

Routing and Broadcasting in Ad-Hoc Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Marc Heissenbüttel
von Frutigen

Leiter der Arbeit:
Prof. Dr. T. Braun
Institut für Informatik und angewandte Mathematik

Routing and Broadcasting in Ad-Hoc Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Marc Heissenbüttel
von Frutigen

Leiter der Arbeit:
Prof. Dr. T. Braun
Institut für Informatik und angewandte Mathematik

Von der Philosophisch-naturwissenschaftlichen Fakultät
angenommen.

Bern, den 16. Juni 2005

Der Dekan:
Prof. Dr. P. Messerli

Preface

The following work was performed during my employment as research and lecture assistant at the Institute of Computer Science and Applied Mathematics (IAM) of the University of Bern. The research was conducted within a long term research project of the Swiss National Science Foundation called the National Center of Competence in Research - Mobile Information and Communication Systems (NCCR-MICS).

I would like to thank Prof. Dr. Torsten Braun, head of the Computer Network and Distributed Systems group (RVS), for supervising this work and for his insightful advises. Prof. Dr. Torsten Braun encouraged and motivated me to publish my research results and he provided me the opportunity to present the work on various conferences, for which I thank him.

I would also like to thank Prof. Dr. Roger Wattenhofer, responsible for the Koreferat of this work. Also, Prof. Dr. Oscar Nierstrasz who was willing to be the co-examinator of this work deserves many thanks.

Many thanks go to my colleagues of the RVS group and of the IAM for our various interesting discussions about all kinds of topics and for making the institute a very pleasant and friendly place to work at. Special thanks go to David Steiner, Marc Steinemann, Matthias Scheidegger, Florian Baumgartner, Ruy De Oliveira, and Attila Weyland.

There are many students who worked with me and helped a lot in developing and implementing. Among them I especially thankful to Thomas Bernoulli, Markus Wälchli, David Jörg, Thomas Huber, and Tobias Roth. I would also like to thank our secretary Ruth Bestgen and our system administrators Peppo Brambilla and Roland Balmer for making things a lot easier.

And finally, I am deeply grateful to my relatives and friends for having a great time together during the time of the thesis: Sibyl Cherpes-Heissenbüttel, Marlies Heissenbüttel, Jimmy Cherpes, Hans U. Schmid, Marc-André Mittermayer, Marc Danzeisen, Stephan Zürcher, Andreas Bochsler.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Problem Statement	2
1.3	Beaconing	4
1.4	Beacon-Less Routing (BLR)	4
1.5	Dynamic Delayed Broadcasting (DDB)	6
1.6	Ant-Based Mobile Routing Architecture (AMRA)	7
1.7	Contributions	8
1.8	Thesis Outline	9
2	Ad-Hoc Networks	10
2.1	Introduction	10
2.2	Ad-Hoc Networks	10
2.2.1	Overview	10
2.2.2	Characteristics of Ad-Hoc Networks	11
2.2.3	Examples of Ad-Hoc Networks	12
2.3	Routing Protocol Considerations	14
2.4	Network, Mobility, and Radio Propagation Models	15
2.4.1	Network Models	16
2.4.2	Mobility Models	18
2.4.3	Radio Propagation Models	20
3	Related Work	24
3.1	Introduction	24
3.2	Topology-Based Routing Protocols for Ad-Hoc Networks	25
3.2.1	Introduction	25
3.2.2	Proactive Routing Protocols	25
3.2.3	Reactive Protocols	26
3.2.4	Hybrid Protocols	27
3.3	Position-Based Routing Protocols for Ad-Hoc Networks	28
3.3.1	Introduction	28
3.3.2	Positioning	28
3.3.3	Basic Concepts	30
3.3.4	Examples of Position-Based Routing Protocols	35
3.3.5	Discussion	38
3.3.6	Beacon-Less Protocols	40
3.4	Link Incidents and Beaconing	40
3.5	Broadcasting in Ad-Hoc Networks	42

3.5.1	Introduction	42
3.5.2	Taxonomy	43
3.5.3	Discussion	46
3.6	Ant-Based Routing Protocols	47
3.6.1	Introduction	47
3.6.2	Ant Colony Optimization	48
3.6.3	Ant-Based Routing in Fixed, Wired Networks	49
3.6.4	Ant-Based Routing in Ad-Hoc Networks	52
3.6.5	Conclusions	55
3.7	Conclusions	55
4	Beaconing in Position-Based Routing Protocols	57
4.1	Introduction	57
4.2	Effects of Beaconing	58
4.2.1	Direct Effects	58
4.2.2	Indirect Effects	59
4.3	Analytical Evaluation	60
4.3.1	Probability Density Function of the Speed	61
4.3.2	Relative Speed of Two Nodes	62
4.3.3	Size of Uncovered Area	64
4.3.4	Probability of Outdated Entries in Neighbor Table	64
4.4	Simulations	66
4.4.1	Parameters and Scenarios	66
4.4.2	Optimal Position-Based Routing	68
4.4.3	Fixed Beacon Intervals	70
4.4.4	Adaptive Beacon Intervals	71
4.4.5	Receiver Threshold	74
4.4.6	Estimation of Link Availability	75
4.4.7	Reactive Beaconing	76
4.5	Conclusions	77
5	Beacon-Less Routing (BLR)	79
5.1	Introduction	79
5.2	BLR Protocol Operation	79
5.2.1	Greedy Mode	80
5.2.2	Backup Mode	83
5.2.3	Unicast Mode	85
5.2.4	Reactive Local Routing (RLR)	87
5.2.5	Options	88
5.3	Analytical Evaluation	89
5.3.1	Probability for at Least One Potential Forwarder	89
5.3.2	Expected Number of Hops before Greedy Mode Fails	89
5.3.3	Density Function for the Progress of One Node	91
5.3.4	Expected Progress	92
5.3.5	Distance Until Greedy Routing Fails	96
5.4	Simulations	96
5.4.1	Parameters and Scenarios	96
5.4.2	Evaluating Different Parameter Values	97
5.4.3	Impact of Backup Mode and Unicast Option	102
5.4.4	Comparison with GFG/GPSR	104

5.5	Implementation in a Linux Testbed	109
5.5.1	Implementation	109
5.5.2	Challenges	112
5.5.3	Experiments	114
5.6	Conclusions	118
6	Dynamic Delayed Broadcasting (DDB)	121
6.1	Introduction	121
6.2	Description of DDB	122
6.2.1	Introduction	122
6.2.2	Minimizing the Number of Transmissions	122
6.2.3	Maximizing Network Lifetime	128
6.2.4	Optimizations	129
6.3	Analytical Evaluation	130
6.3.1	DFD Function	130
6.3.2	Expected Size of Covered Area	132
6.4	Simulations	137
6.4.1	Parameters and Scenarios	137
6.4.2	Evaluating Different Versions of DDB	138
6.4.3	Efficiency of Packet Delivery	142
6.4.4	Mobile Networks	144
6.4.5	Congested Networks	146
6.4.6	Irregular Transmission Range	149
6.4.7	Network Lifetime	150
6.5	Conclusions	152
7	Ant-Based Mobile Routing Architecture (AMRA)	153
7.1	Introduction	153
7.2	Ant-Based Mobile Routing Architecture (AMRA)	154
7.2.1	Introduction	154
7.2.2	Overview of Protocol Operation	154
7.2.3	Topology Abstraction Protocol (TAP)	155
7.2.4	Mobile Ant-Based Routing (MABR)	156
7.2.5	Straight Packet Forwarding (StPF)	160
7.2.6	Routing of Data Packets and Ants	160
7.2.7	Example of Routing with AMRA	161
7.2.8	Looping Packets	163
7.3	Simulations	163
7.3.1	Small Network with Uniform Node Distribution	164
7.3.2	Large Network with Irregular Topology	166
7.3.3	Results Obtained from a Java-Simulator	170
7.4	Conclusions	173
8	Conclusions	176
9	Future Work	179
	Bibliography	182
	Acknowledgment	205

List of Figures

2.1	Delaunay triangulation	17
2.2	Gabriel graph	17
2.3	Relative neighborhood graph	17
2.4	Snapshot of restricted random waypoint model	20
2.5	Transmission ranges with the RIM model	23
3.1	Forwarding strategies	30
3.2	Simple network graph defeating MFR and CTD	31
3.3	Triangulation defeating compass routing	32
3.4	Triangulation defeating closest to destination algorithm	32
3.5	Routing on faces	34
3.6	Greedy and perimeter routing	36
3.7	TRR with anchor points	37
3.8	Suboptimal path taken by position-based routing protocol	39
3.9	Ants finding shortest paths	48
3.10	Simple network with pheromone tables of node <i>A</i>	50
3.11	Cost asymmetric paths	51
4.1	Uncovered area	64
4.2	Expected percentage of outdated neighbors	65
4.3	Performance of GFG/GPSR with varying mobility and node density	67
4.4	Comparison of GFG/GSPR, BNU, and BL	69
4.5	Time-based beaconing	70
4.6	Distance-based beaconing	72
4.7	Linear and polynomial mapping of speed to beacon interval	73
4.8	Speed-based beaconing	74
4.9	Rx threshold for beacons	75
4.10	Prediction with time-based beaconing	76
4.11	Reactive beaconing	77
5.1	Forwarding areas	81
5.2	Additional delay vs. progress vs. distance	82
5.3	Routing with BLR in greedy mode	83
5.4	Routing with BLR in backup mode	85
5.5	Routing with BLR in unicast mode	86
5.6	Destination position of PREQs	87
5.7	Using known nodes for unicast	88
5.8	Probability for at least one potential forwarder	89
5.9	Number of hops until greedy mode fails	90

5.10	Upper half of forwarding areas	91
5.11	Upper half of sector	91
5.12	Expected progress for the forwarding areas with a given number of neighbors	93
5.13	Expected progress of forwarding areas	95
5.14	Distance until greedy routing fails	96
5.15	Comparison of different forwarding areas	98
5.16	Comparison of different <i>Max_Delay</i> values	99
5.17	Comparison of DFD functions	100
5.18	Comparison of different packet sizes	101
5.19	Overhead with isotropic and irregular transmission ranges	102
5.20	Impact of backup mode	103
5.21	Impact of unicast mode	104
5.22	Comparison of BLR, GFG/GPSR, BL, and BNU	106
5.23	Delay of BLR, BL, and BNU	106
5.24	Comparison of BLR and GFG/GPSR with irregular transmission ranges	107
5.25	Implementation of BLR in the protocol stack	109
5.26	Running processes in the BLR application	111
5.27	Topologies for the experiments	115
5.28	Chain topology with <i>Max_Delay</i> = 5 ms and <i>Max_Delay</i> = 25 ms	115
5.29	Pairs topology	116
5.30	Contention topology with <i>Max_Delay</i> = 5 ms	117
5.31	Backup topology with <i>Max_Delay</i> = 5 ms	118
6.1	Example of the broadcast algorithm	123
6.2	Additional covered area	124
6.3	Delay introduced by the DFD function	126
6.4	Grid cells	126
6.5	Condition for the DFD function	131
6.6	Analytical, exponential, and linear DFD functions	132
6.7	Expected additional coverage	136
6.8	Expected delay through DFD	137
6.9	Comparison of different DFD functions	139
6.10	Impact of <i>Max_Delay</i>	140
6.11	Impact of rebroadcasting threshold <i>RT</i>	141
6.12	Impact of the different components	143
6.13	Algorithm efficiency	144
6.14	Mobility with 9 neighbors	146
6.15	Mobility with 19 neighbors	147
6.16	Mobility with 49 neighbors	148
6.17	Congested network with 19 neighbors	148
6.18	Congestion in sparse and dense networks	149
6.19	Impact of irregular transmission range	150
6.20	Network lifetime	151
7.1	Architecture of AMRA	155
7.2	Zones relative to logical router in the center	156
7.3	Logical router and its outgoing links	157
7.4	Ants routed with right- and left- hand rule	161

7.5	Ants laying trails over LL_1 and LL_2 to $Z_{3,3}$	162
7.6	Delivery ratio	165
7.7	End-to-end delay	165
7.8	Looping packet on the highway	167
7.9	Results in a large static network	168
7.10	Path of AMRA and GFG/GPSR in irregular topology	168
7.11	Reaction to radical topology changes	169
7.12	Irregular network with varying number of ants	171
7.13	Irregular network with varying data traffic and no explorer packet	172
7.14	Complex network with 10000 nodes and 19 cities	173
7.15	Hop count in a very large network	174

List of Tables

4.1	Expected speed for different v_{min} and v_{max}	63
4.2	Beacon interval at different speeds	73
5.1	Comparison of BLR and GFG/GPSR in highway scenario	108
5.2	Comparison of BLR and GFG/GPSR in highest speed scenario .	108
7.1	Routing table	158
7.2	Parameters	164
7.3	Parameters for large networks	166

Chapter 1

Introduction

1.1 Overview

In this thesis, we study broadcast and routing protocols for ad-hoc networks. We are especially interested in protocols that make use of location information. We propose three novel position-based protocols for ad-hoc networks, two routing protocols and a broadcast protocol.

Ad-hoc networks consist of a collection of wireless hosts that are free to move randomly. These networks operate without the support of any fixed infrastructure or centralized administration and are completely self-organizing and self-configuring. Nodes are connected dynamically and in an arbitrary manner to form a network, depending on their transmission ranges and positions. Nodes can communicate directly with all nodes within transmission range. As transmission ranges are limited, two nodes may not be able to communicate directly and they must rely on other nodes to forward their packets. A source sends a packet to one or more of its neighbors, which in turn forward the packet to their neighbors, and so on until the destination is finally reached. Thus, nodes must cooperate to provide connectivity and paths are normally multihop.

Routing deals with finding appropriate paths between source and destination nodes, possibly over many intermediate nodes. Traditional routing protocols for fixed wired networks are not adequate for ad-hoc networks and perform poorly because of these networks' distinct characteristics such as the rapidly changing topology, the broadcast propagation medium, and the existence of unidirectional links. Many routing protocols designed for ad-hoc networks have been proposed in the literature. We can distinguish between topology-based and position-based routing protocols. Like routing protocols in the Internet, topology-based routing protocols use routing tables and information about available links to forward packets based on the destination address. These topology-based routing protocols are adequate for many kinds of ad-hoc network such as networks with only a few hundred nodes and spontaneous networks where people meet at a convention center and want to share data. On the other hand, these routing protocols show poor adaptation in networks with frequently changing topology. These changes result in slow convergence behavior or even inconsistent routing tables. Furthermore, these protocols scale poorly with a very large number of nodes as the signaling traffic and the number of required control packets

becomes prohibitive.

In position-based routing protocols (also known as geographical, geometric, or location-based routing protocols), the nodes' geographical positions are used to make routing decisions. A node forwards a packet to the neighbor that is geographically closest to the destination position in a greedy manner. If this greedy routing fails, the packet is forwarded further in a recovery mode. Therefore, a node must be able to determine its own position and the position of the destination node. This information is generally provided by a global navigation satellite system and a location service, respectively. The location service is responsible for maintaining nodes' positions and replying to requests for destination node locations from source nodes. Furthermore, nodes obtain knowledge of their neighbors through beacons, short hello messages broadcasted periodically from each node. Position-based routing algorithms do not require the establishment of any route prior to data transmission, and nodes neither keep track of installed routes nor store routing tables, as packets are simply sent to neighbors in the direction of the destination, making nearly stateless routing feasible.

These characteristics make position-based routing protocols especially suitable for ad-hoc networks with highly dynamic topologies and/or a large number of nodes, where topology-based protocols have their limitations. Typical ad-hoc networks that have such characteristics are sensor networks and vehicular ad-hoc networks. In sensor networks, the nodes are often tiny and have highly restricted power and resource constraints. Possibly tens of thousands of sensors are distributed densely over a specific area for applications like object tracking and information collecting. In vehicular ad-hoc networks, traffic flows are between vehicles; mobility is normally high and topology changes are frequent. Vehicular networks are envisioned and already deployed to enable on-board safety systems, virtual traffic signs, and real-time information on traffic and congestion.

In the remainder of this first chapter, we first state the problems that are investigated in this thesis. The four following sections correspond with the four main chapters, which describe the work carried out during the thesis. In the two last sections, we briefly summarize the main contributions and give a short outline of the thesis.

1.2 Problem Statement

Even though position-based protocols are more suited for these kinds of networks with highly dynamic topologies and a large number of nodes than topology-based protocols, they still have some drawbacks. These drawbacks can be broadly classified in two categories.

1. Drawbacks caused by the required control traffic.
 - The periodical broadcasting of hello messages, called beaconing, is a proactive mechanism of position-based protocols and is performed independently of actual data traffic. Even if no data is being transmitted, nodes constantly exchange beacons, which wastes scarce network resources such as battery power and bandwidth. This may be

a critical factor, especially in sensor networks with very limited devices. Furthermore, the beacons can interfere with data packets that must then be retransmitted, introducing additional delay.

- Even though the protocols are often claimed to be nearly stateless as they do not have to maintain paths, the protocols are still stateful in the sense that nodes must store local information about the network, namely the positions of their neighboring nodes as provided by beacons. This information may become outdated and inaccurate because of mobility, or nodes toggling between active and sleep modes. Stale neighbor information leads to wrong routing decisions, which may cause significant network performance degradation. For example, a node may try to forward a packet to a node that is no longer reachable.

2. Drawbacks originating from using the positions and distances of nodes as the only criteria for routing.

- Routing a packet along the line-of-sight between the source and destination may often not be possible in realistic networks. Thus, greedy routing of position-based protocols will fail and a recovery mechanism must be applied. The path chosen by this recovery mechanism may be very suboptimal and have a much higher hop count.
- Each packet is sent completely independent of all others, i.e., if the protocol forwards them along a very long path even though a much shorter one exists, all subsequent packets follow the longer path. The algorithm has no way to adapt and to learn from experience.
- Packets are routed only on the basis of location information, and normally the length of the traveled path is minimized. Other criteria like delay, link capacity, and traffic load are not taken into account. Thus, the protocols do not account for end-to-end performance and Quality-of-Service as the shortest path may not be the appropriate one because of congestion and long delays.

We can summarize these drawbacks by saying that protocols are stateful concerning neighborhood topology and stateless about the topology of the network on a large scale. This is however exactly the opposite to what seems intuitive and logical. The neighborhood changes frequently and at unpredictable times and, thus, protocols should avoid to maintain state about the local network topology. On the other hand, the overall node distribution in the network remains quite static and only varies slowly over time, e.g., because people tend to stay in towns and move along streets. Therefore, it is beneficial to accumulate such information at the nodes to facilitate communication on a large scale.

Each of the three position-based protocols proposed in this thesis aim to overcome one of these two types of drawbacks. One of the proposed routing protocols and the broadcast protocol address the first type of drawbacks caused by the required control traffic and the associated local statefulness. The second proposed routing protocol targets the second type of drawbacks, which is caused by the sole use of position information for forwarding decisions and the statelessness about global network topology.

1.3 Beaconing

Although some shortcomings of existing protocols caused by beacons were discussed in the previous section, we also carried out an assessment of the possible impacts by theoretical analysis and simulations. Thus, we provide a more formal justification for the proposed stateless routing and broadcast protocols, which avoid beacons and do not suffer from these drawbacks.

Beaconing is the proactive broadcasting of short hello messages to advertise a node's position to its neighbors. Knowledge of the neighboring node's position is required by conventional position-based protocols to make forwarding decisions. Commonly, a node assumes that a link to a neighbor exists if it has received a beacon from that neighbor within a certain time-interval. Beaconing uses scarce network resources and is also a major source of wrong routing decisions, as the stored network states can become stale, and thus the topology perceived by a node may be different from the actual physical network topology. The view of the network topology is, strictly speaking, always outdated if nodes move, as the positions indicated by the beacons do not correspond to the actual positions. The most severe impact occurs if a node tries to forward a packet to a node that it considers a neighbor but the node has left transmission range. We show by analytical and simulation results that these wrong routing decisions and outdated neighbor information can significantly degrade the performance of the network. We also propose and evaluate some obvious extensions to existing position-based proposed routing protocols to mitigate to observed drawbacks. The main findings of this chapter can be summarized as follows:

- Packet loss in uncongested networks is mainly due to outdated neighbor information and wrong routing decisions.
- The delay can be increased by an order of magnitude or more in highly dynamic networks.
- Already minor extension can significantly improve the performance of position-based protocols. However the fundamental flaw of these protocols remain, namely their statefulness.

1.4 Beacon-Less Routing (BLR)

The goal of the design of BLR was to avoid completely the periodical broadcasting of hello messages. Forwarding decisions should be based solely on the information given in the data packet itself, and no additional control messages should be required. Thus, BLR is a real stateless protocol, which eliminates the drawbacks of beaconing mentioned above.

BLR consists of four components; greedy mode, backup mode, unicast mode, and reactive local routing. Packets are routed in greedy mode whenever possible. Greedy routing is performed in a distributed manner without any information about neighboring nodes, neither their positions nor even their existence. Because nodes do not rely on information about neighbors, beaconing can be disposed completely. If a node has a packet to send, it simply broadcasts the packet and every neighboring node receives it. The protocol ensures that just one of the receiving nodes relays the packet further. This is accomplished by

computing a Dynamic Forwarding Delay (DFD) at each node depending on its position relative to the previous and to the destination node. The node located at the most “optimal” position introduces the shortest delay and thus transmits the packet first. Other nodes overhear the further relaying and cancel their scheduled transmissions of the same packet. To ensure that all nodes detect the forwarding, only nodes within a certain forwarding area apply DFD and take part in the contention to forward the packet. Nodes outside the forwarding area just drop the packet. To the best of our knowledge, BLR is the first routing protocol where the forwarding decision is taken not at the sender or source node of a packet but at the receivers. The broadcasting of packets over all hops makes BLR susceptible to packet duplication. Packet duplication occurs for each node in the forwarding area that does not detect a subsequent rebroadcasting of previously transmitted packet. In realistic environments, many factors may prevent nodes from receiving successfully a packet transmitted by a node within transmission range such as interferences, obstacles, and the error prone wireless propagation medium. In order to reduce the number of broadcast transmissions of greedy mode, and thus the duplicated packets, BLR has an option to forward packets in unicast mode if neighbors are known.

If no node is located within the forwarding area, greedy routing fails. The node that was not able to relay the packet in greedy mode routes the packet further in backup mode. Therefore, the node broadcasts a request for beacon packets. All nodes that receive this packet reply with a beacon indicating their position. The packet is then forwarded to the replying node that is closest to the destination. If none of the neighbors is closer to the destination than the requesting node, the packet is routed according to the “right-hand” rule, a concept known for traversing mazes. As soon as the packet arrives at a node closer to the destination than where it entered backup mode, the packet switches back to greedy routing.

The intended position of the destination, as indicated in the packet header, may not correspond with the current position of the destination node because of mobility. To cope with position inaccuracies, reactive local routing is applied in the vicinity of the destination if the packet cannot be delivered to the destination. Therefore, the packet is flooded within a restricted area around the thought position of the destination node. If the destination has not moved too far away, there is a high probability of delivering the packet.

Many further optimizations for BLR apart from the above three components are presented later in this thesis. The implementation and evaluation of all optimizations is outside the scope of this thesis. We focus on the ones we believe are most promising. BLR is first studied by analytical means and then the performance and behavior are evaluated by simulations. Furthermore, we implemented BLR in a Linux testbed consisting of laptop computers equipped with WLAN-cards and GPS devices, and performed several real-world experiments. Considering the drawbacks of existing position-based protocols that require beaconing, we can summarize the main benefits of BLR as follows:

- BLR eliminates control traffic completely as no hello messages are required in greedy mode. This conserves scarce resources such as battery power and bandwidth in ad-hoc networks. Control traffic is only required if the greedy mode fails.
- BLR is almost completely stateless and does not store network topology

information. Thus, it remains almost unaffected by even very high rates of topology change and proves highly scalable in terms of the number of nodes.

- BLR performs routing on the actual topology and thus minimizes the risk of wrong routing decisions because of outdated topology information, which would significantly degrade performance.

1.5 Dynamic Delayed Broadcasting (DDB)

The principle of Dynamic Delayed Forwarding (DFD) was also used to design a broadcast protocol named Dynamic Delayed Broadcasting (DDB). It has the same objectives as BLR, such as the elimination of any control traffic to save scarce resources, and to be stateless in order to avoid outdated network topology information.

The broadcasting of packets throughout the network is a frequently performed task in ad-hoc networks. Optimized broadcasting is achieved by dynamically, and in a completely distributed way, delaying transmissions at the receiving nodes by DFD and determining whether to rebroadcast a packet at all. Depending on the metric used to calculate the delay, DDB can be optimized for many objectives. We propose two schemes that aim to reduce the number of overall transmissions and to extend the network lifetime, respectively. In the first scheme, the delay introduced by DFD at each node depends on the additional area that would be covered by the node's transmission. Larger additional covered areas yield smaller additional delays. Consequently, nodes at the boundary of the sender's transmission range, which cover a significant additional area, rebroadcast the packet first. Nodes close to the sender, which introduce a large delay, overhear the subsequent transmissions of the distant nodes. There is a high probability that these nodes will not rebroadcast the packet at all as their transmission would not cover any additional area, or only a very small one. The second scheme extends the network lifetime by conserving power at the nodes with almost depleted batteries. Nodes delay the packet forwarding depending on their battery status. Nodes with an almost depleted battery schedule the rebroadcasting of the packet with a large delay whereas nodes with high remaining battery power forward the packet almost immediately. Again, nodes do not rebroadcast a packet if they cannot cover any additional area. Consequently, energy is conserved at almost depleted nodes, which increases their lifetime and in turn prolongs the connectivity of the network. Both DDB schemes can also operate without location information, if incoming signal strength can be used to approximate distances between nodes. First, some general properties of DDB are derived analytically, and then simulations are conducted to evaluate its performance. As DDB is based on the same concept of DFD, it shows also benefits similar to BLR.

- DDB does not require any control traffic. Rebroadcast decisions are taken in a completely localized manner and are locally optimal without any knowledge of the neighborhood.
- As DDB is nearly stateless, it remains unaffected by frequently changing topologies.

- DDB can easily be optimized for different objectives by using different metrics.

