and (ii) after the link validity time, the node still has enough energy (E_{Min_2}) to return back to the base station. This means, in equation (5.4), E_{min} is composed of two parts: $E_{min} = E_{Min_1} + E_{Min_2}$, and usually E_{Min_2} dominates because movements cost more energy than packet transmission for mobile nodes.

5.3.2 Link Validity Time Estimation

Even if UAVNet is an example of wireless ad-hoc networks, the mobility of UAVs is not random. Instead, the movements of UAVs should be controlled and follow certain steering rules. Considering these non-random mobility characteristics, UAVNet performs special movement behaviors. In this context, our algorithm includes the estimation of the validity time of a link between two connected UAVs, and this information will be used in the routing decision. After a node has been selected as the relay node for a sender, the sender will finish the transmission of subsequent packets using unicast to that node. Therefore, the duration of this unicast transmission needs to be determined beforehand. A Link Validity Estimation (LIVE) protocol will run at every node to estimate the validity time (T_{LV}) of each link with its 1-hop neighbors. This value will be used to decide how long the unicast transmission will last. When this link validity time expires, the sender will start another broadcast process to find a better forwarding node.

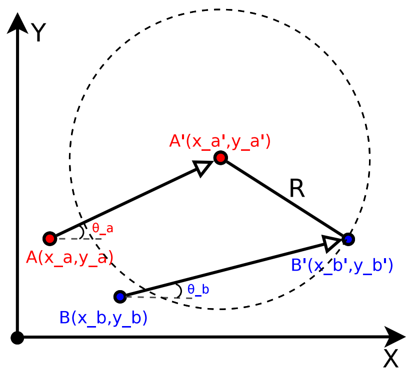


Figure 5.3: Link validity estimation calculation

Let us assume that every node knows its moving direction and speed. Using the information collected from the neighbors (position and mobility information), every node can calculate the distances to neighbors and this will enable it to predict the validity time of each link with neighbors. As shown in Figure 5.3, suppose that two nodes A and B are flying with speed V_a, V_b and direction θ_a , θ_b. Given the initial location of A (X_A,Y_A) and B(X_B,Y_B), A and B can easily calculate the link validity time (T_{LV}) of the link between them using the following equation:

$$\begin{aligned}&[(X_B + V_b \times \cos\theta_b \times T_{LV}) - (X_A + V_a \times \cos\theta_a \times T_{LV})]^2 \\ &+[(Y_A + V_a \times \sin\theta_a \times T_{LV}) - (Y_B + V_b \times \sin\theta_b \times T_{LV})]^2 \\ &= RadioRadius^2\end{aligned} \tag{5.5}$$

5.4 Performance Evaluation

5.4.1 Simulation Description and Evaluation Metrics

In this section, we evaluate the performance of TLG through OMNeT++ simulations by using the framework proposed in [118] [152]. We perform the experiments using both scalar data and multimedia data (source nodes generate scalar data and multimedia data) separately to evaluate our protocol based on both QoS and QoE metrics. Since in certain applications, QoS metrics alone can not reflect the user's perception, we configure the source nodes to transmit also the multimedia data and collect the QoE metrics to capture the subjective aspects associated with the humans' experience.

In both simulations, 31 nodes are randomly placed over a flat area, where the simulation runs for 300 s. The source node generates constant bit rate (CBR) UDP packets and video sequences in two experiments. We use the CSMA implementation from Castalia simulation framework [33] as the MAC protocol. The physical parameters of the antenna, such as transmission power, antenna gain, and receiver sensitivity are set to obtain a nominal transmission range of around 11 m. The results are averaged over 20 simulation runs with different random-generated seeds to provide a confidence interval of 95% (vertical bars in the figures). It is important to highlight that we focus on the new formula to calculate DFD in this work, and thus we assume the link validity time between nodes are fixed and assign a constant value for T_{LV}. Table 5.1 shows the simulation parameters.

Table 5.1: Simulation parameters

Parameter	Value	Parameter	Value
Field Size	40 × 40 m	Radio model	CC2420
BS location	(38,38)	Video sequence	Hall
Source location	(5, 5)	Frame rate	26 fps
Node deployment	Uniform	Video encoding	H.264
UDP source rate	2 Pkt/s	Video format	QCIF (176×144)
Transmission power	-10 dBm	T_{LV}	4 s
Path loss model	Lognormal	DFD_{max}	0.1 s

To prove that TLG achieves the best performance only when multiple metrics are considered, we give a detailed study on the impact of different coefficients (α, β, γ) in the DFD formula (5.1). A large coefficient value in equation (5.1)

Table 5.2: Combinations of coefficients in equation (5.1)

Combination #	α (Energy)	β (Link Quality)	γ (Progress)
1	0	0	1
2	0.1	0.05	0.85
3	0.1	0.1	0.8
4	0.1	0.15	0.75
5	0.1	0.2	0.7
6	0.1	0.25	0.65
7	0.1	0.3	0.6
8	0.1	0.35	0.55
9	0.1	0.4	0.5
10	0.1	0.45	0.45
11	0.1	0.5	0.4
12	0.1	0.55	0.35
13	0.1	0.6	0.3
14	0.1	0.65	0.25
15	0.1	0.7	0.2
16	0.1	0.75	0.15
17	0.1	0.8	0.1
18	0.1	0.85	0.05

means the corresponding metric is of more importance when calculating the forwarding delay function.

We define 18 combinations with different values of α (energy), β (link quality), γ (geographic progress) to show the importance to consider multiple metrics. Table 5.2 shows the values of each combination. For example, combination #1 assigns coefficients α = 0, β = 0, and γ = 1, which means the DFD calculation using this coefficient combination just consider the geographic progress of node, ignoring energy and link quality. Combination #10 assigns coefficients α = 0.1, β = 0.45, and γ = 0.45, which means that the DFD calculation using this coefficient combination treats the importance of link quality and progress equally. To show the superiority of TLG over the routing protocols that consider a single metric, we compare the performance of TLG with the well-known GPSR and BLR protocols.

We use the Packet Delivery Ratio (PDR) and goodput as QoS metrics when the source generates scalar data, and two well-known objective QoE metrics, i.e. Structural Similarity (SSIM) and Video Quality Metric (VQM) when multimedia data is produced from the source. SSIM measures the structural distortion of the video. SSIM has values ranging from 0 to 1, and a higher value means better video quality. VQM measures the "perception damage" of video experienced, and a value closer to 0 means a video with a better quality.

5.4.2 Simulation Results with Scalar Data Transmission

First, we analyze the performance of TLG when the source node sends UDP packets with a constant packet rate of 2 packets/s. PDR and goodput are measured at the destination. Results are shown in Figure 5.4. We can observe that combination #1 has the worst performance of PDR and goodput. This is because combination #1 gives all the weights to progress and therefore ignores link quality and energy ($\alpha = \beta = 0$, $\gamma = 1$). This means that a node considers only progress when calculating the DFD function. Therefore, a node always chooses the neighbor that is closest to the destination as next hop. However, the most distant neighbor has the highest probability of suffering from a bad channel quality and thus leads to higher packet loss rate. Therefore, packet delivery ratio and goodput of combination #1 are the worst.

Combinations #2 to #18 have identical coefficients for energy ($\alpha = 0.1$) since energy is not a vital metric in our experiments, and they differ in the weights for link quality (β) and progress (γ). We can find out that the combination #18, which gives more importance for link quality, has also a bad performance. This is because it gives severely unbalanced weights to progress and link quality ($\beta = 0.85$, $\gamma = 0.05$). This coefficient combination means that a node will always choose the neighbor with the best channel quality as next hop, which is the closest neighbor. However, this behavior might encounter the problem that all nodes make short progress at each hop by choosing the closest neighbor, even if there might be more distant neighbors that successfully receive the packets. This means that a packet will need more hops to reach the destination and a longer delay will occur in a sparse environment. Another reason for the bad performance of combination #18 is that, during the unicast transmission phase to the selected forwarder, there will be higher interference introduced by the closer nodes. On the other hand, combinations #2 to #18 perform better than combination #1. This is because they have different weights for link quality and progress. Then, by tuning the coefficients for link quality and progress, TLG can achieve the best trade-off between large progress and good link quality.

We can also observe that the combinations that assign fairly balanced weights to progress and link quality perform better, i.e., combinations #7 to #14. This is because under these situations, TLG will make a joint fair consideration of link quality, distance progress and remaining energy when calculating DFD with no preference to any factor. This could avoid the occurrence of the bad situations, such as choosing the most distant neighbor, which has a poor link quality, or choosing the nearest neighbor with small progress. The best performance is achieved for combination #13, which can improve the performance of PDR and goodput by nearly 50% against the worst combination #1.

However, it is interesting to notice that the best performance is not achieved by the combination with the most balanced coefficients for progress and link quality, which is combination #10 ($\beta = \gamma = 0.45$). Instead, combinations with slight imbalance between progress and link quality, such as #13, #7, #8, #14, produce

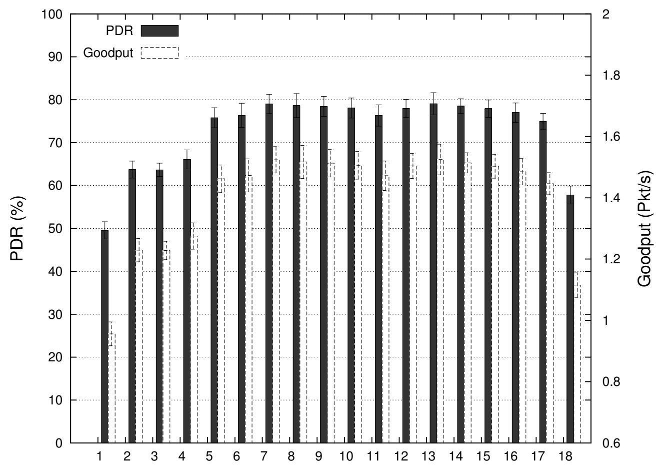


Figure 5.4: PDR and Goodput of different coefficient combinations

the best performance (from the perspective of PDR and throughput). A deep investigation into the coefficients of those combinations can reveal the fact that if a node wants to achieve the best performance, it has to give certain preference to one of the competing factors. If all the competing factors have the same weight, such as combination $\#10$ ($\beta = \gamma = 0.45$), then the best performance can not be reached. This may be because with a balanced coefficients for progress and link quality, the packet forwarding progress at each hop can not be maximized. This will lead to a lower PDR result. However, the coefficient imbalance must not be too large, otherwise the performance will degrade significantly, as for $\#1, \#2, \#3, \#4, \#18$. Therefore, depending on the application requirements, users could assign different priorities to progress, link quality, or remaining energy, to give a controlled preference to the interested factor.

To show that TLG outperforms existing approaches that consider single metrics, we compare TLG with the well-known GPSR and BLR protocols. The implementations of GPSR and BLR use a default beacon interval of 4 s, which equals to T_{LV}. The greedy mode of BLR is implemented such that a node can always find relay candidates. Figure 5.5 shows the PDR and goodput of three protocols when the source generates UDP packets. We choose only the worst ($\#1$) and the best ($\#13$) coefficient combinations of TLG to show its advantage. TLG performs much better than GPSR, which can deliver only 20% of the packets. This is because GPSR greedily chooses the neighbor that is closest to the destination as next hop. However, the furthest neighbor has the highest probability to suffer from a bad connection with the packet sender, which leads to packet loss. The non fully-covered network might be another reason for GPSR's bad performance. BLR performs better than GPSR, because it does neither have to discover and maintain routes nor to maintain a neighbor table that may be outdated and inconsistent. We can also see that BLR is better than the worst case ($\#1$) of TLG, this may be because BLR defines a "forwarding area" such that only the nodes within the region are the candidates. In TLG, any nodes that are closer to the destination could be the candidates, which increases the coordination overhead and thus reduces the performance. However, BLR is still worse than the best case of TLG (#13), since it uses only progress to compute DFD.

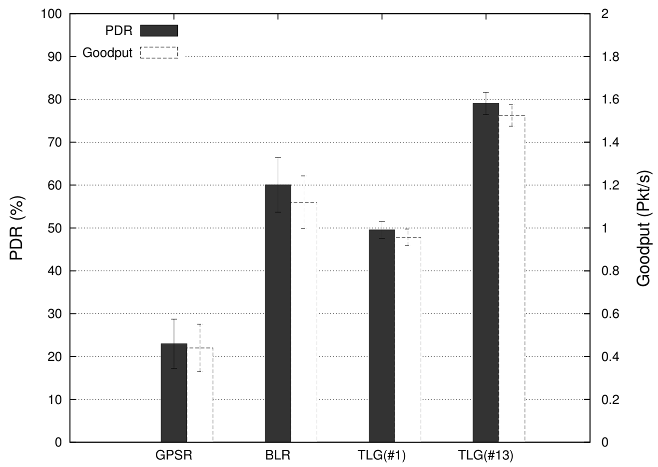


Figure 5.5: PDR and goodput of three protocols

5.4.3 Simulation Results with Video Transmission

In some applications, multimedia data can give the end-user a better understanding of what is happening in the monitored area. Therefore, in this section, we configure the source node to transmit multimedia data and the performance of different protocols are evaluated from the user's perspective via objective QoE metrics. Same as before, we compare TLG to GPSR and BLR using video data. The Hall video sequence [62] was chosen as the video source and it uses the QCIF format [43] with H.264 encoding technique [60]. Details of the video transmission can be found in Table 5.1.

Figure 5.6 shows the SSIM and VQM of GPSR, BLR, and TLG (only the worst and the best cases of combination #1 and #13). We find that TLG outperforms GPSR and BLR, by nearly 30% in the best case. This is because TLG simultaneously considers geographic progress and link quality to calculate DFD at each hop. This enables the packet sender to always select a node with good balance of distance advancement and reception link quality as packet forwarder. A node with only good distance advancement while undergoing a very bad link quality will not chosen. For multimedia transmission, it is more essential to pick up a node with good link quality as forwarder, since multimedia data is more prone to channel quality variation, and a slight link variation might cause significant visual fluctuation at the end-user side. Therefore, TLG enables the transmission of video content with QoE level assurance from a user's perspective.

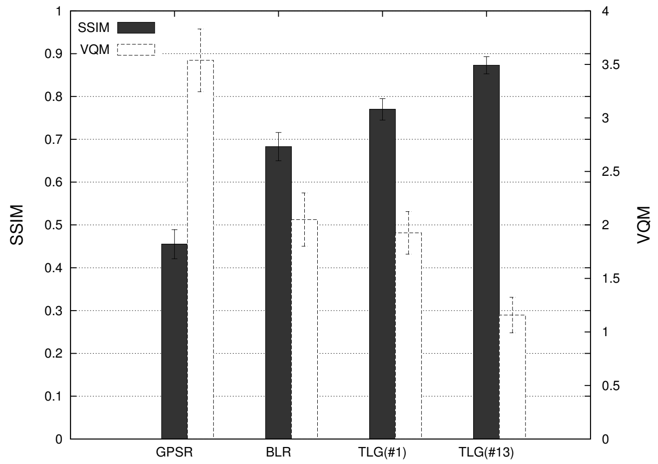


Figure 5.6: SSIM and VQM of three protocols

5.5 Conclusions

In this chapter, we proposed a new opportunistic routing protocol called TLG: Topology and Link quality-aware Geographical opportunistic routing protocol for wireless mobile ad hoc networks. TLG uses the concept of DFD to coordinate multiple receivers of a packet and each qualified receiver applies a forwarding delay timer before it rebroadcasts the received packets. The node with the smallest delay will rebroadcast the packet first and it will be selected as the forwarder. The calculation of this delay timer at each node is based on multiple types of network metrics, such as remaining energy, link quality, and progress. To validate the performance of our proposal, we evaluated TLG using both scalar and video data. QoS and QoE measurements are collected respectively to analyze protocol performance. The simulation results show that TLG achieves the best performance when multiple metrics are used to calculate DFD, and it could improve QoS metrics by nearly 40% and QoE metrics by nearly 30% compared to other routing protocols that consider single metrics.

Chapter 6

Context-aware Adaptive Opportunistic Routing Protocol (CAOR)

6.1 Introduction

In this chapter, we further extend our previous work of an opportunistic routing protocol TLG, which considers topology, link quality, and geographic information to choose packet forwarders. In this chapter, we propose a generalized concept of context-aware opportunistic routing, which is able to take any types of context information into account as far as the context is beneficial to the routing decision. Our proposal enables the protocol to add or remove context information according to different application requirements, and leave the choice of context to the applications. Additionally, our protocol supports the run-time adjustment of the parameters of multiple context according to their instantaneous values. This enables the protocol to adapt its behavior according to the latest situations of the network.

Our protocol CAOR - ***C**ontext-aware **A**daptive **O**pportunistic **R**outing* [154] [151] chooses packet forwarders by simultaneously exploiting multiple types of cross-layer context information, and the choice of the context is left to the application requirement. Different application scenarios change their preferences of context to meet their specific requirements. In the current implementation, CAOR chooses link quality, geographic progress, residual energy, and relative node mobility as the interested context. With the help of the Analytic Hierarchy Process (AHP) theory, CAOR is able to adjust the weights of context information based on their instantaneous values to adapt the protocol behavior at run-time. Moreover, CAOR uses an active suppression mechanism to reduce packet duplication, which might be a serious issue in beaconless routing. To validate the performance of CAOR, we perform different types of simulation and evaluation results show that CAOR can provide efficient routing in highly mobile environments. Additionally, the adaptivity feature of CAOR is verified.

6.2 Motivation

Context of an identity is the describing information of the entity itself and its environment. In communication networks, context can be, e.g. location, energy level, connectivity, etc. If the network is dynamic, the context can vary dynamically, for example due to node mobility or wireless link quality variation in multihop wireless communication networks. Context-based routing enables network nodes to make routing decisions according to the context values. Existing works on context-aware routing are limited to the consideration of energy (energy-aware routing) or location (geographic routing). The routing concept proposed in this chapter takes several types of context into account simultaneously. Therefore, the route selection depends on multiple parameters and requires finding a local optimum solution based on the current context values and the current route decision preferences. This means that parameters have to be combined into a multi-dimensional cost function. This cost function is applied to the route/forwarder selection process, using network-specific context criteria.

The feature of adaptivity to varying situations is important in dynamic environments. When multiple types of metrics are considered to choose packet forwarders, the weights assigned to different metrics play an important role and have significant impacts on routing results. Therefore, the weights of different routing metrics should be controlled adaptively according to their real-time values. For example, energy is an critical factor in WSN and its protocol design. If a node has detected that its residual energy is below a certain threshold value, it should respond actively to avoid further energy consumption. To do this, it could increase the weight of energy, such that energy plays a more important role in the forwarder selection process. However, most of the existing works on context-aware routing just show the importance of being adaptive, without the support of adaptivity at run-time. Adaptive applications usually require a selection or decision making from certain alternatives. Therefore, a decision-making algorithm is needed to provide adaptivity. The Analytic Hierarchy Process (AHP) theory [123] is a multi-criteria based decision-making algorithm that considers different options, and it has been applied for the calculation of combined metrics [25]. Therefore, in this work, we apply AHP as the decision-making solution to adaptively adjust the weights of different context information.

Coordination among multiple packet forwarding candidates is important to avoid duplicates in opportunistic routing. BLR and TLG implement the coordination mechanism through a timer, such that when a node has its timer expired, it retransmits the packet. Other candidates cancel their timers when overhearing the retransmission, such that only one candidate will actually forward the packet. However, if other candidates do not overhear the retransmission due to some reasons, such as channel fluctuation or interference, they will retransmit the same packet, which leads to duplicates. Therefore, an efficient coordination mechanism is needed in beaconless routing approach.

6.3 Protocol Description

In this section, we present the design of our proposal of the context-aware adaptive opportunistic routing (CAOR). Compared to the our previous protocol contribution, which is a topology and link quality-aware geographic opportunistic routing protocol (TLG), CAOR has the following five new features:

- A DFD-based forwarder selection mechanism, which takes into account general context information.
- A multiplicative calculation of DFD, which avoids possible problems of using an additive function.
- Incorporating relative node mobility to choose packet forwarders.
- The exploration of AHP to dynamically adapt the weights of context information according to their real-time values to modify protocol behaviors.
- A lightweight active suppression mechanism to reduce duplicate transmissions.

6.3.1 Context Information

Context refers to any information that can be used to characterize the attributes of an entity. In the scope of this work, context means any information with impact on routing and they could be classified into three types: local, link, and global context. Local context includes local attributes of nodes such as location, mobility and residual energy. Link context includes various properties associated with wireless links such as quality and bandwidth. Global context includes attributes of networks such as topology and traffic load. Existing solutions integrated few of these context information with fixed parameter values and thus are not able to cope with varying network conditions. Context-awareness implies that an entity performs actions by considering the context information of itself and its neighborhood. When referring to packet routing in mobile ad-hoc networks, context-aware routing means that the routing decision is made by utilizing the context information described above.

6.3.2 Dynamic Forwarding Delay (DFD)

We assume that in CAOR each node is aware of its location via a GPS device or another location service. Whenever a source wants to send a packet, it broadcasts it, including the location of itself and the destination. Due to their location-awareness, nodes that receive the packet check whether they are closer to the destination or not. If yes, they are qualified and use the same idea of Dynamic Forwarding Delay (DFD) (section 5.3.1 in TLG) to start a timer before rebroadcasting the packet. Multiple neighbors compute and initiate their DFD simultaneously, and the one with the shortest DFD wins the competition and retransmits the packet first. By overhearing the retransmission, other candidates stop their timers and drop the packet. The retransmission is also used as a passive acknowledgment to inform the source about the winner of the competition such that it can send subsequent packets using unicast.

CAOR leaves the decision of which context information to be included in the routing process as an open choice, such that any context can be added into the DFD calculation as far as it has impact on routing decisions. The context information of a node is defined as $(x_1, x_2, ..., x_n)$. Currently CAOR incorporates four types of context information: x_1 = *Link Quality*, x_2 = *Progress*, x_3 = *Remaining Energy*, and x_4 = *Link Validity Duration* to choose forwarders. Depending on the applications, additional context may be added. For example, buffer information is critical for multimedia applications and thus should be included. An additive function is not a good option for combining context: even if one criterion has a very bad value, which should make the candidate undesirable, this may be compensated by good values of the other criteria as long as all context information are accumulated. Instead, a multiplicative operation is immune to this problem, since an unacceptable value of any context can simply map the whole function into a bad result. Therefore, CAOR uses a *multiplicative* combination of the four types of context information to define the DFD, as shown in Eq. (6.1):

$$DFD(x_1, x_2, ..., x_n) = DFD_{Max} \times \prod_{i=1}^{n} w(x_i) \cdot DFD(x_i) \tag{6.1}$$

$DFD(x_i)$ is the i-*th* component of DFD related to context x_i and $DFD(x_i) < 1$. $w(x_i)$ is the weight factor of that context and $\sum_{i=1}^{n} w(x_i) = 1$. The weight $w(x_i)$ is a function of x_i, which means that CAOR determines the weights of different types of context information according to their values. DFD_{Max} is the predefined maximum delay allowed at each node. Details of the DFD components are presented in Eq. (6.2) - (6.4). Since exponentially distributed random timers can reduce the number of responses, which leads to a lower feedback latency and better feedback suppression than others, we define different DFD components using exponential distribution [66].

Link Quality

Compared to the definition of link quality in TLG (section 5.3.1), CAOR uses an exponential distribution to calculate link quality. The calculation of DFD(link quality) is shown in Eq. (6.2). The link quality is usually measured at the physical layer. For example on sensor nodes, the CC2420 radio chip measures the physical layer information and provides the Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) for each received packet. These parameters directly reflect the immediate link quality. In our study, we use LQI as the indicator of link quality between two nodes. The threshold values of different parameters in Eq. (6.2) are identical with that in Eq. (5.2).

$$DFD(x_1) \sim DFD(\text{link quality}) = \sqrt{\frac{e - e^{(\frac{LQI_t}{LQI_{Max}})}}{e - 1}} \tag{6.2}$$

Progress

Eq. (6.3) defines how DFD(progress) in Eq. (6.1) is calculated. Compared to the calculation of progress in TLG (Eq. 5.3), CAOR uses an exponential distribution.

$$DFD(x_2) \sim DFD(\text{progress}) = \sqrt{\frac{e - e^{(\frac{P_i}{2R})}}{e - 1}} \tag{6.3}$$

in which P_i is the progress of node i, and R is the radio range. The definition of progress is identical with that of TLG, which can be found in section 5.3.1.

Link Validity Time Estimation

Geographic routing selects forwarders solely based on the position information of nodes. However, in the presence of node mobility, if only position information is used, it is possible to miss some good candidates due to mobility. Some mobile nodes, whose current locations are not preferable, might move to new locations and have the potential of being better forwarders. Therefore, node mobility information (moving direction and speed) should also be included in the design of routing protocol to further improve system performance.

To make use of the mobility information, CAOR exploits the relative movement between two mobile nodes. CAOR prefers nodes with promising movement tendency, even if their current locations are not favorable. As shown in Figure 6.1, A should take C as a relay, since C is moving towards D and it can bring the packets closer to D, even if B has a more favorable location at this moment. Without considering the relative mobility, A will take B as its next-hop due to its location advantage, even though B will bring the packet further away from the destination.

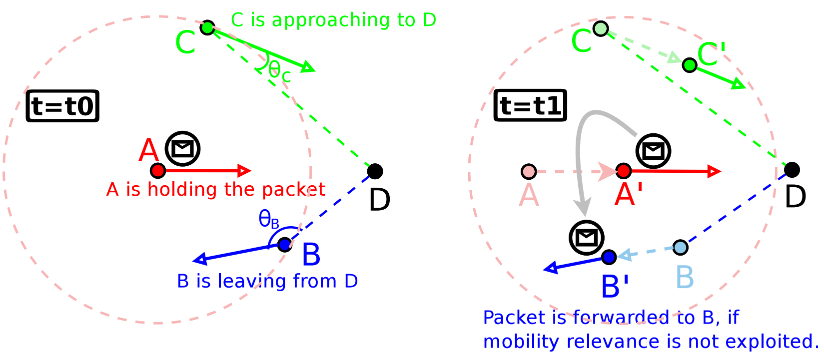


Figure 6.1: Inefficient packet relays without mobility relevance

In order to achieve the proper utilization of relative mobility, a node has to notify its neighbors about its location whenever it initiates a packet transmission. If a source wants to send a packet, it adds its location and mobility (speed and direction) into the header and broadcasts it. After receiving a broadcast, if a node is qualified, it starts a link validity estimation (LIVE) process to estimate the validity period of that link. $DFD(\text{LIVE})$ is defined in (6.4), where θ is the angle between a node's moving direction and the line connecting the destination and itself (as shown in Figure 6.1). θ = 0 means that the node is moving exactly towards the destination, which should lead to the shortest delay; while θ = 180 means the node is moving towards the opposite direction of the destination, which will generate the longest delay. T_{LV} is how long the link is expected to hold, and is calculated according to (6.5). For example in Figure 5.3, if nodes A and B are moving with (speed V_a, direction ω_a) and (speed V_b, direction ω_b), given their initial location of (X_A,Y_A) and (X_B,Y_B), they can compute the validity period of the link connecting them.

$$DFD(x_4) \sim DFD(\text{LIVE}) = \frac{1}{\sqrt{\frac{e-e^{(\frac{\theta}{180})}}{e-1}} \times T_{LV}} \tag{6.4}$$

$$\begin{aligned} &[(X_B + V_b \cdot \cos\omega_b \cdot T_{LV}) - (X_A + V_a \cdot \cos\omega_a \cdot T_{LV})]^2 \\ &+[(Y_A + V_a \cdot \sin\omega_a \cdot T_{LV}) - (Y_B + V_b \cdot \sin\omega_b \cdot T_{LV})]^2 = R^2 \end{aligned} \tag{6.5}$$

After the packet transmission, the source should also know the value of the "LIVE" of this link such that it can unicast subsequent packets within the validity period of this link. When this link validity time expires, the source will start another broadcast process to search for another forwarder. If either end of the link changes its mobility before the expiration of the calculated T_{LV}, that node has to inform this to the other by piggybacking its movement data in the packet header and broadcast the packet. In this way, the other node of the link is aware of the fact the link will be broken and it should update the validity time of that link timely. If the other node is the packet sender, then it should stop unicast transmission and start a new forwarder selection process. It is important to make sure that the notification of the link breakage notification is received by the other side of the link. Otherwise it might happen that the source node of a data transmission is not aware of the leaving of the endpoint of the link, and it keeps sending unicast packets. However, the receiver has left the communication and all the packets will be lost since the leaving of the receiver.

Energy

Eq. (6.6) defines the energy part of the DFD function. A node with a high remaining energy value (E_r) generates a small $DFD(\text{remaining energy})$ value, which means a small contribution to the DFD. Compared to the calculation of energy in TLG (Eq. 5.4), CAOR uses an exponential distribution.

$$DFD(x_3) \sim DFD(\text{remaining energy}) = \sqrt{\frac{e - e^{(\frac{E_r}{E_0})}}{e - 1}} \tag{6.6}$$

The threshold values of different parameters in Eq. (6.6) are identical with that in Eq. (5.4).

6.3.3 Run-time Self-adaption of Context Weight by AHP

When multiple types of context information are integrated, the assignment of the weights of different context information will significantly affect the performance of the protocol. The weights of various context information should depend on their values. This is because a fixed assignment of weights only reflects the relative importance of context information and fails to consider the real-time values of the context. For example in battery sensitive applications, a fast drop in battery level indicates the imminent exhaustion of energy, thus the protocol should prevent the node from packet forwarding by reducing the weight of battery to save energy. Besides, some context information have conflicting objectives, for example *link quality* and *progress*. A good value of *link quality* indicates short progress, which conflicts with a good value of *progress*. Optimizing all context information simultaneously is not possible. We must trade off the achievement of one objective (i.e., the optimization of one context) against another one. In this context, the problem of determining the weights of different context can be regarded as a decision-making problem, which should be solved by some decision-making approaches. In this work, CAOR applies the Analytic Hierarchy Process (AHP) theory [123] to adapt the context weights according to their values.

In the context of this work, CAOR uses AHP and resolves the problem of context weight assignment into the sub-problem of comparing the importance of each context-pair. CAOR takes the real-time values of the four interested context, namely *Progress (PG)*, *Link Quality (LQ)*, *Residual Energy (RE)*, and *Link Validity (LV)*, as inputs, and applies the AHP theory to generate the weights of four criteria. An AHP hierarchy for next-hop selection is shown in Figure 6.2, and it has three steps:

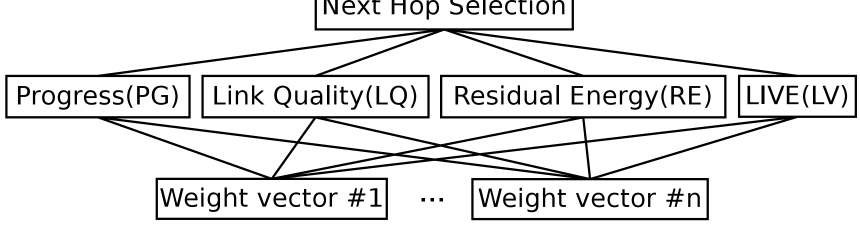


Figure 6.2: AHP hierarchy for next-hop selection in CAOR

Table 6.1: Pairwise context importance

c_{ij}	**Importance relation**
3	i is much more important than j
2	i is more important than j
1	i is as important as j
1/2	i is less important than j
1/3	i is much less important than j

(1) Construct a Comparison Matrix

A comparison matrix shows the pairwise relative importance between every two context types. As shown in Eq. (6.7), $A = (c_{ij})_{n\times n}$ is a comparison matrix, where n denotes the number of elements to be compared. The value of c_{ij} shows how important the i-th element is compared to the j-th element. Table 6.1 defines *five* importance levels.

$$A = (c_{ij})_{n\times n} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \tag{6.7}$$

In CAOR, every candidate constructs its own comparison matrix by comparing the importance of all context-pairs according to their instantaneous values. We define certain thresholds for each context to divide its value into three category ranges (the first range means poor, the second range means intermediate, and the third ranges means good), as shown in Table 6.2. Then, the importance comparison of two contexts is based on the ranges in which the context values are. In general, context i is more important than context j if its value is in a more critical range. For instance, context i has a value in the first category range, and context j has a value in the third category range, then context i is in a much more critical range and the importance between i and j is 3. As a concrete example, after a successful packet reception, if a candidate derives that for the received packet, its current context value ranges are: *link quality* ∈ (LQI_{good}, LQI_{max}), *progress* ∈ (0, $\frac{2}{3}R$), *remaining energy* ∈ ($\frac{1}{3}E_0$, $\frac{2}{3}E_0$), and *LIVE* ∈ (0, 3), then according to Table 6.1, its comparison matrix can be constructed, as shown in Eq. (6.8).

(2) Calculate the Weight Vector

With the comparison matrix $A = (c_{ij})_{4\times 4}$ generated, the next step is to calculate its eigenvalue equation, namely $AW = \lambda_{max}W$, where W is a non-zero vector called eigenvector, and λ_{max} is a scalar called eigenvalue. By solving the eigenvalue equation, we derive the eigenvector W, which is also the weight vector of the context information (according to the eigenvalue method (EM) included in [123]).

Table 6.2: Threshold definition of context information

Context	***Ranges***
Link Quality (LQ)	$(0, LQI_{bad})(LQI_{bad}, LQI_{good})$ (LQI_{good}, LQI_{max})
Progress (PG)	$(0, \frac{2}{3}R)(\frac{2}{3}R, \frac{4}{3}R)(\frac{4}{3}R, 2R)$
Residual Energy (RE)	$(0, \frac{1}{3}E_0)(\frac{1}{3}E_0, \frac{2}{3}E_0)(\frac{2}{3}E_0, E_0)$
LIVE (LV)	$(0, 3)(3, 5)(5, \sim)$

Based on the comparison matrix of (6.8), its weight vector can be derived as $W = (0.351, 0.109, 0.189, 0.351)^T$.

$$A = (c_{ij})_{4\times4} = \begin{pmatrix} & PG & LQ & RE & LV \\ PG & 1 & 3 & 2 & 1 \\ LQ & 1/3 & 1 & 1/2 & 1/3 \\ RE & 1/2 & 2 & 1 & 1/2 \\ LV & 1 & 3 & 2 & 1 \end{pmatrix} \tag{6.8}$$

(3) Check for Consistency

If comparison matrix A satisfies two rules: 1) $c_{ij} = \frac{1}{c_{ji}}$; 2) $c_{ik} \cdot c_{kj} = c_{ij}$, then the matrix is a consistency matrix, and the derived weight vector is correct. However, due to the deviations during the construction of the comparison matrix, it is usually not perfectly consistent. Therefore, a standard consistency check and adjustment process should be made to revise the comparison matrix into a consistent one. AHP uses a *consistency ratio (CR)* to represent the deviation produced when constructing the comparison matrix. *CR* is defined as the ratio of *consistency index (CI)* to *random index (RI)*, which is defined in Table 6.3. *CI* can be achieved by Eq. (6.9).

Table 6.3: Random index (RI)

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.96	1.12	1.24	1.32	1.41	1.45	1.49

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{6.9}$$

where λ_{max} is the eigenvalue of the constructed comparison matrix and n is the number of elements to be compared. Therefore, if CI = 0, then the constructed comparison matrix is a consistent one; otherwise it is not. When $CR = CI/RI \leq 0.1$, then the inconsistency of the constructed comparison matrix is acceptable. Otherwise, the pairwise comparison should be adjusted until matrix A satisfies the consistency check.

Following the process of consistency checks, we can find out that the comparison matrix A in Eq.(6.8) has CI = 0.00333 and CR = 0.003472 $\leq$ 0.1. Therefore, the inconsistency of the constructed pairwise comparison matrix is acceptable. When the consistency check is passed, the generated context weights are validated.

6.3.4 Reducing Packet Duplicates

Packet duplication can be a serious problem of beacon-less routing. This is because in beacon-less routing, there is no centralized coordinator to indicate explicitly which receiver will be the only forwarder of a packet. Multiple receivers may attempt to forward the packets at the same time due to the hidden terminal problem. For example, if a source node broadcasts a packet, node A and B both receive it and they will compete to be the forwarder. They will start the calculations of their DFD, and the node with a short delay timer will broadcast the received packet first (assume it is node B). However, this broadcast may be not successfully received by node A (either because they are not in the radio range of each other or because the connection between them is interrupted by unknown interference). In this case, node A will regard itself as the only candidate and will also broadcast the same packet, which creates duplicates. For wireless sensor networks, where resources and bandwidth are both scarce, duplicates should be controlled strictly to improve system performance. To achieve this goal, CAOR applies two mechanisms to reduce duplicates.

Reducing Duplicates at Intermediate Nodes

When an intermediate node wants to forward a carried packet, it applies a three-way handshake to find the relay. Unlike the Request-To-Send (RTS)/Clear-To-Send (CTS) mechanism in IEEE 802.11 MAC, a CAOR source broadcasts a data packet as an implicit RTS and waits for responses from candidates. Using a data packet as implicit RTS message has another advantage. In lossy wireless environments, the packet size usually has a direct relationship with the error probability. Concretely, bigger packets have less probability of being received than smaller ones. Based on this observation, discovering the neighborhood using one small control packet may cause a routing protocol to select a next-hop which is not able to receive the bigger data packet. Therefore, CAOR uses normal data packets as implicit RTS control messages instead of using specific small-size control packets.

After receiving a data packet, qualified neighbors start their DFD calculation individually. When the DFD timer expires, instead of re-broadcasting the data packet immediately, a candidate first broadcasts a Clear-To-Broadcast (CTB) packet that contains its location to notify the source node about its existence. A candidate cancels its timer if it receives a CTB packet from another candidate. Since CAOR does not use any “forwarding area”, a candidate may not receive the CTB packet from another one, if they are not within the radio range of each other or if the interference interrupts the packet transmission. As shown in Figure 6.3, node 2 and node 3 can not overhear the CTB packet of each other. When the source successfully receives the first CTB packet, it selects the candidate, from which the

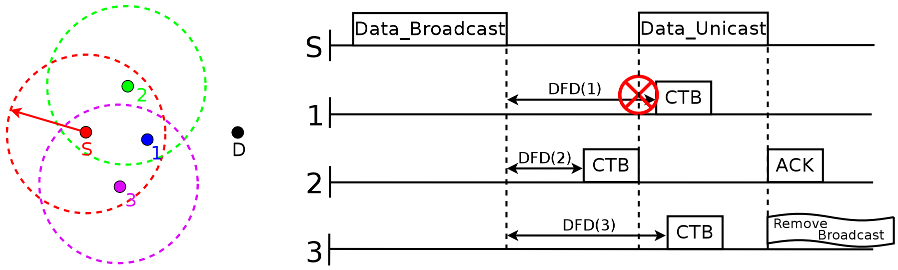


Figure 6.3: CAOR handshake

CTB packet comes, as its next-hop to send the subsequent packets using unicast. To avoid duplicates, the source node ignores further CTB packets that arrive later. After receiving the unicast transmission, each candidate can infer whether it has been selected as the forwarder or not by checking whether itself is the target receiver of the unicast transmission. If a node finds out that the unicast is not for itself, it can infer that another node has won the competition, and it drops the packet. If a node detects that the unicast is for itself, then it means itself has won the candidate selection before and it is the forwarder now. In this case, it will send an acknowledgment and repeat the whole process. This unicast transmission is kept until the moment when a unicast acknowledgment is not received by the source node. In this case, the source node believes that the link with the selected forwarder is invalid anymore. Therefore, the source node stops the unicast transmission, and repeats the broadcast procedure to find new packet forwarders.

Reducing Duplicates at the Destination

When receiving a packet for the first time, the destination should notify its neighbors by broadcasting a message including the sequence number of the received packet. Neighbors that still hold the packet check if this broadcast comes from the destination: if yes, then they cancel the timer and delete the packet; if not, the timers continue to count off. For nodes that do not have that packet, the received broadcast message is simply dropped, if it comes from the destination; otherwise, a timer is triggered.

6.3.5 Overhead of Context Awareness and the AHP Operation

CAOR utilizes various types of context information. These contexts are required locally (node's residual energy and geographic progress) or derived from the received broadcast packets (link quality and node mobility), which means CAOR has no additional communication overheads. Meanwhile, computation and storage costs are needed indeed to store and manipulate these contexts. However, since CAOR includes neither prediction/estimation operation, nor routing table maintenance, the overhead of context manipulation and AHP calculation is tolerable. In this work, we follow the common consensus that the wireless radio transceiver dominates the energy consumption of low power wireless devices, and thus, ignore the energy depleted in the operations uncorrelated with radio transceivers.

6.4 Performance Evaluation

To evaluate the performance of CAOR, we perform extensive simulations in OMNeT++ using the extended version of the evaluation framework [118]. We compare CAOR against the well-known geographic routing protocol GPSR, the beaconless routing protocol BLR, and the TLG protocol [157]. Packet delivery ratio (PDR), packet duplicates, average end-to-end delay and network lifetime are collected as performance metrics.

6.4.1 Simulation Settings

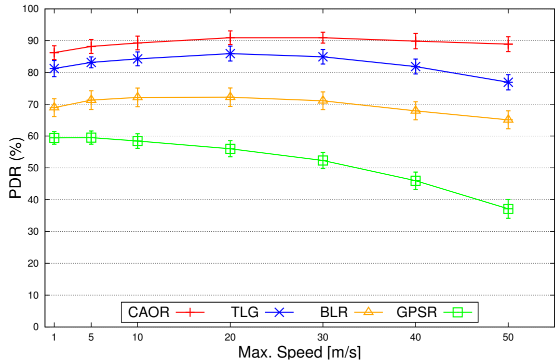
We randomly deployed 31 nodes in a flat area of size $50m \times 50m$, including 1 source, 1 destination, and 29 intermediate nodes. Source and intermediate nodes move under the Random Waypoint mobility model [13], generated by the BonnMotion tool [1]. Because nodes are mobile, we used somewhat larger area than the simulation performed in TLG (section 5.4). The source node generates constant bit rate UDP packets with a default rate of 2 packets/s. A CSMA implementation from Castalia [33] is chosen as the MAC protocol for CAOR, and an irregular radio module from Castalia has been used. A nominal transmission range of 11 m is set. Each simulation runs for 300 s, and the results are averaged over 30 runs with random-generated seeds to provide a confidence interval of 95%. Table 6.4 shows the simulation parameters.

Table 6.4: Baseline simulation parameters

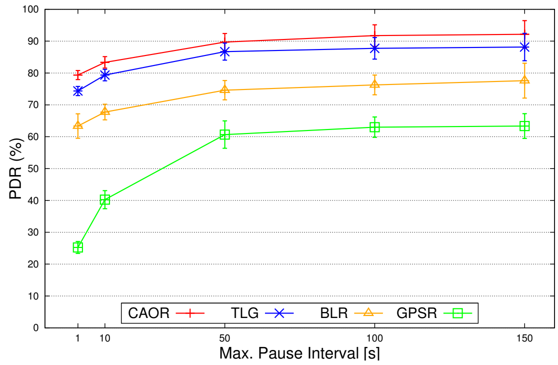
Parameter	Value	Parameter	Value
Source location	(1,1)	Radio model	CC2420
Destination location	(48,48)	Path loss model	Lognormal
Max. Speed	10 m/s	Transmission power	-10 dBm
Max. Pause	50 s	DFD_{max}	10 s
Radio Range	11 m	Data rate	2 Packet/s

6.4.2 Performance Analysis

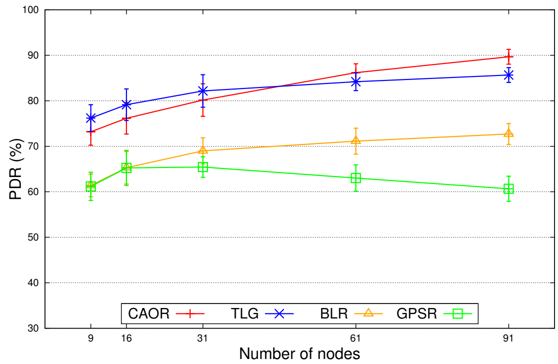
In this section, we show the evaluation results of the four protocols, namely packet delivery ratio, number of duplicated packets, average end-to-end delay and network lifetime of the four protocols with varying maximum speeds, pause intervals, and node densities, separately.



(a) Maximum speed



(b) Maximum pause interval



(c) Node density

Figure 6.4: PDR as a function of (a) Maximum speed; (b) Maximum pause interval; (c) Node density

Packet Delivery Ratio (PDR)

Fig. 6.4a presents the PDR of the four protocols with different maximum moving speeds. GPSR performance is the worst and degrades significantly as speed increases, due to its location sensitivity. GPSR has to maintain neighbor table and routes, which will be outdated more frequently when the maximum moving speed increases. For BLR, TLG, and CAOR, as the speed increases, their performance values first improve (with speed up to $\sim$ 20 m/s) and then become worse. This is because with low mobility, nodes have higher chances to meet a better forwarder to transmit the packets. However, if the speed is too high ($>$ 30 m/s), the contact duration between two nodes is too short, which reduces performance. This is more severe for CAOR and TLG, due to their dependence on the link validity duration. TLG and CAOR outperform BLR, because BLR only considers the progress and it always selects the neighbor closes to the destination as next-hop, which has a higher chance to suffer from a weak radio link. TLG does not address the problem of duplicates. Redundant transmission creates a lot of contention at the MAC layer, which affects its performance. Due to the three-way handshake mechanism, CAOR significantly reduces duplicates and therefore outperforms TLG.

Fig. 6.4b shows the results of PDR as a function of maximum pause time of mobile nodes. As we can see, with a long pause interval, TLG and CAOR perform very well. This is because they rely on the link validity duration, which is long if the nodes have long pause time under the Random Waypoint mobility model. When the pause interval is short, the encounter time between mobile nodes will be short, which brings troubles to other protocols. When nodes frequently change their movements, the routing table of GPSR is not able to keep up with the topology changes. Therefore, GPSR also performs badly when the maximum pause time is short.

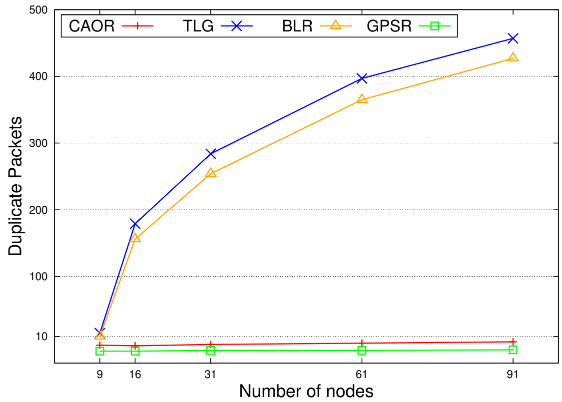
Fig. 6.4c indicates the influence of node density on the PDR of the four protocols. We can observe that, as the number of nodes increases, GPSR performance improves first and then becomes worse for number of nodes $>$ 31. This may be because with a higher node density, additional control packets produce too much congestion, which leads to worse performance. TLG and CAOR are both better than BLR, since they consider multiple context information when choosing the next-hop. The candidate with proper values of progress, link quality with last-hop, enough residual energy, and long validation interval with last-hop will be preferred to be the forwarder. While BLR prefers nodes with the largest progress without considering the quality of the links during the selection of forwarder, nodes with lossy links are likely to be chosen. TLG performs better than CAOR, if node density is low. This is because with few nodes involved in packet transmission, it is more likely that each node has full workload, which leads to fast exhaustion of the limited context capacity. In this situation, CAOR will adapt the weights of the context information at run-time to prevent nodes from competing for packet forwarding, which restricts its performance. However, as the number of nodes increases, there are more candidate nodes with good context values available in the network. Therefore, CAOR can further improve its performance by distributing the packet forwarding tasks to nodes with more sufficient and balanced context values. Additionally, since TLG has no control of packet duplicates, duplicated transmission will be a serious problem for TLG as the node density increases. This will create congestion at the MAC layer and limit its performance improvement. Therefore, CAOR outperforms TLG when node density is high. In this way, the advantage of using AHP to adapt the weights of context is also validated.

Packet Duplicates

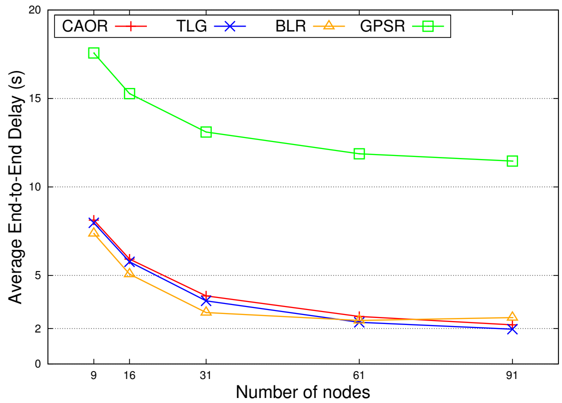
Duplicated transmission is a serious problem for beacon-less routing solutions. As shown in Fig. 6.5a, GPSR is almost free of duplicates, since it uses a neighboring table to control packet transmission. The data transmission at each hop has concrete destination, which avoids the possibility of duplicates. BLR and TLG have a lot of packet duplicates, because they do not have any explicit coordination mechanism to control the competition from multiple forwarding candidates. It happens frequently that multiple receivers of a broadcast transmission can not overhear each other and they all re-broadcast the same packet, which generates a lot of duplicates in the network. BLR has fewer duplicates than TLG, because it defines a "forwarding area" such that the nodes within the area are supposed to hear each other to avoid duplicates. However, the irregular property of the wireless transmission makes more neighbors to consider themselves as the only forwarder by not overhearing each other, even if they are in the same forwarding area. CAOR selects the next-hop based on a three-way handshake mechanism to make sure that only one neighbor will actually forward the packet. It efficiently eliminates the duplicates, at the price of some low-level control overhead.

Average End-to-End Delay

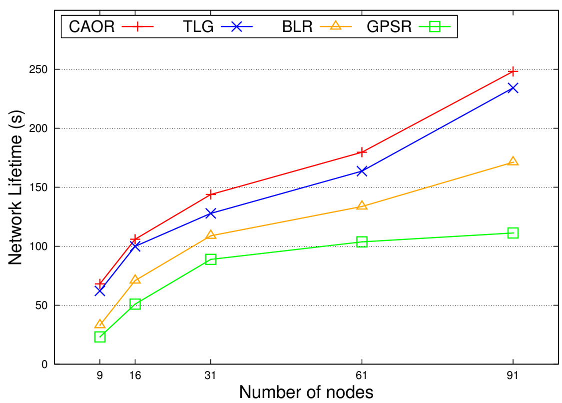
Delay results of the four protocols with different node densities are presented in Fig. 6.5b. GPSR has the worst performance, because it always uses the same nodes recorded in the routing table for packet relay. Due to unexpected link failures or topology changes, packets might be lost or the routing table might be outdated. Therefore, a source node has to find another path and all subsequent packets have to be buffered and delayed. BLR, TLG, and CAOR have only a fraction of the average end-to-end delay compared to GPSR. This is mainly due to two reasons: first, opportunistic routing allows packets to reach the destination in fewer hops than GPSR; second, BLR, TLG, and CAOR do not rely on routing tables, therefore they are immune to packet retransmission caused by outdated routing tables. When node density is low, TLG and CAOR have somewhat longer delays than BLR. This is because they do not choose the most distant neighbor as the forwarder at each hop, which means the packet has to go through more hops to reach the destination. When more nodes are deployed, it is more likely that BLR chooses a more distant neighbor as forwarder. Therefore, that neighbor will bear a poor link and



(a) Packet duplicates



(b) Average end-to-end delay



(c) Network lifetime

Figure 6.5: Influence of the node density on (a) Packet duplicates; (b) Average end-to-end delay; (c) Network lifetime

need re-transmission, which leads to longer delay. That's why when node density increases, the delay of BLR becomes longer. Due to the three-way handshake, CAOR experiences some longer delays than TLG.