1.6 Ant-based Mobile Routing Architecture (AMRA)

The ant-based mobile routing architecture (AMRA) is a two-layered framework with three independent protocols rather than an actual routing protocol. AMRA also makes use of position information available at each node for routing. Packets are routed towards the destination not greedily but perhaps to several intermediate anchor positions, thus avoiding the aforementioned second type of drawback of conventional position-based routing. It is designed for routing in large-scale ad-hoc networks with possibly tens of thousands of nodes and irregular topologies, where routing along a line-of-sight between the source and destination is not possible because of mountains or lakes in between. AMRA is also able to cope with mobility of nodes as long as the overall distribution of the nodes remains rather static. For most realistic scenarios, this is reasonable assumption as nodes are typically located in towns and on/along streets. In such scenarios, AMRA is able to find more optimal paths by memorizing past traffic such that packets are not routed directly towards the destination, as this may be very suboptimal. The memory required to keep track of the traffic can be kept quite small by applying an aggregated and fisheye-like view onto the network.

AMRA is designed with modularity in mind such that each protocol can be replaced individually and almost independently of the others or with only minor modifications. Three specific protocols are presented exemplarily for the AMRA framework. The two protocols used on the upper layer are called Topology Abstraction Protocol (TAP) and Mobile Ant-based Routing Protocol (MABR). TAP provides a simplified logical network topology to the actual routing protocol MABR. MABR uses a paradigm from *swarm intelligence* to achieve optimal routing. StPF (Straight Packet Forwarding) is situated on the lower layer and acts as an interface for MABR to the physical network.

TAP provides in a transparent manner a simplified topology with fixed *logical routers* and fixed *logical links*. Logical routers are defined as fixed geographical areas. All nodes located within a logical router have the same logical view of the network. Each node maintains a routing table, which depends on its current view and its past locations and overheard packets. The rationale for this abstraction of the actual network topology is twofold. First, to scale well in large networks, and second, to provide a rather static topology to the actual routing protocol MABR.

MABR operates in the upper layer on top of this abstract topology and is not required to respond to frequently changing topologies. It maintains probabilistic routing tables at each node, which are updated by data packets and ants. Ants are special control packets that are routed independently of the data packets and explore new paths. Unlike ants, data packets are routed according to these pheromone tables. At each node, the destination coordinates of the data packets determine the intermediate anchor position to which the packet is sent on its way to the destination. Each node overhearing or forwarding a data packet

or an ant updates its routing table by increasing the probability of the path followed by the packet.

StPF is a purely position-based routing protocol and basically any existing protocol can be used as StPF, including BLR as proposed in this thesis. StPF is applied to forward a data packet physically over the logical link determined by MABR to the next anchor point, i.e., data packets are routed to the next logical router purely on the basis of nodes' positions. When the packet arrives at this logical router, MABR must again determine the next logical router. StPF is also the only means by which ants are routed to their destination.

The framework with the three described protocols is evaluated by simulations. AMRA has the following main features.

- AMRA allows nodes to learn by memorizing past traffic such that disadvantageous paths are avoided. Packets are routed along paths with high node density and high-quality paths.
- Because of the abstract topology, it can cope easily with high mobility and is scalable in terms of the number of nodes and the geographical area covered by the network.
- Not only can it take into account position of and distances between nodes, but also different cost metrics.

1.7 Contributions

We can summarize the main contributions of this thesis as follows:

- The impact of beaconing on position-based routing was evaluated by analytical means and by simulations.
- We designed a position-based routing protocol (BLR) and a broadcasting protocol (DDB) for ad-hoc networks. These protocols do not require any neighbor information and are completely stateless.
- We designed the scalable and modular AMRA framework for routing in ad-hoc networks, based on ideas from swarm intelligence and comprising three single protocols.
- We studied analytically some fundamental properties of BLR and DDB and implemented them in a network simulator.
- AMRA was also implemented in a network simulator to assess its performance in large-scale ad-hoc networks.
- BLR was also implemented in a testbed of Linux laptop computers equipped with GPS devices and WLAN cards and several real-world experiments were conducted.

1.8 Thesis Outline

In Chapter 2, we give a basic overview of ad-hoc networks and their characteristics and also discuss network and simulation models used in this thesis to evaluate the proposed protocols. In Chapter 3, we give a comprehensive overview of related work covering routing and broadcasting protocols for ad-hoc networks. Then, we study the impact of beaconing on position-based routing protocols and identify several drawbacks in Chapter 4, such as stale routing information and waste of network resources. To avoid some of these observed drawbacks, we propose the Beacon-Less Routing protocol (BLR) in Chapter 5, which avoids the beaconing mechanism of other position-based routing protocols. Unlike all other routing protocols, forwarding decisions are taken at the receiving nodes in a completely distributed manner by making use of the broadcast property of the wireless medium. We introduce the Dynamic Delayed Broadcasting protocol (DDB) in Chapter 6, based on the same principles as BLR to achieve optimized broadcasting without any control message overhead. The decision of whether or not to rebroadcast a packet is taken completely locally, depending only on the information given in the broadcast packet. In Chapter 7, we introduce the Ant-based Mobile Routing Architecture (AMRA). If routing along a straight line between the source and destination is not possible because of voids in the topology, conventional position-based routing protocols perform very suboptimally. AMRA employs ideas from swarm intelligence to find optimized routes in such scenarios by increasing the probability of shorter paths. Finally in Chapter 8, we summarize the main findings and conclude the thesis. In the last Chapter 9, we discuss further improvements to the proposed protocols and also outline more generally possible future directions for research.

Chapter 2

Ad-Hoc Networks

2.1 Introduction

This chapter explains some basics regarding ad-hoc networks and the way protocols are evaluated and judged commonly in the research community.

Ad-hoc networks have several salient characteristics that make them quite different from other networks. The perhaps two most notable are the dynamic topology and the absence of any centralized infrastructure. These characteristics pose large challenges for protocols on any layer of the network stack. (Mobile) ad-hoc networks are often cited explicitly in the literature. However, the paradigm of ad-hoc networking has also been applied in various other contexts and distinct names for these networks emerged. We briefly summarize some of the well-known networks based on the ad-hoc networking paradigm. We start this chapter by discussing the characteristics of ad-hoc networks and giving a general overview of networks relying on the ad-hoc networking paradigm in Section 2.2.

In Section 2.3, we consider design and evaluation criteria especially aimed at protocols for ad-hoc networks. As protocols are typically evaluated first by simulations, one has to take care that the parameters defining the simulation model capture many of the possible scenarios encountered in reality. Furthermore, the performance and behavior of the protocols have to be judged objectively by well-defined metrics in order to be able to compare them among each other.

For the theoretical analysis and the simulations, different models have been proposed to reproduce reality. Many models are quite simple and allow to derive some general properties easily, while others try to reflect reality more accurately. In Section 2.4, we discuss various important network, mobility, and radio propagation models, which are used for the simulations and the analytical evaluations of ad-hoc network protocols.

2.2 Ad-Hoc Networks

2.2.1 Overview

All kinds of wireless networks have become increasingly popular over the past years. Due to the continuous technological advances, today's wireless and portable devices are small, light-weight, and have high computing capabilities.

The proliferation of these devices and the demand of users to communicate continuously are the driving forces behind the deployment of wireless networks. We can distinguish between two fundamentally different types of wireless networks. The first are known as infrastructure or cellular networks in which mobile hosts communicate with base stations connected to a fixed network infrastructure. If a node moves out of range of a base station, a handoff to a new base station within transmission range occurs. Data is routed through the wired network and only the last hop is wireless. The second type of wireless networks are ad-hoc networks, often also-called wireless multihop networks, which is probably the more descriptive term. These networks are simply formed by a collection of wireless hosts, which are often free to move randomly, and without any established infrastructure or centralized administration. Naturally, these networks must be self-configuring and self-organizing. Since source and destination nodes may not be within transmission range, the paths are normally multihop. All hosts act as routers and forward packets on behalf of other nodes to provide communication throughout the entire network.

Ad-hoc networks have several distinct advantages over infrastructure networks for many applications and in many scenarios. Some of these reasons are their ease and speed of deployment, their robustness, as they do not depend on any infrastructure, and the low costs. Ad-hoc networks may also be used in combination with cellular networks to form so called hybrid or multihop cellular networks. Hybrid networks can help to extend the coverage and to increase the redundancy and the performance of cellular networks.

The increased interest in ad-hoc networks is also reflected by the formation of the “Mobile Ad-hoc Network” working group within the Internet Engineering Task Force IETF [120]. This working group currently focuses on routing protocols suitable for wireless routing applications. Many protocols have been proposed as Internet Drafts and some reached RFC-status. The Internet Research Task Force IRTF [121] has also established a research subgroup “Ad-hoc Network Systems” that investigates some specific areas in the context of ad-hoc networks like inter-layer protocol interaction, Quality-of-Service routing, routing scalability, and network auto-configuration.

Ad-hoc is Latin and can be literally translated as “for this”. The meaning is “for this purpose only”. The term (mobile) ad-hoc network also is often used as an umbrella term for any kind of self-organizing, infrastructureless, and wireless multihop networks. In this thesis, we follow this definition and often refer to these kinds of networks simply as ad-hoc networks.

2.2.2 Characteristics of Ad-Hoc Networks

Ad-hoc networks have several distinct characteristics that make them quite different from conventional cellular networks. These characteristics are mainly due to the absence of any fixed infrastructure, the wireless propagation medium, and the limited resources of the mobile devices. The most salient characteristics which should be considered by protocol designers were addressed within the “Mobile Ad-hoc Networks” working group and are listed in RFC 2501 [160].

- **Dynamic topologies:** Major reasons for topology changes are the mobility of the nodes, the adjustment of transmission and reception parameters,

and the sleep cycles of nodes to save energy. Therefore, the network topology may change frequently and at unpredictable times.

- **Limited and variable capacity of the links:** The capacity of wireless links is and will presumably remain considerably lower than of wired links. Furthermore, the capacity of the links may vary over time because of the changing propagation conditions and the varying distances between nodes.
- **Power-constrained operation:** Nodes in an ad-hoc network typically rely on batteries for their operation. Even though, battery capacity has doubled in energy density every 35 years [198], it still does not satisfy today's demands. Unfortunately however, a breakthrough is not expected in the near future [83, 225].
- **Limited physical security:** Mobile wireless networks are generally more prone to security threats than fixed wired or cellular networks [266]. The wireless propagation medium introduces vulnerabilities to malicious attacks varying from passive eavesdropping to active interference. Unlike cellular networks, ad-hoc networks do not have a centralized administration that can act as a trusted third party.

Protocols designed for fixed and cellular networks are not appropriate to cope with these characteristics and new protocols tailored especially for ad-hoc networks are required. The protocols that we propose in this thesis are mainly concerned with the first three characteristics and implement mechanisms to operate efficiently in such scenarios. They do not address security issues as this is out of scope of this thesis.

2.2.3 Examples of Ad-Hoc Networks

In the following, a brief description of some typical networks which are based on the paradigm of ad-hoc networks is given. These networks are often not pure ad-hoc networks, but rather hybrid networks as some nodes are connected to infrastructure networks. Most of these terms are not strictly defined, vary over time, or are used in different context by different authors. We try to give the most common used description of these different terms as they are used in our view nowadays.

- **Mesh networks:** Mesh networks may be used as a last mile solution where cabling is impossible, too expensive, or just as an alternate infrastructure in the event of failure [26]. Nodes are deployed densely all over a certain area in order to enable broadband wireless access from home. The term is also used in the context of wireless metropolitan area networks, where wireless hotspots are interconnected to offer users wireless access. Several companies [156, 171] have already deployed such networks in various cities. Today, most often IEEE 802.11b [115] is used as the underlying wireless technology. It is likely that new standards like the different variants of IEEE 802.16 [117] and IEEE 802.20 [118], which offer higher data rates and higher transmission ranges, will further boost this development and will complement and/or partially replace IEEE 802.11b networks.

- **Wireless personal area networks:** A personal area network is a computer network used for communication among computer devices close to one person, including telephones and personal digital assistants. The reach of a personal area network is typically a few meters only and can be used for communication among the personal devices themselves and also for connecting to the Internet. Several commercially available wireless technologies like Bluetooth [16] and other technologies which are currently under standardization in the IEEE 802.15 [116] Working Group are typically used for these wireless personal area networks.
- **Vehicular ad-hoc networks:** Vehicular ad-hoc networks are used for on-board safety systems, virtual traffic signs, real-time congestion and traffic information, and commercial applications which require vehicle-to-vehicle or vehicle-to-roadside networking [242, 97, 138]. Vehicular ad-hoc networks have some distinct features compared to other 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.
- **Sensor networks:** Low power and energy efficient radios and processors have made all types of sensor networks a reality [3]. The tasks of the sensor networks are including object tracking, information collecting and querying, or producing a response to a certain event. The data collected and often already partially processed by the sensors is transmitted to a sink node that communicates with a monitoring center. The number of nodes in sensor networks can be several orders of magnitude higher than in the previously discussed ad-hoc networks. Sensor nodes are very limited devices and have strict power, communication, computation, and memory constraints. Furthermore, sensor networks mainly use a broadcast or geocast communication paradigm, whereas most ad-hoc networks are based on point-to-point and unicast communication. For sensor networks, the power conservation may be the most important design parameter.
- **Spontaneous networks:** Spontaneous networking has got a lot of attention recently [67]. Spontaneous networking can be described as the integration of services and devices with the objective that services are offered instantaneously to users without any manual intervention [199]. To achieve this objective, these networks also have to account for effective service discovery among devices [205].
- **Military and rescue networks:** In scenarios like disaster rescue and military operations, one cannot rely on centralized administration or the availability of a communication infrastructure. The existing networks may be destroyed, may not be reliable in enemy regions, or just may not be accessible. To facilitate the operation, it is important to be able to deploy quickly a communication infrastructure. Consequently, ad-hoc networks have already been used early in military operations.
- **Packet radio networks:** The idea of self-organizing wireless networks is not really new. Already back in the 1970's, a multihop multiple-access packet radio network [130] was developed under the sponsorship of the

Defense Advanced Research Projects Agency (DARPA). Packet radio networks are somehow synonymous to ad-hoc network even though mobility was not a major concern, as devices were heavy and not very mobile.

2.3 Routing Protocol Considerations

Routing deals with finding appropriate paths between source and destination nodes, possibly over many intermediate nodes. Depending on the underlying communication paradigm, we distinguish between unicast, multicast, anycast, geocast, and broadcast routing. All these types of routing have one sender, but they differ in the number of destination nodes and the way the destination nodes are determined. In unicast communication, there is exactly one specific destination node. In anycast routing, packets are delivered to exactly one destination among several possible destinations. Multicast is the delivery to multiple destinations which are aggregated in a multicast group. If the multicast group is defined as the set of all nodes within a specified geographical region, we speak of geocasting. Broadcast aims at delivering packets to all nodes in the network. In this thesis, we are mainly concerned with unicast routing protocols and if not noted otherwise, we simply refer to unicast routing protocols as routing protocols.

Traditional routing protocols for fixed wired networks such as RIP [163] and OSPF [175] are not adequate for the characteristics encountered in ad-hoc networks and perform poorly. The challenge of any routing protocol for ad-hoc networks is that they must be able to cope efficiently with their salient characteristics.

RFC 2501 [160] describes differences between ad-hoc and fixed wired networks and discusses the resulting impact on the design and evaluation of network control protocols, focusing on routing protocols. An ad-hoc network is characterized by several defining parameters that should be considered during the design, the simulation, and the comparison of routing protocols. A networking context or a scenario is defined as a set of characteristics describing an ad-hoc network and its environment. In the literature, these parameters are typically used and varied to determine the performance of protocols in different network scenarios. In the following, we briefly describe some of these parameters.

- **Network size:** This is simply measured as the number of nodes which are member of the ad-hoc network.
- **Network connectivity:** Network connectivity is measured by the average number of neighbors of a node and depends on the node density and the transmission range. This parameter is also referred to as the degree of a node.
- **Topological rate of change:** The rate with which the network topology is changing, i.e., the rate with which links break and new links come up. These changes are not only caused by mobility. Nodes toggling into and out of sleep states, interferences with other transmissions, and changing propagation characteristics can also cause changes in the topology.
- **Fraction of unidirectional links:** Unidirectional links may be the rule and not the exception caused by varying transmission ranges, different

SINR (Signal to Interference plus Noise Ratio), etc. Thus the effectiveness and efficiency of protocols in the presence of unidirectional links may be crucial.

- **Traffic patterns:** This includes all different kinds of connection types and the distribution of the traffic load. Connections may be short- or long-lived, data may be transmitted in bursts or constantly and smoothly, some nodes may send/receive more traffic than others.

Quantitative and qualitative metrics are also proposed in RFC 2501 [160] in order to judge and compare ad-hoc network routing protocols. Some of them apply also to routing protocols for fixed, hardwired networks, whereas others are more specific for ad-hoc networks. As stated in RFC 2501, it is crucial that the metrics are independent of any specific routing protocol. The most often used quantitative metrics are the following:

- **End-to-end data throughput and delay:** These metrics are the two most important statistical measures of the effectiveness of routing performance and may include, e.g., mean, variance, and distribution.
- **Packet delivery ratio:** The effectiveness is often not only measured as the absolute throughput, but also as the fraction of successfully received data packets at the destination and transmitted packets at the source.
- **Percentage out-of-order delivery:** This measure is of particular importance to certain transport and application layer protocols such as TCP [197] and RTP [216] which prefer in-order delivery of packets as sent by the source.
- **Overhead:** This may be viewed as an internal measure of the routing protocol's effectiveness, often also called normalized routing load. Depending on the efficiency, a certain amount of overhead is required to achieve a certain level of data routing performance. The overhead can be measured in bits, but often it is measured as the average number of control and data packets transmitted per data packet delivered. This measure tries to capture a protocol's channel access efficiency. This is of special importance with contention-based MAC layer protocols such as the DCF (Distributed Coordination Function) of IEEE 802.11b [115], where the cost of channel access may be disproportional high for short control packets.

2.4 Network, Mobility, and Radio Propagation Models

In this section, we give an overview of network, mobility, and propagation models. Network models are mainly used for theoretical analysis and evaluation of algorithms. The mobility and radio propagation models are used for simulations to obtain wireless hosts movement patterns and to describe accurately radio characteristics such as path loss, interference, and fading, respectively.

2.4.1 Network Models

Wireless network are often modeled as graphs with nodes as vertices in the plane and edges between two nodes if they are within transmission range, i.e., if a link exists between the nodes. At any given point in time, the network graph depends on factors such as the nodes' positions, the transmission power levels and coverage patterns, and the SINR, which all determine the communication range. Thus the network graphs may change with time if any of these parameters change. Given a set of vertices in the plane, different network graphs differ in the set of edges between the vertices.

By far the most common graph to model wireless networks and approximate the actual network is the unit disk graph. Its simplicity is the reason for its wide spread use which eases the theoretical analysis. In the unit disk graph, there is an edge between two vertices if and only if the Euclidean distance between them is at most a certain fixed transmission range. The transmission range is constant for all nodes and is scaled to 1. The unit disk graph has the property that edges are undirected and, thus, all existing links are bidirectional. Furthermore, it inherently presumes a single communication channel, e.g. as for IEEE 802.11 [115] and omni-directional antennas, which may not always be the case in realistic scenarios. Other examples for network graphs include the minimum power graph [7, 112, 229], the quasi unit disk graph [148], etc., which try to model reality more accurately.

It was however also argued in [65, 226] that these models derived from wired network models with vertices and edges might not be an accurate representation for a wireless network as they do not account for several distinct peculiarities. For example, the transmissions over wireless links may not be possible if there are ongoing transmissions on other nearby links. Additionally, costs of transmissions are associated with nodes rather than links, unlike in fixed wired networks, as one transmission is received at all neighbors.

Many algorithms depend on some properties of the network graphs to operate effectively and efficiently. Numerous position-based routing algorithms require a planar graph to guarantee loop-freedom for example. A planar graph is a graph which can be drawn with no edges crossing. Obviously, edges may cross in the actual network graph modeled by the unit disk graph (and also in the quasi unit disk or the minimum power graph). Therefore, they do not belong to the class of planar graphs. Often a subgraph with certain properties such as planarity is extracted from the actual network graph, which includes only a subset of all edges E . We briefly give the definition of certain graphs, which all are planar and connected, used in related work as described in Section 3.3.

- A convex subdivision is a graph such that each face of the graph is a convex polygon, except the outer face which is the complement of a convex polygon.
- In a triangulation, every face is a triangle, again except the outer face.
- The Delaunay triangulation is a triangulation in which there is a triangle of edges between three points U, V, W if and only if the disk determined through them contains no other points, cp. Fig. 2.1.
- In the Gabriel graph, two points U and V are interconnected by an edge if and only if there is no other point in the circle between these two points

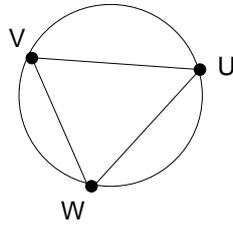


Figure 2.1: Delaunay triangulation

centered at their middle point and whose diameter is equal to the distance between them, cp. Fig. 2.2.

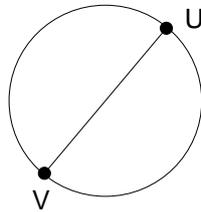


Figure 2.2: Gabriel graph

- In the relative neighborhood graph, there is an edge between two nodes U and V if and only if there is no other node located either closer to U or V Fig. 2.3.

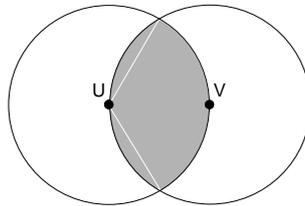


Figure 2.3: Relative neighborhood graph

- The minimum spanning tree consists of a subset of edges of minimum costs, which form a tree.

Another structure that is often encountered in related work and will also be used in Chapter 6 is the (minimum) connected dominating set. A dominating set is a set of nodes such that all nodes are either in the set or a neighbor of a node in the set. The set is connected if any two nodes in the set are connected over a path consisting solely of nodes in the set. The minimum connected dominating set is the connected dominating set with the fewest nodes. Connected dominating sets are frequently used as a backbone, e.g., it is sufficient if all nodes from the minimum connected dominating set broadcast the packet in order to broadcast a packet to all nodes in the network.

An important feature is whether these graphs are locally computable for practical purposes, i.e., whether a node is able to determine with only the knowledge of the position of itself and its one-hop neighbors if a certain edge belongs to the respective graph. Unfortunately, except for the Gabriel graph and the relative neighborhood graph, these graphs are not locally computable. Furthermore, the edges of the planar graphs, the convex subdivisions, and the triangulation are not necessarily a subset of the edges of the actual network graph. Links may be arbitrarily long whereas in the actual network topology the length of the links is restricted to the maximal transmission range. That means in these classes of graphs, links may exist over which communication is not possible because the nodes are too distant from each other. For these two reasons, the Gabriel and the relative neighborhood graphs are the choice for practical position-based routing protocols, whereas results on the other graphs are more of theoretical interest. In view of these considerations, a very important result was derived in [23]. It was proven that the intersection of the Gabriel and the unit disk graph is connected and planar, provided that the unit disk graph is connected. It even contains the minimum energy path for any path loss exponent ≥ 2 .

Many other graphs were proposed that could be used as underlying graphs for position-based routing protocols. [77] introduced the restricted Delaunay graph, where the length of the path between any two nodes is only a constant times the minimum length possible. If position information about nodes is not available, [248] proposed an algorithm that allows local construction of a planar subgraph based on a general order over the neighbors' link qualities. Similarly, [152] proposed a topology control algorithm, which does not use actual position information of nodes, but depends only on directional information.

2.4.2 Mobility Models

Numerous mobility models have been proposed to determine the movement patterns of nodes. By far the most often used is the random waypoint mobility model as introduced below. We also describe the restricted random waypoint mobility model in more detail, which we use in this thesis as well. This model allows for more realistic movement patterns, taking into account topologies with highways and cities. A survey including other common mobility models used in ad-hoc network research can be found in [33, 212].

Random Waypoint Mobility Model

In the random waypoint mobility model, all nodes are randomly placed over the whole simulation area at the beginning. Each node starts moving to a randomly selected destination position in the simulation area with a speed uniformly distributed in the interval between v_{min} and v_{max} . When it arrives at the destination position, it stops for *pause.time*, chooses a new destination position and continues as previously described. A common way to control the mobility, and therefore the rate of topology changes, is by the *pause.time* parameter. The longer a node remains static when it reaches one of its destination positions, the lower the average nodal speed in the network is. It was observed in [18] however that pause times over 20 s result in a rather stable network with few link changes per node even at high speeds. Furthermore, these temporar-

ily static nodes form a backbone-like structure, which can be used for routing. The protocols only apparently have to deal with highly dynamic topology for longer pause times. Consequently, we often use only a pause times of 0 s for the simulations in this thesis in order to have a really challenging scenario.

This simple random waypoint mobility model has also other characteristics that have to be taken into account. First, the nodes are uniformly distributed at the beginning over the whole simulation area. With time, the distribution of the nodes is no longer uniform and the node density is higher in the center than at the borders. The reason is that nodes move in a straight line to the destination position often traversing the center if the destination is located in the opposite half of the simulation area. Secondly, the distribution of the nodes' speed is only uniform in the interval $[v_{min}, v_{max}]$ at the beginning of the simulation. With time, more and more nodes move slower and the distribution nodes' speed approach a distribution inversely proportional to the speed. As this stationary distribution of the random waypoint is different from the distribution at the beginning, the simulations need an initial warm-up phase to reach a stable state. The duration of this warm-up phase is difficult to predict and depends on the parameters of the random waypoint mobility model such as *pause_time*, v_{min} , v_{max} , and also the size of the simulation area. Thus, we implemented the stationary distribution of the random waypoint model as described in [179]. The positions and the speed of the nodes are as in the stationary distribution right from the beginning and no longer uniform as in the original unmodified random waypoint mobility model. Furthermore, it was shown in [268] that for a $v_{min} = 0$, the expected average nodal speed approaches 0. Most authors until now used and unfortunately continue to use $v_{min} = 0$. Therefore these results should be considered with caution. For a long simulation time, the networks become almost static independent of the chosen v_{max} .

The random waypoint mobility model is a simple model that allows investigating how a protocol is affected by mobility in general. However, it is not appropriate for more realistic network topologies. For example, packets always can be routed along a roughly straight line from the source to the destination because there are no voids in the topology. These voids may be caused by obstacles or areas where no nodes are located thus preventing packets to be transmitted in this direction. These voids however, are a big challenge for every routing protocol. Protocols that may perform well with a "uniform" distribution of nodes may perform poorly with highly irregular topologies.