Network Lifetime

Network lifetime is defined as the time duration from the beginning of the data transmission until the failure of the first node due to battery exhaustion. As shown in Fig. 6.5c, GPSR is the worst protocol, since it keeps using the same nodes that are recorded in the routing table for packet transmission, which leads to fast energy depletion of that set of nodes. The overhead for maintaining the neighbor table also consumes additional energy. As the number of nodes increases, there are more nodes involved in packet transmission and on average they could share the packet forwarding tasks. Therefore, the network lifetime is improved. BLR does not use any beacons and has no additional control overhead. It finds packet forwarders hop-by-hop in a fully distributed way. Therefore, it has less energy consumption than GPSR and the network lifetime is longer. With increased node density, the network lifetime is also extended. Due to the consideration of nodes' residual energy, TLG and CAOR prefer nodes with sufficient battery level, which enable them to distribute the packet forwarding task uniformly and this prolongs the network lifetime. CAOR performs better than TLG, this is mainly due to three reasons. First, with the help of AHP, CAOR is able to adapt the weight of energy according to its real-time value dynamically to prevent a node suffering battery exhaustion from packet forwarding. On the contrary, TLG uses a fixed context weights and cannot react to their instantaneous values. Second, CAOR uses a multiplicative function to combine contexts, while TLG uses an additive one. Even if TLG also considers energy, due to the additive function when calculating the DFD, a bad value of *remaining energy* can be compensated by a good value of *progress* or other contexts. This allows a node with exhausted battery to be involved in packet forwarding. This further reduces the network lifetime. Last, since TLG has no control of duplicates, the redundant transmission, as shown in 6.5a, will cost more energy and result in a reduced network lifetime. CAOR significantly eliminates the duplicates, therefore saves the scarce energy from being wasted.

6.5 Conclusions

In this chapter, we analyzed the drawbacks of existing candidate list-based opportunistic routing in the presence of dynamics in mobile ad-hoc networks. Based on that, we have proposed and evaluated a novel concept of opportunistic routing protocol called CAOR - Context-aware Adaptive Opportunistic Routing protocol. CAOR abandons the usage of candidate list, and it allows all the qualified nodes to participate into the packet forwarding task. To deal with mobility, CAOR utilizes the relative movement between two mobile nodes and it prefers nodes with promising movement tendency, even if their current locations are not preferable. Compared to our previous protocol contribution of TLG, the novelties of CAOR are multi-fold:

- It extensively exploits multiple context information to make the routing decision. CAOR adopts a timer-based coordination mechanism and the calculation of the timer is based on nodes' link quality, geographical progress, residual energy, and link validity duration.
- It uses a multiplicative calculation of a timer, which avoids possible problems of using an additive function.
- It incorporates the relative node mobility to choose a node with more favorable mobility pattern as the packet forwarder.
- It applies the AHP theory to adjust the weights of various context information according to their importance to adapt the protocol behavior at run-time.
- It uses an active suppression mechanism to reduce packet duplicates to further improve performance.

Simulation results show that CAOR can provide efficient routing performance in mobile scenarios compared to other routing approaches that solely use single metrics and without any adaptivity mechanism. With the ability of self-adaptive and duplicate reduction, CAOR outperforms other beacon-less opportunistic routing protocols under certain network scenarios as well.

Due to the intrinsic limitations, a network simulator can not accurately monitor the energy consumption and computation overhead associated with CPU processing and memory access, which are highly correlated with the specific physical characteristics of the wireless devices. This drawback brings difficulty to measure the overhead to perform the context manipulation and the AHP calculation. However, even if we did not include the overhead and energy measurement of the AHP-related operation, the improved network throughput and reduced network lifetime, from another perspective, confirm the efficiency of using AHP to improve network performance.

Chapter 7

Sensor Context-aware Adaptive Duty-cycled Opportunistic Routing for Wireless Sensor Networks (SCAD)

7.1 Introduction

Wireless sensor networks (WSNs) pose special requirements, such as low-power transmissions, various unknown interferences and resource constraints. In order to be energy-efficient, duty-cycling operation has been widely used to conserve energy. However, the temporary unavailability of sensors can adversely affect both network coverage and connectivity. Opportunistic routing (OR), even though was originally proposed to improve performance of wireless multihop communication, can not be directly used in WSNs due to some intrinsic design features of WSNs:

Energy Efficiency, Reliability vs. Throughput: Opportunistic routing was originally designed to improve the throughput of multihop wireless ad-hoc networks. However, wireless sensor network applications normally have a high preference of reliable transmission with energy efficiency and not high throughput. In this chapter, we show how opportunistic routing can be adapted to improve energy efficiency while keeping performance at a satisfactory level.

Duty Cycling in Wireless Sensor Networks: WSN nodes are usually duty-cycled to ensure long network lifetime. However, duty-cycling limits the number of nodes that can concurrently overhear packet transmissions. As a result, it might prevent spatial reuse in the forwarding process, which is one of the key benefits of OR. However, as we will show in this chapter OR brings low latency to duty-cycled networks: instead of waiting for a given forwarder to wake up, the anycast primitive allows the first awoken node to be the forwarder.

Complexity of Unique Forwarder Selection: Opportunistic routing relies on a consensus protocol to determine a unique forwarder among the receiving nodes. For example, each ExOR packet contains a list of potential forwarders ranked according to their priorities. Due to the small packet size in wireless sensor networks, the usage of such forwarder list is not really feasible. Similarly, assigning time slots to each potential forwarder poses implementation challenges.

7.2 Motivation

The biggest difficulty that prevents the direct application of opportunistic routing to WSN routing is its dependency on candidate lists. This problem, however, can be easily removed when a beaconless opportunistic routing approach is used. Beaconless opportunistic routing forwards packets in a fully distributed way, without using any beacon messages. Each node starts the packet transmissions by broadcasting, and the selection of packet forwarders are determined at the receiver side.

In this chapter, we describe our proposed protocol of SCAD - *Sensor Context-aware Adaptive Duty-cycled* beaconless opportunistic routing protocol for WSNs [150] [149]. SCAD applies the concept of beaconless opportunistic routing into wireless sensor networks. We consider the particular requirements and challenges of WSNs by taking energy efficiency as the key target and tailoring OR to adaptively duty-cycled sensor nodes. SCAD selects packet forwarders by jointly considering multiple types of cross-layer context information, such as link quality, progress, residual energy, and energy draining rate. An adaptive duty-cycling scheme is also implemented in SCAD to tune the sleep intervals of sensor nodes according to traffic loads and energy drain rates. The contributions of SCAD are threefold:

- It smoothly integrates opportunistic routing with duty cycling of wireless sensor nodes.
- It considers sensors' energy drain rates when selecting packet forwarders, which leads to a evenly distributed energy depletion over different nodes.
- It adapts duty cycles of sensor nodes according to real-time traffic loads and energy drain rates.

Extensive OMNeT++ based simulations showed that SCAD outperforms other protocols significantly. It provides satisfactory performance while keeping network alive longer in both static and mobile scenarios. Real-world implementation and evaluations on the TinyOS sensor operation system running on Tmote Sky nodes validated the performance of SCAD in different network topologies [149].

7.3 Protocol Description

This section introduces the packet forwarding mechanism of SCAD. SCAD is a cross-layer routing approach that utilizes multiple types of context information to forward packets. It combines opportunistic routing with duty cycling of wireless sensor nodes to improve packet transmission in WSNs. SCAD includes an adaptive duty-cycle mechanism to control sleeping intervals of sensors to achieve a balance between energy efficiency and performance.

7.3.1 Packet Forwarding Scheme in SCAD

We assume that each node is aware of its location and that of the destination. This can be done via GPS or other location technologies. SCAD does not use any beacons to maintain network topology. Instead, the packet forwarding decision is made in a hop-by-hop fashion. In SCAD, packet transmission is triggered by broadcasting of data packets. Whenever a node has data to send, it adds the locations of itself and the destination into the packet header, broadcasts it, and waits for the responses from its neighbors. The neighbors that successfully receive this packet first check their relative closeness to the destination by comparing their distances to the destination with that of the packet sender. If they do not provide any distance improvements, they just drop the packet. Otherwise, they follow the same procedures as in TLG (Chapter 5) and CAOR (Chapter 6), which start a DFD timer (section 5.3.1) and wait for the expiration of the timer. When the timer expires, the node will forward the packet (by broadcast) and repeat the same process until the packet reaches the destination. During the count-down of the timer, a neighbor might overhear the forwarding of the same packet from another node. This indicates that the timer of another neighbor expired earlier, and it should cancel its timer and drop the packet.

When the original packet sender overhears this forwarded transmission, it considers this reception as a response from the packet receiver and infers that one of its neighbors has successfully received the packet. The sender might receive multiple responses from different neighbors, if these neighbors can not overhear each other. In this case, the sender will only accept the first response and ignore the rest. When the first forwarded transmission from a neighbor arrives at the original sender, the sender will acknowledge it by sending the subsequent packets by unicast to that specific neighbor for a certain time interval. During that interval, the sender keeps a unicast connection between the selected forwarder and itself to enhance performance. After the validity interval of this unicast transmission, a packet sender should trigger the same broadcast-based process to find another forwarder. The duration of the unicast phase depends on the validity of the link between the sender and the selected forwarder.

Figure 7.1 shows the packet forwarding process in SCAD. When S broadcasts a packet, its neighbors nodes 2 and 4 miss the packet due to the irregularity property of wireless transmission, even if they are within the radio range of S. Node 5 drops the packet, since it does not provides any distance progress. Nodes 1 and 3 compete for forwarding by starting their DFD timers. The node whose DFD timer expires first wins the selection and forwards the packet. The same procedure will be repeated until the packet arrives at the destination.

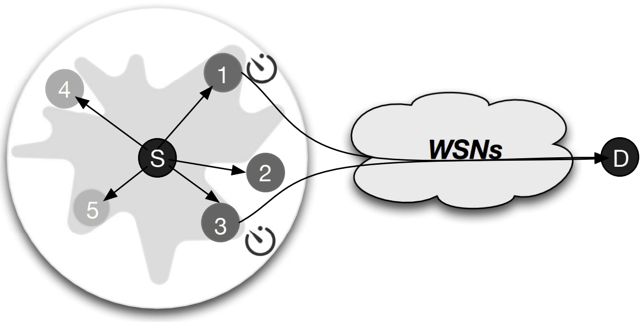


Figure 7.1: Packet forwarding scheme of SCAD

7.3.2 Dynamic Forwarding Delay (DFD)

Whenever receiving a broadcast packet, neighbors that provide distance improvements will start a timer before forwarding this packet. The goal of this timer is to select a forwarding node while avoiding collisions caused by the concurrent transmissions from multiple neighbors. In SCAD, we use the same concept of *Dynamic Forwarding Delay (DFD)* as in TLG (section 5.3.1), by integrating multiple cross-layer context information, such as *geographic progress*, *link quality*, *residual energy* and *energy drain rate*. The DFD is calculated according to (7.1):

$$\begin{aligned} \text{DFD} = (\alpha \times \text{Link Quality} + \beta \times \text{Progress} + \\ \gamma \times \text{Energy Objective Function}) \times DFD_{Max} \end{aligned} \tag{7.1}$$

Link Quality, Progress, and Energy Objective Function are the utility functions relevant to link quality, progress, and energy. α, β, and γ are the weights of each context and $\alpha + \beta + \gamma = 1$. Depending on the application requirements, SCAD assigns different weights for different metrics. DFD_{Max} is the predefined maximum delay at each node. The calculation of link quality and progress are similar to that in our previous protocols of TLG and CAOR (section5.3.1). Therefore, we skip the descriptions of link quality and progress definition, and only introduce the new definition of energy objective function.

Energy Objective Function

Energy is an essential issue in wireless sensor networks, since sensor nodes are usually battery-powered with limited capacity. Therefore, instead of achieving high throughput, WSN applications usually have a higher preference of energy efficiency and demand reliable transmissions, instead of high throughput performance. To this end, real-time energy information should be utilized in routing protocols. However, if the residual energy (RE) is the only energy-relevant metric, it may happen that a node with the biggest residual energy accepts all the packet forwarding requests and absorbs all the packets around, and its energy will deplete quickly. To mitigate this problem, the energy drain rate (EDR) should also be considered as another energy-relevant metric. EDR reflects the speed at which the battery is running out of energy. Each node observes its EDR by averaging the amount of energy consumed over a given time interval (called *sliding window*). We use EDR_i to indicate how much energy is consumed by node i within the last interval on average. RE_i refers to the residual energy of node i. To get precise values, two vital measurements have to be performed at each node: the ongoing traffic rate and the energy consumption rate. Given these two values, we define a ratio in Eq. (7.2), called *Energy Objective Function (EOF)*, to represent the global impact of energy on packet routing.

$$EOF_i = \frac{EDR_i}{RE_i} \tag{7.2}$$

We can observe that, the inverse of EOF_i ($\frac{RE_i}{EDR_i}$) indicates how long the sensor node could keep alive with its residual energy, assuming the current traffic load. Of course, a node with high value of residual energy or a low value of energy drain rate will generate a small value of EOF_i, which produces small input to Eq.(7.1) and is then preferred as a good forwarding candidate. Every node calculates its EDR by utilizing the well-known Exponential Weighted Moving Average (EWMA) method [4] according to Eq.(7.3), where edr_k denotes the real energy drain rate observed at the end of the k-th interval, EDR_{k-1} is the calculated energy rate of interval k-1, and μ is the weighting factor.

$$EDR_k = \mu \times EDR_{k-1} + (1 - \mu) \times edr_k \tag{7.3}$$

7.3.3 Adaptive Duty Cycling

The sleeping intervals of sensor nodes should be controlled individually according to real-time network conditions. An identical sleep interval often leads to heterogeneous energy consumption such that nodes located in heavy traffic regions are prone to suffer from frequent data transmissions, which will lead to fast energy depletion and short network lifetime.

Merely increasing the sleep interval makes the sensor nodes sleep longer and save energy, however, it will degrade performance. To achieve a balance between energy efficiency and performance, SCAD adapts the duty cycles of sensors according to both estimated energy drain rates and monitored traffic load variances. SCAD integrates the duty-cycling features of MaxMAC [73] and IDEA [40]. MaxMAC aims to maximally adapt to changes in network traffic loads at run-time. It improves the system throughput by assigning additional wake-ups when the rate of incoming packets reaches certain threshold values. However, MaxMAC does not consider the energy drain rate, such that after a highly intensive wake-up period, nodes do not rest longer to compensate their fast energy depletion during the past intensive wake-up periods. On the other hand, IDEA increases the sleep interval of a sensor node if its energy drain rate within a certain interval becomes high.

In SCAD, nodes start from a state with a default wake-up interval of *Base Sleep Interval (T_B)*. Each node persistently estimates the incoming traffic rate within a certain time period, called *sliding window*. When nodes observe increasing traffic load, they will reduce the sleep interval by assigning additional wake-ups to be more actively involved in the packet transmission such that network throughput can be improved. In addition to estimating the traffic rates, SCAD nodes also measure the energy consumption within the *sliding window* to estimate the energy drain rate (EDR). When an intensive traffic load passes away (the time interval that nodes promise to stay in the new state, referred to as *LEASE*, expires), nodes check their EDR values (Eq.7.3). If the calculated energy drain rate is above a certain threshold, nodes deduce that they have been intensively active for a long period, and they will increase the sleep intervals to rest longer and compensate their previous vast energy consumption.

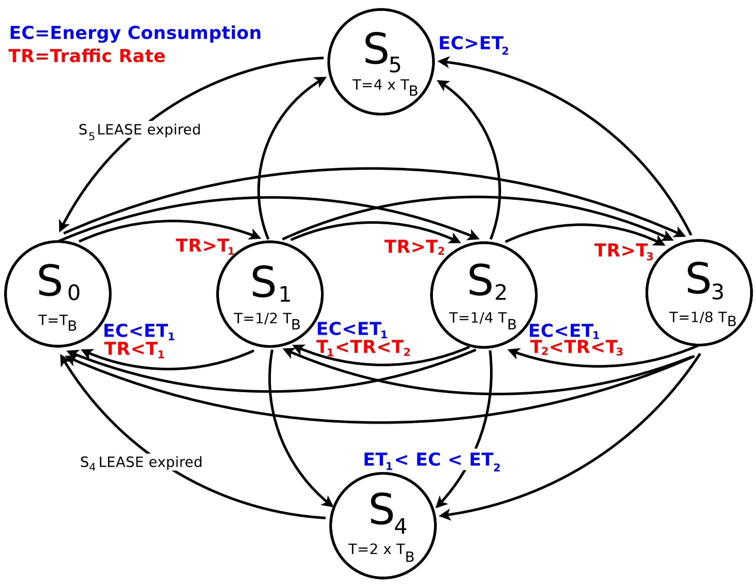


Figure 7.2: State diagram of the duty-cycling adaptation scheme

Figure 7.2 illustrates the state diagram of the duty-cycling adaptation scheme of SCAD. The state transition is triggered by either exceeding a traffic rate threshold or an energy consumption rate threshold. Nodes switch among different states when the estimated traffic rate reaches the predefined threshold values T_1, T_2, and T_3. When switching to higher states (states with higher state numbers), SCAD schedules extra wake-ups and applies a wake-up interval of $\frac{1}{2}T_B$ (at state S_1), $\frac{1}{4}T_B$ (at state S_2), or $\frac{1}{8}T_B$ (at state S_3). SCAD conserves energy at the cost of some performance, if there is a conflict between energy-efficiency and performance. This is illustrated by the introduction of states S_4 and S_5. For example, at state S_2, when the LEASE of S_2 expires, nodes may transit to states S_1, S_3, S_4 or S_5, depending on the measured values of energy consumption (EC) and traffic rate (TR) within the last sliding window. If EC is above threshold ET_2, it means too much energy has been depleted in this sliding window and the node should sleep longer to compensate its energy consumption. Therefore, the node transits to state S_5, where the wake-up interval is enlarged to four times of T_B. If energy consumption is between threshold ET_1 and ET_2 ($ET_1 < EC < ET_2$), then the node transits to state S_4. As an example, the possible state transition from state S_2 is shown in Algorithm 1.

Algorithm 1 Algorithm to decide the state transition from S_2

```
Let EC = energy consumption, ET = energy threshold, and TR = traffic rate
if EC > ET_2 then S_2 → S_5
end if
if ET_1 < EC ⩽ ET_2 then S_2 → S_4
end if
if EC ⩽ ET_1 then
    if T_1 < TR ⩽ T_2 then S_2 → S_1
    end if
    if T_2 < TR ⩽ T_CSMA then S_2 → S_3
    end if
end if
```

As indicated in the past paragraphs, state transitions in SCAD are triggered by the monitored values of traffic rate and energy draining rate during a sliding window. Once certain threshold values have been reached, the node switches from one state to another. Therefore, the choice of these threshold values affects the performance of the protocol. In fact, the definition of the thresholds ranges also determines how many states the protocol has. In the current implementation, we use three traffic rate threshold values T_1, T_2, and T_3 to divide the state-diagram into four state S_0, S_1, S_2, and S_3. Then depending on the energy drain rate threshold values EC_1 and EC_2, two additional states S_4, and S_5 are included for saving energy by increasing the sleep intervals. A small value of the traffic rate threshold will make the protocol switch frequently from lower states to higher states, which is more suitable for applications focusing on high throughput. A large traffic rate threshold value makes the protocol less sensitive to traffic load changes, and therefore more suitable for delay-tolerant applications, where energy-efficiency is of higher priority. The LEASE parameter, which defines how long the node will stay in a new state, also affects protocol performance. A large value means the node will stay longer at a new state independent of the monitored traffic load, which reduces the adaptivity of the protocol. In real application scenarios, the choices of these parameters should be considered thoroughly according to the specific requirements, in order to improve protocol performance.

7.3.4 Duty Cycling *vs.* Opportunistic Routing

As stated before, opportunistic routing benefits from overhearing of broadcast transmissions such that unexpected packet receptions can be opportunistically exploited to improve system performance. However, in wireless sensor networks, sensor nodes are configured to periodically switch off their radio transceivers to save energy. Duty cycling reduces the number of nodes that can overhear the broadcast transmission, therefore limits the benefits of opportunistic routing. In fact, two factors affect the probability of multiple reception of a broadcast transmission by multiple forwarding candidates: node density and wake-up rates of nodes. Both a high node density and a short wake-up interval (high wake-up rate) will increase the probability of a packet being received by multiple candidates.

To quantify the probability of multiple receptions (or the probability of a broadcast transmission be overheard by multiple qualified candidates simultaneously), we assume N to be the number of qualified candidates that wake up exactly once during a time period T, and that node i ($i \subseteq N$) wakes up within T and remains active for a period p, as shown in Fig. 7.3a. The probability that a broadcast transmission can be received by multiple nodes (a subset of N) can then be calculated by Eq.(7.4).

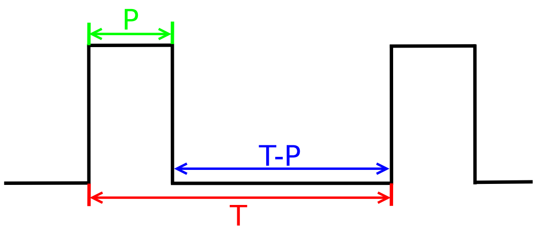
Suppose that node i receives the packet, then the situation that all the other (N-1) nodes do not receive the same packet happens only when all the other (N-1) nodes are at their idle states, which has the probability value of $(1 - \frac{p}{T})^{N-1}$. Fig. 7.3b shows the probability of a packet being received by multiple forwarding candidates as functions of node densities and wakeup intervals. As we can see, the probability of multiple receptions decreases with an increasing wakeup interval, and increases with a larger node density. Therefore, in order to increase the probability of multiple reception, a short wakeup interval or a large node density is preferred.

$$P_{multiplereception} = 1 - (1 - \frac{p}{T})^{N-1} \tag{7.4}$$

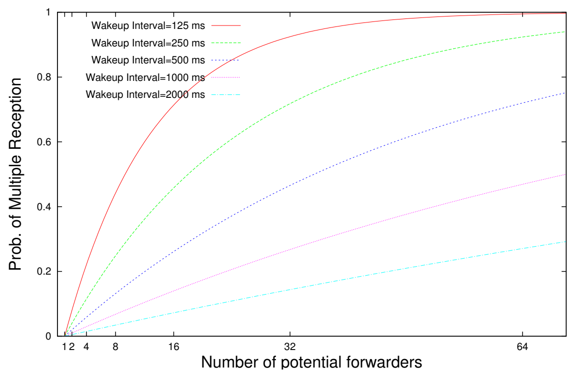
7.4 Run-time Energy Profiling of Tmote Sky Mote with TinyOS

In a network simulator, the energy consumption of sensor nodes can be calculated easily via a certain energy model. However, in real-world implementation, one big challenge of an energy-aware routing protocol is to accurately measure sensors' energy level at run-time. Hardware platforms such as Tmote Sky mote or standard WSN operating systems such as TinyOS, do not provide any energy measurement functionality. Hardware-based energy measurement mechanisms are typically difficult and expensive to add to existing hardware since they require significant modifications. Therefore, energy consumption estimation can only be done using software in a real-world deployment.

Energy consumption of a sensor node is due to the operations of its on-board hardware components. Tmote Sky sensor nodes include several components: the MSP 430 micro controller unit (MCU), the CC2420 wireless radio transceiver, the LEDs, the EEPROM flash memory, and a number of sensors. Each component has different working states/modes, and every component switches from one state to another under certain situations. Different states consume different amounts of



(a) Duty cycle



(b) Probability of multiple receptions

Figure 7.3: Result of duty-cycle on opportunistic routing: (a) Example of duty cycle interval; (b) Probability of multiple receivers versus the node density and wake-up intervals

energy. Software-based run-time energy profiling records state transitions experienced by hardware components at run-time. Specifically, the energy consumed by the sensor mote is calculated by tracking the time spent in different operating modes by different components. The energy consumption of these components is then calculated by using the current draw data in each power state, which are provided from the data sheets or can be obtained by separate calibration measurements. The energy consumption ($E_{consumed}$) of a sensor mote is then estimated using a linear energy model at run-time. Real devices can have a non-linear power dissipation behavior dependent on time and supply voltages. However, using a power model based on non-linear functions is computation expensive. In this work, we measure the real-time residual energy of sensor motes using a linear energy model at runtime. Such a model has been proven to be sound with battery-powered devices as long as the voltage supply remains within the required range. With the consumed energy calculated, the residual energy ($E_{residual}$) can be computed by: $E_{residual}$ = $E_{initial}$ - $E_{consumed}$.

Our energy estimation scheme is triggered every time by a state transition of a hardware component. When a component transits from state i to state j, the energy profiling mechanism records the time stamp when the transition happens. As the component transits again from state j to another state, a time difference is produced and logged to the total time that the component has been in state j. The energy profiling scheme keeps recording all the state transition information of each component, and records the time duration of the state at which the component is in. The energy consumption is then calculated by summing up the energy cost in each state by different components.

In this work, we implement on-line energy profiling of the Tmote Sky node in TinyOS by logging the time spent by the main energy-consuming components in different power states. The energy consumption of sensor nodes are caused by their on-board components, such as micro-controller for task processing, on-board sensors for object sensing, external flash memory for data reading/writing, and wireless radio transceiver for packet transmission and reception. Due to the fact that radio transceiver dominates the energy consumption of a sensor node, and also because our protocol SCAD does not include high computation overhead, we ignore the energy consumption from MCU and memory access, and consider the energy consumption of the wireless radio transceiver. Therefore, the energy consumption estimation is calculated as in Eq.(7.5). The current draw data of different operating states (receive, sleep, and transmit with different transmission power levels) of CC2420 radio transceiver is listed in Table 7.1, which is derived from CC2420 data-sheet [131].

$$E_{consumed} = V \times (I_{sleep} \times \Delta t_{sleep} + I_{receive} \times \Delta t_{receive} + I_{transmit} \times \Delta t_{transmit}) \tag{7.5}$$

where V is the supply voltage. I_{sleep}, $I_{receive}$ and $I_{transmit}$ are the current draws of the radio transceiver in sleep, receive, and transmit mode. Δt_{sleep}, $\Delta t_{receive}$, and $\Delta t_{transmit}$ are the time intervals of the radio transceiver in sleep, receive, and transmit mode.