Restricted Random Waypoint Mobility Model

To simulate realistically large networks with irregular topologies, we use the restricted random waypoint mobility model [15]. This model defines rectangular city areas and highways connecting cities. Node movement within a city is according to the standard random waypoint mobility model. Nodes move to one of the cities connected via a highway with a certain probability. The node speed on the highway is higher than for trips within the same city and also defined by a minimum and maximum speed. In reality, most people often move within relatively small geographical areas and only rarely travel long distances to other cities. On the other hand, some people may travel frequently also over long distances between cities such as commuters, couriers, and truck drivers. The restricted random waypoint mobility model tries to capture this behavior

by introducing two kinds of nodes that correspond to the rarely and frequently traveling people called ordinary nodes and commuters, respectively. These two kinds of nodes differ in their frequency to move to another city and their pause time between the trips. Commuters are also used to cause enough traffic on the highways and provide connectivity among the cities. In Fig. 2.4, a snapshot of the restricted random waypoint mobility model with four cities and three interconnecting highways is shown with a typical node distribution after some simulation time. In order to route from city 1 to city 4, the packet has to pass over city 2 and 3 and cannot be routed directly.

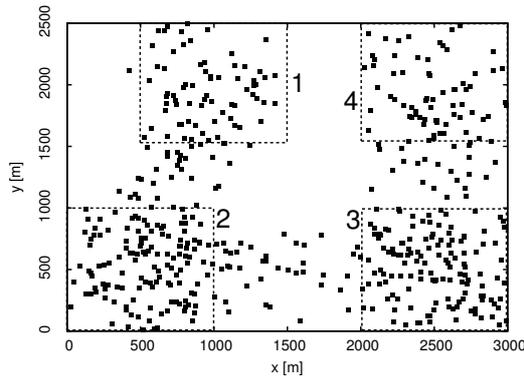


Figure 2.4: Snapshot of restricted random waypoint model

2.4.3 Radio Propagation Models

Network simulators use radio propagation models to calculate the incoming signal strength at the nodes based on various parameters such as transmission power, antenna gain, interferences, wavelength, etc. Many different models have been proposed which differ in their complexity and accuracy. The free space and the two ray ground models are the most widely applied propagation models. They are also the simplest ones and always yield isotropic transmission ranges, i.e., they basically model a unit disk graph. We further describe the radio irregularity model (RIM) which accounts for real world influences that can cause transmission ranges to be highly irregular. A survey of various radio propagation models can be found in [181, 177].

Free Space Model

The free space propagation model assumes that there is only one clear line-of-sight path between the transmitter and the receiver with a minimal path loss factor of 2. The following equation as derived in [72] is used to calculate the received signal power P_r at a distance d from the transmitter

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

where P_t is the transmitted signal power and λ is the wavelength. G_t and G_r are the antenna gains of the transmitter and receiver respectively. The parameter $L \geq 1$ is the system loss and generally set to 1. If omnidirectional antennas are used, G_t and G_r are also set to 1. The transmission radius r is then simply the distance d at which the received signal power $P_r(d)$ equals the receiver sensitivity. The receiver sensitivity is the minimum power required to successfully receive and decode a packet. Nodes within the transmission range r receive all packets. If the distance $d > r$, nodes are not able to receive packets at all.

Two-Ray Ground Reflection Model

The two-ray ground reflection model [263] was introduced to give more accurate predications at long distances than the free space model. This model considers not only the signal received along the line-of-sight but also a signal received along a ground reflection path. The received signal power at distance d is calculated as

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}$$

where h_t and h_r are the heights of the transmitting and receiving antennas, respectively. The two-ray ground reflection model does not give good results for short distances in the vicinity of the transmitter. Thus, the two-ray ground reflection model is only used for distances d larger than a threshold distance d_t , while the free space model is used for distances d smaller than d_t . At d_t , the two models must yield the some receiving power level $P_r(d)$ for a continuous transition. This immediately allows the calculation of the threshold distance d_t .

$$d_t = \frac{4\pi h_t h_r}{\lambda}$$

The two-ray ground reflection model also yields a circular transmission range because the path loss is isotropic, i.e., equal in all directions.

Radio Irregularity Model (RIM)

The radio irregularity model for radio propagation was introduced in [270] and is based on real world measurements to more accurately model the irregularity of transmission ranges. RIM is an extension to an underlying radio model and accounts for main properties of devices and radio signals such as non-isotropic path losses, continuous variation, and heterogeneous signal sending power. Even if the transmission range is highly irregular with RIM, the mean power density at a distance d from the transmitter is the same as in the underlying propagation model. Two parameters are used in the RIM model. *DOI* is used to control the degree of irregularity of the transmission range, while *VSP* is used for the variance of sending power at the devices. When these parameters values are set to 0, the RIM is reduced to the underlying isotropic model. The received signal power P_r is calculated as follows in decibel notation. In this thesis, we recalculate the P_r for a node whenever it has moved 50 m to account for the changed environmental factor.

$$P_r = P_t^{VSP} - PL^{DOI}$$

where P_t^{VSP} and PL^{DOI} are the adjusted sending power and the adjusted path loss, respectively. P_t^{VSP} is used to account for heterogeneous signal sending power due to differences in hardware and battery status and is calculated as

$$P_t^{VSP} = P_t \cdot (1 + rand \cdot VSP)$$

where P_t is sending power level as used in the underlying propagation model and $rand$ is a random variable, which follows a normal distribution. The parameter VSP is defined as the maximum percentage variance of the signal sending power among different devices.

The adjusted path loss PL^{DOI} is used to reflect the two main properties of radio irregularity; non-isotropy and continuous variation. The path loss PL denotes the signal attenuation and is obtained, in decibel notation, from P_t and P_r as

$$PL[dB] = P_t[dBm] - P_r[dBm] = 10 \log \frac{P_t}{P_r}$$

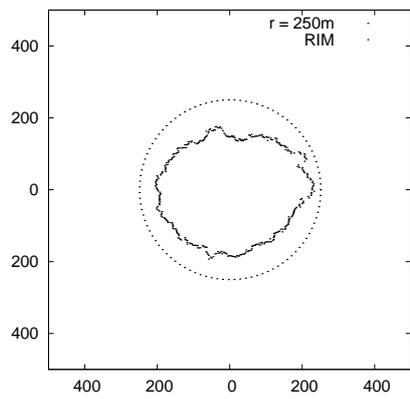
The PL^{DOI} is given by

$$PL^{DOI} = PL \cdot K_i$$

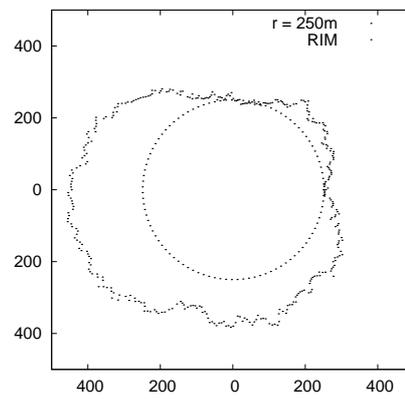
where PL is the path loss as calculated by the underlying propagation model. K_i are the coefficients to represent the difference in path loss in different directions and are calculated in the following way

$$K_i = \begin{cases} 1 & : i = 0 \\ K_{i-1} \pm rand \cdot DOI & : 0 < i < 360 \text{ and } i \in \mathbf{N} \end{cases}$$

which generates random numbers K_i according to the Weibull distribution, where $rand$ is a random variable uniformly distributed in the interval $[0, 1]$. For non-integer angles, we interpolate the K_i value based the two adjacent integer angles. Unlike the free space and two ray ground reflection propagation models, the transmission radius is highly irregular and depends on the direction. In Fig. 2.5, two typical transmission ranges are depicted for a DOI of 0.01 and a VSP of 0.5. These are the most extreme values used in [270]. We can see that the transmission ranges may vary quite a lot among different nodes and with time. The dotted circles indicate the transmission range of the underlying isotropic propagation model, i.e., the two-ray ground reflection model in this case.



(a) Example 1



(b) Example 2

Figure 2.5: Transmission ranges with the RIM model

Chapter 3

Related Work

3.1 Introduction

In this chapter, we give an overview of the related work important for the work carried out in this thesis. The chapter is again divided in a way such that each section corresponds to one of the main four chapters of this thesis.

We start by describing current state of the art routing protocols for ad-hoc networks. The routing protocols can be broadly classified in two categories, namely topology-based and position-based protocols. Topology-based protocols use routing tables and information about available links to forward packets based on the destination address. Since topology-based protocols are not subject of this thesis, only the most well-known topology-based routing are presented in Section 3.2. Position-based routing protocols, on the other hand, use the nodes' geographical positions to make forwarding decisions. Packets are sent to a neighboring node, which reduces the distance to the destination. Nodes progressively forward the packets from one neighbor to the next until the packets reach the destination. We give a comprehensive overview of the applied concepts in position-based routing protocols and describe explicitly some of the more relevant protocols in Section 3.3. In Section 3.4, we describe related work that addresses drawbacks of link-incidents and beaconing on routing protocols, and ways to alleviate their impact, e.g., by selecting reliable neighbors, introducing new metrics, or proactively repairing existing paths. Since no standard broadcast protocol has yet emerged for ad-hoc networks, we give a rather broad overview over all kinds of broadcast protocols in Section 3.5. In Section 3.6, we describe the paradigm of ant colony optimization and how it is applied to routing in fixed wired networks. Then, we discuss ant-based routing protocols for ad-hoc networks, which have been proposed only recently. Finally, Section 3.7 concludes this chapter and summarizes the main results.

3.2 Topology-Based Routing Protocols for Ad-Hoc Networks

3.2.1 Introduction

Topology-based routing protocols make use of information about available links between nodes. This information is then used by the nodes to forward packets. A tremendous number of topology-based protocols have been proposed and thus it is not surprising that taxonomies to categorize them have also been discussed. The perhaps most often employed taxonomy is the classification in proactive, reactive protocols, and hybrid protocols. In proactive protocols, nodes in the network periodically refresh routing information so that nodes have consistent and up-to-date information from each node to every other node in the network at all times. On the other hand, reactive protocols only acquire routes on demand. If a node has to send a packet and does not know a path to the destination, it triggers a route discovery mechanism. Currently, the "Mobile Ad-Hoc Network (manet)" working group within the Internet Engineering Task Force IETF [120] is about to standardize two routing protocols, a Reactive MANET Protocol (RMP) and a Proactive MANET Protocol (PMP), using mature components from previous work on reactive and proactive protocols. Hybrid routing protocols employ proactive and reactive concepts for routing within a local and more global scope, respectively. An overview of topology-based protocols and other classifications can be found in [93, 111, 273].

3.2.2 Proactive Routing Protocols

In proactive protocols, routes are maintained between host pairs at all times by exchanging route information periodically or each time a change occurs in the network topology. Therefore, routes are immediately available if a packet needs to be sent. A shortcoming is the maintenance of unused paths causing large overhead, which wastes scarce network resources. Other issues are scalability and the time required for the algorithms to converge in case of frequent topology changes.

Traditional link-state and distance vector-protocols fall into this category, such as OSPF [175] and RIP [163]. However, these protocols are not designed for the encountered characteristics in ad-hoc networks. Already small inaccuracies in the routing tables can cause disconnections or loops. Furthermore, if topology changes frequently, a storm of link status change messages and triggered updates rises for OSPF and RIP, respectively. In the following sections, we describe in more detail the two proactive protocols for ad-hoc networks, which were upgraded to RFC-status. Other well-known proactive protocols are, e.g., DSDV [194], WRP [176], FSR [190], CGSR [42], and STAR [128].

Optimized Link State Routing (OLSR)

The optimized link state protocol OLSR was proposed in [45] (RFC 3626 [44]). OLSR is a variant of the classical link state protocols with optimizations to meet the requirements of ad-hoc networks. The main difference to traditional link-state protocols is the concept of multipoint relays MPRs, which aims at efficiently distributing topology information by reducing the number of required

link-state packets. MPRs are a minimal set of one-hop neighbors such that all two-hop neighbors are reachable through these MPRs. Broadcast messages are only forwarded by MPRs. This reduces not only the number of transmissions for topology information broadcasts, but also reduces the size of the broadcast packets, since nodes only need to list their MPRs in the link-state messages. OLSR basically consists of three phases; neighbor sensing based on periodic exchange of hello messages, efficient flooding of control traffic using the MPRs, and computation of an optimal route using a shortest-path algorithm.

Topology Broadcast based on Reverse-Path Forwarding (TBRPF)

In [11], Topology Broadcast based on Reverse-Path Forwarding (TBRPF) was described (RFC 3684 [188]) that is similar to OLSR. The main difference to OLSR is that OLSR supports only source trees to its two-hop neighbors whereas in TBRPF each node computes a source tree, which provides paths to all reachable nodes. Each node periodically broadcasts part of its source tree to its neighbors as an update. These updates are not further forwarded but may cause a change in the receiving node's source tree that is again propagated in the next update message. Differential updates are used to minimize the overhead. Neighbor sensing is done with hello messages, which are broadcasted to inform about changes in the neighborhood topology. Each node is also able to report additional topology information to improve robustness in order to support highly mobile networks.

3.2.3 Reactive Protocols

In reactive protocols, the computation of a path is performed only on-demand. The source initiates route discovery when a path to the destination is needed in order to transmit user data. The advantages of reactive protocols are the power and bandwidth efficiency compared to proactive protocols. However, this point has to be reconsidered because results in [5, 264] indicate that battery power consumption is about the same for proactive and reactive protocols. This is due to the fact that just listening to the medium is almost as costly as receiving a packet with today's devices. The main drawback is the long latency until a route is acquired and established between source and destination. We briefly summarize the reactive protocols AODV, which already has RFC-status, and DSR, which is expected soon to be upgraded to RFC-status. Other reactive protocols include TORA [189], ABR [238], FRESH [62], OLIVE [123].

Ad-Hoc On-Demand Distance Vector (AODV) Routing

AODV was proposed in [195] (RFC 3561 [196]). When a node has to send a packet to a destination for which it does not have a valid route, the node broadcasts a route request message. Each node forwards the route request message and caches the node from which it received the message. If the destination receives a route request message, it sends a route reply back to the originator of the request message establishing a bidirectional route between source and destination node. Consequently, nodes are not aware of the whole path, but only of the immediate next hop towards the destination and the source. Intermediate nodes may also generate a route reply message if they know a route to

the destination. Route Error messages are used to notify other nodes of link breaks in existing routes. AODV uses the concept of sequence numbers to avoid the formation of loops. Each destination includes a monotonically increasing sequence number with each route information it sends to a requesting node. A node that has two different routes to a destination always has to use the one with the larger sequence number.

Dynamic Source Routing Protocol (DSR)

The Dynamic Source Routing Protocol (DSR) [129] is currently available as an Internet draft, but is expected to become an RFC later. DSR 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 receives 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 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 if they know a route to the destination. Route maintenance is used to detect if a link along a route is broken. When route maintenance indicates that a route is broken, the source node can either use another route it knows or invoke route discovery again.

3.2.4 Hybrid Protocols

Hybrid protocols apply principles of proactive routing for the local neighborhood and reactive routing for distant nodes, respectively, for the following two reasons. First, changes in the topology are only important for nodes close-by and have little impact on nodes on the other side of the network. Secondly, most communication takes place between nodes that are close to each other. We briefly describe perhaps the most well-known hybrid protocol ZRP, which was available as an Internet Draft [92] and is planned to be proposed as RFC. Other hybrid protocols include SHARP [203] and LANMAR [191].

Zone Routing Protocol (ZRP)

ZRP [96] is not really a routing protocol per se, but rather a framework for hybrid routing. It is composed of the Intrazone Routing Protocol (IARP [91]), the Interzone Routing Protocol (IERP [90]), and the Bordercast Resolution Protocol (BRP [89]), where the intrazone and interzone routing protocols can basically be any proactive and reactive routing protocol, respectively. Each node defines its zone as all nodes that are within a certain number of hops, called zone radius. The intrazone routing protocol is used by a node to communicate with the nodes in its zone whereas interzone routing protocol is applied to detect routes to nodes in other zones. If a node requests a route to a distant node, a route request is issued and forwarded by the border resolution protocol. This protocol optimizes the forwarding of the request by making use of the fact that

nodes have knowledge of the nodes in their zones. When a node receives a request for a destination, it first checks if the destination is located within its zone. If not, the node forwards the route request but only to its peripheral node by using the border resolution protocol, i.e., the nodes that are at the border of its zone, which have a minimal hop count equal to the zone radius from the current node. Lately, the authors of ZRP proposed the Independent Zone Routing (IZR) [208], an enhancement of the ZRP framework, which allows adaptive and distributed configuration for the optimal size of each node's routing zone radius on a per-node basis.

3.3 Position-Based Routing Protocols for Ad-Hoc Networks

3.3.1 Introduction

Position-based routing protocols forward packets based on the nodes' physical locations. Forwarding decisions are based solely on the position of the current node, the positions of neighboring nodes, and the position of the destination. A node that wants to forward a packet to a destination node chooses one of its neighbors as a next hop according to some criterion like the one closest to the destination. When the packet reaches a dead end where this simple greedy routing fails, a recovery mechanism is applied. Unlike topology-based protocols, position-based protocols require only little control traffic and are nearly stateless. Position-based routing protocols do not require the establishment or maintenance of routes and thus eliminate the overhead of frequent topology updates and route acquisitions of topology-based routing protocols. For these reasons, they are generally considered scalable and more robust to changes in the network topology than topology-based protocols. Assuming that location information is available, these characteristics make them a preferred choice for large and highly dynamic networks

Many position-based routing protocols have been proposed in the literature. Unlike for topology-based protocols, only very few taxonomies for position-based routing protocols have been proposed and none of these are widely accepted. We start by describing how position information can be acquired in ad-hoc networks by the nodes. Then, we summarize basic concepts applied in position-based routing algorithms and briefly describe several protocols based on these concepts. In a next section, we describe in more detail the two most relevant routing protocols for our work. At the end, we discuss the main advantages and shortcomings of position-based routing protocols. Surveys of position-based routing protocols can be found in [71, 82, 227, 167].

3.3.2 Positioning

Position-based routing protocols require very little position information about the network to forward packets; namely knowledge of the position of the current node, the destination, and the neighbors is needed.

1. **Current Node:** The position of the current node can be provided in several ways. The most common and easiest way relies on GPS [170]

or its European counterpart Galileo [25] to provide location information. The availability of small and cheap GPS receivers that operate on low power provides justification for applying position-based routing in ad-hoc networks. Even sensor nodes equipped with GPS receivers are now available [46]. GPS and Galileo allow nodes to determine longitude, latitude, and altitude with a certain degree of accuracy. For reasons of simplicity, we do not consider the altitude in this thesis, i.e., nodes are located in a two-dimensional plane. We assume an isotropic transmission range with radius r . The accuracy depends on the number of satellites in the line-of-sight of the node and commonly is in the range of some few meters. Furthermore, GPS-technology allows nodes to be synchronized as they can determine the time with an accuracy in the order of 200 *ns*. Differential GPS [173] can be used for more accurate positioning in the order of centimeters.

However, in several scenarios, nodes may not be able to receive GPS-signal or are simply not equipped with GPS-receivers. For such scenarios, many approaches have been proposed to provide relative positions of nodes. They either use received signal strength together with the time of arrival [210], the time difference of arrival [200], or the angle of arrival [184] to estimate nodes' positions. Other methods of determining a node's position are based on radio-location [29] or hello messages from an available fixed infrastructure [28]. Furthermore, several hybrid schemes have been proposed [183] where only a few designated nodes are aware of their positions and all other nodes derive their positions by taking into account the hop count to the designated nodes. Routing without absolute location information has become a hot topic recently, where each node computes its virtual position relative to some other nodes, e.g. [174, 12, 204, 56, 35].

2. **Neighbors:** The positions of the neighboring nodes are obtained through beacons broadcasted periodically by every node. Beacons are small hello packets containing information about the node's current position. Nodes keep track of the positions of other nodes and update their neighbor tables upon receiving a beacon. If a node does not receive a beacon from a specific node within a certain period of time, it assumes that the neighbor is no longer available and removes this node from the neighbor table. The reasons can be manifold such as mobility, sleep cycles, interferences with other transmitting nodes, power switch off, adjustment of transmission and reception parameters, failure.
3. **Destination:** Source nodes are assumed to be able to determine the position of the destination nodes. The source nodes then tag the packets with the destination coordinates. Consequently, all nodes receiving these packets are aware of the coordinates to which the packet has to be sent as well. In this thesis, we limit our focus on position-based routing and simply assume that the location of the destination is known by any means. In some networks and for some applications, the position or direction of the destination may be known in advance or may be given implicitly. For example in vehicular ad-hoc networks, the message may be routed back along the street to alert other drivers of an incident on the road. Sensor networks are typically concerned with events associated with a

geographical location and data requests are sent to nodes within a certain area. Sensor nodes then forward their collected data to a sink. For ad-hoc networks where the position of the destination is not given a priori or inherently, the position can be determined through any location service such as VHR [81], GLS [151], GRSS [113],[232],[95], [1]. A comprehensive survey of location services for ad-hoc networks can be found in [32].

3.3.3 Basic Concepts

In the following, we discuss routing concepts proposed in related work, which show different properties on different kind of network graphs, cf. Section 2.4.1. We say that a graph *defeats* a routing algorithm if there exists a source and destination pair such that a packet never reaches the destination when beginning at the source. Otherwise the algorithm *works* for this kind of graph. In *randomized algorithms*, a neighboring node is chosen randomly among all possible neighbor candidates.

Basic Loop-Free Algorithms

We first discuss basic routing strategies that are inherently loop-free. Many algorithms have been proposed based on the notion of *progress* which was introduced in [180, 112, 237]. Progress is defined as the projection of the distance traveled over the last hop from P to any node A onto the line from P to the final destination D . We speak of forward progress, if the projection of the receiving node is closer to the destination than the previous node (A' , B' , C' in Fig. 3.1). Conversely, we speak of negative progress if the distance to the destination becomes larger (E'). In [237], Most Forward within Radius (MFR)

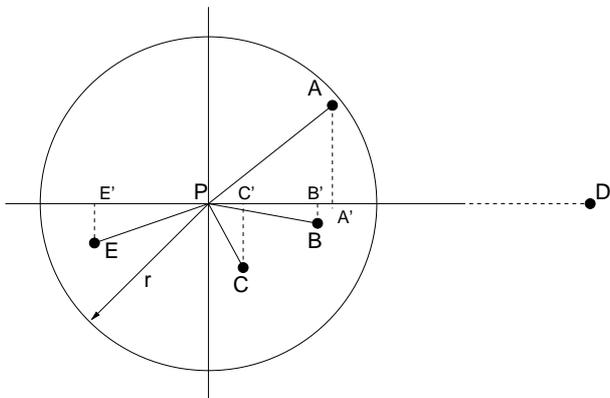


Figure 3.1: Forwarding strategies

was introduced to minimize the number of hops. A node forwards the packet to the node among its neighbors with the largest progress (A in Fig. 3.1). Instead of using the progress as the decision criterion for choosing the next hop, [69] applied a greedy principle where the node closest to the destination (CTD) is chosen as the next hop (node B). This forwarding strategy minimizes the traveled Euclidean distance of a packet. Under the assumption that nodes are able to adjust their transmission power, Nearest with Forward Progress (NFP) was

proposed in [112] in order to minimize the interference with other nodes and the overall power consumption (node C). Similarly [229] proposed to forward packets to the nearest node among the neighboring nodes which is closer to the destination. It was shown in [228] that the greedy forwarding strategies based on progress (MFR) or its complement based on distances (CTD) yield in most cases the same path to the destination and only differ slightly. Therefore, we refer to both strategies as greedy forwarding for simplicity reasons.

These simple greedy algorithms are most efficient if a direct path along a line-of-sight between the source and the destination exists, i.e., they can also operate efficiently for all network graphs, in particular on the unit disk graph. A further advantage is that these algorithms are inherently loop-free. This comes at the cost that they are also defeated by any kind of network graph because they cannot recover from local minima, namely in cases where no neighboring node has forward progress or is closer to the destination. For example in Fig. 3.2, there exists a path from P to D but since no node within the transmission range of P has forward progress or is closer to the destination, the packet reached a local minimum and cannot be routed further.

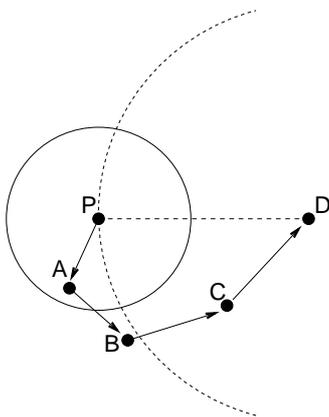


Figure 3.2: Simple network graph defeating MFR and CTD

Basic Non Loop-free Algorithms

Other basic routing algorithms not making use of the notion of progress prevent a packet from reaching a local minimum and cannot be further forwarded. On the other hand, they do no longer guarantee loop-freedom for unit disk graphs but only for certain kinds of subgraphs of the network graph, e.g., the Gabriel graph. That means that not all available links can be used for forwarding.

Compass Routing was introduced in [21], in which a node forwards a packet to the neighboring node minimizing the angle between itself, the previous node, and the destination (e.g., node B in Fig. 3.1). It was shown in [22] that this protocol does not guarantee loop-freedom for unit disk graphs but only on Delaunay triangulations. In Fig. 3.3, a convex subdivision is depicted, but since this is not a Delaunay triangulation a packet can get trapped in a loop if sent from node P towards node D . P forwards the packet to node A minimizing the angle to node D . A in turn forwards the packet to B , which again minimizes

the angle to node D . The packet continues to travel on the outer faces of the graph, eventually arriving again at node P .

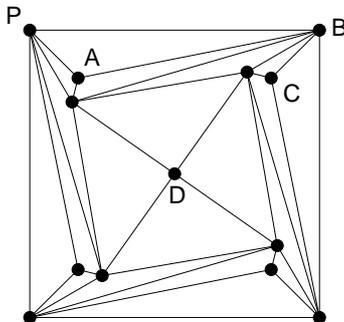


Figure 3.3: Triangulation defeating compass routing

In the routing algorithm of [21], the packet is sent to that neighboring node which is closest to the destination. A node can always forward a packet, however the algorithm does not prevent packets from looping. Consider the network graph in Fig. 3.4 where the packet is sent from P to A and back to P . Note that the circle does not indicate the transmission range but shows the distances of the different nodes to the destination D . It was shown that this routing algorithm works also on Delaunay triangulations.

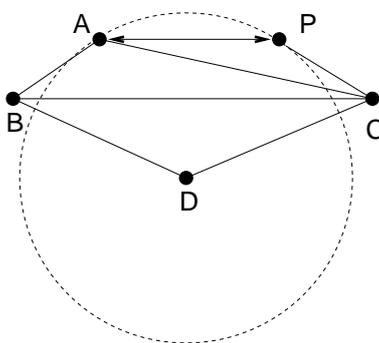


Figure 3.4: Triangulation defeating closest to destination algorithm

Variations of the Basic Algorithms

Many variations, extensions, and combinations of these basic loop-free and non loop-free forwarding strategies have been proposed. We will briefly describe some of them.