Table 7.1: Energy model of CC2420

State	Current Draw
Receive	19.7 mA
Sleep	0.01 mA
Transmit (0 dBm)	17.4 mA
Transmit (-1 dBm)	16.5 mA
Transmit (-3 dBm)	15.2 mA
Transmit (-5 dBm)	13.9 mA
Transmit (-7 dBm)	12.5 mA
Transmit (-10 dBm)	11.2 mA
Transmit (-15 dBm)	9.9 mA
Transmit (-25 dBm)	8.5 mA

During the implementation of the run-time energy profiling mechanism, to avoid scattering the code of energy consumption estimation over the existing structure of TinyOS application codes, a dedicated TinyOS module, called EnergyAnalyzer.nc, is introduced. EnergyAnalyzer.nc module is directly connected to the drivers of different Tmote Sky components. EnergyAnalyzer is a new standalone module, to which other TinyOS components make calls to register their state transitions, if any. The device drivers of TinyOS have to be modified to capture the state transitions and record the relevant time stamps. The implementation architecture is shown in Figure 7.4.

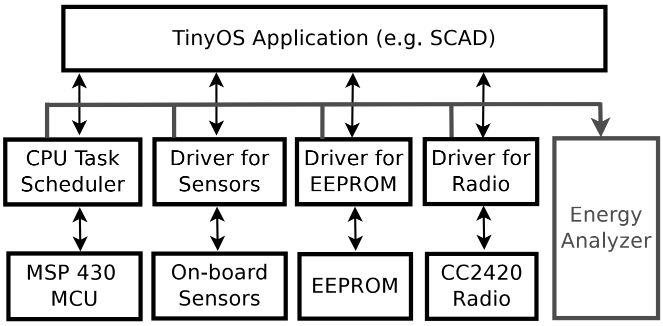


Figure 7.4: Architecture of SCAD's energy profiling mechanism in TinyOS

7.5 Performance Evaluation using Simulation

In this section, we describe the simulation conducted to validate the performance of SCAD. Different simulation results of different scenarios are presented and discussed. We used two different network scenarios: one with static topology where nodes are statically deployed; another one with mobile nodes moving according to certain mobility models.

7.5.1 Simulation Scenarios and Parameters

We take SCAD as an integrated approach of routing and MAC layers and test it as one entity. We evaluate SCAD through the OMNeT++ simulation framework [118]. The built-in module of "Resource Manager" provides real-time residual energy information. The CC2420 radio module and the Lognormal propagation model are used as physical layer implementations. We performed two types of simulation: static topology and mobile topology. We compared SCAD against the well-known geographic protocol GPSR, the beaconless protocol BLR, TLG, and the opportunistic routing protocol ExOR. Packet delivery ratio (PDR), average end-to-end delay, network lifetime, and hop-count are collected as evaluation metrics.

Both the static and mobile simulations consist used 60 nodes, which are uniformly deployed in a 300×300 m flat area. One source node is located at (5, 5), and one destination node is located at the corner with coordinates (290, 290). In the static scenario, the intermediate nodes are statically deployed in the networks. In the mobile scenario, the intermediate nodes are moving following the Random Waypoint mobility model with the default maximum speed of 5 m/s and maximum pause interval of 5 s. The transmission power of the nodes is configured to have a transmission range of around 50 m. Our simulation runs for 1000 s, and results are averaged over 30 runs with different seeds to provide a confidence interval of 95$\%$.

Since the calculation of DFD relies on multiple types of context information, their weights (α, β, γ) in Eq.(7.1) play a vital role. A large value of weight means the corresponding context is more critical in the delay calculation. Considering the essential requirement of energy-efficiency in WSNs, SCAD assigns the highest priority to the context of energy. The other two contexts, namely progress and link quality, are equally important. In the simulation, we use weight values of γ (energy) = 0.6, and α (link quality) = β (progress) = 0.2. The weighting factor (μ) and the duration of the sliding window (T) that are relevant to the EOF (energy objective function) calculation in Eq.(7.2) are 0.4 and 5 s separately. Table 7.2 shows the simulation parameters. The configuration parameters of the MAC layer of the duty-cycle adaptivity mechanism of SCAD are presented in Table 7.3.

Table 7.2: Simulation parameters

Parameter	Value	Parameter	Value
Field Size	300 $\times$ 300 m	Radio model	CC2420
Source, Destination	(5,5) (290,290)	Path loss model	Lognormal
Source number	1	UDP source rate	2 packet/s
Max. Speed	5 m/s	Max. Pause Interval	5 s
(α, β, γ) in Eq. (7.1)	(0.2, 0.2, 0.6)	DFD_{max}	0.1 s
μ, T in Eq. (7.2)	0.4, 5 s	Radio range	50 m

Table 7.3: MAC layer parameters of SCAD

Parameter	Value
Sleep Interval of S_0, S_1, S_2, S_3, S_4, and S_5	[500, 250, 125, 62.5, 1000, 2000] ms
Traffic Rate Threshold of T_1, T_2, and T_3	[2, 4, 8] packet/s
Energy Consumption Threshold of EC_1 and EC_2	[2, 4] J
LEASE interval of S_1, S_2, S_3, S_4, and S_5	1 s

7.5.2 Simulation Results of the Static Scenario

We first present the results of the static scenario. Figure 7.5 shows the results of the five protocols under the default simulation settings.

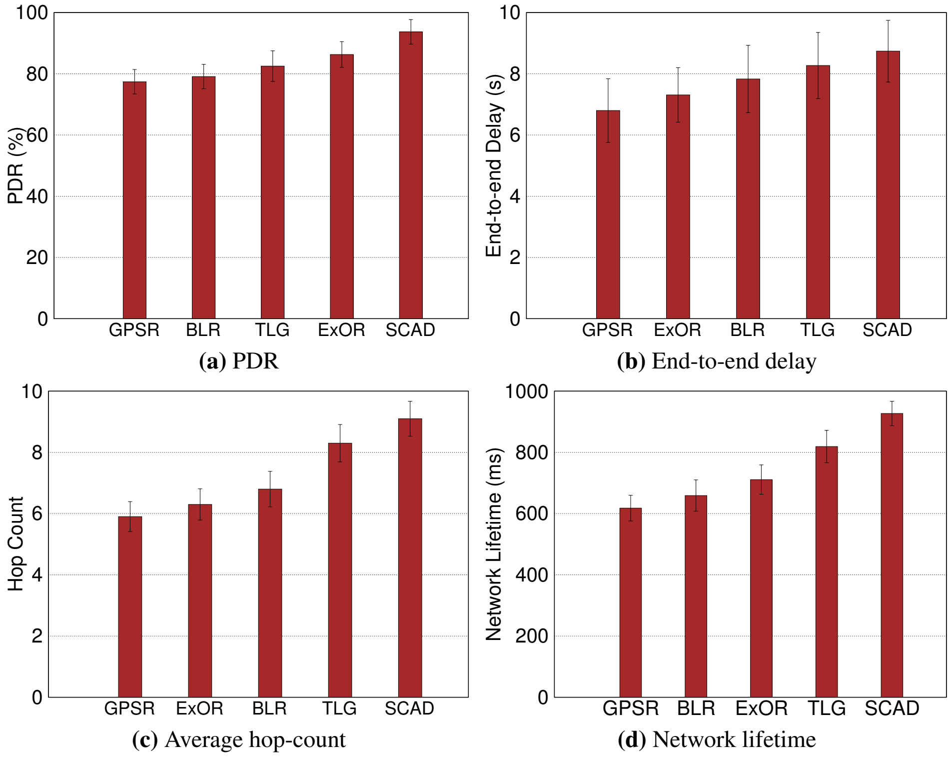


Figure 7.5: Results of (a) PDR; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the simulation evaluation in the ***static*** scenario

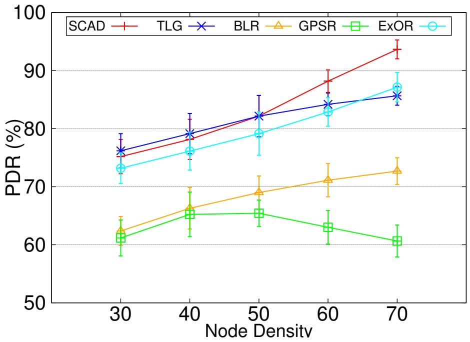
PDR values of the five protocols are presented in Fig. 7.5a. We can see that BLR performs better than GPSR, since its packet transmission does not depend on the pre-calculated end-to-end path, which is unreliable in lossy wireless environments. However, BLR solely considers geographic progress when selecting forwarders, and it always chooses the neighbor closest to the sink as its next-hop relay. That neighbor has a high chance to suffer from a weak radio link, which leads to transmission failures. TLG improves BLR by modifying the DFD calculation. It considers not only progress, but also link quality and remaining energy of relay candidates. This enables TLG to choose a forwarder with a good balance between link quality and distance improvement. ExOR has the candidate list prepared according to candidates' priorities before data transmission. In a static topology, candidates have high chances to keep their rankings when packets arrive. Therefore, nodes will wait for their turns to forward packets accordingly, which leads to good performance. SCAD focuses more on energy efficiency by introducing adaptive duty cycling. This enables SCAD nodes to keep their radios on when network traffic load is high. Therefore, SCAD has the best PDR results.

Fig. 7.5b denotes results of average end-to-end delay. GPSR forwards packets based on the route information in the routing table. However, in unstable environments, such as low-power WSNs, link breaks happen frequently. Consequently, a source has to find another route and all subsequent packets will be delayed. BLR, TLG, and SCAD are free of packet re-transmissions caused by unstable links, and packet forwarders are selected from the neighbors that successfully receive the packets. TLG and SCAD have longer delays than BLR, since they do not choose the most distant neighbors at each hop like BLR. In the end, packets have to go through more hops to reach the sink. SCAD undergoes longer delays than TLG, because it applies a duty cycling mechanism to save energy. Duty-cycling limits the number of nodes that are alive to concurrently overhear broadcast transmissions, which introduces additional delays. ExOR nodes forward packets according to their rankings in the candidate list. Therefore, ExOR has no additional delays at each hop, resulting in shorter delays than BLR, TLG, and SCAD.

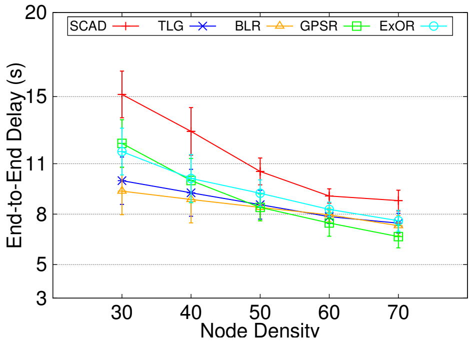
Fig. 7.5c indicates the average hop count that a packet takes to reach the sink. GPSR, ExOR, and BLR have the best results. This is because they all pick up the nodes providing the best geographic advancements as packet forwarders, which results in short paths with few hops. TLG and SCAD choose packet forwarders by considering not only distance improvement but also and link quality. Usually good link quality implies short distance progress. Therefore, they make intermittent progress at each hop, which leads to bigger hop count values.

Fig. 7.5d presents network lifetime results. We consider network lifetime to be the timespan from network deployment until the first node failure due to battery exhaustion. GPSR produces the worst result, since it keeps using the same nodes for packet transmissions after they are selected, which leads to fast energy depletion. BLR has no control overhead. Therefore, nodes live longer than in case of GPSR. ExOR does not consider energy, therefore, it performs worse than TLG. However, since BLR nodes forward packets strictly following the priority list, duplicate transmissions are removed. Therefore, it has less energy waste than BLR, which results in a better network lifetime. TLG considers the remaining energy, and it distributes energy consumption to different nodes according to their residual energy. However, if only the residual energy is considered, a node with sufficient remaining energy will accept packet forwarding requests from multiple neighbors and be involved in multiple packet transmissions. In this way, its energy will deplete quickly. Thanks to considering both residual energy and energy drain rate, SCAD can avoid over-dissipation of specific nodes by taking into account the current traffic load and by utilizing the energy drain rate. This enables SCAD to evenly distribute the energy expenditure to different nodes. Therefore, SCAD has a longer network lifetime than TLG. It is worth noting that BLR consumes much more energy and thus has shorter network lifetime than TLG and SCAD. This is because BLR prefers neighbors closest to the destination as forwarders. The longer hop distance consumes much more energy, since energy cost on data transmission is proportional to the transmission distance between sender and receiver. Thanks to the adaptive duty-cycle, SCAD can provide high performance when the traffic load is high, and it also makes the nodes sleep longer to compensate its fast energy dissipation to prolong the network lifetime.

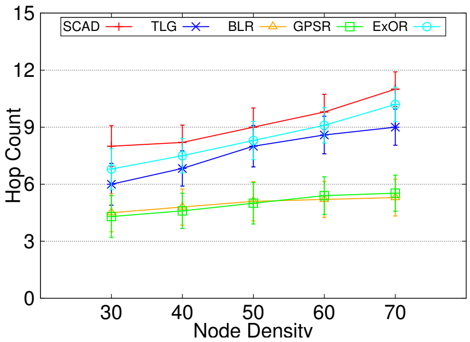
Figure 7.6 presents the results of the five protocols as a function of node density, to show the effect of node density on protocol performance. We vary the node number from 30 to 70, in steps of 10.



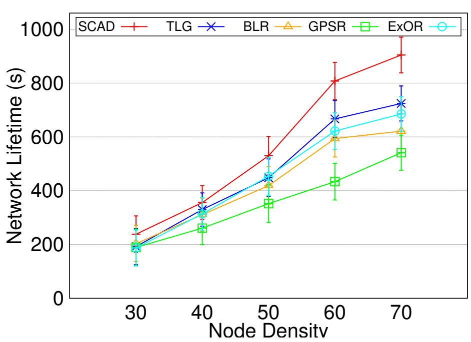
(a) PDR



(b) End-to-end Delay



(c) Average hop-count



(d) Network lifetime

Figure 7.6: The effect of ***node density*** on (a) PDR; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***static*** scenario in the ***simulation*** evaluation

Fig. 7.6a shows the effect of node density on PDR. As we can observe, SCAD performs poorly when node density is low. This is because its duty-cycling scheme reduces the number of alive nodes that can overhear a broadcast transmission. Therefore, the benefits of OR can not be fully exploited. When more nodes are deployed, SCAD performance catches up. This is because when more forwarding candidates are available, the probability that a broadcast transmission being received by multiple nodes is increased. This result also confirms our statement in section 7.3.4. BLR and GPSR both perform worse than TLG and GPSR, since their forwarder selection rules do not consider link quality. GPSR performance improves first with increasing number of nodes, and then becomes worse. This might be due to the severe congestion caused by additional control overhead, which results in performance degradation at high node densities. Since BLR has no control overhead, its performance keeps increasing when more nodes are deployed in the network. ExOR performs also quite well, and since the network is static, the priority list remains valid for longer periods. Therefore, candidates forward packets following their orderings in the candidate list, which leads to good performance.

Fig. 7.6b presents the end-to-end delay results. As expected, when few nodes are deployed, all five protocols perform badly. Low node numbers result in long distances between node pairs, which leads to low link quality and further performance degradation. When more nodes are available, the delays improve.

Fig. 7.6c indicates the average hop count results. We can observe that GPSR and BLR have the best results, since at each hop, they always take neighbors with the largest geographic progress as forwarders, which leads to the smallest hop count. TLG and SCAD consider link quality at each hop, and thus, they prefer nodes located close to the packet sender as forwarders. This makes packets go through more hops to reach the destination.

Fig. 7.6d depicts the results of network lifetime with different node densities. GPSR has the shortest network lifetime, since its route information does not consider energy. The usage of control messages costs additional energy. Even if BLR has the shortest hop count, it generates a bad network lifetime, since it does not consider energy. Frequent packet re-transmissions, which are caused by poor links, waste energy. But since BLR has no control overhead, it produces better network lifetime results than GPSR. TLG and SCAD use the latest energy information, and thus, they can distribute packet forwarding tasks to different nodes, such that the energy consumptions are distributed over multiple nodes. Therefore, they have better network lifetime results than GPSR and BLR. SCAD performs better than TLG, because of its additional consideration of energy consumption rate. With this capability, SCAD nodes achieve a good balance between performance and energy efficiency. The performance of SCAD is maximized especially when node density is high.

7.5.3 Simulation Results of the Mobile Scenario

We next evaluate SCAD in a mobile environment, and here we present the results of the five protocols when nodes are mobile. In this experiment, the destination node is fixed at the corner of the field, while source nodes and intermediate nodes are mobile and their mobility follows the Random Waypoint mobility model [36].

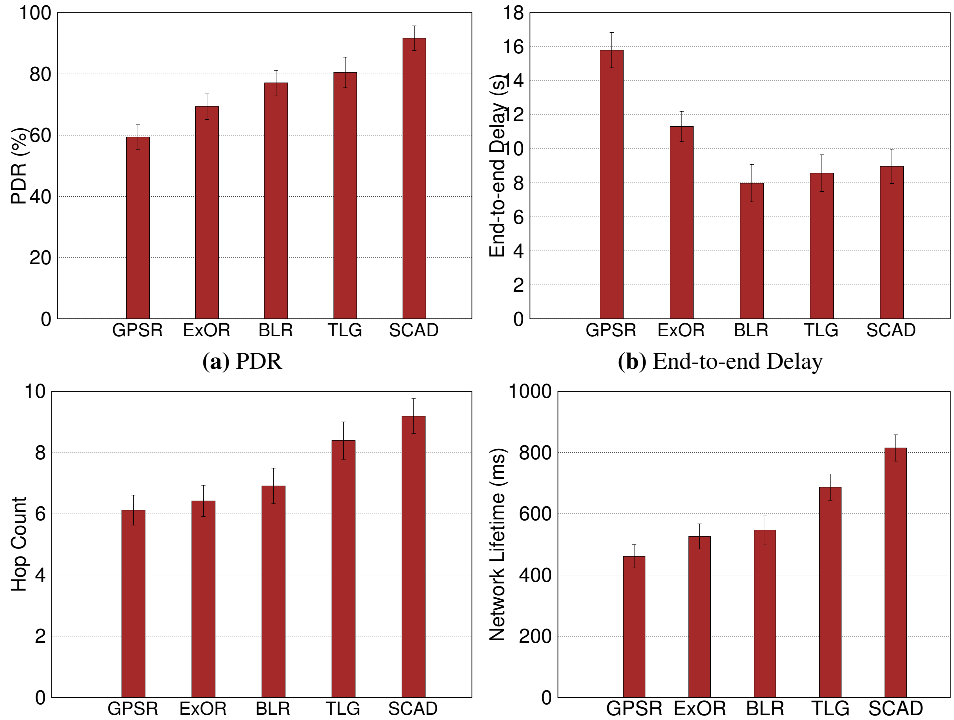
Figure 7.7 shows protocol performance under the default mobility setting with the maximum moving speed of 5 m/s and maximum pause time of 5 s).

Fig. 7.7a presents the PDR results. Compared to the static case, GPSR and ExOR performance degrade significantly: PDR of GPSR drops from 78% to 60% and PDR of ExOR drops from 85 % to 70 %. However, performance of BLR, TLG, and SCAD remains almost the same as in the static scenario. This is because GPSR and ExOR rely on network topology, which changes when nodes move. GPSR forwards packets based on routing tables, and ExOR selects packet forwarders according to a pre-generated candidate list. Both routing tables and candidate lists are built on top of the knowledge of network topology. Once the network topology changes, this information will be invalid, which degrades performance of packet transmissions. BLR, TLG, and SCAD are stateless protocols, and their packet transmissions are independent of any topology information. Therefore, their performance is immune to mobility.

Fig. 7.7b shows the delay results. Compared to the static topology, GPSR and ExOR change from the best two protocols to the worst two protocols. This is due to the reason described above. Outdated topology degrades performance to packet transmissions, which results in longer delays for these two protocols. BLR, TLG, and SCAD are stateless protocols, and their performance keeps similar values as in the static case.

Fig. 7.7c indicates the average hop count that a packet travels to reach the destination. Similar to the static case, GPSR, ExOR, and BLR have the best performance, since they always prefer nodes providing the largest geographic advancements as forwarders. Therefore, a packet needs fewer hops to reach the destination than TLG and SCAD. TLG and SCAD take link quality as a selection criterion, and they choose nodes that satisfy both requirements of link quality and geographic progress as forwarders. Therefore, packets have to go through more hops.

Fig. 7.7d presents the network lifetime results. Compared to the static scenario, all protocols have reduced network lifetime. This is because node movements cost a significant amount of energy. The performance ordering of the five protocols are the same as before. GPSR produces the shortest network lifetime, because it keeps using the nodes in the routing table for packet transmissions, which leads to fast energy depletion. Thanks to considering residual energy and energy drain rate, SCAD evenly distributes energy expenditure to different nodes. Therefore, it has the best network lifetime.



(a) PDR

(b) End-to-end Delay

(c) Average hop-count

(d) Network lifetime

Figure 7.7: Simulation results of (a) PDR; (b) End-to-end delay; (c) Hop-count; (d) Network lifetime in the ***simulation*** *evaluation in* ***mobile*** *scenario*

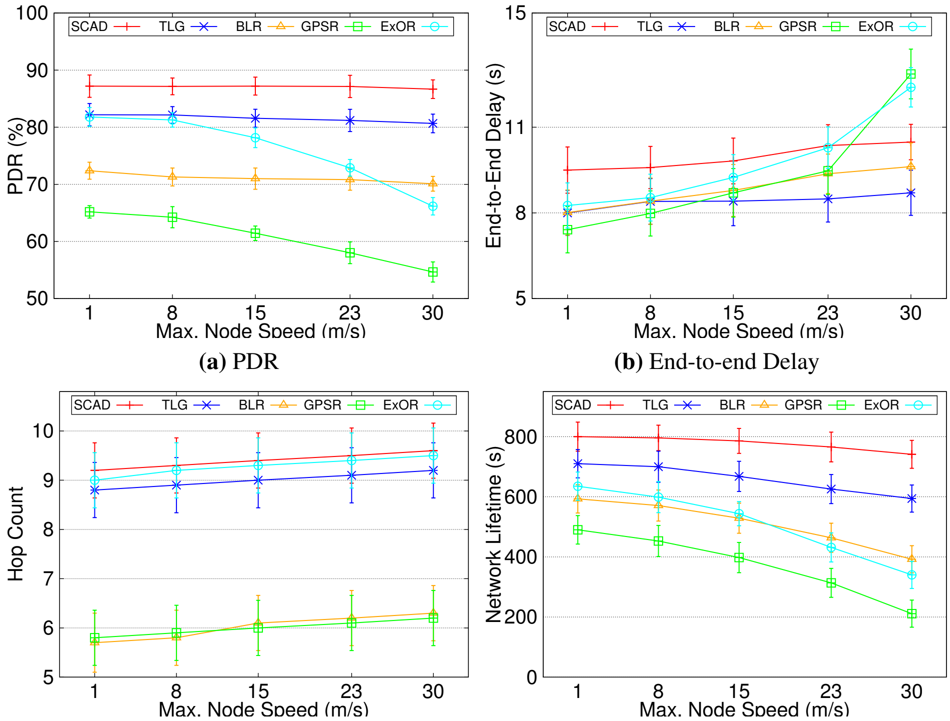
Figure 7.8 presents the effect of node speed on protocol performance. We vary the maximum node speed from 1 to 8, 15, 23, and 30 m/s.

Fig. 7.8a shows PDR of five protocols under different speeds. An obvious observation is that performance of GPSR and ExOR degrade significantly when node speed increases. GPSR transmits packets according to routing tables, which will be outdated frequently when node speeds increase. Therefore, nodes have to update routing tables and retransmit packets, which lead to degraded performance. ExOR uses candidate lists, which are similar to routing tables, but include multiple forwarders ranked according to their priorities. However, these candidate lists are produced statically prior to data transmissions. Therefore, they will be invalid when nodes move and change their positions. The faster nodes move, the higher is the percentage of invalid list entries. Therefore, ExOR performance degrades significantly at high node speeds. On the other hand, performance of BLR, TLG, and SCAD are not sensitive to speed changes. This is because their forwarder selections are made after data transmissions and the decisions are based on instantaneous link quality. They do not rely on any prepared routing table or priority list, and thus are immune to topology changes.

Fig. 7.8b denotes the average end-to-end delay results. GPSR and ExOR have the best results at low speeds. However when speeds increase, they will suffer from the problems described above, which leads to longer delays. As mentioned before, BLR, TLG, and SCAD are stateless and are immune to topology changes. Therefore their delays remain almost constant when node speeds change. Among these three protocols, SCAD has the worst result, since its duty-cycle operation reduces the number of active neighbors that can concurrently overhear broadcast transmissions, which leads to long delays.

Fig. 7.8c indicates the average hop count results. We can observe that GPSR and BLR always have the best results, since at each hop, they always take the neighbors with largest progress as forwarders, which results in the shortest hop count.

Fig. 7.8d plots the results of network lifetime for different speeds. As we can see, when speeds increase, all protocols suffer degraded performance. This is because the faster nodes move, the more energy will be consumed. However, an interesting observation is that different protocols have different performance degradation rates. GPSR and ExOR have the biggest performance degradations, which is due to their dependence on network topology. ExOR is overtaken by BLR when node speed is above 15 m/s. For high speeds the candidate list of ExOR will be invalid and duplicate transmissions will be frequent. This leads to energy waste and results in reduced network lifetime.