In the Random Progress Method (RPM) introduced in [180], any node among the neighbors with forward progress is chosen with equal probability as the next hop. [155] proposed to drop a packet if the best choice for a current node is to return the packet to the node from which it was received. Randomized compass routing chooses randomly from the two neighboring nodes minimizing the angle between itself, the previous node, and the destination in clockwise and counterclockwise order. In greedy compass routing, the node among these two

nodes is selected that minimizes the distance to the destination. Randomized compass routing [21] and greedy compass [22] routing were proven to work for all convex subdivisions and all triangulations, respectively.

In [228] extended knowledge of the neighborhood was used including all one- and two-hop neighbors. A node selects the best candidate node among its one-hop and two-hop neighbors according to the respective criterion. The message is then forwarded to the best one-hop neighbor, which is also a neighbor of the candidate node. Analogously, it was proposed in [45] to extend the knowledge of the neighborhood to more than just the one-hop neighbors, resulting in more optimal routing decisions. [154] proposed to memorize past traffic so that a message is not forwarded twice to the same neighbor. This prevents packets from loops and allows protocols to recover from local minima. In order to detect loops, it is often proposed to store the previous nodes in the packet header in order to avoid a packet being sent back to one of these nodes. The size of packets is increased with each hop to store the followed path. The problem of looping is only solved partially since loops may consist of arbitrary number of hops as it is shown in [21]. Therefore, an “infinite” amount of memory is required to store all previously visited nodes in the header of a packet in order to prevent loops in all cases.

Face Routing

In the seminal work of [143], the concept of face routing was introduced. Face routing works for any planar graph. If combined with a planarization algorithm such as for the Gabriel graph, face routing works for any network graph, i.e., in particular the unit disk graph. The concept of face routing has become the basis of many other position-based routing protocols. The idea is based on the famous right-hand rule to find a way out of a maze. This rule states that when arriving at node A from node B , the next edge traversed is the next one sequentially counterclockwise from the edge connecting A and B . Consequently, the right-hand rule traverses the interior of a face in clockwise edge order. Analogously, an exterior face is traversed in counterclockwise edge order.

The face routing algorithm starts at P in Fig. 3.5 and determines the face F_0 incident to P intersected by the line segment \overline{PD} joining P and D . Then the algorithm starts exploring the edges of this face in clockwise order by the right-hand rule and keeps track of edges traversed that intersect with \overline{PD} . Upon returning to P , the algorithm returns to the intersection closest to the destination traveling again along the boundary of the face. The two phases are marked with “Exploring” and “Forwarding” arrows, respectively. At this point, face routing switches to the face F_1 of the geometric graph and starts traversing the second face. This process is repeated until the destination is reached.

Algorithms Based on Face Routing

In [143], it was also proposed to switch to the next face at the first intersection found at the boundary of the face. This eliminates the need to traverse completely the face before deciding where to switch faces. As face routing is not average case efficient, GFG was presented in [23, 24] and combines face and greedy routing. Greedy routing, where packets are forwarded to the neighbor closest to the destination, is applied as long as there is any neighboring node

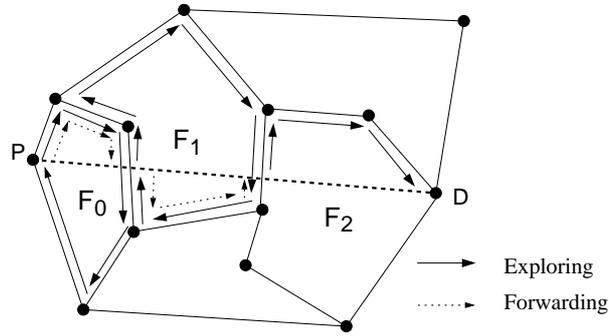


Figure 3.5: Routing on faces

closer to the destination than the current node. Face routing is only applied if the packet can no longer be forwarded with greedy routing. GFG uses the face routing variant where faces are switched at the first encountered intersection. GFG can however not provide asymptotic optimality. The first asymptotically optimal algorithm was AFR proposed in [147], which optimizes face routing, i.e., reduces the number of hops. It avoids routing beyond a certain radius by branching the graph within an ellipse of exponentially growing size. AFR and also the original face routing algorithm however are not average case efficient. This problem was solved by GOAFR [149], which is shown to be the first ad-hoc routing algorithm to be average case efficient like GFG/GPSR, but also retains the asymptotical optimality of AFR. Later, the authors of GOAFR introduced GOAFR+ [146] that is even more efficient than GOAFR on average and is still asymptotically optimal in the worst case. Unlike AFR and GOAFR, GOAFR+ does not necessarily explore the complete face boundary in face routing mode before switching back to greedy mode. GOAFR+ also allows dropping the requirement on network graphs for GOAFR and AFR that any two nodes may not be closer than a fixed constant. An extensive comparison of different combinations of face, greedy, and adaptive face routing was done in [149]. Furthermore, in [49] it was proposed to forward packets with GFG only to neighbors belonging to a connected dominating set to reduce the average hop-count.

Other Position-Based Algorithms

There have been many other complex position-based routing algorithms proposed which are not based on the concept of face routing. In EASE [85], each node caches all positions of previously seen nodes and associates a time-stamp with these positions. A node consults its cache to obtain estimates of the destination's current location. As a packet travels towards its destination, each intermediate node is able to improve successively the estimation of the destination's precise location. In GRA [124], each node does not only have knowledge of its neighbors, but also stores the positions of all other nodes it is aware of, together with the next hop to reach these nodes. When a packet is received, the packet is forwarded to the node that is closest to the destination node's position. Especially tailored for routing in sensor networks, [119] proposed a new paradigm to perform distributed monitoring of the environment called directed diffusion. Nodes are application aware and data is named, thus, the caching and

processing of data can be performed in the network. A further well-known protocol is DREAM [9]. Unlike other position-based algorithm, DREAM includes a location service in order to determine the position of destination. Each node proactively disseminates its location through the network where the frequency and the number of hops over which the packet will be transmitted depends on the speed of the node and the distances to the other nodes. If a node has a packet to transmit and is far away, it still knows the destination's approximate position, and broadcasts the packet. All neighbors within a certain directional range towards the destination forward the packet further. As the packet travels closer to the destination, intermediate nodes which have more recent information about the position of the destination ensure that eventually the packet can be delivered. LAR as proposed in [141] is not a position-based routing protocol as defined above, however, it also uses position information and is therefore described in this section. LAR is similar to the topology-based protocol DSR [129] and uses a route request mechanism to discover a source route to the destination. LAR makes use of location information only to reduce the flooding of the route request packets. A node requesting a path to a destination may have information about previous positions of the destination and thus can estimate the area within which the destination is currently located, called the expected zone. The source node determines the request zone, which is an area that includes itself and the expected zone. Only nodes within the request zone forward a route request.

Limitation of Position-Based Routing

A very important general result on position-based routing algorithms was given in [22]. The authors showed that no deterministic stateless routing algorithm works for all convex subdivisions. In [147], it was proven that any position-based routing algorithm has expected cost of at least the square of the best route's cost. These two results provide some theoretical boundaries for the design of position-based routing algorithms.

3.3.4 Examples of Position-Based Routing Protocols

We briefly discuss the two most relevant routing protocols for our thesis in more detail, namely GFG/GPSR [23, 134] and Terminode routing [15]. GFG/GPSR is used as the representative for position-based routing protocols in Chapter 4 to study the impact of beaconing. Furthermore, we also used GFG/GPSR in Chapters 5 and 7 for comparison with the performance of BLR in Chapter 5 and AMRA in Chapter 7. We used GFG/GPSR because it is appropriate for the scenarios we investigated. Although more advanced position-based routing protocols exists, they may incorporate features that could be incorporated in the proposed protocols as well. Furthermore, the performance gain of these other protocols may be limited or may be restricted to certain scenarios only, not relevant for this thesis. Terminodes is used for comparison in Chapter 7 as it is to the best of our knowledge the only position-based protocol with the same objective as AMRA, namely to not route packets directly towards the destination, if not possible, and avoid that greedy routing fails in the first place.

Greedy and Face Routing (GFG/GPSR)

Perhaps the most cited position-based routing protocols is GPSR [134] which is basically only an extension of GFG [23] with MAC layer enhancements however. Thus in the following we refer to these algorithms together as GFG/GPSR. A packet is routed in a greedy manner towards the position of the destination. Each node selects the node among all its neighbors that is geographically closest to the packet's destination. This process is repeated until the packet reaches the destination. If a node does not have any neighbor closer to the destination, it switches 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 where face routing is applied is called perimeter mode routing in [134]. As soon as the packet arrives at a node closer to the destination than where it switched to face routing, the packet switches back to greedy routing. It was shown that GFG/GPSR guarantees delivery for static and connected networks. However, if nodes are mobile, packets may still loop in the network. In Fig. 3.6, the packet is first routed in greedy mode to node X , which has no closer neighbor within transmission range to destination D , and enters face routing mode. At node Z , the packet is again routed in greedy mode.

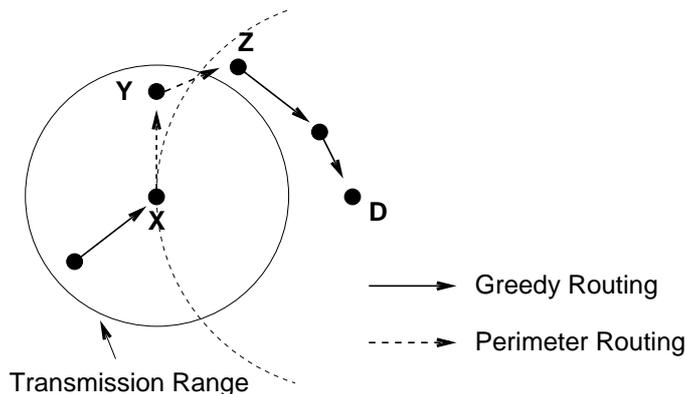


Figure 3.6: Greedy and perimeter routing

The following practical mechanisms were proposed in [134]. Beacons are transmitted at an interval of 1.5 s . A node removes a node from its neighbor table, when it did not receive a beacon for 6.75 s , i.e., if it has missed more than four consecutive beacons. The beacons are randomly jittered by 50% of the beacon interval, in order to avoid possible synchronization of beacons between neighboring nodes [70]. Furthermore, some changes are implemented in the MAC layer to optimize and make IEEE 802.11b [115] more robust in mobile scenarios. If a packet cannot be delivered on the MAC layer within the maximum number of retransmissions to the next hop, normally seven retransmissions with IEEE 802.11, the following operations are triggered. The unreachable node is removed from the neighbor table and all further packets with the same next hop already in the queue of the MAC-interface are passed back to the routing

protocol. Finally, nodes operate in promiscuous listening mode to receive all packets, which allows piggy-backing beacons on data packets.

Terminode Routing

Terminode routing was proposed in [15, 14] and consists of Terminode Remote Routing (TRR) and Terminode Local Routing (TLR), which are position- and topology-based protocols, respectively. In order to optimize routing in case of voids in the network topology, TRR finds a list of anchor points. The data packets are routed basically with GFG/GPSR over these anchor points to the destination circumventing voids. TRR employs the Friend Assisted Path Discovery (FAPD) to discover a loose source path route consisting of several intermediate anchor points, i.e., geographical positions. Each node maintains a list of nodes, called friends, to which it maintains a good path. These nodes do not need to be in the vicinity. They can even be distributed all over the network. To find an anchored path to the destination, a node asks its friends that in turn ask their friends and so on. FAPD makes use of the concept of small world graphs [172]. Small world graphs are sparse, clustered, and have a small diameter as long links exist between the clusters.

To best of our knowledge, Terminodes Routing [15] is the only position-based routing protocol that not always forwards packets in a greedy manner directly towards the destination, but allows forwarding them along an anchored path not along the line-of-sight to the destination. Basically, data packets are routed by GFG/GPSR towards the coordinates of the first anchor. After having arrived approximately at these coordinates, the packet is routed to the second anchor point and so on. After the last anchor, the packet is routed to the position of the destination. In Fig. 3.7, to route a packet initially towards D may be suboptimal because a direct path does not exist due to an obstacle in-between. Thus, node S sends the packet first to anchor point $AP1$. At $AP1$ the packet is redirected towards $AP2$ and then finally to the destination D .

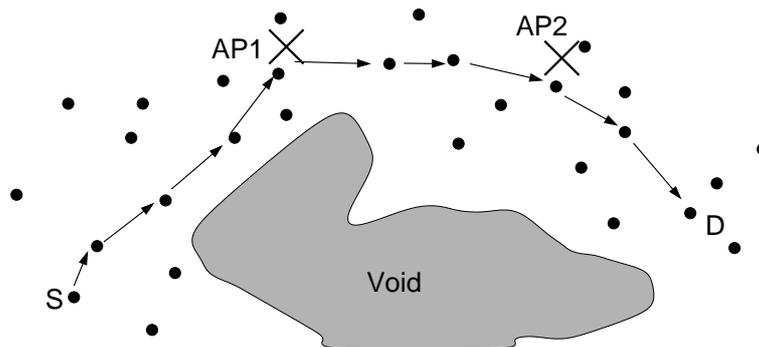


Figure 3.7: TRR with anchor points

If a packet is close to the destination, the packet is routed with TLR. TLR is used to deal with inaccurate position information of the destination because the destination may have moved since the last location update. Each node periodically broadcasts a hello message and includes its one-hop neighbors. Con-

sequently, each node is aware of its two-hop neighborhood and a data packet can still be delivered if the destination node has not moved too far away.

3.3.5 Discussion

Unfortunately, no comprehensive comparison studies for position- and topology-based routing protocols have been performed yet. We can find results where position- and topology-based routing protocols were simulated in [34, 73, 13, 134] and surveys on position-based routing protocols in general in [71, 82, 227, 167]. We briefly list some of the discovered advantages of position-based routing protocols over topology-based routing.

- The number of routing control messages can be reduced significantly which leaves more resources for data traffic. In most protocols the only control traffic are the broadcasted beacons.
- Each node has to store only little information about the network. Basically, it only needs to maintain a list of all its neighbors.
- The risk that stored information becomes stale is reduced since nodes are almost state-less. Stale routing information is in the best case useless or in worse cases can cause wrong routing decisions and loops.
- Position-based routing naturally supports geocasting, which is important for several applications in sensor and vehicular ad-hoc networks.

Even though position-based routing protocols have many advantages and are superior to topology-based protocols for many scenarios, they also suffer from several drawbacks. The first two drawbacks in the following list are similar to drawbacks encountered in topology-based protocols. The reason is that position-based routing protocols reduce, but do not completely eliminate the proactive control traffic and the statefulness. The third to seventh listed drawbacks are caused by the way position-based routing protocols forward packets and do not exist in topology-based protocols.

1. Position-based routing protocols require the proactive transmission of hello messages, which not only wastes scarce battery power, but also interferes with regular data transmissions. Data packets are destroyed and need to be retransmitted, consuming even more battery power, reducing the capacity of the network, and introducing additional delay.
2. Even though the protocols are often claimed to be nearly stateless, nodes must store local information about the network, namely the positions of their neighboring nodes. This information may become outdated and inaccurate because of mobility, or nodes toggling between active and sleep modes. Stale neighbor information leads to wrong routing decisions as packets may be forwarded to unreachable neighbors.
3. Routing a packet along the line-of-sight between the source and destination may often not be possible in realistic networks. In such scenarios, the performance of position-based routing protocols may degrade severely as greedy routing fails and a recovery mechanism has to be applied, where

packets are forwarded according to the face routing algorithm. The followed path may then be very suboptimal as shown in an example in Fig. 3.8.

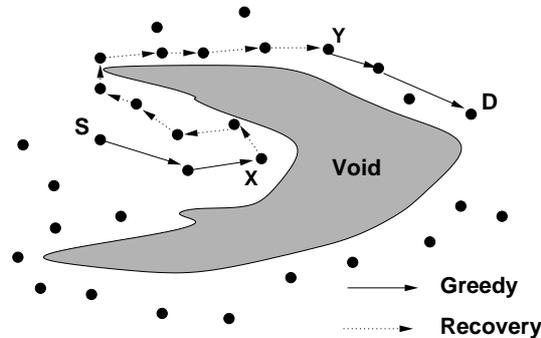


Figure 3.8: Suboptimal path taken by position-based routing protocol

4. Each packet is sent completely independent of all others, e.g., if greedy routing fails and packets are forwarded with face routing along a very long path even though a much shorter exists, all subsequent packets will follow the longer path. The protocols have no way to adapt and to learn from experience.
5. Packets are routed based solely on location information and other criteria such as delay, link capacity, and current traffic load are not taken into account. Even when routing along a straight line to the destination is possible, it may be advantageous to take another path to avoid areas with congestion and long delays.
6. Face routing that is used to recover from local minima does not make use of all available links in order to guarantee loop-freedom. More precisely, packets are sent to close neighbors instead of the far neighbors as chosen by greedy routing which increases the hop count, thus, also the delay, which in turn increases the congestion in the network.
7. Even though, face routing provides loop-freedom in static network, it does not for mobile networks. In highly dynamic networks, a large fraction of the packets is dropped due to loops in the recovery mode where face routing is applied. Looping packets also have an indirect impact on other packets as they may congested the network.

We can conclude that basically all these drawbacks exist also in the protocols described in this section such as GFG/GPSR, GOAFR, and the Terminode routing protocol. There are only two exceptions. First, Terminode routing does not suffer from the third drawback as it avoids to route in directions with no connectivity. Secondly, GOAFR was also shown to provide worst-case guarantees in scenario where greedy routing fails and face routing has to be applied. However, we also have noticed that many of these drawbacks only apply for certain kind of ad-hoc networks and may be completely negligible in others.

3.3.6 Beacon-Less Protocols

Three position-based routing protocols, similar to the BLR protocol proposed in Chapter 6, have been independently developed and almost simultaneously published in three other papers [17, 74, 272]. In their work, the authors focused on different aspects. [17] considered wireless sensor networks and how routing protocols are influenced by nodes toggling between active/sleep modes to conserve energy. Their routing algorithm applies a cross-layer design and is incorporated with the DCF of IEEE 802.11 [115]. Similarly, [74] also proposed to incorporate the routing protocol and the MAC layer. Furthermore, they analyzed the behavior and performance of their routing protocol in vehicular ad-hoc networks by accounting for realistic movement patterns of vehicles on a highway. An in-depth analytical analysis of the multihop and of the energy and latency performance was conducted in [272] and the companion paper [271], respectively.

3.4 Link Incidents and Beaconing

One of the main characteristics of ad-hoc networks is the constant and often unpredictable changing network topology. New links may appear if two nodes move into each other's transmission range, or existing links may break if nodes are no longer within transmission range. Furthermore, sleep cycles of nodes, interferences, and other factors may also cause link incidents. In topology-based protocols, broken links cause existing routes to fail which then must be reestablished causing additional transmissions and introducing delay. For very large and dynamic networks, communication may become almost impossible.

One of the main advantages of position-based routing protocols is that no paths are established, thus, no paths can fail. When a neighbor becomes unreachable, a node simply forwards the packet to another neighbor. Thus, link breaks can be repaired completely locally without causing any transmissions or affecting other nodes. Even if link breaks do not have such severe consequences as in topology-based protocols, they still can have a major impact on the performance of position-based protocols. The reason is due to the difference of the perceived topology and the actual physical network topology. A node stores all its neighbors in a table, i.e., all nodes from which it received beacons. Beacons are broadcasted at periodical intervals, known as beacon-intervals, and a node assumes that a neighbor is no longer within transmission range when it did not receive a beacon from this specific node for a certain interval, called the neighbor time-out interval. This neighbor time-out interval is a major cause for the difference between the perceived and the physical topology. A node may assume that a neighbor is still within transmission range as long it is listed in its neighbor table even though the node may no longer be reachable. Whenever an unreachable node is selected as a next hop, the packet cannot be delivered and has to be rerouted to another neighbor. More inaccurate neighbor tables cause more overhead as delivery to the next hop fails more frequently.

To the best of our knowledge, practically no work has yet been done in the area of neighbor table accuracy and beaconing optimization for position-based routing protocols. Thus, the related work in this section is only loosely related to the subject of this thesis, i.e., position-based protocols. In a first step, we

discuss approaches for topology-based protocols that deal with link incidents and ways to predict and prevent them. Afterwards, we briefly mention other approaches that are in some ways related to beaconing.

Several approaches are described in the literature to mitigate the drawbacks of link incidents for topology-based protocols. AODV [195] implements a local route repair mechanism, which aims to replace a particular broken link with an alternate path between the two nodes minimizing the latency and induced routing overhead of link incidents. To avoid complete disruption of communication, [241] investigated the expected lifetime of routes in order to reschedule the route discovery before actual link breakage. Unlike these protocols, several other protocols were proposed, which take the stability of links and paths into account to minimize the number of link incidents in the first place. In [238], the lifetime of a link is taken into account during route discovery, whereas [61] also considers feedback from the link layer about signal strength as primary routing metric. In [79, 80, 169] results from analytical derivations and observations made by simulations are used to design new routing metrics which favor more stable paths. Based on link availability estimations, a metric for path selection in terms of reliability and resilience is introduced in [127] and refined in [126]. If nodes are equipped with GPS receivers or any other technology that provides absolute or relative positions of nodes, information about the velocity and direction are also often known. This information can be utilized to estimate the expiration time of a link and to reconfigure routes timely as proposed in [235]. Unlike [235], where GPS-information is only applied to maintain routes, [230] additionally makes use of location information in the routing decision itself to establish paths in a depth-first search way. In [39], factors that influence the utility of hello messages were studied for determining link connectivity in topology-based protocols.

Unlike topology-based protocols, almost no research has been performed on link incidents and inaccurate neighbor tables in position-based routing protocols. To the best of our knowledge, the only exception is GPSR [134], where the authors compared the packet delivery ratio and routing overhead for different time intervals between beacons. The transmission of beacons may be considered as simple database updates at neighboring nodes. Even though beaconing was not explicitly studied, the determination of the “best” time when to update information stored at distant databases was studied in other contexts. Numerous location management schemes were proposed for cellular networks in the literature, see e.g. [269] for an overview. Location management schemes deal with when to update databases to keep track of a node’s position if it has moved to a new cell. Furthermore, dissemination and replication of data in ad-hoc networks was studied in [135]. The authors propose different strategies when to trigger updates. While we consider the case of updating only neighboring nodes rather frequently, the other approaches focus on information that is transmitted infrequently to certain central and distant nodes. Unlike neighbor information of position-based routing protocols, entries do not need to be periodically refreshed to remain valid. An other interesting approach was proposed in [2]. Each node continuously sample its location and constructs a model of its movement. Nodes flood their model through the network. Whenever a node’s distance from its actual location to its predicated location is larger than a certain threshold, the updated predication model is flooded again in the network. Only recently, [223] studied the effects of inaccurate location information caused by mobility on position-based routing protocols and proposed two mobility prediction schemes

to mitigate these problems.

3.5 Broadcasting in Ad-Hoc Networks

3.5.1 Introduction

Broadcasting in ad-hoc networks is different from broadcasting in wired networks for various reasons. The network topology may change frequently caused by mobility or by changes in the activity status of nodes. Broadcast protocols also have to cope with limited system resources in terms of bandwidth, computational, and battery power. Unlike wired networks where the total cost of the broadcast is normally calculated as the sum of all link costs, ad-hoc networks can make use of the broadcast property of the wireless medium. This allows to cover all neighbors with one single transmission. Consequently, the costs are typically not associated with the links between nodes but with the nodes themselves. Broadcasting is a common operation in ad-hoc networks and many protocols rely on the successful delivery of packets to each node in the network. For example, several routing protocols use broadcasting to detect routes from the source to the destination node such as AODV [196] and DSR [129]. Other applications that require broadcasting are the paging or sending of an alarm signal to particular hosts. Furthermore, many applications make use of geocasting such as in sensor networks or vehicular ad-hoc networks. Geocasting may be considered basically as broadcasting of a packet to all nodes within a certain geographical area [161, 162, 142, 8, 140, 178].

Broadcasting in ad-hoc networks is most simply and commonly realized by flooding whereby nodes rebroadcast each received packet exactly once. Duplicated packets are uniquely identified by the source node ID and a sequence number. Assuming we have a completely connected network, there may be up to as many transmissions as there are nodes in the network. Especially in dense networks, flooding generates a large number of redundant transmissions where most of them are not required to deliver the packet to all nodes. Nodes in the same area receive the packet almost simultaneously due to the highly correlated timing of retransmissions. This excessive broadcasting causes heavy contention and collisions, commonly referred to as the *broadcast storm problem* [182], which consumes unnecessarily scarce network resources.

Two important objectives of any broadcast algorithm in ad-hoc networks are the reliability and optimizing resource utilization. First, reliability is concerned with the successful delivery of a packet to all nodes in the network. Even in a completely connected network, the packet may often not be delivered to all nodes since broadcast packets are normally not acknowledged and the broadcast storm makes the transmissions highly unreliable. Secondly, the use of network resources should be minimized without affecting the reliability. Interestingly, these objectives are complementary. Reducing the number of transmissions may also increase reliability as it alleviates the broadcast storm.

In mobile networks with constantly changing topologies, it is impossible to broadcast optimally a packet network-wide. In static networks, this may be possible, often however with a prohibitive amount of control traffic only. Thus, most practical broadcast algorithms for ad-hoc networks try to approach network-wide optimal broadcasting by optimizing the local broadcasting of packets.

Many broadcast protocols have been proposed in order to cope with the broadcast storm problem and optimize broadcasting in ad-hoc networks. We provide a taxonomy and review existing broadcast algorithms for ad-hoc networks. In a second step, we discuss some general properties and summarize the shortcomings of these broadcasting protocols. A survey of broadcast protocols can be found in [258, 260, 233]

3.5.2 Taxonomy

The chosen taxonomy divides the protocols in seven categories ranging from simple flooding to sophisticated protocols that make use of directional antennas. Obviously, several other categorizations are possible and the classification of the protocols may not be unambiguous as some protocols may fall into different categories.

Simple Flooding

Flooding is the most simple way to broadcast. It was argued in [187] that it might also be the only way to reliably deliver a packet to every node in highly dynamic or very sparse networks. This is not limited to broadcasting, but also holds for multicast and unicast transmissions. In such environments, the overhead of another protocol may be even higher than that of simple flooding to cope with the frequently changing topology and to maintain paths or neighbor information. In such scenarios, other protocols may not be able to deliver the packets at all.

Probability-Based Approaches

In [182], each node rebroadcasts a packet with a certain probability p and drops the packet with a probability of $1 - p$. If the probability of forwarding a packet is 1, this scheme is identical to simple flooding. [182] also proposed a counter-based scheme, where a node only rebroadcasts a packet if it has received copies of this packet less frequently than a fixed threshold. In [239], the threshold is no longer fixed but dynamically adapted to the number of neighbors. [94] evaluated probabilistic broadcasting in more depth and proposed to account for nodes' neighbor counts and local congestion levels. In [36], the authors proposed to adjust the probability with which a node rebroadcasts a packet depending on the distance to the last visited node. The distance between nodes is approximated by comparing the neighbor lists. Probability-based schemes were evaluated theoretically and by simulations in [209].

Location-Based Approaches

In the location-based schemes proposed in [182], the forwarding decision is based solely on the position of the node itself and the position of the last visited node as indicated in the packet header. Nodes wait a random time and only forward a packet if the distance to all nodes from which they received a copy of the packet is larger than a certain threshold distance value. The random waiting time is required to give nodes sufficient time to receive redundant packets and to avoid simultaneous rebroadcasts at neighboring nodes such that only nodes that cover significantly large additional area rebroadcast the packet. Instead of using

the distance of nodes as a measure for the additional area covered, [182] also proposed an area-based method, which directly determines the possible covered area from the distances between nodes. Recently, [259] validated the simulation results of [182] by analytical models.

Neighbor-Designated Approaches

Neighbor-designated schemes are characterized by the fact that nodes are aware of their neighborhood. The basic idea in all proposed approaches is that each node selects a set of forwarders among its one-hop neighbors so that all two-hop neighbors can be reached through the forwarders. A node only forwards packets from the set of neighbors out of which it was selected as a forwarder, i.e., rebroadcasting nodes are explicitly chosen by upstream senders. Most of these neighbor-designated approaches are quite similar. We describe the multipoint relaying protocol (MPR) proposed in [150] in more detail as it will be used in Section 6.4 for the evaluations and as it is the broadcast mechanism used in the OLSR routing protocol as defined in RFC 3626 [44]. Nodes periodically broadcast beacons including a list of all their neighbors. Consequently, each node has knowledge of its two-hop neighbors. A node selects some of its one-hop neighbors to rebroadcast all packets they receive from it. The chosen nodes are called Multipoint Relays (MPRs). Each MPR also choose a subset of its one hop neighbors to act as MPRs. As always only a subset of all one-hop neighbors rebroadcast a packet, the total number of transmissions is reduced. In order to guarantee that still all nodes in the network receive a broadcasted packet, the MPRs must cover all two-hop neighbors. The protocol select such one-hop neighbors as MPRs that most efficiently reach all nodes within the two-hop neighborhood, i.e., the one-hop neighbors are a minimal set of neighbors which cover all two-hop neighbors. After a node has selected its MPRs, it lists them in the beacons. When a node receives a beacon, it checks if it was selected as MPR from this node, and if so, it must rebroadcast all data packets received from that node. In [158], the set of forwarders excludes all one-hop neighbors that are covered by three or more forwarders. Furthermore, the idea of passive acknowledgments [130] was used to avoid transmission of acknowledgements. Nodes do not send an additional acknowledgment to confirm the reception of a packet, which may become another bottleneck of congestion and collisions called *ACK implosion problem* [166]. The rebroadcast of the packet itself is taken as the confirmation and a NACK packet is transmitted in case a node does not overhear the rebroadcasting from all nodes it expected [166]. In [153, 192] the set of forwarders was reduced by excluding the one-hop neighbors that were already covered by the node from which the broadcast packet was received. In [157], two-hop neighbor information is piggy-backed on packets and permits to eliminate the two-hop neighbors already covered by the last visited node. In [193], the forwarding nodes are selected from a larger set, where a node includes all neighbors of a node with higher priority, e.g., based on IDs, in its cover set. A special class of neighbor-designated approaches are based on connected dominating sets, where only nodes of the dominating set rebroadcast a packet [145, 76].

Self-Pruning Approaches

Unlike the neighbor-designated method, each node decides for itself on a per packet basis if it should rebroadcast the packet. In [153], a node piggy-backs a list of its one-hop neighbors on each broadcast packet and a node only rebroadcasts the packet if it can cover some additional nodes. Several of these approaches are based on (minimal) connected dominating sets.

As the problem of finding such a set is proven to be NP-hard [164], several distributed heuristics are proposed. [262] proposed an algorithm, which only requires two-hop neighbor information. A node belongs to the dominating set, if two unconnected neighbors exist. Furthermore, two rules are proposed to reduce the size of the connected dominating set, which requires an order on the IDs of the nodes. This idea was further improved in [231], where the degree of a node was used as primary metric instead of their IDs. Unlike [262] where two-hop information is required, one-hop neighbor information is sufficient if nodes are aware of their positions in order to determine if two neighbors are connected [231]. Under the assumption that each node knows its accurate position, connected dominating sets and the concept of planar subgraphs are used in [217] to reduce the communication overhead for broadcast messages. In [260], a generic scheme was proposed based on two conditions, namely on neighborhood connectivity and history of the already visited nodes. Each node receives information of its k -hop neighborhood by exchanging $(k - 1)$ -hop information with its one-hop neighbors by periodical hello messages. Information about a node's property, such as ID or node degree, and a list of already visited nodes is added to the broadcast packets. Based on this information a node decides whether to forward a packet or not. In [75], they show that minimum latency broadcasting is also NP-hard and propose an algorithm where latency and the number of transmissions are within a constant time of their respective optimal values. The algorithm constructs a broadcast tree rooted at the source node and afterwards schedules the transmission times for all nodes following a greedy strategy. To be able to cope more efficiently with mobility, [261] proposed to use two different transmission ranges for the determination of forwarders and for the actual broadcast process. The difference between these two transmission ranges is based on the update-frequency and the node movement. They further proposed a mechanism to ensure consistency between the different views of different nodes on the network. A comprehensive performance comparison of various of these broadcast protocols based on self-pruning is given in [47].

Energy-Efficient Approaches

The algorithms discussed above can be also considered as energy-efficient as they aim to reduce the total number of transmissions to deliver the packet to every node in the network. Thus, they also reduce the total energy consumption at the same time. In case nodes can adjust their transmission power however, the number of transmissions may not be proportional to the energy spent. Transmissions to close-by nodes cost much less energy than transmissions to far away nodes, especially in view of realistic path loss factors of approximately 4. Consequently, several transmissions over multiple short hops may save energy compared to one transmission over a long distance.

The problem of transmitting a packet energy-efficiently to all nodes in the

network where nodes have adjustable transmission radii was considered in several papers. [257] proposed a broadcast incremental power algorithm, which constructs a tree starting from the source node and adds in each step a node not yet included in the tree that can be reached with minimal additional power from one of the tree nodes. In [245], theoretical bounds on the performance of the broadcast incremental power algorithm of [257] were provided. [38] considered the minimum energy broadcasting problem and proposed a localized protocol where each node requires only the knowledge of the positions of itself and the neighboring nodes. This eliminates two drawbacks of [257] as the algorithm requires almost global knowledge to construct the tree efficiently and it is difficult to maintain in case of mobility. [30] showed the NP-completeness of minimal power broadcast. They also proposed a distributed algorithm for energy efficient broadcast. Starting from an initial link-based minimal spanning tree, the total energy to maintain the connectivity of this broadcast tree is reduced gradually by exchanging some existing branches by new branches. In [132] it was shown that minimizing the total transmit power does not maximize the overall network lifetime. Note that energy efficiency is not necessarily directly related to network lifetime. If the same nodes always forward packets, broadcasting may be energy-efficient, but the battery at these nodes depletes quickly. In [132], the algorithm constructs a static routing tree, which maximizes network lifetime by accounting for residual battery energy at the nodes. [247] presented a distributed topology control algorithm, which extracts network topologies that increase network lifetime by reducing the transmission power. A comparison of several power-efficient broadcast routing algorithms is given in [131].

Directional Antenna-Based Approaches

Directional antennas are used to improve the performance of broadcasting by reducing interferences, contention, etc. It was shown in [246] that MAC protocols that utilize directional antennas can improve the performance of broadcast traffic in ad-hoc networks. In [114], each node is assumed to have a beamwidth of 90° and packets are only forwarded in the 270° direction other than that in which the packet arrived. If nodes are aware of their neighborhood through hello messages, nodes may explicitly send the packet to nodes that are farthest from the current node. In [37], directional antennas are used to transmit broadcast packets to all neighbors in a connected planar subgraph of the complete network graph, namely the relative neighborhood graph. A comparison study of the performance of various directional antennas algorithms is provided in [133].

3.5.3 Discussion

Different kinds of broadcast protocols show different kinds of advantages and shortcomings. Comprehensive comparison studies were conducted in [258, 94, 47, 131].

The majority of the proposed protocols are either neighbor-designated, self-pruning, or energy-efficient schemes that all belong to the stateful protocols. That means they require at least knowledge of their one-hop neighbors, sometimes even global network knowledge is required. Therefore, they also show similar drawbacks as stateful position-based unicast routing protocols, which

require local neighbor information to forward packets. Like position-based unicast routing protocols, stateful broadcast protocols require the proactive distribution of information with hello messages. The proactive communication and computation overhead of these protocols consumes unnecessarily scarce network resources like battery power and bandwidth. These costs occur even if no packets are broadcasted. Furthermore, their performance suffers significantly in highly dynamic networks as the frequent topology changes induce an excessive, or even prohibitive, amount of control traffic, which occupies a large fraction of the available bandwidth. Furthermore, stateful algorithms may also never converge and reach a consistent state, if changes occur too frequently. Their inability to cope with frequent topology changes together with the proactive transmission of control messages, which wastes network resources, make stateful protocols unsuitable for certain kind of ad-hoc networks such as sensor and vehicular ad-hoc networks. Stateful protocols also have some distinct advantages. Packets can be forwarded almost immediately without introducing additional delay and they are barely affected by high traffic loads and collisions as shown in [258].

The probability- and location-based schemes, as well as simple flooding belong to the category of stateless algorithms, as they do not require nodes to have any neighbor knowledge. As they do not maintain neighbor information, they are almost immune to frequently changing network topologies. A further advantage is their simplicity. The main drawbacks of stateless protocols are twofold [258]. First, the number of rebroadcasting nodes can be disproportionately high in networks with a high node density. Secondly, the random delay introduced at each node before rebroadcasting a packet is highly sensitive to the local level of congestion. The main reason for this is that these stateless protocols use fixed parameters, e.g., a distance-threshold, to rebroadcast a packet. These algorithms are highly sensitive to the chosen threshold values and may perform well in some scenarios, and very poorly in others. For example, packets may die out in sparse networks and the number of transmissions may not be reduced significantly in dense networks for too low and too high parameter values, respectively.

We may conclude that stateless protocols would be a preferred choice for sensor networks, vehicular ad-hoc networks, and other ad-hoc networks with dynamic topology and/or strictly limited resources, if they could achieve nearly the same level of performance of stateful protocols over a wide range of network conditions.

3.6 Ant-Based Routing Protocols

3.6.1 Introduction

Ants, as well as termites, some bees and wasps, belong to the category of social insects. Their behavior can be described as social due to the way they fulfill complex tasks. Each individual insect is a simple individual. They are rather unintelligent and almost blind, have only limited memory, etc. Though, each of these limited individuals is not aware of the problem to solve as a whole, together in cooperation they are able to solve complex tasks through *stigmergy* first described in 1959 in [84]. Sorting brood, finding shortest paths, building

nests are some examples [19]. Stigmergy can be described as “communication through the environment” and is the key to these astonishing abilities. The behavior of the individuals is influenced by what others do or have done. More specifically, how the environment was altered by other insects while contributing their part to solve the task [50].

The principle of stigmergy is illustrated in the following section 3.6.2 by means of an example, namely the ant colony optimization. In Section 3.6.3, we briefly present the most well-known routing algorithms inspired by ant colony optimization for fixed, wired networks. Then, we give an overview of proposed routing algorithms inspired by swarm intelligence for ad-hoc networks in Section 3.6.4.

3.6.2 Ant Colony Optimization

We explain the problem solving paradigm enabled through stigmergy by means of ants finding shortest paths between the nest and a food source, known as ant colony optimization. Ants mark the followed path with a chemical substance called pheromone, which can be perceived by other ants, while foraging and looking for food. Other ants are attracted by these pheromone trails and in turn reinforce them even more. If no longer reinforced, the pheromones evaporate slowly and trails vanish again. With time, the shortest paths emerge as a result of this auto-catalytic effect. The experiment depicted in Fig. 3.9 was conducted in [10] where two paths of different length are offered to the ants between their nest and a food source.

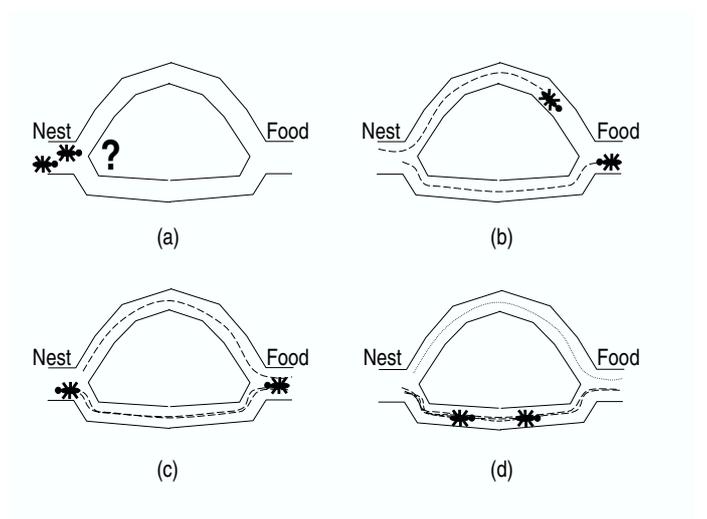


Figure 3.9: Ants finding shortest paths

Suppose that two ants leave the nest at the same time and forage for food. Initially, there are no pheromones on either path and ants have no knowledge of the location of the food. Hence, an ant arriving at the intersection chooses each of the paths with an equal probability of 0.5. We assume that one ant travels the upper and one the lower path to the food source in Fig. 3.9(a).

Clearly, the ant on the lower path arrives at the food source earlier shown in Fig. 3.9(b). When returning to the nest, this first ant finds a pheromone trail on the lower path, namely the pheromones that were deposited by itself. Because the other ant did not yet arrive, no pheromones are laid on the upper path. Consequently, the ant chooses again the lower path with the higher pheromone concentration to return in Fig. 3.9(c). The second ant returning from the food source to the nest, accordingly senses one pheromone trail on the upper path and a pheromone concentration twice as high on the lower path. As ants are more attracted to higher pheromone concentrations, the second ant probably also selects the lower path. Consequently, more and more subsequent ants choose the lower path as depicted in Fig. 3.9(d). On the other hand, the pheromones on the upper path are no longer reinforced, decays, and eventually the trail will vanish. This example mimics the behavior of a type of ant known as *Lasius Niger*, which deposits pheromones when traveling in both directions, to and from the nest. Other species of ants adopt different forms of pheromone laying [40]. For example, some species deposit pheromones only during their return trip or lay more pheromones for richer food sources.

The laying and decay of pheromones result in a positive and negative reinforcement of the available paths, respectively. The decay of the pheromones on the existing trails is necessary and avoids possible stagnation.

3.6.3 Ant-Based Routing in Fixed, Wired Networks

Ant colony optimization was applied to various optimization problems such as the traveling salesman problem [58], graph coloring problem [244], and vehicle routing problem [27]. Overviews of optimization algorithms based on the ant colony optimization can be found in [57, 234]. Recently, several routing protocols have been proposed inspired from social insects behavior for fixed, wired communication networks and for ad-hoc networks. The basic principle of all these algorithms is that current traffic conditions and link costs are measured by transmitting “artificial ants” (mobile routing agents) into the network. These ant packets mark the traveled path with an “artificial pheromone”, i.e., update the routing table depending on the collected information. Therefore, they increase the probability of choosing a certain link for a given destination. Results from ant-based routing applications in fix wired and rather static network are very promising. The pheromones may be used as a measure for any metric under consideration such as delay, bandwidth, jitter. In the following, we describe in more detail some well-known ant-based routing algorithms for infrastructure networks. Overviews can be found in [220, 136].

Ant-Based Control (ABC)

Ant-Based Control [215, 214, 213] is the first ant-based routing algorithm proposed in the literature, based on ideas of [4] where the authors considered the problem of placing calls in a circuit-switched network. The network is represented by an undirected graph where the switching-stations are modeled as nodes and their connecting links as edges. The routing tables in the nodes are replaced by pheromone tables with a row for each node in the network and a column for each neighbor. The entries indicate the strength of the pheromone trail on the link from the current node to the destination via that neighboring

node. Ants are sent periodically from any source node to a randomly chosen destination node. The routing policy for the ants is not deterministic, but rather probabilistic. The ants heading for a given destination select their next hop with a probability according to the values in the pheromone table. Pheromone laying is modeled as an increase of the probability for the entry of the previously visited node and the ant's source node, i.e., pheromone trails are setup in a backward manner, whereby the increase depends on the age of the ant and on the pheromones already present on the link. Unlike the ant packets, a deterministic routing policy is applied for the calls. To find a route for a call from a source to a certain destination, the neighbor with the highest probability in the pheromone table is always chosen as the next hop. Finally, if the destination is reached and the complete route determined, the call is placed on the network if all involved nodes have some spare capacity, otherwise it fails.

In Fig. 3.10, a simple network with four nodes is depicted together with the pheromone table of node *A*. The entries in the pheromone table are given as probabilities. Thus, an ant launched at node *A* with destination node *D* has a 51% and 49% chance to travel via node *C* and *B* respectively. Unlike ants, a call is always routed over the node with the highest probability, i.e., node *C* in this situation. Suppose that an ant just arrived over *B* from node *D* at node *A*. The ant will alter node *A*'s pheromone table corresponding to its source node, i.e., node *D*, by increasing the probability for its last visited node *B* to, e.g., 55%. Subsequent calls are now routed no longer over node *C*, but over node *B*.

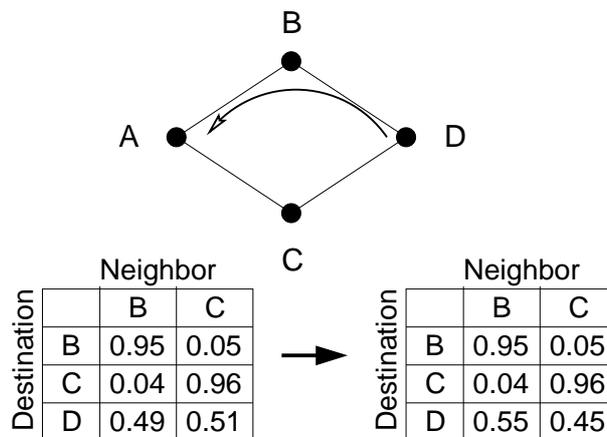


Figure 3.10: Simple network with pheromone tables of node *A*

AntNet

AntNet as proposed in [53, 55, 54, 52] addresses routing in packet-switched networks. In these packet-switched networks, the cost associated with links may be highly asymmetric. That means that the path from a source node to a destination node may be of different quality, i.e., may have different costs, than in the opposite direction. But ants as in the ABC protocol are only able to update the pheromone table in one direction, namely toward their sources, which is not appropriate in such scenarios. Consider the example given in

Fig. 3.11. The thin arrows indicate cost-efficient links whereas the thick arrows indicate high costs. Thus, if node *A* launches ants heading for node *D* that choose the most cost efficient path, they are routed over node *C*. Arriving at node *D*, the ants would reinforce the path in the opposite direction they have traveled, i.e., also over node *C*, although the path from *D* to *A* is more cost efficient over node *B* than over node *C*. However, the routing table at node *D* indicates a high probability for routing over node *C* because most ants arrived along this path from node *A*. Node *D* does not route data packets along the more cost-efficient path over node *B*. AntNet uses forward and backward ants to

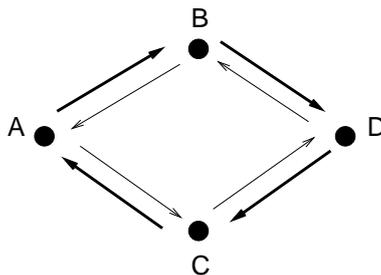


Figure 3.11: Cost asymmetric paths

cope with this asymmetry of packet-switched networks. Each node maintains a probabilistic routing table with a row for all network nodes and a column for outgoing links. Forward ants are launched towards a destination and while moving collect information about the quality of the followed path, e.g., in terms of delay. They do not alter the routing table however. At each node, the forward ant is routed according to the probabilities in the routing table and the current network conditions by taking into account the current length of the outgoing link queues. Once the forward ant has arrived at the destination, it is killed and its accumulated memory is transferred to a backward ant, which travels back to the source node along the same path of the forward ant in the opposite direction. At each visited node, the backward ant updates the routing table entries for the (forward ant's) destination node by increasing the probability. The increase is a function of the quality of the path followed by the forward ant, i.e., the trip time from the current node to the destination node. In Fig. 3.11, the forward ant launched at node *A* would most probably follow the path over node *C* to node *D*, but not yet update the routing tables. The backward ant returning from node *D* to node *A* updates the entries along the traveled path of the forward ant for node *D*. At node *C* simply the direct link to *D* is reinforced, while at node *A* the probability of choosing node *C* as a next hop when heading for destination *D* is increased. In AntNet, each node also maintains a table where the best trip times and estimated means and variances of previous ants are stored to judge accurately the goodness of a path followed by an ant.

Other Ant-Based Algorithms

In [110, 108, 109], each node records the amount of incoming traffic over the different links for different sources. Forward ants are routed to their destinations according to this incoming traffic. This allows a forward ant following approx-

imately the path of a backward ant in the opposite direction. In this way, no backward ants are required and the forward ants can update the routing table immediately. While in ABC the ants only update the entry corresponding to the source nodes, the smart ants in [20] update the entries in the pheromone table for each node they have passed and not only for the source node. This extension was shown to decrease significantly the time until shortest paths emerge and routing tables stabilize. In [254, 250, 256, 255], AntNet was enhanced with genetic algorithm capabilities to solve problems of multipath routing in circuit-switched networks. This algorithm was generalized in [249] to a framework and further enhanced with other functions such as fault location detection [252], fault tolerance [253], and QoS [251]. [236] compares the performance of two ant-based algorithms, where ants are forwarded uniformly and also according to the probability given in the routing table to one of the neighbors. [159] proposes a multicast routing protocol where ants are used to find a multicast tree. Furthermore, [43] proposed an ant-based algorithm for QoS multicast routing, where delay and the jitter of edges linking two nodes are used to route the ants. The routing and allocation of wavelengths in an optical network was described in [243] where ants are used to minimize the number of required wavelengths over one link. In [221] more than one colony of ants is used, each laying its own pheromone. Consequently, nodes maintain multiple probabilistic pheromone tables, which prevents all data traffic from being directed over only one optimal path. [218] proposed a topology control algorithm where nodes collect local topology information via sending and receiving ants in order to determine its appropriate transmission power.

3.6.4 Ant-Based Routing in Ad-Hoc Networks

Like conventional link-state and distance vector routing protocols, the ant-based algorithms for fixed, wired networks are also hardly directly applicable in ad-hoc networks without modifications. Ant-based algorithms have the inherent property to require the network to be rather stable for a long time until shortest paths emerge. At the beginning, all links are initialized with the same probability and packets perform a random walk over the network. Therefore the performance, e.g., in terms of delay and throughput, is far from good at the beginning and stabilizes around the optimum only after a certain time. The required time obviously depends on the size of the network, the number of emitted ants, the rate of topology changes, etc. Thus, in most approaches proposed in the literature, ants are emitted from the beginning to train the network and data packets are only transmitted after this training phase. For example in [54], the ant generation interval at each node is set to 1 s for a network with 14 static nodes interconnected by 21 bidirectional links. Despite this rather small network, 100 s learning time is assigned before traffic flow starts. In [53], due to the larger networks with up to 57 nodes and 162 bidirectional links, there are 500 s of simulation time with no data traffic and ants generated every 0.3 s at each node to build initial routing tables.

This long time required for ant-based algorithm to stabilize and find good routes makes them inapplicable, at least not directly, to ad-hoc networks where the topology may be highly dynamic. Most links between nodes do not exist long enough to receive a large enough amount of pheromones for a certain destination. Considering the required time until paths stabilize in static networks,

these solutions are prohibitive for ad-hoc networks with frequent changing topology. The algorithm may never stabilize or find short paths. This is one of the reasons why most of the ant-based routing algorithms for ad-hoc networks follow different paradigms than the protocols for the fixed, wired network. Most of these ant-based protocols are similar to conventional topology- or position-based routing protocols using ants only for some minor optimizations.

GPS Ant-Like Routing Algorithm (GPSAL)

In [31], the GPS Ant-Like Routing algorithm was introduced which makes use of location information for routing and employs ants only to accelerate the dissemination of routing information, or more precisely, the positions of nodes. All nodes have a routing table where known hosts are stored with their locations and a timestamp of this location. Nodes send out ants that carry the routing table. When a node receives an ant, it compares the timestamp of the entries in its routing table with that of the ant and updates the older with the newer entry. Thus, an ant leaving a node always carries the most updated routing table from the point of view of the nodes already visited. Furthermore, a shortest path algorithm is applied to determine the best possible route to a destination. Unfortunately, the authors of [31] remain vague how exactly this is achieved.

Ant-Colony-Based Routing Algorithm (ARA)

The Ant-Colony-Based Routing Algorithm was proposed in [87, 88, 86]. The routing algorithm is similar to other conventional topology-based routing protocols such as AODV [196] and consists of three phases; a route discovery phase, a route maintenance phase, and a route failure-handling phase. Ants are only emitted on demand, i.e., if a node has to send a packet to a destination for which it does not have a path. A node broadcasts a forward ant that is flooded throughout the network. Each intermediate node stores an entry in the routing table for the forward ant. This entry contains the ant's source address, the previous hop, and a pheromone value that depends on the number of hops to the source node. When the destination node receives a forward ant, it creates a backward ant. The backward ant returns in the opposite direction over the path taken by the forward ant. Like the forward ants, the backward ants create entries in the routing table at intermediate nodes consequently establishing bidirectional paths between source and destination nodes. Data packets are routed probabilistically based on the amount of pheromones on the outgoing links for the respective destinations. In the route maintenance phase, data packets are used to maintain the paths established by the ants. When a node relays a data packet, the path to the destination via the next hop is strengthened as well as the backward path via the previous hop to the source of the data packet by increasing the pheromone value of the respective entries. The pheromone values increase linearly per packet, but decrease exponentially over time. MAC layer feedback is used in the route failure handling phase. If the MAC layer is not able to deliver a packet to the next hop, ARA deactivates this link by setting the pheromone value to 0 and tries to send the packet over an alternate link.