(a) PDR **(b)** End-to-end Delay

(c) Average hop-count **(d)** Network lifetime

Figure 7.8: The effect of ***node speed*** on (a) PDR; (b) End-to-end Delay; (c) Hop-count; (d) Network lifetime in ***mobile*** scenario in the ***simulation*** evaluation

7.6 Performance Evaluation using Real-World Experiments

To validate the performance of our proposal in real-world environments, we implemented our protocol in the TinyOS sensor operating system and run different experiments on both small-scale and large-scale testbeds with static and dynamic topologies. In the small-scale experiments, we used a linear topology and a merging topology of node deployment over a desktop (shown in Figure 7.9). In the large-scale testbed experiments, we have four topologies: a linear topology, a merging topology, an aggregation topology (shown in Figure 7.13), and a mesh topology (shown in Figure 7.18). In the mesh topology of the large-scale experiments, we further divided our experiments into two scenarios. One scenario is with a static topology, which means nodes are statically located at fixed positions and the network connectivity does not significantly change over time. Another scenario is with dynamic topology, where the connectivity among nodes is constantly changing over time. Due to the lack of mobile sensor nodes in the indoor testbed, we implemented the dynamic topology via the run-time adjustment of sensor's transmission power, such that the connectivity among nodes changes over time.

We choose the geographic routing protocol GPSR, the beaconless protocol BLR, and our work of TLG [158] as the baseline protocols for performance comparison. Results of throughput, end-to-end delay, network lifetime, and hop-count are collected as performance metrics. We use the cumulative distribution function (CDF) of the four performance metrics to present the average protocol performance over multiple experiment runs. Take throughput as an example: given a certain throughput value x, the corresponding CDF value represents the percentage of the experiment runs finished with a throughput larger than that value x.

7.6.1 Technical Challenges in Real-World Implementation

We run experiments on two different scenarios: one with sensor nodes deployed on a desktop table, and another one with sensor node deployed in an indoor testbed distributed in four floors of a building. Most beaconless routing protocols in the literature have been designed assuming an almost perfect wireless channel propagation model. In many cases, their designs have neglected the presences of interferences, collisions, packet loss, and all of the typical issues that are present in every real deployment of a wireless network. This causes problems such as selecting neighbors with weak or unreliable links, additional overheads due to retransmissions, and so on. Thus, in a real testbed, the performance of these algorithms can be unsatisfactory.

Due to the fact that wireless channel are unstable and their radio quality varies in nature, we place nodes closer to each other in both the small-scale experiment in section 7.6.2 and the large-scale experiment in section 7.6.4 such that the distance between two neighboring nodes is smaller than the transmission range of a sensor node. In this way, a node will have multiple potential forwarders within its radio range with different channel qualities.

7.6.2 Small-scale Desktop Experiment

We evaluate the SCAD prototype in two different topologies on a small scale indoor testbed deployed on a desktop consisting of few Tmote Sky motes.

For existing geographical routing protocols, we assume that nodes know their position and that of the destination by means of a positioning system. In our implementation, each node is preloaded with the necessary location information under different evaluation topologies. This is done such that there is no additional needs for location service during the experiments, and we focus on performance comparison of different routing approaches.

Since the calculation of DFD relies on multiple types of context information, their weights (α, β, γ) in Eq.(7.1) play a vital role. A large weight value means the corresponding context is more critical in the delay calculation. Considering the essential requirement of energy-efficiency in WSNs, SCAD assigns the highest priority to the context of energy. The other two context information, namely progress and link quality, are of the same importance. In the experiment, we use weight values of γ (energy) = 0.6, and α (link quality) = β (progress) = 0.2. DFD_{max} is set to 0.1 s. The weighting factor (μ) and the duration of the sliding window (T) that are relevant to the EOF calculation in Eq.(7.2) are 0.4 and 5 s separately.

In the experiments, the source node generates constant bit rate (CBR) packets towards the destination with a rate of 1 packet per second (1 packet/s). We choose this low data rate to remove any possible network interference, congestion, or collision, such that the effect of unreliable links can be fully isolated from inter-packet interferences or buffer overflows. During the experiments, our run-time energy profiling mechanism is working to provide nodes with run-time residual energy information. We set the transmission power of -25 dBm, which represents a transmission range of around 2.5 m. We repeat the experiments independently for 10 times, and the presented results are the average of these runs.

7.6.3 Small-scale Desktop Experiment Evaluation Results

In this subsection we present the real-world experiment results of the four protocols in the small-scale desktop topology, where sensors are deployed on a tabletop.

Linear Topology

The first experiment was conducted in a simple network setting: five nodes of A, B, C, D, and E are placed in a line with an inter-distance of 1 m, as shown in Figure 7.9 (a). Node A is the source , E is the destination, and B, C, and D are the intermediate nodes. Results of the four protocols in the linear topology are shown in Figure 7.10.

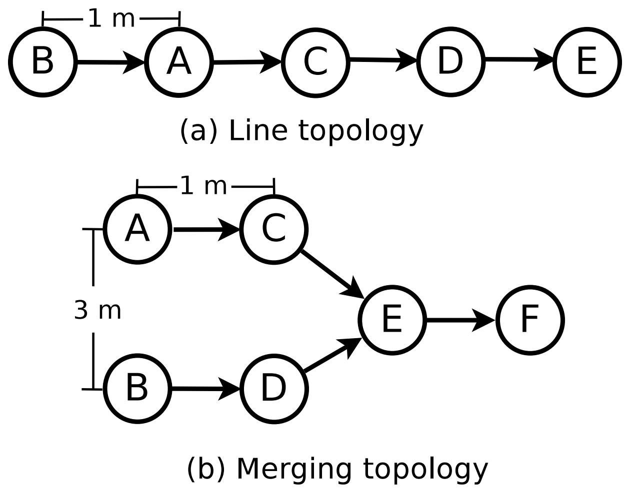
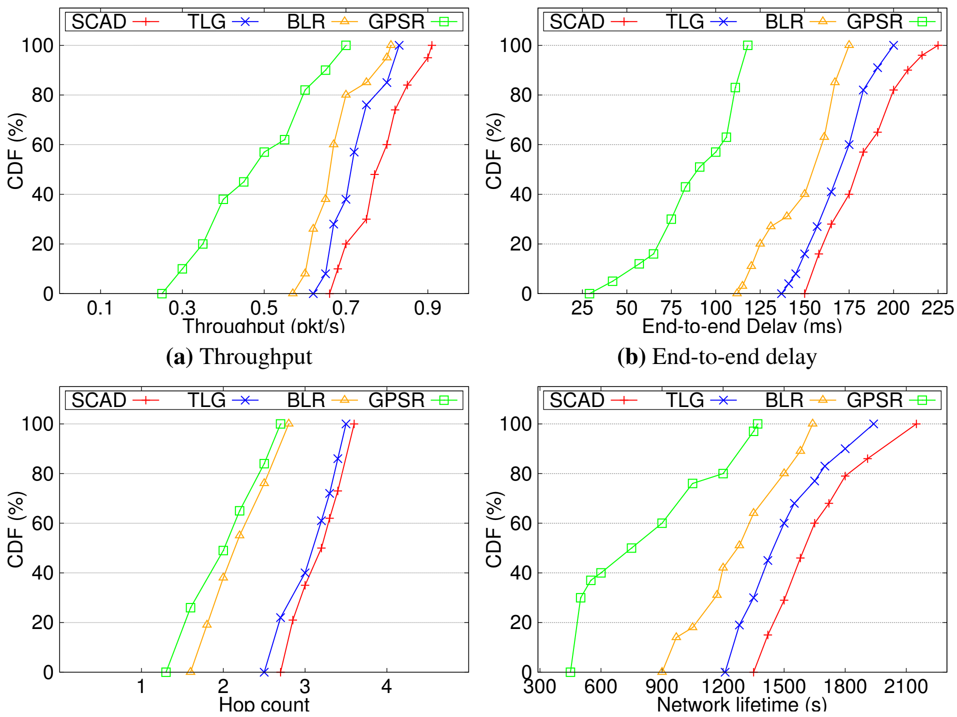


Figure 7.9: Small-scale desktop experiment topologies

Fig. 7.10a shows the CDF results of throughput. As we can see, GPSR has the worst result. Because it uses hop count as the only metric and it prefers a short path with very poor radio links rather than a long path with high-quality links. In lossy environments, where link qualities vary significantly, transmission failures might happen frequently, which leads to retransmissions. This reduces the available bandwidth and decreases network throughput. Additionally, GPSR chooses forwarders according to the routing table, however, the routing table is not able to keep updated with the variation of radio links in lossy environments. BLR, TLG, and SCAD perform far better than GPSR. As we can see from the CDF plots, 70% of the experiments under these three protocols achieve a throughput value of 0.7 packet/s. BLR is the worst one of these three since it chooses the neighbor closest to the sink as its next-hop. Therefore, it also prefers a short path with poor links. TLG improves BLR by modifying the DFD calculation. It considers not only progress, but also link quality and remaining energy of candidates when calculating DFD. This enables TLG to choose a forwarder with a good balance between link quality and distance improvement. SCAD has the best result, and more than 90 % of its experiments finished with a throughput value of more than 0.85 packet/s. This is because SCAD focuses more on energy efficiency by introducing adaptive duty cycles. This enables SCAD nodes to keep their radios on when the network traffic load is high. Therefore, SCAD has the best throughput figure.

Fig. 7.10b denotes the CDF results of the end-to-end delay. For example, the plot shows that 50% of the packets transmitted via GPSR can reach the destination within 100 ms. GPSR has the shortest delay, because GPSR does not require any extra time to select the next-hop, and the route information is prepared in the routing table at the moment of packet arrival. BLR, TLG, and SCAD are free of packet retransmissions caused by unstable links, since their packet forwarding procedures are made after packet transmissions, and the packet forwarder is selected from the



(a) Throughput

(b) End-to-end delay

(c) Average hop-count

(d) Network lifetime

Figure 7.10: CDF as function of (a) Throughput; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the ***Linear*** topology in the ***Small-scale Desktop*** experiment

neighbors that successfully receive the packets. However, they all introduce certain delays at each hop to choose forwarders. Therefore, their delay results are worse than that of GPSR, even if sometimes GPSR has to refresh the routing table. TLG and SCAD have longer delays than BLR, since they do not choose the most distant neighbors at each hop. In the end, the packet has to go through more hops to reach the sink. SCAD has a longer delay than TLG, because it applies a duty cycle mechanism to save energy. Duty-cycling limits the number of nodes that are alive to concurrently overhear the broadcast transmissions, which degrades performance. However, WSN applications usually target at energy efficiency and long network lifetime, a bit longer delay is usually tolerable.

Fig. 7.10c indicates the CDF results of the hop count that a packet takes to reach the destination. As expected, GPSR and BLR have the lowest number because they choose forwarders by only considering the distance improvement. They always select the neighbor closest to the destination as next-hop, and thus, they need less hops to reach the destination. This observation is also confirmed in the figure such that 80% of the packets transmitted using GPSR and BLR reach the destination in 2.5 hops. TLG and SCAD choose next-hops by jointly considering distance improvement and link quality. Distance improvement and link quality are contrary metrics, since a node with good link quality usually provides small distance advancement. Therefore, TLG and SCAD always choose a node with good link quality while also providing a good distance improvement as the packet forwarder, which takes the packet more hops to reach the sink.

Fig. 7.10d presents the CDF results of the network lifetime with different protocols. The network lifetime is defined as the time duration from network initialization until the moment when the first battery depletion happens. As we can see from the figure, GPSR has the worst result, and 50% of its experiments have a network lifetime of only 750 s. This is because GPSR keeps using the same nodes that are recorded in the routing table for packet transmissions. The heavy overloads undertaken by those nodes make their energy consumed very fast, which becomes the bottlenecks of the network lifetime. BLR has a better result, 80% of its experiment runs have a network lifetime of below 1500 s. This is because BLR has no control overhead, and thus, nodes live longer than for GPSR. TLG has 80% of its experiments finished with a lifetime of 1700 s. This is due to its consideration of remaining energy, and it tries to uniformly distribute energy consumption to different nodes according to their residual energy. In the linear topology, SCAD performs similarly to TLG, and has 80% of the experiment with a network lifetime of less than 1850 s. Thanks to the adaptive duty-cycle mechanism, SCAD can provide high performance when traffic load is high, and it can also make the nodes sleep longer once its energy drain rate is high to compensate its fast energy dissipation to prolong network lifetime.

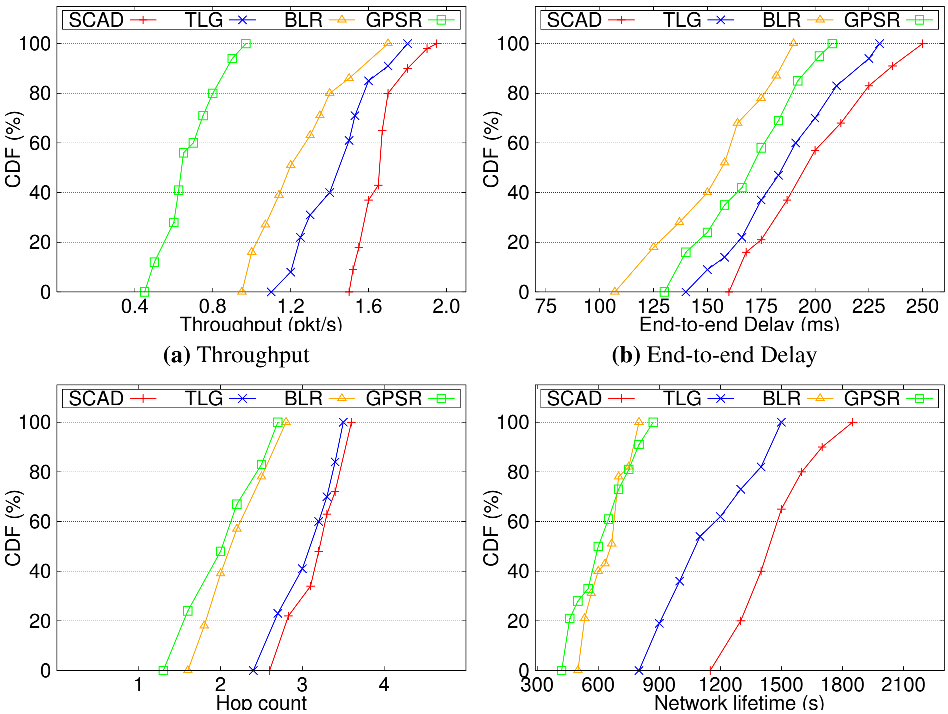
Merging Topology

In a second experiment, we used a merging topology with six nodes to examine the behavior of the four protocols with multiple source nodes. This is a common WSN application scenario, where multiple traffic flows from different source nodes go to the sink at the same time. The topology is shown in Figure 7.9 (b), in which A and B are two source nodes, F is the destination, and C, D, and E are the intermediate nodes. The distance between two neighboring nodes is also 1 m. Two data flows (A-C-E and B-D-E) are distant from each other such that the data transmission on one flow can not be overheard by nodes of another data flow. Figure 7.11 presents the performance of the four protocols in the merging scenario.

Fig. 7.11a shows the CDF results of the throughput in the merging topology. All protocols have increased throughput, since multiple source nodes generate more traffic into the network. As in the linear topology, GPSR still has the worst result. Compared to the results in the linear topology (Fig. 7.10a), GPSR performance improvement in the merging topology is limited. As shown in Figure 7.11a, it has 80% of the experiments ended with a throughput value of 0.8 packet/s. This is because with more traffic, GPSR has higher chances to have congestion. Once the routing table is outdated due to link variations, the source has to search for new routes and update its routing table. The incoming packets will be lost and the network throughput is then decreased. Throughput results of BLR, TLG, and SCAD have more obvious improvements than GPSR. As we can see from the CDF plots, 80% of the experiments under these three protocols achieve a throughput value of 1.4 packet/s. SCAD has the best result, and more than 80 % of its experiments finished with a throughput value larger than 1.7 packet/s. Thanks to the adaptive duty cycle, SCAD nodes could keep their radios on when network traffic load is high, therefore it has the best throughput figure.

Fig. 7.11b shows the CDF results of the end-to-end delay in the merging scenario. One big difference from the linear scenario is that GPSR does not have the best delay results anymore. Under high traffic load, the effect of link variation will be more serious. Outdated routing tables have to be updated frequently, which will delay packet transmission and increase delays. The delays of BLR, TLG, and SCAD are quite similar to that of the linear scenario. The delay results of SCAD reach some high values up to 250 ms compared to the linear scenario in some cases. This is because in the current merging topology, node E is the only merging node in the network. In certain situations, node E might pause the forwarding operation due to its high energy consumption rate. If it has been involved in highly intensive forwarding operation, then due to the duty-cycle mechanism, it will rest longer and thus introduce delays for the forwarding of the packets.

Fig. 7.11c shows the CDF results of the hop count that a packet undergoes to reach the destination. Similar to the linear scenario, GPSR and BLR still have the best performance because they choose the neighbor closest to the destination as the next-hop. Therefore, they need less hops to reach the destination. TLG and SCAD choose forwarders by jointly considering distance advancement and link



(a) Throughput **(b)** End-to-end Delay

(c) Average hop-count **(d)** Network lifetime

Figure 7.11: CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Merging*** topology in the ***Small-scale Desktop*** experiment

quality, and usually good link quality implies short distance. Therefore, they make intermediate progress at each hop, which results in a larger hop count.

Fig. 7.11d presents the CDF results of the network lifetime under the four protocols in the merging topology. GPSR and BLR do not consider energy, therefore their behaviors are independent of the residual energy of nodes. Due to increased traffic and possible congestion, nodes with these two protocols will deplete their energy quickly. Therefore, we can observe that GPSR has 80% of its experiments finished with a network lifetime of 700 s, and BLR has 80% of its experiments finished with a network lifetime of 800 s. TLG and SCAD consider the residual energy when choosing packet forwarders. Therefore, they try to uniformly distribute energy consumption to different nodes according to their remaining energy. TLG has 80% of its experiments finished with a network lifetime of 1350 s. However, if only the residual energy is considered, a node with sufficient residual energy will accept packet forwarding requests from multiple neighbors continuously and be involved in multiple packet transmissions for different sources. In this way, its energy will be depleted quickly. For example in the merging topology shown in Figure 7.9 (b), if nodes C and D do not consider their energy drain rates, they will participate in the forwarding tasks of both nodes A and B. Thanks to the consideration of both residual energy and energy drain rate, SCAD can avoid over-dissipation of specific nodes by taking into account the current traffic load and the energy drain rate. This enables SCAD to evenly distribute the energy expenditure to different nodes. Taking the same example as before, if node C detects that its energy drain rate is high, then it will ignore any forwarding request even if it still has enough residual energy. Therefore, SCAD produces a longer network lifetime than TLG, which is confirmed in the plot that SCAD has 80% of its experiments with a lifetime of 1700 s. Thanks to the adaptive duty-cycle mechanism, SCAD provides high performance when traffic load is intensive. It also makes the node sleep longer once its energy drain rate is high, to compensate its fast energy dissipation to prolong the network lifetime.

7.6.4 Large-scale Testbed Experiment

To evaluate SCAD in a more realistic environment, we further conduct experiments using the WISEBED testbed [127] deployed at the Institute of Computer Science and Applied Mathematics, University of Bern. We evaluate the SCAD prototype in four different large-scale scenarios consisting of a certain number of Tmote Sky sensor motes. The motes are deployed over the four floors of the building, therefore the communication among nodes suffers from various unexpected interferences or signal attenuation caused by walls, floors, objects, and people/object movements. The network topologies of the first two scenarios are similar to the ones we used in the small-scale desktop experiments, which are the linear topology with 5 nodes and the merging topology with 6 nodes. The third scenario is an aggregation scenario with seven nodes, where multiple data flows coming from different source nodes merge into one sink node simultaneously. The fourth scenario is a mesh network topology, where multiple (20) sensor nodes are distributed in the testbed and they form a mesh network. The network topologies of the first three scenarios (linear, merging, and aggregation) are shown in Figure 7.13, and the fourth scenario (mesh) is shown in Figure 7.18, separately. We divide the mesh topology experiment into two parts: a static topology and a dynamic topology, to examine the protocol performances under different situations.

Most of the protocol parameters are kept the same as in section 7.6.2. The only difference is that in these experiments, since the networks are much larger than previous ones, we configure each node to use a transmission power of -10 dBm, which produces a transmission range of around 11 m. We repeat the experiments independently for ten times, and the presented results are the average of these runs. For the experiments conducted in this section, the testbed management system of TARWIS [75] [72] is used to facilitate the experiment deployment and evaluation results collection and analysis.

As before, we choose the geographic routing protocol GPSR, the beaconless routing protocol BLR, and the opportunistic routing protocol TLG as the baseline protocols for performance comparison. Results of throughput, end-to-end delay, network lifetime, and hop-count are collected as performance metrics. To present the evaluation results of multiple experiments, we use Cumulative Distribution Function (CDF) to depict the results of throughput, delay, network lifetime, and hop-count. A cumulative distribution function $F(x)$ describes the probability that a real-valued variable X with a given probability distribution will be found at a value less than or equal to x.

Figure 7.12 shows the used WISEBED testbed topology. We first configured three sets of experiments, indicated as the green, blue, and purple nodes-links in the figure. The green network is consisted of five nodes 1, 2, 3, 4, and D, representing the linear topology. The blue network is consisted of six nodes, 1, 2, 5, 6, 7, and D, representing the merging topology. The purple network is consisted of seven nodes 2, 3, 4, 5, 6, 8, and D, representing the aggregation topology. The network topologies of three scenarios are shown in Figure 7.13. The mesh topology-based experiment is presented in Figure 7.17.

7.6.5 Large-scale Testbed Experiment Evaluation Results

In this section we present the evaluation results of the four protocols in the large-scale testbed. We use four different topologies: linear, merging, aggregation, and mesh topology.

Linear Topology

The first experiment includes five nodes of 1, 2, 3, 4, and D, in which node 1 is the source, D is the destination, and 2, 3, and 4 are the intermediate nodes. Results of the four protocols in this linear topology are shown in Figure 7.14.

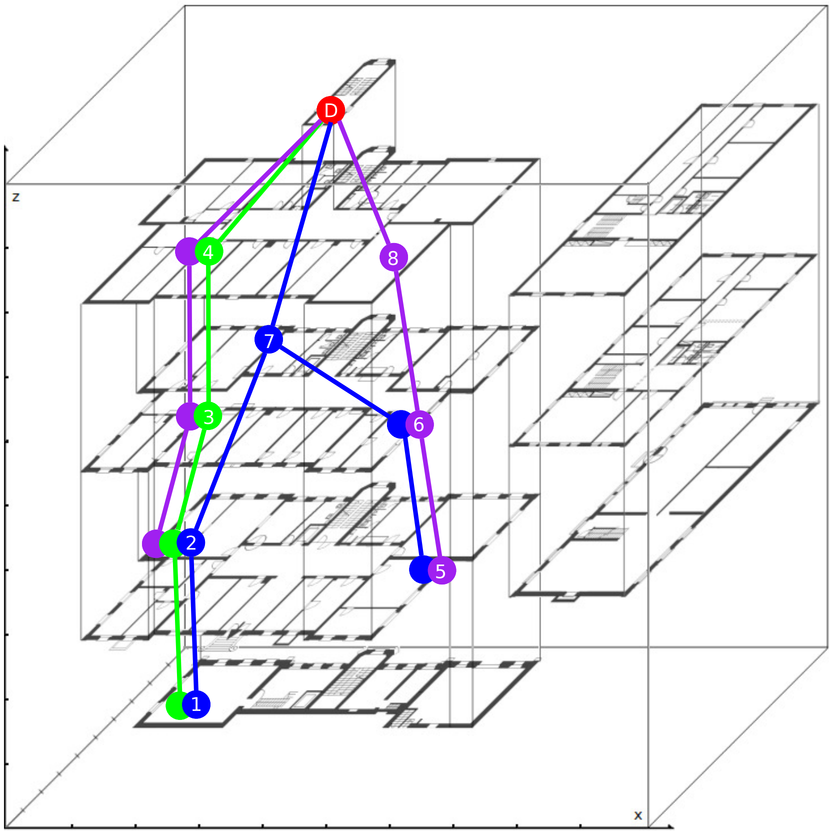


Figure 7.12: Large-scale testbed deployment: Linear, Merging, and Aggregation Topologies

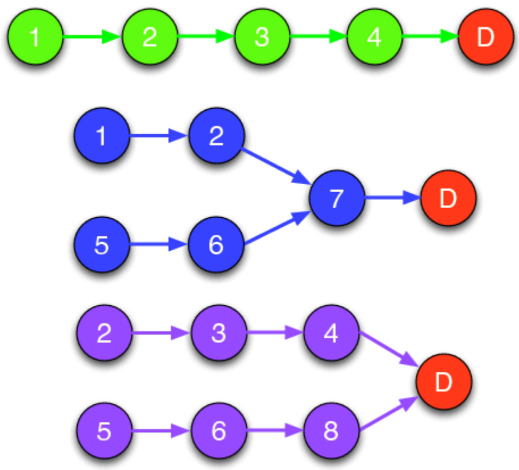
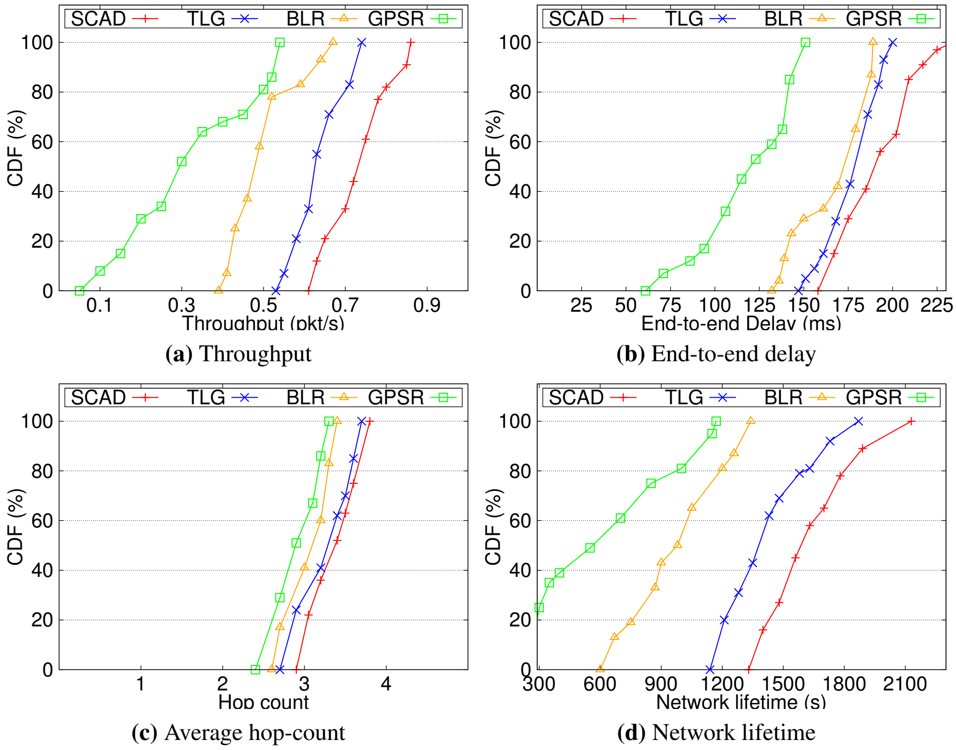


Figure 7.13: Linear, merging, and aggregation topologies of large-scale testbed



(a) Throughput **(b)** End-to-end delay **(c)** Average hop-count **(d)** Network lifetime

Figure 7.14: CDF as function of (a) Throughput; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the ***Linear*** topology in the ***Large-scale Testbed*** experiment

Fig. 7.14a shows the CDF results of the throughput. As in the small-scale experiment, GPSR has the worst result. Compared to the small-scale desktop experiment, the performance of GPSR degrades significantly. As we can see, 60% of the experiments generate a throughput of around 0.3 packet/s. In the desktop experiment, this value is around 0.6 packet/s. This performance degradation is mainly due to the fact that in the larger-scale testbed experiment, nodes are distributed in the building over four floors. The concrete walls, floors and objects create many interferences, which significantly reduce the channel quality of wireless transmission. The wireless signal suffers severe attenuation and multi-path propagation from floors, walls, and roofs, which contributes a lot to performance degradation. Moreover, the signal fluctuation is random such that no prediction can be made to estimate the channel quality due to the dynamics of network topology. Since GPSR solely uses the hop count as routing metric and prefers the shortest path between source and destination pair, the selected path is usually of the shortest length but with very poor link quality. This leads to frequent transmission failures and retransmissions. This wastes the scarce bandwidth resource and decreases the network throughput. Additionally, the channel fluctuation makes GPSR routing table outdated frequently, which further degrades performance. The performance of BLR, TLG, and SCAD also gets reduced because of the channel quality degradation in large-scale distributed testbed environments. Even with degradation, performance of BLR, TLG, and SCAD are still far better than GPSR. As we can see from the plots, 80% of the experiments with these three protocols achieves a throughput value of around 0.6 packet/s. BLR is the worst performing protocol, since it chooses the neighbor closest to the sink as the forwarder. Therefore, it also prefers a short path with poor links. That's why BLR's performance is bit far from SCAD and TLG, which consider link quality in the routing selection. TLG improves BLR by modifying the DFD calculation by taking into account not only progress, but also link quality and the remaining energy of candidates. This enables TLG to choose a forwarder with a good balance between link quality and distance. SCAD has the best result, and more than 80 % of its experiments finished with a throughput value higher than 0.8 packet/s, which is very similar to the result in the small-scale desktop experiment. This means SCAD can scale well and is immune to frequent channel fluctuations. This is because SCAD focuses more on the energy efficiency by introducing adaptive duty cycles, which enable SCAD to keep radios on when traffic load is high. Therefore, SCAD has the best throughput result.