Termite

Termite routing algorithm, as presented in [207] and slightly modified in [206], follows most closely the ant colony optimization. (Even though named Termite, it follows the same principles than the ant-based algorithms ;-). Each node maintains a routing table tracking the amount of pheromones on each outgoing link for all known destinations. When a packet arrives at a node, the pheromones for the source of the packet are incremented by a constant value. Each value in the pheromone table is periodically multiplied by a decay factor. Furthermore, a pheromone ceiling and a pheromone floor are used to prevent extreme differences in the pheromone values. There are four types of control packets; seed, hello, route request, and route reply packets. Seed packets are used to spread actively a node's pheromones throughout the network while hello packets are used to search for neighbors if a node has become isolated. Due to mobility, it may frequently happen that a node does not have a pheromone entry for a destination, thus, route request and route reply packets have to be introduced, similar to AODV [196]. A certain number of route request packets are sent when a node needs to find a path to an unknown destination. The packets perform a random walk, i.e., uniformly choose a next hop, and lay down pheromones on the followed trail. The route request packets are forwarded until a node is found which contains some pheromones for the requested destination or the destination itself. This node issues a route reply packet, which is routed back to the originator of the route request. On its way, the route reply packets add pheromones at the nodes towards its own source. Route reply and data packets are routed probabilistically according to the values in the pheromone table.

AntHocNet

AntHocNet [63, 51] is similar to ARA, but additionally introduces a proactive component. The routes are also only set up reactively when needed. Forward ants are broadcasted by the source and they find multiple paths to the destination. A backward ant traveling back to the source establishes the paths towards the destination by updating the entries in the routing tables. After the route setup phase, data packets are then forwarded probabilistically over available links for load balancing. Unlike ARA, AntHocNet periodically transmits ants during the data session. The ants follow the pheromone trails and have a small probability of being broadcasted at intermediate nodes. Thus, these ants explore paths around the existing ones and they are used to look for path improvements.

Other Algorithms

Ad hoc Networking with Swarm Intelligence (ANSI) as described in [201, 202], is a hybrid protocol similar to ZRP [96]. Proactive ants are periodically emitted to find and maintain paths in the vicinity of a node. Reactive forward and backward ants are used if a node needs to find a path to an unknown destination. The network is clustered into colonies of nodes in [137] to reduce the number of ants. Colony ants are used to find paths between colonies, whereas local ants are used within the colonies. In [165], AODV [195] was extended by ants to reduce route discovery latency. A fixed number of ants forage continuously and

randomly within the network. They keep a history of the recently visited nodes and are only used to disseminate this route information, thus increasing the node connectivity and reducing the amount of route discoveries. Similar to DSR [129], ants are flooded throughout the network in [6] when a node does not have a route to the destination to which it has to send a packet. The ant keeps track of the followed path. A backward ant is created for each forward ant arriving at the destination, which travels the reverse path of the corresponding forward ant and marks the path with pheromones using information about the path quality collected by the forward ant. Data packets are either routed probabilistically, or deterministically based on the highest probability of the next hop. In [219, 224] an energy conservation protocol was introduced which collaborates with a routing protocol. Ants deposit pheromone trails based on the power used to forward packets to the next hop. Nodes located on trails with a lower pheromone concentration are turned off for a certain period of time. [139] studied the performance of different message passing policies to broadcast a packet in an ad-hoc network using swarm intelligence. An ant-based algorithm for solving the minimum power broadcast tree in wireless networks was presented in [48].

3.6.5 Conclusions

The ant colony optimization paradigm has proven of value to various optimization problems. A lot of routing protocols for fixed, wired networks were also proposed based on ant colony optimization, which showed very promising results. Ant packets forage through the network and increase the entries of the routing tables for the followed path depending on the encountered conditions. These protocols were however not tailored for ad-hoc networks and they cannot cope with their salient characteristics such as frequently changing topology. Lately, some ant-based routing protocols have been designed for ad-hoc networks. Basically, all of these proposals are topology-based protocols and incorporate many concepts from conventional on-demand and proactive topology-based protocols which have proven of value, e.g., routes are established only on demand and network topology information is distributed proactively. Even though they improve the resilience of paths and the reliability of the protocols, they also inherit the drawbacks of other topology-based protocols such as large control traffic overhead, consumption of scarce network resources, stored information about the network that can become stale, etc. Thus, they are not quite suited for large networks with highly dynamic topologies such as those considered in this thesis.

3.7 Conclusions

In this chapter, we gave an overview of related work in the area of routing and broadcasting in ad-hoc networks. We started with an overview of topology-based protocols to explain some frequently applied principles. As we are mostly interested in protocols that make use of location information, we discussed in more detail position-based routing protocols as well as their advantages and shortcomings. We saw that they have many advantages mainly due to the reduced communication overhead and the fact that less information about the network has to be stored compared to topology-based protocols. Nevertheless, the same drawbacks remain to a reduced extent. They store local network

topology information and require proactive local communication and, thus, are stateful for the local neighborhood. Furthermore, position-based routing protocols are not able to learn from experience and route packets based solely on geographical information, which is often suboptimal. Afterwards, we described proposed broadcast algorithms for ad-hoc networks. Most of them are stateful and keep track of the network topology in order to optimize broadcasting. They too however suffer from similar drawbacks as previously discussed with position-based routing protocols. Unlike routing protocols, stateless broadcast algorithms have also been proposed for a long time. Nodes decide whether or not to rebroadcast a packet without any knowledge of neighbors and, thus, do not experience the same drawbacks as stateful algorithms. However, it was shown that these algorithms are not able to adapt to varying network conditions. Finally, we gave an overview of the ant colony optimization and the way it is applied to routing in networks. Ant-based routing protocols for fixed wired networks are discussed first to explain fundamental aspects and also because only a few ant-based algorithms have been proposed for ad-hoc networks until now. Ant-based algorithms for ad-hoc networks are similar to conventional topology-based routing algorithms in that they apply the same principles, and show the same characteristics such as a long convergence time. Even though the characteristic of ad-hoc networks make many tasks, especially routing, more difficult than in the fixed, wired networks, they also offer some new opportunities. For example, routing with location information becomes a valid alternative and the broadcast propagation medium allows simultaneous transmission of packet to multiple receivers without spending additional energy.

In Chapter 4, we analyze in more depth the possible drawbacks of the proactive transmissions of hello messages and the statefulness of position-based routing protocols. We are particularly interested in dynamic networks with frequently changing topologies, such as encountered in sensor and vehicular ad-hoc networks, and propose several enhancements to mitigate the observed drawbacks.

Chapter 4

Beaconing in Position-Based Routing Protocols

4.1 Introduction

In this chapter, we study the impact of beaconing on position-based routing protocols. Beaconing, i.e., the periodical broadcasting of hello messages, is used by nodes to announce their presence and location to their neighbors and thereby providing the necessary topological information required for routing. The interval in which beacons are sent is called beacon interval. Each node stores all neighbors and their current positions in a neighbor table, i.e., all nodes within transmission range from whose it received a beacon. If a node does not receive any beacon from one of its neighbors within a certain time interval, called neighbor time-out interval, the corresponding node is considered to have left the transmission range or is unreachable due to any other reason, and is deleted from the neighbor table. Routing of packets is done based on the positions of nodes in the neighbor table. One node is chosen as a next hop according to the applied routing strategy, e.g., the node closest to the destination. Outdated and inaccurate neighbor tables in position-based routing protocols may be considered the analogue of route breaks in topology-based protocols. Even though changes in the network topology do not induce overhead by transmitting control packets as in topology-based routing protocols and only require a local modification of the neighbor table, inaccurate or outdated neighborhood information may severely affect position-based routing protocols. Several topology-based routing protocols make also use of hello messages. Unlike position-based protocols, beacons are however only used to determine the link status among neighbors on established paths in the case when no data packets are transmitted.

We first give an overview of the possible direct and indirect effects caused by the periodical broadcasting of beacons in Section 4.2. Then we try to assess analytically the impact of inaccurate and outdated neighbor information on the performance of the network for position-based routing protocols in Section 4.3. In Section 4.4, we simulate a standard position-based protocol over various sce-

narios to study the effects of outdated neighbor tables and propose and evaluate several optimizations to improve their accuracy. Finally, Section 4.5 concludes the chapter. Further information can also be found in the related publication [105].

4.2 Effects of Beaconing

The periodical broadcasting of beacons has several drawbacks such as unnecessary utilization of network resources and interferences with regular data packets. As beaconing is a proactive component of position-based routing, it is performed independently of actual data traffic. Even in cases where no data is transmitted, nodes constantly broadcast beacons to update their neighbors. We distinguish between direct and indirect effects of beaconing. We classify effects that are caused by the transmissions of beacons as direct effect such as the additionally consumed energy, bandwidth, etc. Indirect consequences comprise all effects that are caused by the fact that a node does never have complete accurate topology information about its neighborhood. These effects are caused by beacons that are broadcasted only periodically. Thus, if nodes are mobile or the topology changes due to any other reason, the topology as perceived by the nodes never corresponds to the actual topology. Inaccurate position information provided by GPS or other position services may further increase the deviation. Even though, the indirect effects are perhaps less obvious, the performance may degrade even more than by the direct effects in terms of increased delay, wasted bandwidth, and battery power.

4.2.1 Direct Effects

We can observe several direct consequences of beaconing. First, additional energy is used to transmit, receive, and process the beacons. Secondly, beacons interfere with regular data transmissions and thus increase the number of collisions and subsequent retransmissions, (if there is no separate signaling channel). This reduces not only the available bandwidth, but at the same time also increases the delay and the congestion in the network. Third, beaconing induces overhead and part of the bandwidth is used for this control traffic and not available for user data.

As we are also strongly concerned with very strict power constraint sensor networks in this thesis, we briefly discuss in more detail the effects of beaconing on the power consumption in more detail. Power consumption is probably the most critical factor for sensor networks that need to operate for years without manual intervention. [66, 68, 267] identified major sources of energy consumption for wireless devices.

- The fixed costs of sending a packet are large compared to the incremental costs.
- The receiving of a message causes high costs, such that if a message is received by some neighbors, the total costs of receiving the message is larger than of sending it.
- After having received the packet header, a node can determine if it is the intended receiver and can discard the packet if it is not. Discarding is a

strategy that allows nodes entering a sleep mode for the duration of the transmission of a packet if it is not the intended receiver. Thus, receiving a packet, passing it to the protocol stack and processing it, costs generally much more than just discarding it at the network interface.

- A node receiving and processing packets destined for other nodes wastes a substantial amount of energy. This is called overhearing, which is, e.g., the case if nodes operate in promiscuous mode.
- Idle listening where a node just listens to the medium to receive possible traffic that is not sent causes high costs.

Let us briefly reconsider the costs of beaconing given these facts. The transmission of beacons is costly even though beacons are small packets. Furthermore, beacons are always broadcasted such that all neighbors receive and need to process the packets and cannot discard them. Many protocols propose to piggy-back beacons on data packets to reduce the total number of transmitted packets. But, it may even increase the power consumption as piggy-backing requires nodes to process every received packet, also unicast packets addressed to other nodes, such that packets can no longer be discarded at the network interface. Furthermore, in scenarios with little data traffic, nodes have to listen to the medium only to receive beacons and can enter less frequently power saving sleep states. This strongly depends on the used MAC protocol however. In view of these results, we may conclude that protocols that use beaconing are highly suboptimal in terms of power consumption and may not be appropriate for sensor networks.

4.2.2 Indirect Effects

In position-based routing protocols, nodes forward packets based on the perceived topology, which typically does not correspond to the actual topology because nodes have moved since their last beacon transmission. Neighbor tables actually do not correspond to the physical topology and are always inaccurate, except for static networks. We can broadly distinguish between three kinds of inaccuracies. First, nodes are listed in the neighbor table with an inaccurate position, but they are still within transmission range. Secondly, a node moved into the transmission range, but it is not visible since no beacons were received yet. These two scenarios have only minor effects on the routing protocol. The routing protocol may take suboptimal decisions and not forward packets over the best-located neighbor. The third scenario has a much stronger impact when nodes are wrongly listed in the neighbor table even though they moved out of transmission range. If such an unreachable node is chosen by the routing protocol, the MAC protocol will not be able to deliver the packet. After several retransmission attempts, the MAC protocol either drops the packet or notifies the routing protocol of the failed delivery and passes the packet back. The routing protocol in turn selects a different next hop and hands the packet over to the MAC protocol again. This process is repeated until the packet can be delivered eventually to the next hop. The rerouting increases the delay, reduces the effective available bandwidth, and consumes energy for the retransmissions. If IEEE 802.11b is used, packets are retransmitted up to seven times before the MAC layer gives up and assumes the next hop to be unreachable. Consequently,

the power consumption is increased by a factor of seven and the effective available bandwidth is only a seventh of the total bandwidth. In order to roughly estimate the induced delay, we consider having IEEE 802.11 on the MAC layer. For each failed transmission the contention window is doubled, starting at a size of 31 up to a maximum of 1023 times the slot time of $20 \mu s$. A node uniformly chooses a backoff time from the contention window for the next transmission. If all seven retransmissions fail because the next hop is out of transmission range, the expected delay is $\frac{31+63+\dots+1023+1023}{2} \cdot 20 \mu s \approx 30 ms$. Note that this additional delay of $30 ms$ is introduced for every select unreachable neighbor, which can happen multiple times at each node before the packet is successfully delivered at the next node. When we refer to neighbor table inaccuracy, we often only refer to this third scenario, which has by far the most severe consequences.

We only mentioned mobility as a possible source of inaccuracy of the neighbor tables. But basically any kind of topology changes have the same effect either caused by nodes that toggle into and out of sleep states, obstacles moving between nodes, interferences, adjustment of transmission and reception parameters, etc. Especially in some kind of ad-hoc networks, like sensor networks where energy conservation is a major issue, putting nodes into sleep mode is often the only way to considerably save energy and may be the major source of topology changes. Thus, we consider speed as a proxy for any kind of topology changes in this thesis.

Topology changes are not the only source of inaccurate neighbor tables. In addition, other factors contribute to inaccuracy. Beacons are broadcasted and most MAC layer protocols do neither require nor provide acknowledgments for broadcast transmissions such that the delivery is not guaranteed. Furthermore, many position-based routing protocols apply a forwarding strategy where packets are forwarded to neighbors close to the destination, cf. Section 3.3.3. Therefore, the chosen neighbor is close to the boundary of the transmission range. This increases the probability that the node has become unreachable. A third factor is that the neighbor time-out interval is often set to a multiple of the beacon interval to avoid that nodes are constantly inserted and removed from the neighbor table if one or two beacons are missed. This longer interval further increases the probability that a neighbor has meanwhile left the transmission range and is no longer available. There are also more practical factors that contribute to inaccurate neighbor tables. Packets transmitted at lower rates typically use more robust modulation schemes and thus can be still decoded at farther distances than packets transmitted at a higher rate. [144] observed that IEEE 802.11b cards transmit broadcast packets constantly at 2 Mbps whereas unicast packets can be transmitted at higher rates. Thus, the set of neighbors may vary for beacons and data packets.

4.3 Analytical Evaluation

After having discussed reasons for outdated and inaccurate neighbor tables and possible implications, we would like to analytically estimate the likelihood of the occurrence of such events. We use the unit disk graph network model, i.e., a fixed isotropic transmission range with radius 1 for all nodes and an unbounded simulation area. Nodes are distributed according to a Poisson point process of constant spatial intensity and move according to the random waypoint model

with zero pause time, i.e., nodes choose randomly some destination and move there with a constant speed chosen uniformly in the interval $[v_{min}, v_{max}]$. The reason for an unbounded area is to simplify our analysis by having a uniform moving direction of the nodes in the interval $[0, 2\pi]$, a uniform distribution of the nodes, and travel distances independent of nodes' locations. All these conditions do not hold in the standard random waypoint model as discussed in Section 2.4.2. Furthermore, we assume that nodes do not change their direction or speed for the time interval under consideration to simplify the analysis. As this time-interval is short and only in the order of a few seconds, this assumption is reasonable for realistic movement patterns. We consider two nodes A and B within transmission range and calculate the probability that they leave each others' transmission range within a certain time interval t , namely the neighbor time-out interval. The crucial point in the derivation is to notice that instead of having both nodes moving, we assume node B being static and node A moving with their relative speed vector. This assumption is valid as nodes move independently of each other and have symmetric transmission ranges. Therefore, we first derive the expected value of the difference of two arbitrary speed vectors in the used mobility model. Then, we calculate the size of the area that was covered by a node's transmission range and is no longer after t when moving at the expected relative speed. The size of this area to the overall transmission range is the probability that a neighbor has left the transmission range.

4.3.1 Probability Density Function of the Speed

We derive the probability density function f_S of the nodes' speed if they choose uniform randomly a speed in the interval $[v_{min}, v_{max}]$. For the probability density function f_S of the speed s , the following holds trivially by definition of the probability density function.

$$\int_{v_{min}}^{v_{max}} f_S(s) ds = 1$$

Since the distance to the next waypoint is independent of the speed, we may assume without loss of generality that all trips have the same distance, say 1. Because a trip with a smaller speed takes inverse proportionally longer than the same trip at a higher speed, we have that $f_S(s)$ must be proportional to $1/s$ in the interval $[v_{min}, v_{max}]$ and 0 otherwise. We immediately have for a certain constant k that

$$\int_{v_{min}}^{v_{max}} \frac{k}{s} ds = 1$$

which yields by integration and some simple algebra

$$k = \frac{1}{\ln\left(\frac{v_{max}}{v_{min}}\right)}$$

Thus, f_S is

$$f_S(s) = \begin{cases} \frac{1}{s \ln \frac{v_{max}}{v_{min}}} & : v_{min} \leq s \leq v_{max} \\ 0 & : \text{otherwise} \end{cases} \quad (4.1)$$

From (4.1), we easily obtain the expected average node speed $E(v_{min}, v_{max})$, which was already derived in [179] and is given here for the sake of completeness.

$$E(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} s \cdot \frac{1}{s \cdot \ln \left(\frac{v_{max}}{v_{min}} \right)} ds = \frac{v_{max} - v_{min}}{\ln \left(\frac{v_{max}}{v_{min}} \right)} \quad (4.2)$$

4.3.2 Relative Speed of Two Nodes

Let the speed vectors \vec{a}, \vec{b} of two arbitrary nodes be given in polar coordinates as (s_a, α) and (s_b, β) with $s_a, s_b \in [v_{min}, v_{max}]$ and $\alpha, \beta \in [0, 2\pi]$. The relative speed vector $\vec{a} - \vec{b}$ in Cartesian coordinates is given by

$$\vec{a} - \vec{b} = (s_a \cos(\alpha) - s_b \cos(\beta), s_a \sin(\alpha) - s_b \sin(\beta))$$

The velocity of the relative speed vector is the norm of $\vec{a} - \vec{b}$ which is

$$|\vec{a} - \vec{b}| = \sqrt{s_a^2 + s_b^2 - 2s_a s_b \cos(\alpha - \beta)}$$

We do not need to consider the corresponding angle of $\vec{a} - \vec{b}$ as the transmission ranges are isotropic and moving directions are uniform over the whole interval $[0, 2\pi]$. It is well known that for a random vector $\mathbf{X} = (X_1, \dots, X_n)$ with the joint density function $f_{\mathbf{X}}(x)$ and a function $\varphi : \mathbf{R}^n \rightarrow \mathbf{R}$, the expected value is

$$E_{\varphi}(\mathbf{X}) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \varphi(x_1, \dots, x_n) f_{\mathbf{X}}(x_1, \dots, x_n) dx_1 \dots dx_n$$

If the X_i are independent, this yields

$$E_{\varphi}(\mathbf{X}) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \varphi(x_1, \dots, x_n) f_1(x_1) \dots f_n(x_n) dx_1 \dots dx_n$$

where the f_i are the probability density functions of X_i .

Thus, the expected value $E_{rel}(v_{min}, v_{max})$ for the norm $|\vec{a} - \vec{b}|$, which is the expected relative speed of two arbitrary nodes, is given by

$$E_{rel}(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_0^{2\pi} \int_0^{2\pi} \sqrt{s_a^2 + s_b^2 - 2s_a s_b \cos(\alpha - \beta)} f_S(s_a) f_S(s_b) f_A(\alpha) f_B(\beta) d\alpha d\beta ds_a ds_b$$

where f_S is the density function of the speed as given in (4.1) and f_A, f_B are the density function of α and β , respectively. As the moving direction of nodes is uniform in the interval $[0, 2\pi]$, we have that $f_A(\alpha) = f_B(\beta) = \frac{1}{2\pi}$.

$v_{min}[\frac{m}{s}]$	$v_{max}[\frac{m}{s}]$	$E_{rel}(v_{min}, v_{max})$	$E(v_{min}, v_{max})$
1	10	5.69	3.91
1	20	9.64	6.34
1	40	16.68	10.57
10	20	18.83	14.43
10	40	29.55	21.64

Table 4.1: Expected speed for different v_{min} and v_{max}

We can simplify this formula by substituting $\alpha - \beta$ by γ . The probability density function f_{Γ} of $\gamma = \alpha - \beta$ is given by

$$f_{\Gamma}(\gamma) = \begin{cases} \frac{2\pi+\gamma}{4\pi^2} & : -2\pi \leq \gamma < 0 \\ \frac{2\pi-\gamma}{4\pi^2} & : 0 \leq \gamma \leq 2\pi \\ 0 & : \text{otherwise} \end{cases} \quad (4.3)$$

This yields for the expected relative speed $E_{rel}(v_{min}, v_{max})$ the following integral where s_a, s_b and γ are distributed according to f_S in (4.1) and f_{Γ} in (4.3), respectively.

$$E_{rel}(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_{-2\pi}^{2\pi} \sqrt{s_a^2 + s_b^2 - 2s_a s_b \cos(\gamma)} f_S(s_a) f_S(s_b) f_{\Gamma}(\gamma) d\gamma ds_a ds_b$$

what is

$$E_{rel}(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_{-2\pi}^0 \sqrt{s_a^2 + s_b^2 - 2s_a s_b \cos(\gamma)} \cdot \frac{2\pi + \gamma}{4\pi^2 s_a s_b \ln\left(2 \cdot \frac{v_{max}}{v_{min}}\right)} d\gamma ds_a ds_b + \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_0^{2\pi} \sqrt{s_a^2 + s_b^2 - 2s_a s_b \cos(\gamma)} \cdot \frac{2\pi - \gamma}{4\pi^2 s_a s_b \ln\left(2 \cdot \frac{v_{max}}{v_{min}}\right)} d\gamma ds_a ds_b$$

Unfortunately, we can not solve this integral analytically and give the values obtained by numerical integration for some specific speed intervals only in Tab. 4.1. Even though a speed interval of $[1, 40] m/s$ seems to be a very high node mobility scenario, the expected average speed $E(v_{min}, v_{max})$ of the nodes is only $10 m/s$, because most nodes move slower than the arithmetic middle of $20.5 m/s$. On the other hand, the expected relative speed $E_{rel}(v_{min}, v_{max})$ is approximately the arithmetic middle and 50% higher than the expected speed $E(v_{min}, v_{max})$ of one single node.

4.3.3 Size of Uncovered Area

We want to determine the size of the area $A(r, d)$ that was initially covered of a node's transmission range and is no longer after it has moved a certain distance d from A to A' . The size of the area $A(r, d)$ is depicted in Fig. 4.1. We

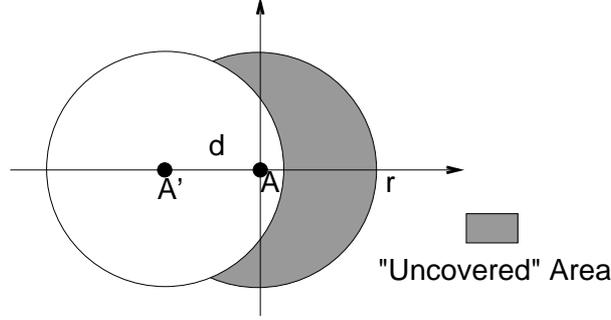


Figure 4.1: Uncovered area

immediately obtain for the size of $A(r, d)$ above the x-axis, which is just half the size of $A(r, d)$, that

$$\frac{A(r, d)}{2} = \int_{-\frac{d}{2}}^r \sqrt{r^2 - x^2} dx - \int_{-\frac{d}{2}}^{-d+r} \sqrt{r^2 - (x+d)^2} dx$$

which yields by integration

$$\frac{A(r, d)}{2} = \frac{d}{2} \sqrt{r^2 - \frac{d^2}{4}} + r^2 \arcsin\left(\frac{d}{2r}\right)$$

and finally we obtain for $A(r, d)$

$$A(r, d) = \frac{d}{2} \sqrt{4r^2 - d^2} + 2r^2 \arcsin\left(\frac{d}{2r}\right) \quad (4.4)$$

4.3.4 Probability of Outdated Entries in Neighbor Table

We can now calculate the probability p that a node B is no longer within transmission range of a node A after t as follows. The given speed interval immediately yields the expected relative speed $E_{rel}(v_{min}, v_{max})$. We obtain the expected distance, which a node moves relative to any arbitrary other node within t , by multiplying E_{rel} with t . From the expected distance d and the transmission range r , we immediately obtain $A(r, d)$ from (4.4), i.e., the size of the area uncovered within the neighbor time-out interval t . As node B is static and uniformly and independently distributed, the probability p that B is out of transmission range after node A has moved to A' equals the ratio of $A(r, d)$ to the size of the whole transmission range $r^2\pi$.

$$p = \frac{A(r, d)}{r^2\pi} = \frac{A(r, E_{rel}(v_{min}, v_{max}) \cdot t)}{r^2\pi}$$

In other words, p percent of all entries in the neighbor table are not valid and correspond to nodes, which are no longer available. In Fig. 4.2 the respective values are given for transmission radii of $r = 250\text{ m}$ and $r = 100\text{ m}$ and different speed intervals.