Fig. 7.14b presents the CDF results of the end-to-end delay of the four protocols. As we can see, all the four protocols have increased delays. For example, the plot shows that around 60% of the packets transmitted using GPSR can reach the destination within 135 ms. The shortest delay is still observed at the GPSR protocol, because it does not require any extra time to find next-hops, and the route information is prepared in the routing table prior to data transmission. Due to the fact that BLR, TLG, and SCAD select their forwarders in a fully distributed way, and they use a timer-based forwarder selection mechanism at each hop, they introduce a certain delay at each hop. Therefore, the delay results of these three protocols are worse than that of GPSR. As in the small scale network with a linear topology (Figure 7.10b), TLG and SCAD have longer delays than BLR, since they care more about link quality instead of distance progress at each hop. This leads to the fact that the packet has to travel along more hops to reach the destination. Due to the duty cycling mechanism, SCAD makes some nodes to sleep adaptively. Duty-cycling reduces the number of nodes that can overhear the broadcast transmission, which generates slightly longer delays.

Fig. 7.14c indicates the CDF results of the hop count that a packet takes to reach the destination. GPSR and BLR still perform better than SCAD and TLG, because they choose forwarders by only considering the distance improvement. GPSR and BLR always select the neighbor closest to the destination as next-hop, and thus, they need less hops to reach the destination. TLG and SCAD choose forwarders by jointly considering distance improvement and link quality. Distance improvement and link quality are contrary metrics, since a node with good link quality usually provides small distance advancement. Therefore, TLG and SCAD always choose a node with good link quality while also providing a big distance improvement as forwarder, which takes a packet more hops to reach the sink. In this topology, 80% of the packets transmitted using GPSR and BLR reach the destination with around 3 hops, and 80% of the packets transmitted using SCAD and TLG reach the destination with around 3.8 hops. A clear difference between this result and the result in the desktop small-scale experiment is that the hop-count difference between TLG, SCAD and BLR, GPSR is reduced. The reduction of the difference is due to fact that channel fluctuation has more severe influence on protocols that do not consider link quality in their route selection mechanisms, such as GPSR and BLR.

Fig. 7.14d depicts the CDF results of network lifetime under different protocols. The implemented on-line energy profiling mechanism will be deployed to provide the routing layer with the latest residual energy information. As shown in the figure, in a real-world testbed, the lifetime of GPSR is reduced significantly compared to the desktop scenario, and 60% of the experiments have a network lifetime reduced from 900 s to only 700 s. This is because GPSR keeps using the same nodes that are recorded in the routing table for packet transmissions, which leads to fast energy depletion. BLR has a better result, and 80% of its experiments have a network lifetime of 1500 s. This is because BLR has no controlling overhead, therefore nodes live longer than GPSR. TLG has 80% of its experiments with a network lifetime of 1700 s. This is due to its consideration of remaining energy, and it tries to uniformly distribute the energy consumption to different nodes according to their residual energy. In the linear topology, SCAD performs similar to TLG, and it has 80% of the experiments with a network lifetime of 1850 s. Thanks to the adaptive duty-cycle mechanism , SCAD provides high performance when traffic load is high. It can also make the nodes sleep longer once its energy drain rate is high to compensate its fast energy dissipation to prolong the network lifetime.

Merging Topology

In this experiment, we deploy six nodes to make a merging topology where the traffic from two separate source nodes will be merged at one intermediate node and then be forwarded to the destination node. Our goal is to examine the behavior of the four protocols under the scenario of multiple data sources and observe the behavior of the merging node. The six nodes are deployed over the four floors, in which nodes 1 and 5 are two source nodes, D is the destination, nodes 2, 6, and 7 are the intermediate nodes (as shown in Figure 7.13). The distance between two neighboring nodes is around 8 m. Figure 7.15 presents the performance of the four protocols in the merging scenario.

Fig. 7.15a shows the CDF results of throughput in the merging topology. Due to the fact that now there are two independent data sources, the throughput of the network increased significantly for all protocols. GPSR still has the worst result, and compared to the single source case in the linear topology, its performance improvement is limited since with more traffic, GPSR has higher chances to have congestions, which decreases the performance. Compared to the merging scenario in the desktop experiment, in a lossy large-scale testbed environment, GPSR routing tables are outdated more frequently, and the source has to search for new routes. Therefore, the performance of the larger-scale testbed experiment is worse than that of the small-scale desktop experiment. This is confirmed by the results where, in the desktop tests, GPSR has its 80% of the experiments with a throughput of around 0.85 packet/s, while in the experiment runs, GPSR has its 80% of the experiments with a throughput of around 0.65 packet/s. The throughput values of BLR, TLG, and SCAD have more significant improvements than GPSR compared to the single source linear topology in the testbed experiments. As we can see from the CDF plots, 80% of the experiments under these three protocols achieve a throughput value of 1.2 packet/s, 1.4 packet/s, and 1.6 packet/s. Thanks to the adaptive duty cycle, SCAD nodes could keep their radios on when the network traffic load is high, therefore it has the best throughput figure.

Fig. 7.15b denotes the CDF results of the end-to-end delay in the merging scenario. One significant difference from the single source linear scenario is that, GPSR does not have the best result of delay anymore. This is because under high traffic load (multiple data flows merge at the merging node 7, which increase the traffic load at node 7), congestion will happen with a higher probability and collisions will be more severe. Thus, GPSR has to set up new routes and retransmit the packets, which produces long delay. Compared to the merging scenario in the desktop experiment, channel variations due to various indoor interferences and signal attenuations degrade performance of all protocols. Therefore, protocols of BLR, TLG, and SCAD have increased end-to-end delays.

Fig. 7.15c presents the CDF results of the hop count that a packet takes to reach the destination. Since now a packet has to go through more hops than in the single source linear topology, all protocols show a larger values of hop-counts. Similar as in the linear scenario, GPSR and BLR have the best performance because

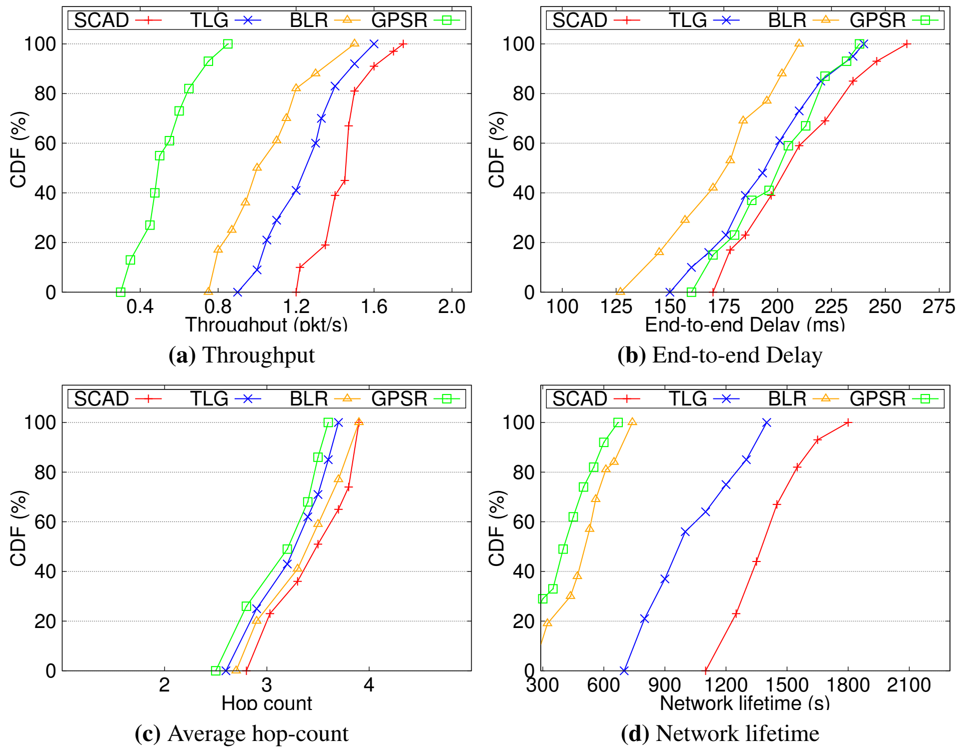


Figure 7.15: CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Merging*** topology in the ***Large-scale Testbed*** experiment

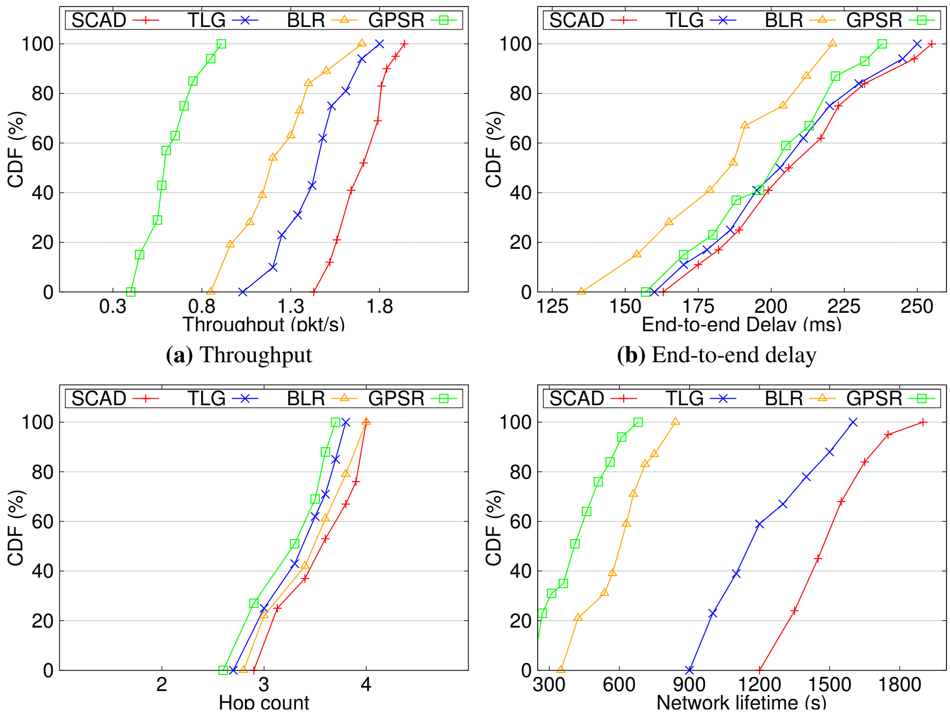
they choose neighbors closest to the destination as packet forwarders. Therefore, they need less hops to reach the destination. TLG and SCAD consider distance advancements and link quality to choose forwarders. Therefore, they make intermediate progress at each hop, which results in a larger hop count.

Fig. 7.15d depicts the CDF results of the network lifetime with the four protocols in the merging topology. This measurement is interesting since it reflects the behavior of different protocols under multiple data sources. Moreover, it helps to monitor the behavior of the merging node, which is most probably the bottleneck of the network in terms of lifetime. All the sensor nodes are configured with the same initial energy level, and the run-time energy profiling mechanism will be triggered necessarily to provide real-time residual energy information to the routing layer of TLG and SCAD. GPSR and BLR do not consider any information related to energy. Therefore, their behavior are completely independent from the residual energy of nodes. When more traffic is injected into a node with low battery level, that node will not deny the forwarding task due to its ignorance of energy information. Additionally, increased traffic and possible congestion will make GPSR and BLR nodes deplete their energy quickly. Therefore, we can observe that GPSR has 80% of its experiments finished with a lifetime of 580 s, and BLR has 80% of its experiments with a lifetime of 610 s. However, TLG and SCAD consider the residual energy when choosing the packet forwarder. Therefore, they are able to uniformly distribute the energy consumption to different nodes according to their remaining energy levels. TLG has 80% of its experiments with a network lifetime of 1300 s, which is already far better than GPSR and BLR. However, if only the residual energy is considered, a node with sufficient residual energy will accept all the packet forwarding requests from multiple neighbors continuously and be involved in multiple packet transmissions for different sources. In this way, its energy will be also depleted quickly. For instance, in the merging topology of Figure 7.13, merging node 7 may receive packet forwarding requests from node 2 and 6 sequentially. If it has enough energy, it will accept both requests and be involved in the packet transmission tasks. However, if it considers the energy drain rate information additionally, it will notice that it has spent a certain amount of energy during a certain time interval, and will deny any further forwarding request in order to have a long break. Thanks to the consideration of both residual energy and energy drain rate, SCAD is able to avoid over-dissipation of specific nodes by taking into account the current traffic load and by utilizing the energy drain rate, especially under the scenarios of high traffic load. This enables SCAD to evenly distribute the energy expenditure to multiple nodes. Therefore, SCAD has a longer lifetime than TLG, which is confirmed in the plot that SCAD has 80% of its experiment with a network lifetime of 1600 s. Thanks to the adaptive duty-cycle capability, SCAD provides a better performance when traffic load is high. On the other hand, SCAD also makes nodes sleep longer once their energy drain rates are high, to compensate their fast energy dissipation and prolong network lifetime.

Aggregation Topology

In this experiment, we deploy seven nodes of 2, 3, 4, 5, 6, 8 and D, in which node 2 and 5 are the sources , and D is the destination. The packets from source 2 and 5 are targeted to the sink node D. Results of the four protocols in the aggregation topology are shown in Figure 7.16.

Fig. 7.16a shows the CDF results of the throughput in the aggregation topology of the large-scale testbed experiment. Similar results can be observed as in the merging scenario of testbed experiment, since these two scenarios are somehow similar. Due to the fact that GPSR uses hop count as routing metric and prefers a short path with poor radio links, its performance is the worst. In a real-world testbed environment, transmission failures happen frequently and thus require routing table updates and packet retransmissions. Compared to the merging scenario, one more node is deployed in this scenario, and thus, it takes certain parts of the packet forwarding tasks. For instance, the possible congestion at the merging node 7 in the merging topology will be removed or reduced. Therefore, the throughput will also be improved. This is confirmed by the figures of CDF of GPSR in these two topologies. In the merging topology, GPSR has 80% of the experiments finished with a throughput value of around 0.68 packet/s, while in the aggregation topology, this number is around 0.82 packet/s. BLR, TLG, and SCAD still perform far better than GPSR. As we can see from the CDF plots, BLR has 80% of its experiments achieving a throughput of 1.32 packet/s, TLG has 80% of its experiments achieving a throughput of 1.82 packet/s, and SCAD has 80% of its experiments achieving a throughput of 2.32 packet/s. BLR is the worst of these three protocols, since it prefers short paths with poor links. TLG considers not only progress, but also link quality and remaining energy of candidates when calculating DFD. This enables TLG to choose a forwarder with a good balance between link quality and distance improvement. SCAD has the best result, because it focuses more on the energy efficiency by introducing adaptive duty cycles to save energy. An interesting observation of this aggregation topology is that, the performance differences of BLR, TLG, and SCAD become larger. This might be because with this topology, multiple sources could have their data simultaneously transmitted over two paths individually. On each path, the benefit of considering link quality and adaptive duty cycle will be accumulated separately, and in the end, the benefits on the two paths will also be added. This produces enlarged performance differences among protocols BLR, TLG, and SCAD.



(a) Throughput

(b) End-to-end delay

(c) Average hop-count

(d) Network lifetime

Figure 7.16: CDF as function of (a) Throughput; (b) End-to-end delay; (c) Average hop-count; (d) Network lifetime in the ***Aggregation*** topology in the ***Large-scale Testbed*** experiment

Fig. 7.16b denotes the CDF results of the end-to-end delay. The results of the four protocols are very similar to that of the merging topology. GPSR has the shortest delay, because GPSR does not need extra time to get a path for packet transmission. BLR, TLG, and SCAD add additional delays at each hop to choose forwarders. Therefore, they have larger delays than GPSR. The delays of TLG and SCAD are larger than that of BLR, due to their consideration of link quality. The link quality constraint makes them choose a node, which is of good link quality and is located very close to the packet sender. Therefore, their packets need more hops to reach the destination. SCAD has longer delays than TLG, because its duty cycle mechanism limits the number of nodes that can concurrently overhear broadcast transmissions. However, compared to the delay results in the merging topology (Fig. 7.15b), the delay of SCAD in the aggregation topology has been reduced significantly. This is because in the merging topology, node 7 is the only merging node in the network. Therefore, whenever it needs a longer sleep after an intensive packet transmission, all the packets going to the destination will be delayed and this will significantly increases the end-to-end delay. However, in the aggregation topology, node 4 or 8 can both work as the merging node. The introduction of node 4 (or 8) gives more options to packet forwarding. Therefore, when one of the merging nodes, for example node 4 needs to sleep longer, the other "back-up" merging node 8 will take over the forwarding task. Thus, the delay introduced by having only one merging node is decreased significantly. This explains why we can observe an obvious reduction of the end-to-end delay of SCAD in the aggregation topology, compared to the merging topology.

Fig. 7.16c indicates the CDF results of the hop count that a packet takes to reach the destination. The performance of the four protocols is also very similar to the merging scenario. GPSR and BLR have the best performance because they choose forwarders by only considering the distance improvement. Therefore, they need less hops to reach the destination. TLG and SCAD choose forwarders by considering not only the distance improvement, but also the link quality and energy information. The consideration of link quality makes TLG and SCAD choose nodes located closely as forwarders, which takes a packet more hops to reach the destination.

Fig. 7.16d presents the CDF results of the network lifetime with different protocols. Similar as in the merging scenario, GPSR has the worst performance since it has no consideration of any energy information. This is the same reason why BLR has a similar bad performance as GPSR. Compared to the merging topology, TLG and SCAD have a better performance, their network lifetimes are increased. This is because of the introduction of one more merging node, which takes over some forwarding tasks and relieves congestion that might happen at the only one merging node in the merging topology.

Mesh Topology

In the last experiment, we uniformly deploy 20 nodes in the testbed and the nodes form a mesh network. Node deployment is shown in Figure 7.17 (depicted links are just some examples of connected links), in which D is the destination, 9, 15, 16, 18, 19 are the source nodes, and 14 other nodes are intermediate nodes. The corresponding network topology is presented in Figure 7.18. As before, the source nodes generate the constant bit rate (CBR) packets towards the destination with a rate of 1 packet/s. Transmission power of nodes has been configured to enable nodes to reach their one-hop neighbors.

We divided our experiments into two parts: the first one with static topology and the second one with dynamic topology. Dynamic topologies can be caused by several reasons, such as node mobility, interference, or the adjustment of radio transmission power. Due to the absence of mobile sensor nodes in the indoor testbed, we implemented the dynamic topology artificially by adjusting the transmission power of the sensor nodes during the experiments at certain scheduled time points. When the transmission power is modified, the connectivity among nodes changes, which can be regarded as a dynamic topology. We use the dynamic topology to examine the performance of different protocols under realistic dynamic environments. The CC2420 radio chips are able to provide 31 transmission power levels by setting the TXCTRL register appropriately, which are shown in Table 7.4. We used two different transmission power levels during the experiments: one transmission power from the default value of -10 dBm and another smaller value of -25 dBm, which produce a transmission range of 11 m and 2.5 m separately. Starting from the default value of -10 dBm, two values are interchanged every 180 seconds.

Table 7.4: CC2420 Transmission Power Level [131]

Power Level	TXCTRL Register	Output Power	Current Consumption
31	0xA0FF	0 dBm	17.4 mA
27	0xA0FB	-1 dBm	16.5 mA
23	0xA0F7	-3 dBm	15.2 mA
19	0xA0F3	-5 dBm	13.9 mA
15	0xA0EF	-7 dBm	12.5 mA
11	0xA0EB	-10 dBm	11.2 mA
7	0xA0E7	-15 dBm	9.9 mA
3	0xA0E3	-25 dBm	8.5 mA

Figure 7.19 presents the CDF of throughput, end-to-end delay, average hop count, and network lifetime of the four protocols under the static mesh topology in large-scale testbed experiments. Fig. 7.19a shows the throughput results, from which we can find out that the performance ordering of the four protocols stays the same as in the other three topologies. Due to the fact that now there are five source nodes in the mesh topology, the throughput of all the protocols are improved. How-

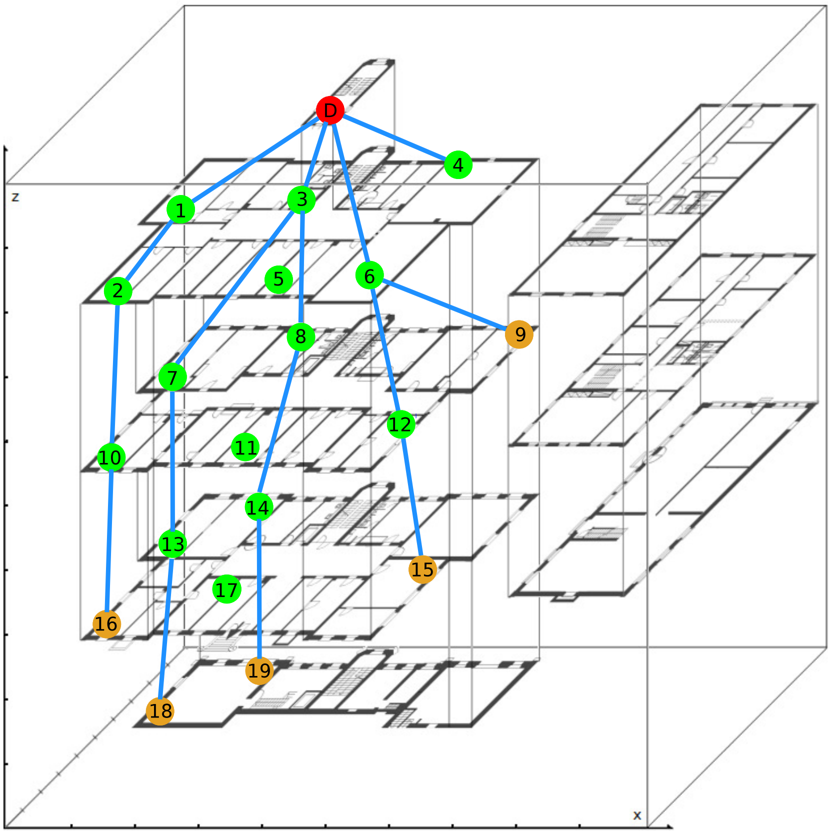


Figure 7.17: Large-scale testbed deployment: Mesh Topology

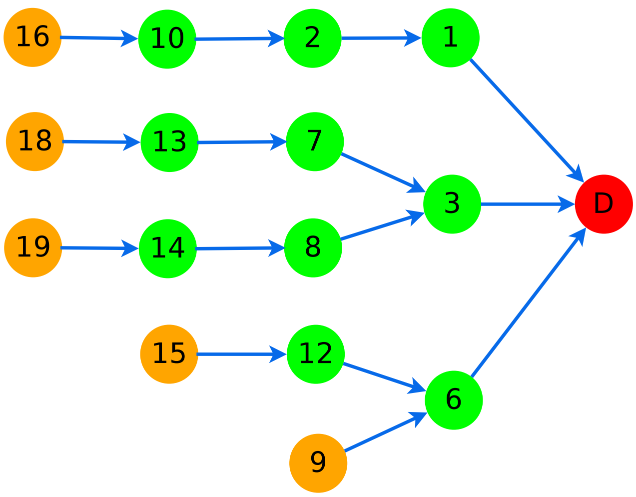
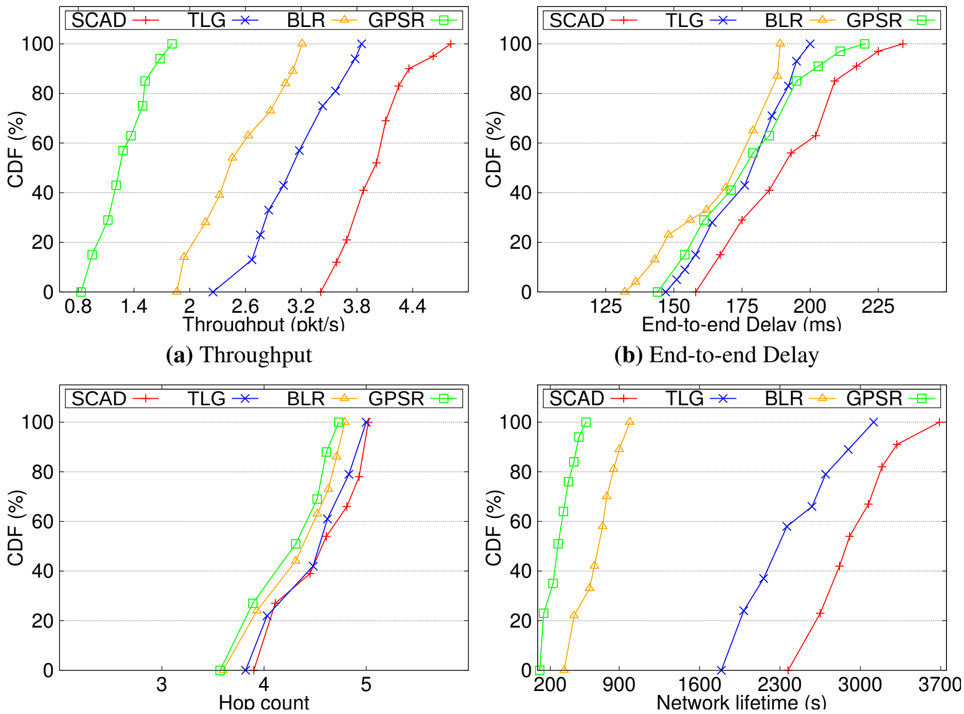


Figure 7.18: Mesh topology of the large-scale testbed



(a) Throughput

(b) End-to-end Delay

(c) Average hop-count

(d) Network lifetime

Figure 7.19: CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Static Mesh*** topology in the ***Large-scale Testbed*** experiment

ever, compared to the throughput results in the merging topology (Fig. 7.16a), the amount of improvements are different for the four protocols. The throughput improvements from the aggregation topology to the mesh topology are: GPSR increases from $\sim$ 0.7 packet/s to $\sim$ 1.5 packet/s; BLR increases from $\sim$ 1.4 packet/s to $\sim$ 3.1 packet/s; TLG increases from $\sim$ 1.55 packet/s to $\sim$ 3.5 packet/s; and SCAD increases from $\sim$ 1.9 packet/s to $\sim$ 4.3 packet/s. The reason of the different improvements is due to the routing mechanisms of different protocols. GPSR relies on a routing table, which is pre-generated via control messages. When the number of source nodes increases, the probability of collision or interference increases, which leads to packet retransmissions, and thus, degrades routing performance. On the other hand, BLR, TLG, and SCAD are beaconless protocols, they are immune to topology changes. Therefore, their performance improvements are more significant than that of GPSR.