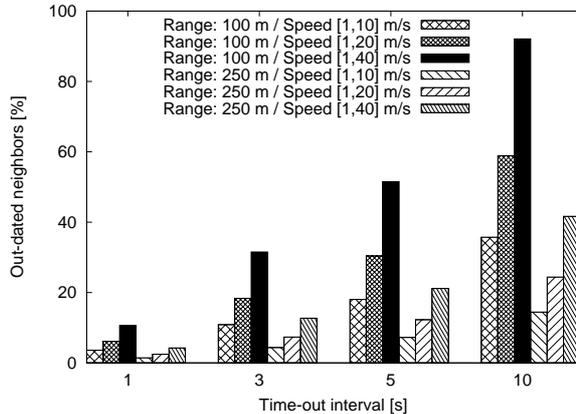


Figure 4.2: Expected percentage of outdated neighbors

Even for a large $r = 250\text{ m}$ and for slow $v_{max} = 10\text{ m/s}$, the percentage of outdated entries is in the order of 10% for time-out intervals of 5 s or more. For very high-speed scenarios with $v_{max} = 40\text{ m/s}$ and long time-out intervals of 10 s, we may expect more than 40% of the nodes listed in the neighbor table to be actually unreachable. We can observe that the number of outdated neighbors is almost inverse proportional to the transmission radius. A 2.5 times smaller transmission radius yields an about 2.5 times higher probability p . A similar proportionality also holds for the relative speed E_{rel} and the number of outdated neighbors. This rough estimation of the percentage of outdated neighbor entries does not account for other factors as discussed in Section 4.2. The probability for an entry in the routing table to become outdated depends also on the distance to the respective node, which was not considered in the analysis. Therefore, as routing protocol typically select a next hop close to the boundary of the transmission range, the percentage of unreachable next hops selected by the routing protocol will be even higher.

For symmetry reasons, we can calculate completely analogously the size of the area newly covered of a node's transmission range, i.e., the number of nodes that are within transmission range but are not yet discovered because no beacons were received from them yet. The only difference is that t is no longer the time-out interval but the beacon interval. As the beacon interval is normally much shorter, the number of undiscovered neighbors is only a fraction of the number of outdated neighbors.

These considerations give an indication for the severeness of the problem of inaccurate neighbor tables, which occur frequently and have a non-negligible impact on the performance of position-based routing protocols. This also provides justification to reconsider position-based routing protocols and try to assess the impact by simulations and evaluate simple optimizations, which help to improve the accuracy of neighbor tables.

4.4 Simulations

In this section, we simulate a standard position-based routing protocol over various scenarios and propose and evaluate several optimizations. Therefore, we first try to identify an appropriate simulation scenario, which produces significant results and permits to assess more easily the goodness of the optimizations. Afterwards, we evaluate the performance of two optimal routing protocols, which have completely accurate neighbor tables by using the global simulator data and thus never select unreachable nodes as next hops. The significantly better performance compared to GFG/GPSR motivates to propose and evaluate possible optimizations, whose objectives are the improvement of the neighbor table accuracy.

4.4.1 Parameters and Scenarios

In all simulations of this chapter we used GFG/GPSR as the underlying position-based routing protocol, cf. Section 3.3.4. In accordance with the parameter values chosen in [134], the beacon interval and the neighbor time-out interval are set to 1.5 s and 6.75 s respectively. As also proposed in [134], we implemented changes in the MAC layer protocol to optimize routing and make IEEE 802.11 more robust in mobile wireless scenarios. The most important optimization is that a packet is not dropped if it cannot be delivered, but handed back to GFG/GPSR for rerouting.

The simulations were conducted using the Qualnet network simulator [211] and the results are averaged over eight simulation runs. Radio propagation is modeled with the isotropic two-ray ground reflection model, cf. Section 2.4.3. The transmission power is set to 15 dBm and the receiver sensitivity to -81 dBm corresponding to a nominal transmission range of 250 m. We use IEEE 802.11b DCF with RTS/CTS operating at a rate of 2 Mbps on the MAC layer. (RTS/CTS is often used in wireless multihop networks because it is commonly assumed to alleviate the hidden node problem. However, in the view of the results in [125], there arise some doubts regarding this assumption.) The nodes are placed in a rectangular area of 600 m x 3000 m. The simulations last for 900 s and the nodes move according to the random waypoint mobility model. We implemented the stationary distribution of the random waypoint model as described in Section 2.4.2. Thereby the simulations do not need an initial warm-up phase to reach a stable state. In order to avoid possible synchronization of beacons between neighboring nodes [70], the beacons are randomly jittered by 50% of the respective beacon interval. The interface queue length is set to 1500 bytes. We have one Constant Bit Rate (CBR) traffic flow with 64 Byte packets at a rate of 2 packets per second between a randomly selected source and destination. We choose this low traffic scenario to prevent congestion and interference in order to isolate the effects of inaccurate neighbor tables on the routing protocol.

We first conducted several simulations with the standard GFG/GPSR protocol, i.e., with a beacon interval and neighbor time-out interval of 1.5 s and 6.75 s respectively and without any optimizations. Thereby we are able to identify a challenging scenario such that the impact of the proposed optimizations can be observed more easily. In Fig. 4.3(a) and Fig. 4.3(b), the delivery ratio and the average end-to-end delay are depicted for a speed interval of $[1, 40]$ m/s. We also ran simulations with a speed interval of $[1, 20]$ m/s. The results showed

the same trends with the difference that the delay and the packet loss rate were about 30% and 50% lower respectively. As already previously mentioned, there are three reasons why we decided to use this high-speed interval. First, even though the speed interval may seem high, the average speed of the nodes is only approximately 10 m/s . Secondly, we wanted to have a challenging scenario for the routing protocol to observe more definitely the differences in the results. And finally, we consider mobility as a proxy for any kind of topology changes, which could also be caused by sleep cycles of nodes, interferences, adjustment of transmission and reception parameters, etc. as discussed in Section 4.2.

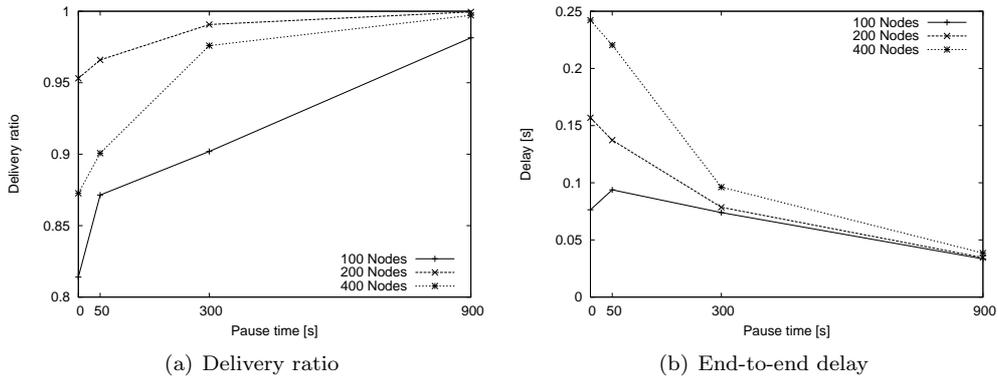


Figure 4.3: Performance of GFG/GPSR with varying mobility and node density

As expected, the performance suffers in case of low pause times for all different kind of node densities. The optimum is for 200 nodes what is approximately 111 nodes per square kilometer (the value also used in [134]). In case of 100 nodes, the density is too low and GFG/GPSR is not able to achieve a high packet delivery ratio because the network is temporarily disconnected and also because face routing is often applied to forward packets, which then may loop. We observed that in highly mobile networks, a large fraction of the dropped packets is due to cycles when face routing is applied in recovery mode. Actually, face routing only guarantees delivery for static networks. The shorter end-to-end delay with 100 nodes is because packets routed over longer paths with longer delays are more likely to be dropped when face routing is applied. Packets received at the destination have often traveled only short paths and thus show a short delay. More surprisingly is the fact that the performance also suffers with a higher node density of 400 nodes. The reason is that the selected next hop is generally farther away and thus has a higher probability of having left the transmission range causing wrong routing decisions. Therefore, we choose to run all following comparative simulations with 400 nodes in a speed interval of $[1, 40]\text{ m/s}$ and a pause time of 0 s , unless given otherwise. The minimum speed was set to 1 m/s as for a minimum speed of 0 , the average speed of the nodes also approaches 0 , cf. Section 2.4.2.

4.4.2 Optimal Position-Based Routing

We first evaluate two protocols, called BNU (Beacons Not Used) and BL (Beacon Less), which use perfect neighborhood information provided by global data of the simulator to determine the next hop. This enables us to do a kind of “best-case” simulation analysis for position-based routing. Except for these accurate neighbor tables, BNU and BL are identical to GFG/GPSR. These optimal protocols allow assessing explicitly the impact of inaccurate neighbor tables and beaconing on the performance. In BNU, nodes broadcast beacons as with GFG/GPSR, but the position information is not used, but taken from the global data. Comparing BNU and GFG/GPSR, we can quantify the influence of inaccurate and outdated neighbor tables. In BL on the other hand, the beacon mechanism is disabled completely. The performance difference between BNU and BL is an indicator for the performance loss solely due to the additional traffic caused by beacons, e.g. by collisions with data packets.

The end-to-end delay and the number of retransmitted RTS packets on the MAC layer are depicted in Fig. 4.4. An RTS packet is transmitted by the source prior to the data packet transmission to mitigate the hidden node problem in IEEE 802.11 [115]. The intended receiver acknowledges the RTS with a CTS packet. Afterwards the actual data packet is transmitted by the source and acknowledged by the receiver. An RTS retransmission occurs if the source does not receive the CTS from the next hop within a certain time-out interval. In our scenario with very little traffic, RTS and CTS should not collide with other packets. Thus, RTS retransmissions are an indication for the unavailability of the next hop. If the routing protocol selects an unavailable next hop, the MAC layer protocol retransmits seven RTS before giving up the delivery of the packet and handing the packet back to the routing protocol. Consequently, RTS retransmissions are a direct indication for the accuracy of the neighbor tables. Fig. 4.4(c) and Fig. 4.4(d) show the same results as Fig. 4.4(a) and Fig. 4.4(b) on a different scale for clarity reasons, as the difference between BL and BNU are hardly visible. The delivery ratio for both protocols, BL and BNU, was always 100% and thus not shown. Only very infrequently one packet was lost. The delay of BNU and BL is approximately 10 *ms* independent of the pause time. This is much shorter than of the GFG/GPSR with delays between 60 *ms* and 210 *ms*. The much higher end-to-end delay of GFG/GPSR is directly correlated with the number of retransmitted RTS packets, where up to 60'000 RTS packets are sent. As an unreachable neighbor causes seven retransmissions, we can assume $60000/7 \simeq 8600$ wrong routing decisions, where the next hop was not available. With 1400 packets transmitted in total, each packet is tried approximately $8600/1400 \simeq 6$ times to be routed to an unavailable neighbor. Each of these wrong routing decisions adds on average 30 *ms* delay as seen before. Thus, $6 \cdot 30 \text{ ms} \simeq 180 \text{ ms}$ of the total end-to-end delay is caused by wrong routing decisions due to the outdated neighbor tables. On the other hand, we did not observe any RTS retransmissions for BL and only very few for BNU. The reason is that no unreachable nodes are listed in the neighbor table and packets are always routed to neighbors within transmission range. The few RTS retransmissions for BNU are due to collisions of RTS packets with beacons. These retransmissions are also the reason for the 10% higher delay of BNU compared to BL. The delay of the two optimal protocols remains almost constant for all mobility rates. These results indicate that outdated neighbor

tables do not only cause long delays but are also a main reason for packet loss in uncongested networks. For all position-based protocols, which only use local information to forward packets, the delivery ratio and the delay of the two optimal protocols are an upper and lower bound, respectively.

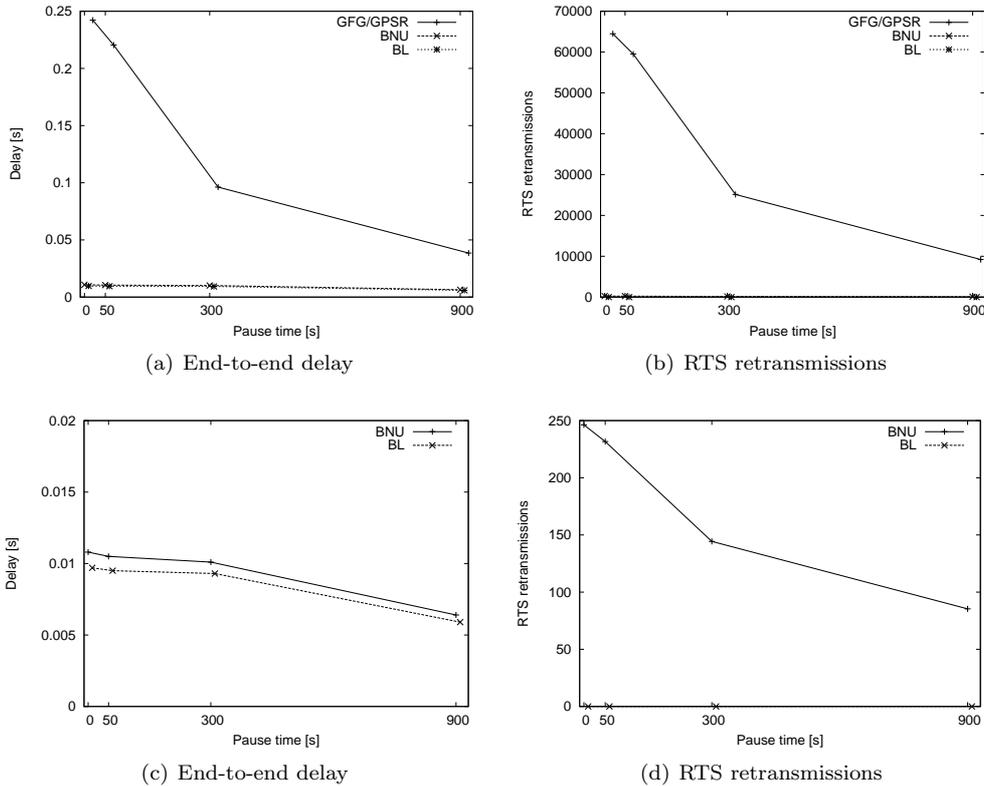


Figure 4.4: Comparison of GFG/GSPR, BNU, and BL

These results provide justification to investigate more in depth approaches to increase the accuracy of the neighbor tables of GFG/GSPR as the performance gap to protocols without wrong forwarding decisions is significant. A number of possible approaches are described and evaluated in the following sections. In a first step, we simply study the performance of different fixed beacon and neighbor time-out intervals. Then, we improve the accuracy of neighbor information by adapting the beacon interval according to the nodes' mobility. In a third approach, nodes close to the boundary are not considered neighbors as they have the highest risk to become outdated. The fourth approach adds additional information to the beacons such that nodes can estimate future positions of neighbors. In a last approach, we make beaconing reactive. A node requests its neighbors to transmit a beacon only when it has a packet to send. For reason of simplicity, each approach is considered separately, even though it is possible to use them in combination.

4.4.3 Fixed Beacon Intervals

We try to assess the impact of different fixed beacon intervals B and fixed neighbor time-out intervals D on the performance, i.e., all nodes have the same beacon and time-out intervals during the whole simulation. We denote the ratio of the neighbor time-out D to the beacon interval B as k . Similar simulations were conducted in [134] where it was found that the values $B = 1.5$ s and $D = 6.75$ s are appropriate for GFG/GPSR. Many other authors also have chosen high ratios k . In our simulations, we did not only want to study the impact of a longer or shorter beacon interval, but also the impact of the ratio k between beacon and time-out interval. The obtained simulation results for $k = 2$ and $k = 4.5$ are given in Fig. 4.5.

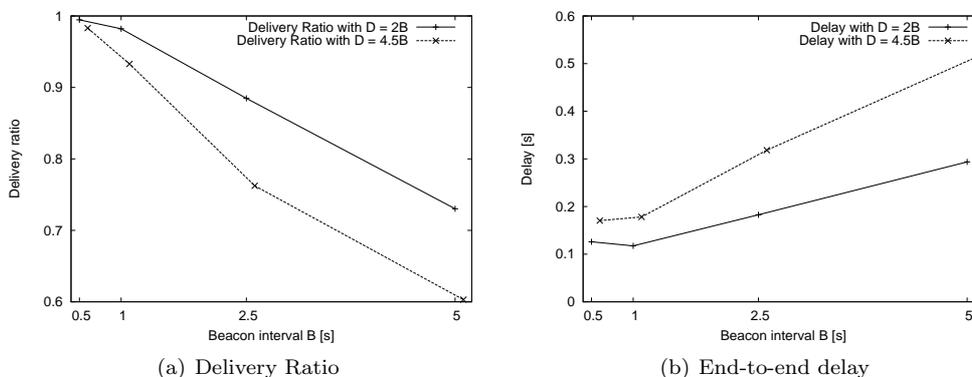


Figure 4.5: Time-based beaconing

The time between two consecutive beacons can be up to $2 \cdot B$ due to the 50% jitter, the time needed by IEEE 802.11 to acquire the medium, and the transmission delay. For shorter time-out intervals than $2 \cdot B$, nodes could be removed from the neighbor table between two consecutive beacons erroneously even when no beacons were missed. The results indicate that a smaller beacon interval generally increases the reliability of the network. A shorter neighbor time-out interval increases the delivery ratio and at the same time also decreases the end-to-end delay. The best results are achieved with $B = 1$ s and $D = 2 \cdot B$. The prolongation of the beacon and the time-out intervals degrades the performance significantly. Shorter intervals however incur additional network load and power consumption by the higher frequency of broadcasted beacons. A pure time-based approach to improve neighbor table accuracy has therefore several shortcomings. On one hand, intervals may be too short and induce unnecessary transmissions if nodes are almost immobile or the network is congested. On the other hand, the interval may be too long for highly dynamic networks with fast moving nodes causing nodes to forward packets frequently to unreachable neighbors, e.g., two nodes on the highway heading in opposite directions may only be within transmission range for a few seconds. In the following, we try to cope with these circumstances by making the beacon and time-out intervals adaptive to the movement and the speed of the nodes.

4.4.4 Adaptive Beacon Intervals

We evaluate two approaches based on traveled distances and nodes speeds. They improve the accuracy of the node positions in the neighbor table by adapting the interval between beacons B and as well the neighbor time-out interval D .

Distance

In the distance-based approach, a beacon is sent whenever a node has moved a given distance d , a “beacon distance”, since its last transmission. Furthermore, we introduce two different methods to determine the neighbor time-out interval. The first one follows the same idea as the time-based approach, i.e., a simple fixed ratio between beacon and time-out interval. A node deletes an entry if it has moved more than k -times the distance d , or after a maximal time-out of 10 s . Consequently, the neighbor time-out interval is the minimum of $[k \cdot d, 10\text{ s}]$. Actually, the term time-out interval is somehow misleading as it is also distance based. However, we keep the term “time-out interval” for all approaches to remain consistent. In the second approach, a neighbor is only deleted from the neighbor table after 10 s independent of the distance the node moved. For the distance-based approach, nodes have to store additionally their positions each time they receive a beacon with the corresponding entry. Similar as in the time-based approach, we conducted simulations with two different values of k . The values are set again to $k = 2$ and $k = 4.5$. With the distance-based approach, we hope to map the movement of nodes to the beacon and neighbor time-out interval. Fast moving nodes send beacons frequently, whereas slow moving nodes send beacons less frequently. Problems with the distance-based approach arise if nodes move at significantly different speeds. Slow nodes only infrequently transmit beacons and a fast moving node passing by may only be within transmission range for a few seconds. Likely, the fast node will not detect the slow moving nodes and thus perceive a reduced connectivity of the network which makes it more difficult to forward packets efficiently.

As depicted in Fig. 4.6(a), we have the best delivery ratios of approximately 94% for distances between $d = 10\text{ m}$ and $d = 20\text{ m}$. The results indicate that it is necessary to make also the time-out interval adaptive to the moved distances. A pure time-based D is not able to cope efficiently with fast moving nodes and entries are deleted too late. For shorter distances, the delivery ratio decreases due to the increased network load caused by the large number of beacon transmissions. A fast node at 40 m/s may transmit up to eight beacons per second. Unlike the time-based approach, a higher $k = 4.5$ does not perform much worse than $k = 2$. The reason is that fast nodes remove entries in the neighbor tables quickly even for $k = 4.5$ whereas in the time-based approach, entries are kept in the neighbor table independent of the node speed. The delivery ratio for $k = 4.5$ is even better than of $k = 2$ for short distances d because beacons collide frequently and cannot be received at the neighbors. With $k = 2$, already one not received beacon may cause the respective node to be removed from the neighbor table, which causes a high fluctuation and many nodes within transmission range are temporarily not listed in the neighbor tables. Thus, nodes perceive a lower connectivity of the network than it actually is. For $k = 4.5$ up to four beacons can be missed before a node is deleted from the neighbor table, which increases the perceived connectivity. In this case, wrongly listed neighbors harm

less than the removal of too many nodes within transmission range with respect to the delivery ratio. Higher k values may increase the delivery ratio, but still the delay is always shorter for smaller k values. This can be explained by the fact that the increased delivery ratio comes at the cost of numerous attempts to forward packets to unreachable neighbors. For shorter k values, the delivery ratio may suffer due to the poor perceived connectivity, but if a packet is delivered, it is so without choosing too many unreachable neighbors. Furthermore, the end-to-end delay is only improved if the neighbor time-out interval is the minimum of the covered distance and the maximal time-out of 10 s and not for a simply time-based time-out.

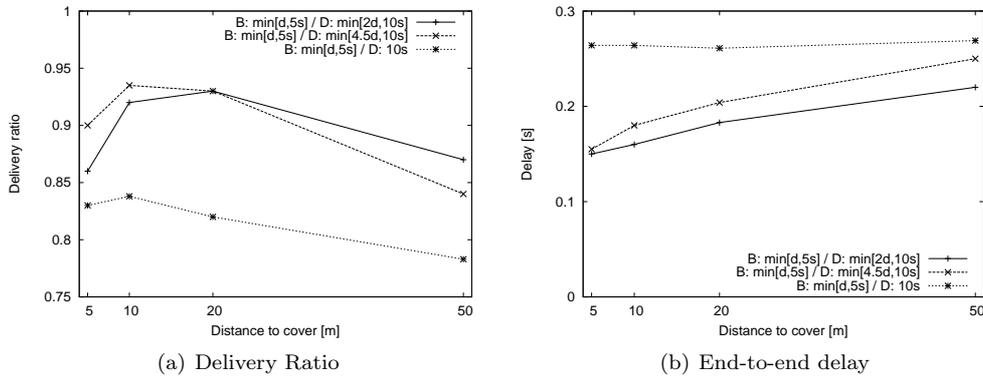


Figure 4.6: Distance-based beaconing

Speed

In the speed-based approach, the beacon interval B and the neighbor time-out interval D are correlated to the speed a node is moving at. Each node calculates its neighbor time-out interval D again as a multiple k of the beacon interval B . Unlike before, nodes send their calculated values of D in their beacons. A receiving node then determines the time-out for this neighbor as the minimum of the neighbor's D as indicated in the beacon and its own D calculated from its current speed. With this enhancement, we hope to overcome the drawback from the distance-based approach where the determination of a correct time-out interval between two nodes moving at different speeds is not solved satisfactory. The beacon interval B can be determined using either a discrete or continuous function of the nodes' speed within a predefined time range $[a, b]$. The continuous function to calculate the beacon interval B is given in (4.5) where v indicates the current speed of a node and v_{max} and v_{min} the maximal and minimal speed a node can move at, respectively.

$$B = a + (b - a) \cdot \left(\frac{v_{max} - v}{v_{max} - v_{min}} \right)^n \quad (4.5)$$

We set the range of the functions to $[1 \text{ s}, 5 \text{ s}]$ and thus have $a = 1$ and $b = 5$. We conducted simulations with three different values for n . For $n = 1$, the mapping of the speed to the beacon interval is linear and for $n = 2$ and $n = 4$ we obtain

polynomial functions. The corresponding graphs are depicted in Fig. 4.7.

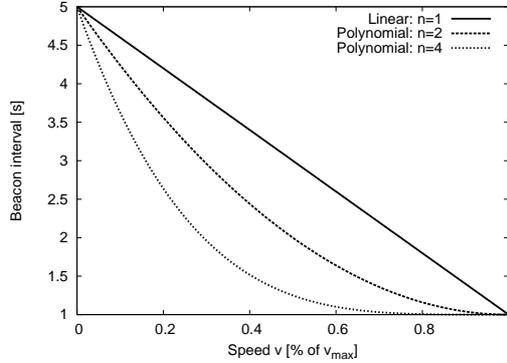


Figure 4.7: Linear and polynomial mapping of speed to beacon interval

The discrete function for the mapping of the speed interval $[1, 40] m/s$ to the beacon interval is given in Tab. 4.2. The end-to-end delay and the delivery

Table 4.2: Beacon interval at different speeds

Speed $[m/s]$	Beacon Interval $[s]$
1-5	5
5-10	3
10-20	2
20-40	1

ratio for these three functions are given in Fig. 4.8 with $k = 2$, i.e., the time-out interval D is always $2 \cdot B$. The discrete and polynomial function with $n = 4$ perform very well whereas the linear function is only slightly better than the standard time-based approaches, cf. Fig. 4.5. With $n = 2$ the performance is about in between the two others as expected, since the graph is still similar to the linear function. The better performance of the discrete and polynomial function with $n = 4$ is due to the distribution of the speed of the nodes. It was shown in [179] that more nodes move at lower speed than at higher in the random waypoint mobility model. This results in an average speed of $10 m/s$, only almost half of the arithmetic middle of the speed interval $[1 m/s, 40 m/s]$. The reason is that fast nodes arrive more quickly at their destination and then have a uniform probability of choosing a low speed. The linear function does not account for this fact. The polynomial function distributes beacon intervals over a larger range for low speeds. The discrete function was defined with the same objective in mind. The delay is reduced to around $120 ms$ and the delivery ratio increased at the same time to 94%. These results are very promising compared to the 87% delivery ratio and $210 ms$ of the standard GFG/GPSR in the same scenario, i.e., the delay and packet loss rate could be about halved.