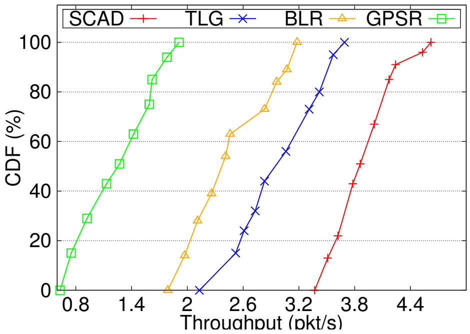
Fig. 7.19b depicts the delay results of the four protocols. Compared to the delay results of the linear topology (Fig. 7.14c), GPSR has increased delays. A possible reason is that with more source nodes and intermediate nodes deployed in the mesh topology, concurrent packet transmissions are more likely to suffer from transmission collisions. However, GPSR forwards packets based on a routing table, which is calculated prior to the transmission of data packets. When collisions occur, source nodes have to update the routing table and retransmit the packets, which leads to increased delay. Beaconless protocols BLR, TLG, and SCAD forward the packets in a hop-by-hop way, and the next-hop forwarder is chosen only from the neighbors that have successfully received the transmissions. Therefore, collisions are reduced significantly. This explains why the delay of GPSR increases significantly and that of BLR, TLG, and SCAD stay similar with the results in the linear topology (Fig. 7.14c).

Fig. 7.19c shows the hop count information of the four protocols. Since the mesh topology is of a larger size, packets from source nodes have to travel through more intermediate nodes to reach the destination. Therefore, an increased hop count value is observed for each protocol.

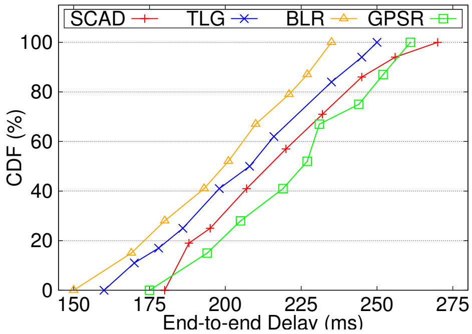
Fig. 7.19d indicates the network lifetime results of the four protocols. Compared to the results in the linear topology, BLR, TLG, and SCAD all have significant increases of network lifetime, however, the lifetime of GPSR is reduced. This is again related to the forwarding mechanisms of the different protocols. GPSR uses the route information stored in the routing table to forward packets. As far as the routing table is not updated, it always uses the same nodes to send unicast transmission. The energy depletion rate of the nodes stored in the routing table is then the bottleneck of the network lifetime. Frequent packet retransmissions, which are caused by transmission collisions, also consume additional energy. BLR, TLG, and SCAD are beaconless protocols, they do not choose fixed nodes as packet forwarders. The next-hop forwarder is chosen after the packet transmission at the receiver side. TLG and SCAD choose forwarders based on several types of context information, and the consideration of energy and link quality enable them to distribute packet forwarding uniformly to different nodes in the network. Therefore, different nodes will take the forwarding tasks, and the energy consumption is distributed among them. Energy is not considered in BLR, and thus, a node with few residual energy can not reject any packet forwarding tasks, even if it has few energy left. That is the reason why BLR has a worse network lifetime result than TLG and SCAD.

Figure 7.20 presents the results of the four protocols in the dynamic mesh topology. Fig. 7.20a shows the throughput results. GPSR performance degrades significantly compared to the results in the static mesh topology shown in Fig. 7.19a, because its routing table will be outdated more often in a dynamic topology. The route information recorded in the routing table is invalid when the connections among nodes change, which leads to updates of routing tables and packet retransmissions. In the mesh topology with a high node density, bandwidth is a scarce resource and will be occupied by frequent packet retransmissions, which degrades the throughput performance of GPSR. Beaconless protocols like BLR, TLG, and SCAD start packet transmissions by initiating broadcast transmissions, which is independent of topology changes. Therefore, their performance values are less affected by the changes of network topology. SCAD has the best result, this is due to its adaptive duty-cycle scheme, which enables nodes to forward the packets when traffic rate is high. In fact, as we can see from Figure 7.18, the mesh topology is a combination of linear topology, merging topology, and aggregation topology. Therefore, the benefits of SCAD in different topologies will be accumulated to generate a significant performance improvement in the mesh topology. Due to the high node density, the benefits of opportunistic routing can still be preserved, even if in some cases the radios of some nodes might be switched off periodically due to the duty-cycle mechanism of SCAD.

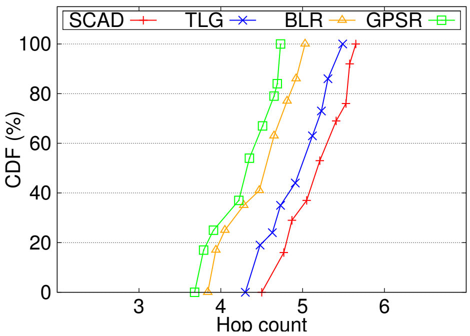
Fig. 7.20b depicts the delay results of the four protocols. An interesting observation is that the end-to-end delay of GPSR increases significantly, and it even overtakes that of SCAD. This is mainly due to the multiple packet retransmissions caused by outdated routing tables and route re-discoveries. When the network topology changes over time, the packets have to be buffered, wait for updates of routing tables, and then be retransmitted. Another reason of the increased delay values are the possible transmission collisions in the large-scale topology, which also leads to packet retransmissions. The increased amount of packet retransmission directly produces an increased end-to-end delay for GPSR. The delay results of the other three protocols also increase for dynamic network topologies, since a changing topology introduces additional delays for BLR, TLG, and SCAD when selecting packet forwarders.



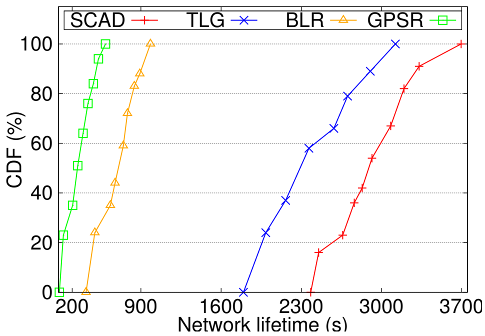
(a) Throughput



(b) End-to-end Delay



(c) Average hop-count



(d) Network lifetime

Figure 7.20: CDF as function of (a) Throughput; (b) End-to-end Delay; (c) Average hop-count; (d) Network lifetime in the ***Dynamic Mesh*** topology in the ***Large-scale Testbed*** experiment

Fig. 7.20c describes the hop count results of the four protocols. As we can observe, GPSR and BLR almost keep the same results as in the static mesh topology, since they always prefer nodes located closer to the destination, which indicates a short path to reach the destination. TLG and SCAD need a bit more hops than in the static mesh topology, because the changing topology makes them always choose closer nodes as forwarders and thus leads to a longer path to reach the destination.

Fig. 7.20d indicates the network lifetime results of the four protocols. GPSR has the worst result. Compared to the static mesh topology, its network lifetime reduces significantly. This is because in a dynamic topology, GPSR routing tables are outdated frequently, which needs additional control messages to update the routing table. Frequent routing table updates and packet retransmissions will cost much energy and produce shorter network lifetimes. BLR, TLG, and SCAD are rather immune to topology changes, since they choose forwarders in a distributed way. For SCAD and TLG, the changing transmission power enables the packets to be received by nodes located in different areas (a distant area if the transmission power is high, or a close area if the transmission power is low), which are covered by different radio ranges. This enables packets to be forwarded by nodes located at different areas, which helps to distribute packet forwarding tasks to nodes in different regions and therefore prolongs network lifetime. Thanks to the adaptive duty-cycle mechanism, SCAD is able to conserve energy of a node whose energy drain rate is high, which further ensures the distribution of energy consumption to nodes with enough energy budget and low energy drain rate. Therefore, SCAD produces a prolonged network lifetime, in the meantime, keeps the throughput at a satisfactory level.

7.7 Conclusions

The quality of a wireless link between two sensor nodes is not static. It varies over time and might be affected by many factors, such as channel fading, interferences, collisions, etc. The time varying nature of wireless channel quality has an impact on the bit error rate of the received packet and a received packet with more bit errors will be discarded, which leads to packet retransmissions. In resource-constrained networks like wireless sensor networks, intensive packet retransmissions significantly occupy the network bandwidth and degrade system performance. Moreover, frequent packet retransmissions cost additional energy and reduce the lifetime of the network. In order to enhance network lifetime, packet retransmission should be minimized by sending packets through the link with good channel characteristics. The lifetime of the network can be further improved by distributing the packet forwarding tasks to different nodes, which have sufficient remaining energy, to distribute the energy consumption to multiple nodes. Therefore, network lifetime optimization is achieved by reducing the number of retransmission and load balancing.

To achieve these goals, this chapter proposes a new cross-layer opportunistic routing protocol called SCAD: *Sensor Context-aware Adaptive Duty-cycled* beaconless opportunistic routing protocol for wireless sensor networks. SCAD adapts the idea of beaconless opportunistic routing to the specific requirements of WSNs by taking energy efficiency as a major concern. SCAD selects forwarders based on multiple types of cross-layer context information, such as node residual energy, geographic progress of the forwarding candidate, received packet link quality, and energy drain rate. Energy is of primary concern in wireless sensor networks and their protocol design, SCAD additionally takes energy as an important issue. SCAD adapts duty cycles of sensor nodes according to the real-time monitored network traffic load and the estimated sensor node energy drain rate to achieve a balance between routing performance and energy efficiency. The effectiveness of SCAD has been validated by experiments in both simulation and real-world implementation in indoor testbeds. Real-world testbed experiments include a small-scale and a large-scale experiment, and the evaluation results confirm that SCAD outperforms other beaconless or geographic routing protocols, and that it provides good routing performance while increasing network lifetime.

Chapter 8

Conclusions and Outlook

In this chapter, we first summarize the contributions of this thesis in the order of their existences in the thesis at section 8.1. Our first contribution is a simulation framework of opportunistic routing protocols. The second contribution is a topology and link quality-aware geographic opportunistic routing protocol. The third contribution is a context-aware adaptive opportunistic routing protocol, which is an extension of our second contribution. The fourth contribution is a sensor context-aware adaptive duty-cycled cross-layer opportunistic routing protocol. Afterwards, we briefly elaborate on possible future work in the field of the routing approaches proposed in the thesis at section 8.2.

8.1 Main Contributions

In this thesis, we focused on the study of opportunistic routing protocols in mobile ad-hoc and sensor networks. We proposed a generalized context-aware opportunistic routing concept and applied the idea into the domain of mobile ad-hoc and sensor networks. We proposed to consider multiple cross-layer network context information to make the routing decision synthetically. Besides, due to the intrinsic nature of dynamics in wireless ad-hoc and sensor networks, we designed an adaptive mechanism in the routing process to adapt protocol behaviors according to the instantaneous values of the interested context information.

In Chapter 4, we analyzed the common features of existing opportunistic routing protocols, and we claimed that there are some shared characteristics that can be abstracted to facilitate the analysis and future developments of opportunistic routing protocols. We therefore presented a simulation framework, which decouples the opportunistic routing schemes into four procedures - *Forwarder Candidate Selection*, *Forwarder Selection*, *Forwarder Role Change Notification* and *Collision Avoidance*. Different protocols should have specific implementation mechanisms of each procedure. In the framework, these four procedures are defined as virtual functions and act as implementation stubs such that different protocols could be implemented by overriding them in the derived function according to their distributed strategies.

In Chapter 5, we proposed TLG - Topology and Link quality-aware Geographic opportunistic routing protocol for mobile ad-hoc networks. Unlike existing opportunistic routing protocols, which select packet forwarders based on a pre-generated candidate list, TLG completely abandons the idea of candidate list and allows all the qualified nodes to participate in the packet forwarding process. Additionally, TLG simultaneously uses multiple network metrics such as network topology, link quality, and geographic location to implement the coordination mechanism among multiple receivers of a broadcast packet to select the forwarder. To evaluate the performance of our proposal, we perform two types of simulations. First we use the scalar data, which means the source node sends a scalar data packet, and the performance is measured from the network point of view. Collected results include packet delivery ratio (PDR), end-to-end delay, average hop count, etc. Besides the scalar data, we also configured the source node to send multimedia data, and the experiments are conducted to collect quality of experience metrics from the end-users' point of view. This gives us the opportunity to test our protocol in scenarios where pure scalar data can not fulfill the requirements. For example, in a disaster rescue task, end-users need to receive the multimedia data (video or picture of the monitored) to support decision making. Simulation evaluations of both scalar and multimedia data validate the performance of our protocol.

In Chapter 6, we further extended our work of TLG by considering more general context information for the routing decision. We proposed a novel concept of opportunistic routing, called CAOR - Context-aware Adaptive Opportunistic Routing for MANETs. In CAOR, depending on the application requirements, any types of context information, which are helpful to make routing decision, could be included. CAOR forwards packets by simultaneously exploiting multiple types of cross-layer context information, such as link quality, geographic progress, residual energy, and node mobility. With the help of the Analytic Hierarchy Process (AHP) theory, CAOR adjusts the weights of context information based on their instantaneous values to adapt the protocol behavior at run-time. Moreover, CAOR uses an active suppression mechanism to reduce packet duplication. Simulation results show that CAOR can provide efficient routing in highly mobile environments, and it outperforms other protocols significantly.

In Chapter 7, we implemented the idea of context-aware opportunistic routing in real-world wireless sensor networks. In order to fully achieve the functionalities defined in the protocol, we have implemented an on-line energy profiling mechanism to get the latest residual energy of sensor nodes, which is needed to provide energy-awareness for routing decisions. Due to the energy constraints of wireless sensor nodes, we adopt a duty-cycle mechanism to save the scarce resource of energy if the detected energy drain rate is high. Unlike the existing duty-cycle mechanism in most of the WSN MAC protocols, we proposed to adapt the sensor's duty cycles based on both monitored traffic rates and estimated energy drain rates. Our goal is to enable the sensor node to fully work once the traffic load is high, and to sleep longer after the traffic peak according to its energy drain rate. If the energy spent in a certain period of time (called sliding window) is larger than a predefined threshold value, then the node will sleep longer and ignore further forwarding requests to compensate its fast energy consumption. In general, we linked the duty cycles of sensor nodes to traffic rates and energy drain rates. Evaluation results from both simulation and real-world implementation prove that SCAD produces the best results, and it provides satisfactory routing performance, while keeping the network alive longer than other protocols.

8.2 Outlook

In this thesis, we proposed the concept of context-aware opportunistic routing, which combines the idea of opportunistic routing with context-aware communication. Our proposal enables network nodes to select packet forwarders opportunistically by taking into account multiple types of context information. Additionally, we integrated the idea of adaptivity into the design of opportunistic routing such that nodes could adjust their behaviors according to the real-time values of the interested context information. To validate our ideas, we performed various types of simulation in static and mobile environments. Besides, we also validated our proposals using real-world implementation experiments in both small-scale and large-scale testbed scenarios. However, the completed real-world experiments were only conducted in static environments without any mobile objects. This is due to the absence of mobile sensors in the testbed. In the future, we might consider to add some mobile sensors into our indoor testbed and perform further experiments including mobile sensor nodes. Packet forwarding will be more challenging with mobile nodes deployed in the real-world testbed, since node mobility will increase the dynamics of the network. On the other hand, the mobile sensors would be implemented by an integration of normal sensors and mobile unmanned vehicles, and the sensors should be carried by the moving vehicles. The introduction of carrying objects will bring new opportunities for the design of routing protocols, along with new challenges ahead. For example, the carrying objects could have more space for additional resources for the carried sensor nodes, such as battery or storage. The integration of sensors and carrying objects should be designed properly to promote overall system performance.

When sensors become mobile, one important issue is to keep the connectivity among the moving objects, such that the network will not be divided into multiple disconnected zones. This results in new problems of connectivity and topology control in mobile ad-hoc sensor networks. Many works have been done in this area. The most common approach, for example in wireless sensor networks, is to adapt the transmission ranges of the sensor nodes to maintain the network topology. However, most of the works focused on the static scenarios, while few efforts have been made into the connectivity and topology control in the environments with mobile objects. We plan to design and implement some novel topology control protocols based on the signal strength of the received packets, such as our preliminary proposal described in [156].

Moreover, another work that deserves more investigation is the validation of the dynamic adjustment of the context parameters (Chapter 6) for a WSN network, in both network simulation and real-world testbed experiment. Currently, we only verify the idea of on-line adjustment of the multiple context parameters in network simulations for a MANET. The same idea should be implemented in a WSN environment (in a simulator and real-world testbed) and tested in a large-scale environment. When moving from a network simulator to a physical testbed, some new challenges might come, which need further investigation. For example, the calculation of the new context weight parameters requires the application of the AHP algorithm to consider the latest context values. The generation of new weights needs certain amount of mathematic computation. In a network simulation environment, this operation can be done without any concern of energy or computing resources. However, in a real-world implementation, for instance sensor motes, a lot of physical constraints might make the realization of the algorithm a nontrivial task. The computation overhead involved in the AHP calculation might reduce the benefits it brings. Therefore, some optimizations of the original AHP algorithm might be needed such that a satisfactory result could be achieved in a real-world implementation. Additionally, a theoretical analysis of the computation complexity of using the AHP algorithm to adjust the context parameters at run-time is also of great interest.

Chapter 9

Acronyms

AHP Analytical Hierarchy Process

ARQ Automatic Repeat reQuest

BLR Beacon Less Routing

CAOR Context-aware Adaptive Opportunistic Routing

CBR Constant Bit Rate

CDF Cumulative Distribution Function

CSMA Carrier Sense Multiple Access

CTS Clear To Send

DFD Dynamic Forwarding Delay

DTN Delay Tolerant Network

DSDV Destination Sequence Distance Vector

EAX Expected Anypath Transmission

EOF Energy Objective Function

ETX Expected Transmission Count

EWMA Exponential Weighted Moving Average

ExOR Extremely Opportunistic Routing

FCC Federal Communications Commission

FEC Forward Error Correction

GOP Group of Pictures

GPSR	Greedy Perimeter Stateless Routing
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LIVE	Link Validation Estimation
LQI	Link Quality Indicator
MANET	Mobile Ad-hoc Network
MAC	Medium Access Control
MCU	Micro Controller Unit
MPR	Multi Point Relaying
OLSR	Optimized Link State Routing
OR	Opportunistic Routing
PDR	Packer Delivery Ratio
QoS	Quality of Service
QoE	Quality of Experience
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time
RTS	Request To Send
SCAD	Sensor Context-aware Adaptive Duty-cycled routing
SNMD	Sensor Node Management Device
SSIM	Structural SIMilarity
TCP	Transport Control Protocol
TLG	Topology and Link-quality aware Geographic routing
UDP	User Datagram Protocol
USB	Universal Serial Bus
VANET	Vehicle Ad-hoc Network
VQM	Video Quality Metric
UAV	Unmanned Aerial Vehicle

UAVNet	UAV Ad-hoc Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network
WMSN	Wireless Multimedia Sensor Network

Bibliography

[1] Bonnmotion. http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/.

[2] Btnode sensor mote. http://www.btnode.ethz.ch/.

[3] Csma. http://en.wikipedia.org/wiki/Carrier-sense-multiple-access.

[4] Exponential weighted moving average. http://lorien.ncl.ac.uk/ming/filter/filewma.htm.

[5] Ieee 802.11 rts/cts mechanism. http://en.wikipedia.org/wiki/IEEE-802.11-RTS-CTS.

[6] Ieee 802.11 standard. http://www.ieee802.org/11/.

[7] Inetmanet framework. https://github.com/inetmanet/inetmanet.

[8] Msp430 processor. http://www.ti.com/lsds/ti/microcontroller/16-bit msp430/overview.page.

[9] Omnet++ inet framework. http://inet.omnetpp.org/.

[10] Omnet++ mixim framework. http://mixim.sourceforge.net/.

[11] Personal area networks. http://en.wikipedia.org/wiki/Personal area network.

[12] Random walk model. http://en.wikipedia.org/wiki/Random-walk.

[13] Random waypoint model. http://en.wikipedia.org/wiki/Random-waypoint-model.

[14] Tdma. http://en.wikipedia.org/wiki/Time-division-multiple-access.

[15] Tmote sky sensor mote. http://www.capsil.org/capsilwiki/index.php/TELOSB.

[16] E. Casilari A. Ariza-Quintana and A. Triviño Cabrera. Simutools '08: Proceedings of the 1st international conference on simulation tools and techniques. In *Implementation of MANET routing protocols on OMNeT++*.

[17] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad hoc networks*, 2(1):1–22, 2004.

[18] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[19] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11 b mesh network. *ACM SIGCOMM Computer Communication Review*, 34(4):121–132, 2004.

[20] Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, 2000.

[21] AHP example to select a leader.

[22] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Comput. Netw.*, 38(4):393–422, March 2002.

[23] Jamal N. Al-Karaki and et al. Efficient virtual-backbone routing in mobile ad hoc networks.

[24] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Ágila Bitsch Link, Stefan Götz, and Klaus Wehrle. Bursty traffic over bursty links. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 71–84. ACM, 2009.

[25] Abdullah M. S. Alkahtani, M. E. Woodward, and K. Al-Begain. Prioritised best effort routing with four quality of service metrics applying the concept of the analytic hierarchy process. *Computers and Operations Research*, 33(3).

[26] Islam T. Almalkawi, Manel Guerrero Zapata, Jamal N. Al-Karaki, and Julian Morillo-Pozo. Wireless multimedia sensor networks: Current trends and future directions. *Sensors*, 10(7):6662–6717, 2010.

[27] I.T. Almalkawi, M. Guerrero Zapata, J.N. Al-Karaki, and J. Morillo-Pozo. Wireless Multimedia Sensor Networks: Current Trends and Future Directions. *Sensors*, 10(7):6662–6717, 2010.

[28] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. BonnMotion: a Mobility Scenario Generation and Analysis Tool. pages 51:1–51:10. International Conference on Simulation Tools and Techniques (SimuTools), Malaga, Spain, March 2010.

[29] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[30] Ana Bildea, Olivier Alphand, Franck Rousseau, and Andrzej Duda. Link quality metrics in large scale indoor wireless sensor networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 1888–1892. IEEE, 2013.

[31] Sanjit Biswas and Robert Morris. Exor: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, August 2005.

[32] Rajendra V Boppana and Satyadeva P Konduru. An adaptive distance vector routing algorithm for mobile, ad hoc networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1753–1762. IEEE, 2001.

[33] A. Boulis. Castalia, a simulator for wireless sensor networks and body area networks, version 2.2. August 2009.

[34] David Braginsky and Deborah Estrin. Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31. ACM, 2002.

[35] Torsten Braun, Marc Heissenbüttel, and Tobias Roth. Performance of the beacon-less routing protocol in realistic scenarios. *Ad Hoc Networks*, 8:96–107, 2010.

[36] Josh Broch, David A Maltz, David B Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97. ACM, 1998.

[37] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320. ACM, 2006.

[38] Claudio Casetti, Matteo Cesana, Ilario Filippini, György Dán, and Ian Marsh. Context-aware information dissemination in vehicular networks. In *Next Generation Internet (NGI), 2011 7th EURO-NGI Conference on*, pages 1–2. IEEE, 2011.

[39] Llorenç Cerdà-Alabern, Vicent Pla, and Amir Darehshoorzadeh. On the performance modeling of opportunistic routing. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, pages 15–21. ACM, 2010.

[40] Geoffrey Werner Challen, Jason Waterman, and Matt Welsh. Idea: Integrated distributed energy awareness for wireless sensor networks. In *Pro-*

ceedings of the 8th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'10), pages 35–48, New York, NY, USA, 2010.

[41] Tsu-Wei Chen and Mario Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 1, pages 171–175. IEEE, 1998.

[42] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, Laurent Viennot, et al. Optimized link state routing protocol (olsr). 2003.

[43] Common Intermediate Format.

[44] CRAWDAD trace repository.

[45] Gianpaolo Cugola and Matteo Migliavacca. A context and content-based routing protocol for mobile sensor networks. In *Wireless Sensor Networks*, pages 69–85. Springer, 2009.

[46] Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. Candidate selection algorithms in opportunistic routing. In *Proceedings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, PM2HW2N '10, pages 48–54, New York, NY, USA, 2010. ACM.

[47] W. Dargie and C. Poellabauer. Fundamentals of wireless sensor networks: theory and practice. *John Wiley and Sons*, pages 168 – 183, 2010.

[48] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[49] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. Least-cost opportunistic routing. In *Proceedings of 2007 Allerton Conference on Communication, Control, and Computing*, volume 5, page 11, 2007.

[50] Valentin Dumitrescu and Jinhua Guo. Context assisted routing protocols for inter-vehicle wireless communication. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 594–600. IEEE, 2005.

[51] A. Dunkels, B. Groenvall, and T. Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. pages 455–462. IEEE Workshop on Embedded Networked Sensors (EmNets), Tampa, Florida, November 2004.

[52] Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, and Zhitao He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 28–32. ACM, 2007.

[53] Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Dover Publications, 1956.

[54] Amre El-Hoiydi and J-D Decotignie. Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*, volume 1, pages 244–251. IEEE, 2004.

[55] J. Eriksson, F. Osterlind, N. Finne, N. Tsiftes, A. Dunkels, and T. Voigt. COOJA/MSPSim: Interoperability Testing for Wireless Sensor Networks. pages 27:1–27:7. International Conference on Simulation Tools and Techniques (SimuTools), Rome, Italy, March 2009.

[56] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.

[57] Holger Füßler, Jörg Widmer, Michael Käsemann, Martin Mauve, and Hannes Hartenstein. Contention-based forwarding for mobile ad hoc networks. *Ad Hoc Networks*, 1(4):351–369, 2003.

[58] Frieder Ganz, Payam Barnaghi, Francois Carrez, and Klaus Moessner. Context-aware management for sensor networks. In *Proceedings of the 5th International Conference on Communication System Software and Middleware*, page 6. ACM, 2011.

[59] Yu Gu and Tian He. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 321–334. ACM, 2007.

[60] H.264/MPEG-4 AVC Encoding.

[61] Zygmunt J Haas, Marc R Pearlman, and Prince Samar. The zone routing protocol (zrp) for ad hoc networks. *draft-ietf-manet-zone-zrp-04. txt*, 2002.

[62] Hall Video Sequence.

[63] Ivaylo Haratcherev, Gertjan Halkes, Tom Parker, Otto Visser, and Koen Langendoen. Powerbench: A scalable testbed infrastructure for benchmarking power consumption. In *Int. Workshop on Sensor Network Engineering (IWSNE)*, pages 37–44, 2008.

[64] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, October 2002.

[65] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

[66] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli. Optimized stateless broadcasting in wireless multi-hop networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, 2006.

[67] Anton Hergenroeder, Joachim Wilke, and Detlev Meier. Distributed energy measurements in wsn testbeds with a sensor node management device (snmd). In *Architecture of Computing Systems (ARCS), 2010 23rd International Conference on*, pages 1–7, Feb 2010.

[68] Barbara Hohlt, Lance Doherty, and Eric Brewer. Flexible power scheduling for sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 205–214. ACM, 2004.

[69] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60. ACM, 1999.

[70] Che-Jung Hsu, Huey-Ing Liu, and Winston Seah. Economy: a duplicate free opportunistic routing. In *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*, page 17. ACM, 2009.

[71] Peizhao Hu, Marius Portmann, Ricky Robinson, and Jadwiga Indulska. Context-aware routing in wireless mesh networks. In *Proceedings of the 2nd ACM international conference on Context-awareness for self-managing systems*, pages 16–23. ACM, 2008.

[72] P. Hurni, M. Anwander, G. Wagenknecht, T. Staub, and T. Braun. TARWIS - A Testbed Management Architecture for Wireless Sensor Network Testbeds. pages 1–5. International Conference on Network and Service Management (CNSM), Short Paper Session, Paris, France, October 2011.

[73] P. Hurni and T. Braun. MaxMAC: a Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks. pages 289–305. European Conference on Wireless Sensor Networks (EWSN), Coimbra, Portugal, February 2010.

[74] P. Hurni and T. Braun. Real-World Experiences with the Maximally Traffic Adaptive Medium Access Control Protocol. Technical Report IAM-11-001, Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland, June 2011.

[75] P. Hurni, G.Wagenknecht, M. Anwander, and T. Braun. A Testbed Management System for Wireless Sensor Network Testbeds (TARWIS). pages 33–35. European Conference on Wireless Sensor Networks (EWSN) Demo Session, Coimbra, Portugal, February 2010.

[76] Philipp Hurni, Benjamin Nyffenegger, Torsten Braun, and Anton Hergenroeder. On the accuracy of software-based energy estimation techniques. In *Wireless Sensor Networks*, pages 49–64. Springer, 2011.

[77] Shweta Jain and Samir R Das. Exploiting path diversity in the link layer in wireless ad hoc networks. *Ad Hoc Networks*, 6(5):805–825, 2008.

[78] Xiaofan Jiang, Prabal Dutta, David Culler, and Ion Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 186–195. ACM, 2007.

[79] Xiaofan Jiang, Joseph Polastre, and David Culler. Perpetual environmentally powered sensor networks. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 463–468. IEEE, 2005.

[80] JN5148 IEEE 802.15.4 Wireless Microcontroller.

[81] David B Johnson, David A Maltz, Josh Broch, et al. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 5:139–172, 2001.

[82] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 6(4):32, 2007.

[83] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000.

[84] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The one simulator for dtn protocol evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*.

[85] Young-Bae Ko and Nitin H Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.

[86] David Kotz and Tristan Henderson. Crawdad: A community resource for archiving wireless data at dartmouth. *Pervasive Computing, IEEE*, 4(4):12–14, 2005.

[87] K.V.Jonsson. A gateway for wireless dissemination of delay-tolerant content. In *aster Dissertation*, Janurary 2008.

[88] Peter Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):47–54, 2001.

[89] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks. pages 115–148. Ambient Intelligence Part II, Springer Verlag, USA, 2004.

[90] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: Accurate and scalable simulation of entire tinyos applications. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137. ACM, 2003.

[91] Qinglan Li, Jonathan Beaver, Ahmed Amer, Panos K Chrysanthis, Alexandros Labrinidis, and Ganesh Santhanakrishnan. Multi-criteria routing in wireless sensor-based pervasive environments. *International Journal of Pervasive Computing and Communications*, 1(4):313–326, 2005.

[92] Y. Li, W. Chen, and Z.L. Zhang. Optimal forwarder list selection in opportunistic routing. In *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*, pages 670–675. IEEE, 2009.

[93] Qilian Liang and Qingchun Ren. Energy and mobility aware geographical multipath routing for wireless sensor networks. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1867–1871. IEEE, 2005.

[94] Gang Lu, Bhaskar Krishnamachari, and Cauligi S Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 224. IEEE, 2004.

[95] Mingming Lu and Jie Wu. Opportunistic routing algebra and its applications. In *INFOCOM 2009, IEEE*, pages 2374–2382. IEEE, 2009.

[96] Behnam Malakooti and Ivan Thomas. A distributed composite multiple criteria routing using distance vector. In *Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on*, pages 42–47. IEEE, 2006.

[97] T.Braun M.Heissenbutel and M.Walchli. Blr: beacon-less routing algorithm for mobile ad hoc networks. In *Computer Communication Journal*, 2004.

[98] Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, and Omprakash Gnawali. Routing without routes: The backpressure collection protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 279–290. ACM, 2010.

[99] S. Morgenthaler, T. Braun, Zhongliang Zhao, T. Staub, and M. Anwander. Uavnet: A mobile wireless mesh network using unmanned aerial vehicles. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1603–1608, Dec 2012.

[100] Simon Morgenthaler, Torsten Braun, Zhongliang Zhao, Thomas Staub, and Markus Anwander. Uavnet: A mobile wireless mesh network using unmanned aerial vehicles. In *3rd International Workshop on Wireless Networking for Unmanned Autonomous Vehicles*, Anaheim, USA, December 2012.

[101] Mirco Musolesi and Cecilia Mascolo. A community based mobility model for ad hoc network research. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 31–38. ACM, 2006.

[102] Mirco Musolesi and Cecilia Mascolo. Car: context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, 8(2):246–260, 2009.

[103] V. A. Siris N. Gazoni, V. Angelakis and B. Raffaele. Pe-wasun '10: Proceedings of the 7th acm workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. In *A framework for opportunistic routing in multi-hop wireless networks*.

[104] Asis Nasipuri, Ryan Burleson, Benjamin Hughes, and Johnny Roberts. Performance of a hybrid routing protocol for mobile ad hoc networks. In *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*, pages 296–302. IEEE, 2001.

[105] NS2 simulator.

[106] NS3 simulator.

[107] O.R.Helgason and K.V.Jonsson. Opportunistic networking in omnet++. In *SIMUTools '08: Proceedings of the 1st International Conference on Simulation Tools and Techniques*.

[108] Chulsung Park, Kanishka Lahiri, and Anand Raghunathan. Battery discharge characteristics of wireless sensor nodes: An experimental analysis. *power*, 20:21, 2005.

[109] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 1, pages 70–74. IEEE, 2000.

[110] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM Computer Communication Review*, volume 24, pages 234–244. ACM, 1994.

[111] Charles E Perkins and Elizabeth M Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pages 90–100. IEEE, 1999.

[112] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.

[113] Rajmohan Rajaraman. Topology control and routing in ad hoc networks: A survey. *SIGACT News*, 33:60–73, 2002.

[114] Ram Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 404–413 vol.2, 2000.

[115] Venugopalan Ramasubramanian, Zygmunt J. Haas, and Emin Gün Sirer. Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '03, pages 303–314, New York, NY, USA, 2003. ACM.

[116] Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. Prentice Hall PTR New Jersey, 1996.

[117] Tobias Reusing. Comparison of operating systems tinyos and contiki. *Sens. Nodes-Operation, Netw. Appli.(SN)*, 7, 2012.

[118] D. Rosario, Z. Zhao, C. Silva, E. Cerqueira, and T. Braun. An omnet++ framework to evaluate video transmission in mobile wireless multimedia sensor networks. In *Proceedings of the 6th International Workshop on OMNeT++*, Cannes, France, March 2013.

[119] Denis Rosário, Rodrigo Medeiros Costa, Helder Paraense, Kássio Machado, Eduardo Cerqueira, and Torsten Braun. A smart multi-hop hierarchical routing protocol for efficient video communication over wireless multimedia

sensor network. In *2nd IEEE International Workshop on Smart Communication Protocols and Algorithms (ICC'12 WS - SCPA)*, Ottawa, Canada, 2012.

[120] Denis Rosário, Zhongliang Zhao, Torsten Braun, Eduardo Cerqueira, and Aldri Santos. Opportunistic routing for multi-flow video dissemination over flying ad-hoc networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium and Workshops on a*, June 2014.

[121] Elizabeth M Royer, P Michael Melliar-Smith, and Louise E Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 3, pages 857–861. IEEE, 2001.

[122] Eric Rozner, Jayesh Seshadri, Yogita Ashok, and Mehta Lili Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks.

[123] T.L. Saaty. *Fundamentals of decision making and priority theory with the analytic hierarchy process*. RWS Publications, 2000.

[124] Juan A Sanchez, Rafael Marin-Perez, and Pedro M Ruiz. Boss: Beacon-less on demand strategy for geographic routing inwireless sensor networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–10. IEEE, 2007.

[125] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys (CSUR)*, 37(2):164–194, 2005.

[126] S.Chachulski. Trading structure for randomness in wireless opportunistic routing. In *Proceeding of ACM SIGCOMM, 2007.*

[127] Seventh Framework Programme FP7 - Information and Communication Technologies. Wireless Sensor Networks Testbed Project (WISEBED), FP7 Project 2008-2011.

[128] S Son, B Blum, T He, and J Stankovic. Igf: A state-free robust communication protocol for wireless sensor networks. Technical report, Citeseer, 2003.

[129] Ahmet Soylu, Patrick De Causmaecker, and Piet Desmet. Context and adaptivity in context-aware pervasive computing environments. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*, pages 94–101. IEEE, 2009.

[130] Diane Tang and Mary Baker. Analysis of a local-area wireless network. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 1–10. ACM, 2000.

[131] Texas Instruments CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver.

[132] Texas Instruments Support Forum.

[133] V. Mazzola M. A. R Dantas U. Correa, C. Montez. Ifip international federation for information processing, 2006. In *Frad-Hoc: A Framework to Routing AD-Hoc Networks*.

[134] U.S. Census Bureau. Tiger, tiger/line and tiger-related products.

[135] George L Usselman. Frequency shift keying, February 8 1949. US Patent 2,461,456.

[136] Y. Zhang V. Kawadia and B. Gupta. Mobisys '03: Proceedings of the 1st international conference on mobile systems, applications and services. In *Master Dissertation*, Janurary 2008.

[137] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM, 2003.

[138] A. Varga. The OMNeT++ Discrete Event Simulation System. pages 319–324. European Simulation Multiconference (ESM), Prague, Czech Republic, June 2001.

[139] Anthony Wood, John A Stankovic, Gilles Virone, Leo Selavo, Zhimin He, Qiuhua Cao, Thao Doan, Yafeng Wu, Lei Fang, and Radu Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *Network, IEEE*, 22(4):26–33, 2008.

[140] Mingjun Xiao, Jie Wu, and Liusheng Huang. Community-aware opportunistic routing in mobile social networks. 2013.

[141] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 70–84, New York, NY, USA, 2001. ACM.

[142] Guang-Zhong Yang and Magdi Yacoub. Body sensor networks. 2006.

[143] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576. IEEE, 2002.

[144] Jiaxi You, Dominik Lieckfeldt, Frank Reichenbach, and Dirk Timmermann. Context-aware geographic routing for sensor networks with routing holes. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6. IEEE, 2009.

[145] Ossama Younis and Sonia Fahmy. An experimental study of routing and data aggregation in sensor networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[146] S. Yousefi, M.S. Mousavi, and M. Fathy. Vehicular ad hoc networks (vanets): Challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766, June 2006.

[147] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

[148] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*, pages 154–161. IEEE, 1998.

[149] Z. Zhao and T. Braun. Real-world evaluation of sensor context-aware adaptive duty-cycled beaconless opportunistic routing. In *submitted to IEEE LCN 2014*.

[150] Z. Zhao and T. Braun. Scad: Sensor context-aware adaptive duty-cycled beaconless opportunistic routing for wsns. In *submitted to IEEE Globecom 2014*.

[151] Z. Zhao, T. Braun, D. Rosario, and E. Cerqueira. Caor: Context-aware adaptive opportunistic routing in mobile ad-hoc networks. In *7th IFIP Wireless and Mobile Networking Conference (WMNC 2014)*, May 2014.

[152] Z. Zhao, T. Braun, D. Rosario, E. Cerqueira, R. Immich, and M. Curado. Qoe-aware fec mechanism for intrusion detection in multi-tier wireless multimedia sensor networks. In *1st International Workshop on Wireless Multimedia Sensor Netrworks (WMSN, WoMob'12)*, Oct. 2012.

[153] Z. Zhao, B. Mosler, and T. Braun. Performance evaluation of opportunistic routing protocols: A framework-based approach using omnet++. In *ACM 7th Latin America Networking Conference (LANC 2012)*, 2012.

[154] Z. Zhao, D. Rosario, T. Braun, and E. Cerqueira. Context-aware opportunistic routing in mobile ad-hoc networks incorporating node mobility. In *IEEE*

Wireless Communications and Networking Conference (WCNC 2014), April 2014.

[155] Zhongliang Zhao and Torsten Braun. Omnet++ based opportunistic routing protocols simulation: A framework. In *Proceedings of the 10th Scandinavian Workshop on Wireless Ad-hoc Networks*, 2011.

[156] Zhongliang Zhao and Torsten Braun. Topology control and mobility strategy for uav ad-hoc networks: A survet. In *Joint ERCIM eMobility and MobiSense Workshop, WWIC 2012*, page 27, 2012.

[157] Zhongliang Zhao, Denis Rosário, Torsten Braun, Eduardo Cerqueira, Hongli Xu, and Liusheng Huang. Topology and link quality-aware geographical opportunistic routing in wireless ad-hoc networks. In *9th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC 2013)*.

[158] Zhongliang Zhao, Denis Rosário, Torsten Braun, Eduardo Cerqueira, Hongli Xu, and Liusheng Huang. Topology and link quality-aware geographical opportunistic routing in wireless ad-hoc networks. In *9th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC 2013)*.

[159] Z. Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 441–450, 2007.

[160] Zifei Zhong, Junling Wang, Srihari Nelakuditi, and Guor-Huar Lu. On selection of candidates for opportunistic anypath forwarding. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(4):1–2, 2006.

[161] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 125–138, New York, NY, USA, 2004. ACM.

[162] Liang Zhou, X Naixue, Lei Shu, Athanasios Vasilakos, and Sang-Soo Yeo. Context-aware middleware for multimedia&# xd; services in heterogeneous networks. 2010.

[163] M. Zorzi and R.R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. *IEEE Transactions on Mobile Computing*, 2(4):337 – 348, Oct.-Dec. 2003.

List of Publications

Refereed Papers (Journals, Conferences, Workshops)

- Z. Zhao and T. Braun, “Real-World Evaluation of Sensor Context-aware Adaptive Duty-cycled Beaconless Opportunistic Routing,” in *39th IEEE Conference on Local Computer Networks (LCN)*, September 8-11, 2014, Edmonton, Canada.

- D. Rosario, Z. Zhao, T. Braun, E. Cerqueira, and A. Santos, “Opportunistic Routing for Multi-flow Video Dissemination over Flying Ad-Hoc Networks,” accepted by *15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2014)*, Sydney, Australia, June 16-19, 2014.

- Z. Zhao T. Braun, D. Rosario, and E. Cerqueira, “CAOR: Context-aware Adaptive Opportunistic Routing in Mobile Ad-hoc Networks,” in *7th IFIP Wireless and Mobile Networking Conference (WMNC 2014)*, Vilamoura, Algarve, Portugal, May 20-22, 2014.

- Z. Zhao, D. Rosario, T. Braun, and E. Cerqueira, “Context-aware Opportunistic Routing in Mobile Ad-hoc Networks Incorporating Node Mobility,” in *IEEE Wireless Communications and Networking Conference (WCNC 2014)*, Istambul, Turkey, April 6-9, 2014.

- Z. Zhao, D. Rosario, T. Braun, and E. Cerqueira, “Topology and Link Quality-aware Geographical Opportunistic Routing in Wireless Ad-hoc Networks,” *9th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC 2013)*, Cagliari, Sardinia - Italy, July 1-5, 2013, ISBN 978-1-4673-2478-6.

- D. Rosario, Z. Zhao, A. Santos, T. Braun, and E. Cerqueira, “A Beaconless Opportunistic Routing Based on a Cross-Layer Approach for Efficient Video Dissemination in Mobile Multimedia IoT Applications,” *Elsevier’s Computer Communications Journal*, doi = ”http://dx.doi.org/10.1016/j.comcom.2014.04.002.

- D. Rosario, Z. Zhao, T. Braun, and E. Cerqueira, “A Cross-layer QoE-based Approach for Event-Based Multi-tier Wireless Multimedia Sensor Networks,”

to appear *International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS), 2014*.

- D. Rosario, Z. Zhao, T. Braun, E. Cerqueira, A. Santos, and Z. Li, "Assessment of a Robust Opportunistic Routing for Video Transmission in Dynamic Topologies," in *IFIP Wireless Days conference (WD 2013)*, Valencia, Spain,November 13 - 15, 2013, pp.1-6.

- D. Rosario, Z. Zhao, C. Silva, E. Cerqueira, and T. Braun, "An OMNeT++ Framework to Evaluate Video Transmission in Mobile Wireless Multimedia Sensor Networks," in *6th International Workshop on OMNET++*, Cannes, France, March 5, 2013, ISBN 978-1-4503-2464-9.

- D. Rosario, P. Lima, K. Machado, E. Cerqueira, Z. Zhao, T. Braun, "Demo Abstract: Disseminating WMSN Data by Using Social Network and Web," in *10th European Conference on Wireless Sensor Networks (EWSN 2013)*, Ghent, Belgium, February 13, 2013.

- S. Morgenthaler, T. Braun, Z. Zhao, T. Staub, and M. Anwander, "UAVNet: A Mobile Wireless Mesh Network Using Unmanned Aerial Vehicles," in *3rd International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles, collocated with IEEE Globecom 2012*, Anaheim, CA, USA, December 3 - 7, 2012.

- D. Rosario, R. Costa, H. Paraense, K. Machado, E. Cerqueira, T. Braun, and Z. Zhao, "A Hierarchical Multi-hop Multimedia Routing Protocol for Wireless Multimedia Sensor Networks," in *Journal of Network Protocols and Algorithms - Special Issue on 2nd IEEE Smart Communication Protocols and Algorithms'*, Vol. 4, Nr. 4, December, 2012, pp. 44-64, Macrothink Institute.

- Z. Zhao, T. Braun, D. Rosario, E. Cerqueira, R. Immich, and M. Curado, "QoE-aware FEC Mechanism for Intrusion Detection in Multi-tier Wireless Multimedia Sensor Networks," in *1st International Workshop on Wireless Multimedia Sensor Networks (WMSN), collocated with IEEE WiMob 2012*, Barcelona, Spain, October 8 - 10, 2012, pp. 689-696.

- Z. Zhao, B. Mosler, and T. Braun, "Performance Evaluation of Opportunistic Routing Protocols: A Framework-based Approach using OMNeT++," in *ACM 7th Latin America Networking Conference*, Medellín, Colombia, October 4 - 5, 2012.

- Z. Zhao, and T. Braun, "Topology Control and Mobility Strategy for UAV Ad-hoc Networks: A Survey," in *Joint ERCIM eMobility and MobiSense Workshop, collocated with WWIC 2012*, Santorini, Greece, June 8, 2012, pp. 27-32.

- Z. Zhao, and T. Braun, “OMNeT++ based Opportunistic Routing Protocols Simulation: A Framework,” in *10th Scandinavian Workshop on Wireless Ad-hoc Networks (ADHOC 2011)*, Stockholm, Sweden , May 10 - 11, 2011.

Unrefereed Papers (Technical Reports, Project Deliverables)

- D. Rosario, Z. Zhao, E. Cerqueira, T. Braun, and A. Santos, “A Link Quality and Geographical-aware Routing Protocol for Video Transmission in Mobile IoT,” Universität Bern, Institut für Informatik und angewandte Mathematik, Bern, Switzerland, March 28, 2013, IAM-13-001.

- S. Fekete, T. Braun, Z. Zhao, and et. al., “Library of Algorithms for Wireless Sensor Networks,” WISEBED Deliverable D3.4, May, 2011.

- S. Fekete, Z. Zhao, and et. al., “Second set of Algorithms Implementation and Library Integration,” WISEBED Deliverable D3.2, June, 2010.

Submitted Papers (Currently Under Review)

- Z. Zhao and T. Braun, “SCAD: Sensor Context-aware Adaptive Duty-cycled Beaconless Opportunistic Routing for WSNs,” Submitted to IEEE Globecom 2014.

Acknowledgment

I would like to thank Dr.Amir Darehshoorzade, a former member of the Computer Science Department in Politecnico University of Catalunya (UPC), for his sharing of the code to calculate the candidate list of different opportunistic routing protocols. Special thanks also go to Dr.Rafael Marin Perez , a former member of the Department of Information and Communications Engineering, University of Murcia, for his sharing of the code of the BOSS protocol. Another thank should go to Prof.Congduc Pham, affiliated in the LIUPPA laboratory of University of Pau, who provides the code of wvsn-castalia simulation model, which is the basis of our proposed wireless multimedia sensor networks simulator. In the last, I would greatly thank Dr. Imad Aad from our CDS research group, for his patient proofreading of this thesis and many amendment suggestions.

Curriculum Vitae

Personal Details

Name	Zhongliang Zhao
Date of Birth	Feburary 6, 1983
Address	Rodtmattstrasse 99
	CH-3014 Bern, Switzerland
Hometown	Hefei City, AnHui Province, People's Republic of China
Nationality	Chinese

Education

2009.10 – 2014.06	Ph.D. student in Computer Science, University of Bern, Switzerland
2006.09 – 2008.08	Master of Science in Telecommunication Networks, Politecnico di Torino, Italy
2002.09 – 2006.07	Bachelor of Science in Computer Science, Southeast University, Nanjing, China