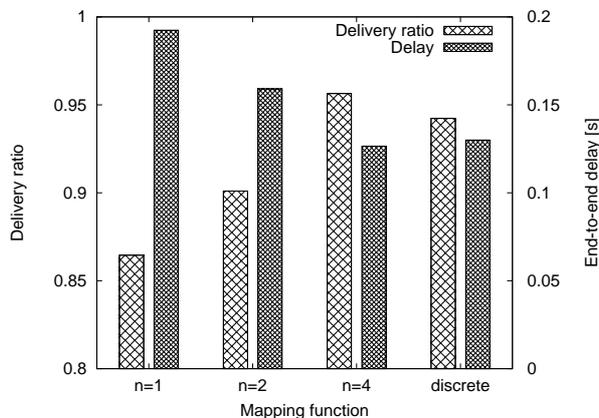


Figure 4.8: Speed-based beaconing

4.4.5 Receiver Threshold

Most position-based protocols like GFG/GPSR forward packets to the neighbor that minimizes the distance to the final destination. Thus, this neighbor is normally located close to the transmission boundary. Exactly these nodes however have the highest probability of becoming unavailable. By introducing a receiver (Rx) power threshold, we can create a circular gray zone at the transmission range boundary. Beacons received at a power level less than this Rx-threshold are not processed, i.e., nodes in this zone are not considered as neighbors and data packets are not forwarded to them. Unlike beacons, data packets received from these nodes are processed and forwarded as normal. In reality, transmission ranges may be highly irregular due to obstacles and interferences. The use of an Rx-threshold instead of a distance-based threshold has the advantage that it allows to cope with irregular transmission ranges.

The physical layer of IEEE 802.11 has a typically receiver sensitivity of approximately -81 dBm . Together with the transmission power of 15 dBm , this determines the maximal transmission range of 250 m in the two-ray ground reflection model. We conducted simulations with several Rx-threshold between 79 dBm and 71 dBm . We can map this power levels to distances of 223, 199, 177, 158, 140 m when using the two-ray ground model where the signal attenuates with $\frac{1}{d^4}$ for distant nodes. Thus, only beacons from nodes closer than these distances are processed.

The delivery ratio first increases and reaches its maximum of 95% for a Rx-threshold of 75 dBm as shown in Fig. 4.9. For higher thresholds, the values degrade again. This behavior is as expected, because a larger threshold reduces the effective transmission range of a node and, thus, the number of neighbors. For a too high threshold, the connectivity of the network is not guaranteed and packets start being dropped because no path exists to the destination. The hop count increases steadily from about six hops to over 10 hops because for a higher Rx-threshold the distance to the neighbors is limited. At the same time, the end-to-end delay is constantly reduced for higher thresholds. The reason is that a higher threshold reduces the wrong routing decisions and thus also the end-to-

end delay, if a packet arrives at the destination. The delay first drops and then remains rather constant as the higher hop count and the time to acquire the medium by IEEE 802.11 at each node introduce delay as well. We may conclude that wrong routing decisions have a bigger influence on the delay than the actual hop count in uncongested networks. A shortcoming of the current Rx-threshold implementation is its inability to select the most appropriate threshold. A fast moving node should only add close nodes in its neighbor table and consequently choose a large Rx-threshold depending on the node density. On the other hand, nodes close to the transmission boundary may be accepted as neighbors for slow moving nodes. Similarly as in the speed-based approach, we could map speed of nodes to Rx-thresholds to solve this problem.

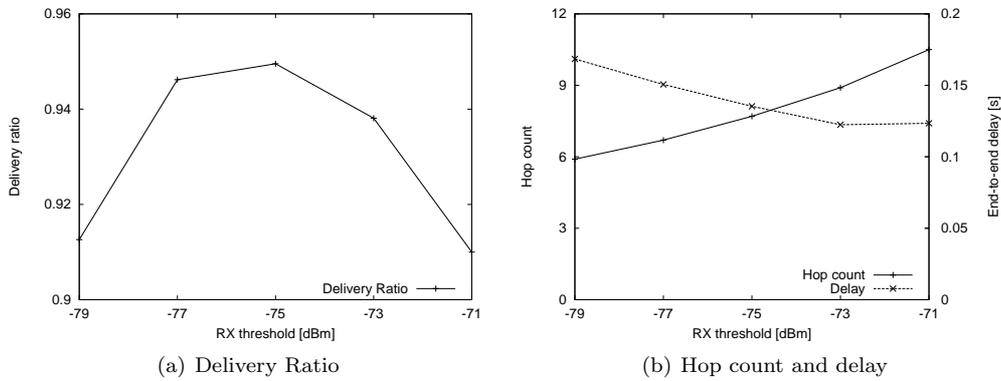


Figure 4.9: Rx threshold for beacons

4.4.6 Estimation of Link Availability

In this approach, the velocity and direction of nodes are used to estimate the time when two nodes are no longer within transmission range. In most cases, a node keeps its speed and direction during a time long enough to reliably predict its future position during the next few seconds. Consequently, each node transmits its current speed and direction in the beacons. Nodes store in their neighbor table all neighbors with their speed vectors and label the entry with the time when the beacon was received. When a node at position A has to transmit a data packet, it calculates the distance to a neighbor that was located at position B and moving with speed vector \vec{b} t seconds ago. This node is predicted to be at position $B' = B + \vec{b}t$. Assuming a circular transmission range r , the neighbor is no longer reachable if the distance $AB' > r$, i.e.,

$$|B + \vec{b}t - A| > r$$

The simulations were conducted with the same beacon and time-out intervals as for the pure time-based approach and also again with two k -values for the ratio between the intervals. Unlike the time-based approach, the delivery ratio and the average end-to-end delay is almost independent whether the neighbor time-out interval is 2 or 4.5 times the beacon interval as seen in Fig. 4.10. The reason is that the prediction of nodes' future positions is quite accurate

also for a time interval of $4.5 \cdot B$. We observe an at least five times shorter end-to-end delay between 25 ms and 50 ms and a less steep increase for longer beacon intervals than in the pure time-base simulations. The link availability is predicted accurately and wrong routing decisions are strongly reduced. Note that this approach again decreases the perceived connectivity because nodes in the neighbor table may be removed early. Furthermore, the delivery ratio increases significantly, e.g. for $B = 1.5\text{ s}$ and $D = 6.75\text{ s}$, we obtained a ratio of approximately 96% compared to 87% with the standard GFG/GPSR without prediction in Fig. 4.3(a).

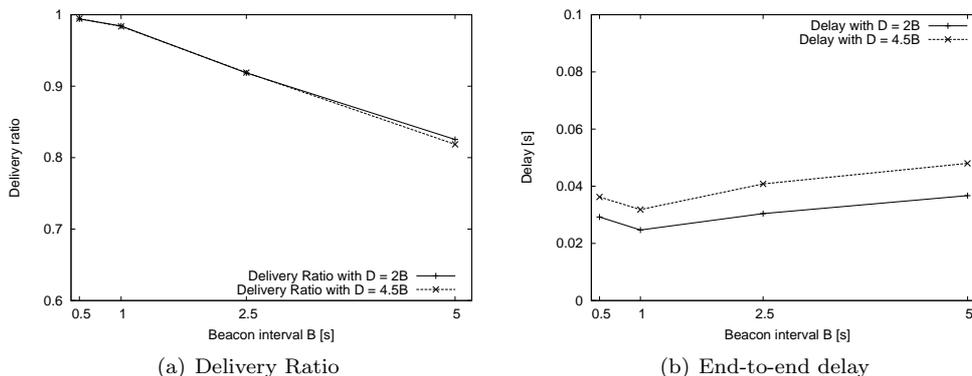


Figure 4.10: Prediction with time-based beaconing

4.4.7 Reactive Beaconing

As already proposed in [134], we make the beaconing mechanism of GFG/GPSR fully reactive. Only when a node has to transmit data packets, it solicits beacons from its neighbors by transmitting a beacon request packet. Each node overhearing this request replies with a beacon to announce its position. Nodes randomly jitter the transmissions of their beacons by 1 ms to avoid that all nodes respond simultaneously and packets interfere at the receiver. We conducted two simulations where the requesting node waits 5 ms and 10 ms for incoming beacons and only then it forwards the data packet to the “best” node. This time has to be set much higher than the jitter of the transmissions as neighbors may have to wait some time to acquire the medium if many neighbors transmit almost simultaneously. The neighbor table is deleted and the whole process is repeated for the next packet. Nodes operate on almost accurate neighbor information as the interval between the beacon and effective packet transmission is very small.

With this reactive beaconing, we achieved a delivery ratio of 95% and an average end-to-end delay of 138 ms when the requesting node waits 5 ms as shown in Fig 4.11. The time saved through the more accurate neighbor table outweighs the additional introduced delay of 5 ms per node to acquire neighbor information. As seen before, we can expect 30 ms delay per attempt to route to an unreachable neighbor. Thus, as long as there is more than one wrong routing decision in seven hops, the reactive beaconing should perform better. For a waiting time of 10 ms , the delivery ratio was even further increased to

98%, but at the same time also the end-to-end delay increased to 220 *ms*. The results are promising, especially if we consider that this is a very basic reactive version where no optimizations are implemented, e.g., no caching of positions or overhead packets.



Figure 4.11: Reactive beaconing

4.5 Conclusions

In a first phase, we discussed the reasons for and the possible impact of inaccurate neighbor tables in position-based routing. We showed a strong relation between inaccurate neighbor tables and the reliability and performance of position-based routing protocols. These considerations were emphasized by a theoretical analysis that indicated that outdated entries in the neighbor tables are the rule rather than the exception in dynamic networks. Factors that amplify the inaccuracy are small transmission ranges, long beacon intervals, and high node mobility. The simulations with two optimal protocols supported the analytical results and showed that the delay can increase by more than an order of magnitude due to inaccurate neighbor tables. Furthermore, packet losses in uncongested networks are also mostly due to outdated neighbor information and wrong routing decisions. These analytical and simulation results indicate that improvements of neighbor information are possible and worthwhile.

In a second phase of this chapter, we then proposed and evaluated several optimizations, which alleviate the drawbacks, of an existing position-based protocol GFG/GPSR. Already with these rather basic optimizations, we were able to achieve significant performance gain. However, the optimizations come at a certain cost and several shortcomings remain. They require either a higher frequency of beacon transmissions or a larger size of the beacons, which results in an increased utilization of network resources and additional overhead. Perhaps even more important is that some of the schemes reduce the connectivity by not using all available links in order to minimize the risk of selecting an unreachable neighbor. Especially in sparse networks where nodes have only few neighbors, the network may become disconnected. An unsolved problem is the case of group

mobility often encountered in reality, where nodes move quickly but their relative positions remain invariant. In such scenarios, no beacons may be required as the topology is almost static. The reactive approach, the approach enhanced with prediction, and the speed-based approach showed the best results in our simulations. For delay critical applications in highly mobile networks, a combination of the prediction-based and speed-based GFG/GPSR may be a preferred choice because of the shortest delays. The reactive GFG/GPSR is more appropriate for low traffic scenarios as it eliminates the proactive broadcasting of hello messages and, thus, conserves scarce network resources. One possible application area are sensor networks with strict constraints on power-consumption and where traffic may be transmitted very rarely.

Position-based routing protocols are appropriate in many scenarios, but also have severe drawbacks in others. Especially for highly dynamic networks or networks with strict power constraints, the required knowledge of the local neighborhood causes two fundamental problems. First, the protocols become stateful and thus are exposed to the risk of outdated information. And secondly, the control traffic consumes unnecessarily network resources such as battery and bandwidth. Furthermore, also the performance of the optimized routing protocol studied in this chapter still remains significantly below the optimum. Therefore, it is worth to reconsider the whole concept of position-based routing because these optimizations are only bug fixes and do not address the root of the poor performance, namely that a transmitting node has to select a “best” next hop on an incomplete and inaccurate topology, especially in highly dynamic networks. In Chapter 5, we propose the routing protocol BLR based on an entirely new paradigm. The forwarding decisions are no longer taken at the sender but in a completely distributed way at the receivers. This paradigm allows the design of a routing protocol, where nodes do no longer require knowledge of their neighbors, neither about their position, nor even about their existence. BLR only requires that nodes are aware of their own position, and the position of the destination, which is given in the packet header. This eliminates the two fundamental drawbacks of other position-based protocols. BLR is completely stateless and therefore information cannot become outdated, and BLR does not require proactive control traffic and conserves network resources.

Chapter 5

Beacon-Less Routing (BLR)

5.1 Introduction

In this chapter, we propose a position-based routing protocol based on a new routing paradigm enabled by the broadcast property of the wireless propagation medium. Forwarding decisions are not taken at the sender of a packet, but in a completely distributed manner at the receivers. A sender does not have to be aware of its neighbors and consequently nodes do not have to transmit proactively beacons as in other position-based protocols. Therefore, we call the protocol Beacon-Less Routing Protocol (BLR). A node that has a packet to forward simply broadcasts it. All receiving nodes delay the further forwarding depending on their positions. The protocol ensures that the “best” node among all receivers rebroadcasts the packet first and suppresses the other nodes. BLR is especially tailored for networks with highly dynamic topologies or strict power consumption constraints. It is stateless and thus immune to topology changes and avoids other drawbacks of beaconing such as unnecessary use of scarce resources.

We first describe the BLR protocol and its components in Section 5.2. Some characteristics and properties of BLR are then studied analytically in Section 5.3. In Section 5.4, we evaluate BLR by simulations over a wide range of network scenarios. Furthermore in Section 5.5, we describe the implementation of BLR in a testbed of Linux laptops equipped with GPS receiver and WLAN cards and give results of the conducted real-world experiments. Finally, we conclude the chapter of BLR in Section 5.6. Further information can also be found in the publications about BLR [99], [100], [101], [104].

5.2 BLR Protocol Operation

Like any other position-based routing algorithms, we assume that nodes are aware of their own positions and that the source has the possibility to locate the position of the destination node. However, as the fundamental difference to other position-based routing algorithms, BLR does not require nodes to have information about their neighboring nodes, neither about their positions nor even about their existence. This allows eliminating completely the periodical proactive broadcast of beacons. We assume an isotropic transmission range

with radius r . As usual, packets are uniquely identified by their source node ID and a monotonically increasing sequence number. Furthermore, there is one parameter taken by the algorithm called *Max_Delay*. It indicates the maximum time a packet can be delayed per hop and is used to calculate dynamically the delay at each node. Packets are forwarded in four different modes of operation. BLR routes packets in greedy mode whenever possible. If greedy routing fails, BLR switches to backup mode to recover and route the packet further. In order to reduce the number of broadcast transmissions of greedy mode, BLR has an option to forward packets in unicast mode if neighbors are known. Additionally to these three modes, BLR implements a local routing algorithm such that packets can still be delivered even if the destination node is not at the indicated position. These three modes of operation and the local routing algorithm are described in detail in the next four sections. Afterwards, we briefly discuss further possible optimizations of BLR.

5.2.1 Greedy Mode

If a source node has a data packet to send, it first determines the destination node's position and stores these geographical coordinates along with its own position in the header of the packet. Since the source does not have any knowledge of neighboring nodes, the only thing it can do is to broadcast the packet. All nodes within transmission range receive the packet. The algorithm takes care that one, and only one, appropriate neighboring node is chosen which forwards the packet. The only available information a receiving node has is its own position and the positions of the previous and the destination nodes from the packet header. From these three positions, a node can easily determine if it is located within a specific area relative to the previous node in the direction of the destination. We call this area the *forwarding area* and nodes located within this area *potential forwarders*. Potential forwarders apply a concept called *Dynamic Forwarding Delay (DFD)* to calculate a short additional delay prior to relaying the packet. Nodes outside the forwarding area do not take any further action and simply drop the received packet. The additional delay calculated by the DFD function is in the interval $[0, Max_Delay]$. The node that computes the shortest DFD forwards the packet first and stores its current position in the packet header. The other potential forwarders overhear the further relaying and are suppressed, i.e., they cancel their scheduled transmission of the same packet. This transmission not only suppresses the other potential forwarders, but simultaneously also acknowledges the previous node, called passive acknowledgement [130]. As most MAC protocols do not provide acknowledgments for broadcast packet, e.g., IEEE 802.11, this use of passive acknowledgements has the advantage that we do not require acknowledgments on the network layer. Furthermore, we can also save one transmission per hop by using passive acknowledgments, which may be important in resource constraint networks. If there are no potential forwarders, the packet is not relayed further and greedy routing fails. Thus, a node that does not detect the relaying of a previously broadcasted packet after *Max_Delay* assumes an empty forwarding area and switches to backup mode. If there are always nodes in the forwarding area, greedy routing continues until the packet arrives eventually at the destination node. The destination node is the last node on the path and does not forward the packet further. Thus, as the only node it has to send explicitly an ac-

knowledge. In the following subsections, we discuss the two most important protocol design aspect of the greedy mode, namely the shape of the forwarding area and the applied DFD function.

Forwarding Areas

The forwarding area is calculated at each receiving node and is determined by, and always relative to the positions of the previous and the destination node. The maximum width of the forwarding area has to be less than or equal to the transmission range r to ensure mutual receptions of packet transmissions among all potential forwarders. Otherwise, packet duplication may occur when some potential forwarders are not suppressed. A second fact that has to be taken into account is the size of the forwarding area. A large forwarding area increases the probability of having a potential forwarder and route packets in greedy mode. Furthermore, if the objective is to reduce the number of hops to the destination, the shape of the area should favor nodes located close to the border of the transmission range, i.e., the shape's center of gravity should be located far from the transmitting node. Considering these facts, we propose three different forwarding areas as depicted in Fig. 5.1, namely the sector, the Reuleaux triangle, and the circle. For the sector and Reuleaux triangle, an apex

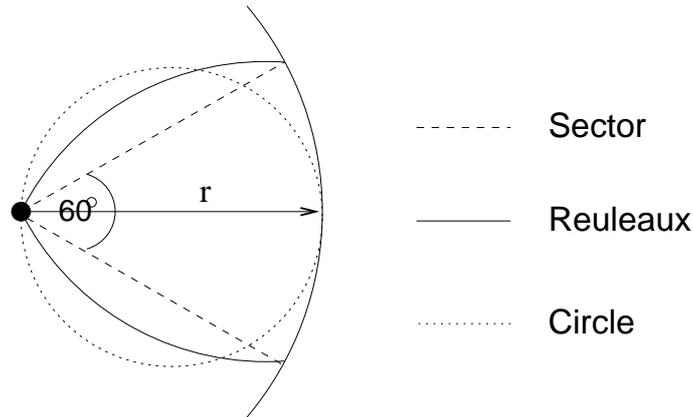


Figure 5.1: Forwarding areas

angle of $\frac{\pi}{3}$ guarantees a maximal distance of r between two arbitrary nodes, whereas the circle simply has a diameter of r . We observe that only a fraction of the whole transmission range is covered by the forwarding areas and is used for routing in greedy mode. The exact numbers for the different forwarding areas are given below.

- Sector: $\frac{1}{6} \simeq 0.17$
- Reuleaux triangle: $\frac{1}{2} - \frac{\sqrt{3}}{2\pi} \simeq 0.22$
- Circle: $\frac{1}{4} = 0.25$

We will see in Section 5.3 that the size and the shape of the forwarding area are critical to the performance of the algorithm.

Dynamic Forwarding Delay (DFD) Functions

With the concept of DFD, nodes calculate a certain delay depending on their position before relaying the packet further. Potential forwarder determine their progress $p \in [0, r]$ towards the destination with respect to the last hop. The progress p is then used to derive the additional delay Add_Delay in the interval $[0, Max_Delay]$.

$$Add_Delay = Max_Delay \cdot \left(\frac{r - p}{r} \right) \quad (5.1)$$

$$Add_Delay = Max_Delay \cdot \left(\frac{e - e^{\frac{d}{r}}}{e - 1} \right) \quad (5.2)$$

These delay functions basically implement MFR, cf. Section 3.3.3. A node with less progress introduces a larger delay than a node with more progress. Consequently, the node with the most progress within the forwarding area forwards the packet at first. The first function (5.1) maps the progress linearly to the range of the delay. Instead of this basic function, more advanced DFD functions are possible. It was shown in [186] that exponentially distributed random timers can reduce the number of responses compared to uniformly distributed timers. This fact is taken into account in (5.2). The resulting Add_Delay is depicted in Fig. 5.2 for all nodes with forward progress. The transmitting node is located at the coordinates $p = 0$ and $q = 0$. Neighbors close to the previous node introduce a long delay, whereas nodes located farther away compute an exponentially shorter Add_Delay .

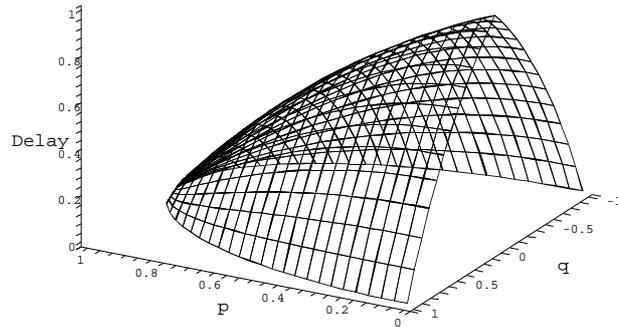


Figure 5.2: Additional delay vs. progress vs. distance

Example

We give a brief example on how routing is performed with BLR. As forwarding area we choose the circle and estimate the delays introduced by the DFD

functions. In Fig. 5.3(a), node S broadcasts a data packet for destination node D . Upon the reception of this packet, nodes A and B determine that they are within the forwarding area and thus use the DFD function to calculate delays $T = 0.1\text{ ms}$ and $T = 0.3\text{ ms}$, respectively. The other neighbors just discard the packet. As node B has more progress, it calculates the short additional delay than A . After 0.1 ms , node B rebroadcasts the packet, which suppresses

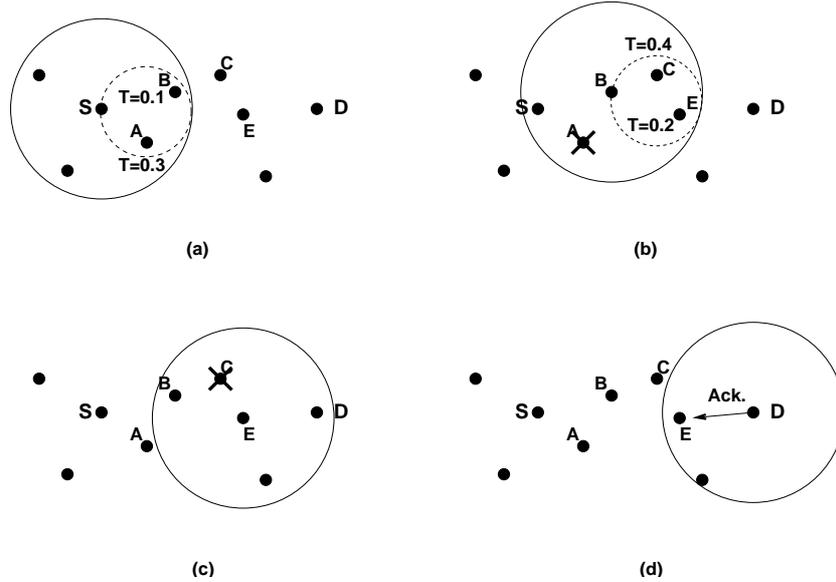


Figure 5.3: Routing with BLR in greedy mode

node A , and simultaneously acknowledges to node S the successful reception in Fig. 5.3(b). The same broadcasted packet is also received at nodes C and E , which have not previously overheard this packet. They calculate a delay of $T = 0.2\text{ ms}$ and $T = 0.4\text{ ms}$ and consequently node E relays the packet further in Fig. 5.3(c). When node D overhears any packet with its destination address, it immediately sends an acknowledgement packet as shown in Fig. 5.3(d), independent of its location. The acknowledgement serves not only the purpose to acknowledge the reception to node E , but also suppress other potential forwarders that scheduled the packet for rebroadcasting.

5.2.2 Backup Mode

If a node does not detect a further forwarding of its previously broadcasted packet within Max_Delay through passive acknowledgment, it assumes that no node is located within the forwarding area towards the destination. Thus, a recovery strategy has to be applied to route the packet further.

The node broadcasts a beacon request packet, which prompts all receiving nodes to transmit a beacon indicating their position. In order to avoid that all neighbors transmit simultaneously and the beacons collide at the receiver, the neighbors also use a DFD function to delay the broadcasting of their beacons by $Add_Delay \in [0, Max_Delay]$. Analogously to the additional delay introduced in greedy mode, the delay is based on the progress, such that neighbors with

more progress towards the destination reply earlier. The corresponding function is given in (5.3) for progress $p \in [-r, r]$.

$$Add_Delay = Max_Delay \cdot \left(\frac{r - p}{2r} \right) \quad (5.3)$$

If there are any neighbors with forward progress, i.e., $p \in [0, r]$, the data packet is sent by unicast to the node with the largest progress and switches immediately back to greedy mode. If no neighbor has forward progress, the requesting node extracts a locally planar subgraph, namely the Gabriel graph, for its one-hop neighborhood. The extraction of the planar subgraph is necessary in order to guarantee loop-freedom as discussed in Section 3.3. The position where the packet entered backup mode is stored in the packet header and the packet is forwarded by unicast according to the right-hand rule. The next node repeats the same procedure, i.e., it broadcasts a beacon request packet, extracts the Gabriel graph from the positions of its neighbors indicated in the beacons, and forwards the packet according to the right-hand rule. A packet is forwarded in backup mode until the packet is received at a node located closer to the destination than where it entered the backup mode. This recovery algorithm is based on face routing, cf. Section 3.3.3, which is also used in GFG/GPSR [24, 134] to recover from local minima. Analogously, the backup mode could be further optimized by a branching ellipse as proposed in GOAFR [149].

The delay increases if a node has to wait at least Max_Delay to receive the beacons from all neighbors. We can minimize the delay, but only at the node that switches to backup mode. This node forwards a packet as soon as it receives a beacon from a node with forward progress. As the forwarding area is small and covers only about half all nodes with forward progress, there is a large probability that there is such a node. And only if no neighbor has forward progress, the node must wait until all neighbors have replied. The reason is that if the packet is forwarded to the first replying node with backward progress, this neighbor may not belong the Gabriel graph and, thus, loop-freedom is no longer guaranteed. Similarly, loops may occur if any other node than the one that switched to backup mode forwards a packet to a neighbor with forward progress. Consider the example depicted in Fig. 5.4, where node S has a packet to transmit for destination D . There are no potential forwarder and S requests its neighbor to transmit a beacon. As soon as node S receives a beacon from node G it forwards the packet by unicast to G and BLR continues to operate in greedy mode. However, let us assume that either node G is not there or its beacon was not overheard at S . After Max_Delay and having received beacons from all neighbors, the packet will be routed over the indicated path. S does not route the packet directly to node B because the connecting link does not belong to the Gabriel graph. Node A continues to route in backup mode as its distance to D is larger than of S . If it switched to greedy mode again because there is a neighbor with forward progress, namely node S , the packet would be sent back and forth between A and S . Furthermore, after node B requested beacons from its neighbors, it does not forward the packet to node E , although node E has forward progress towards the destination. In order to remain loop free, the backup mode must not switch back to greedy mode as long as the packet is not closer to the destination than where it entered the backup mode, i.e., at node F in this scenario.

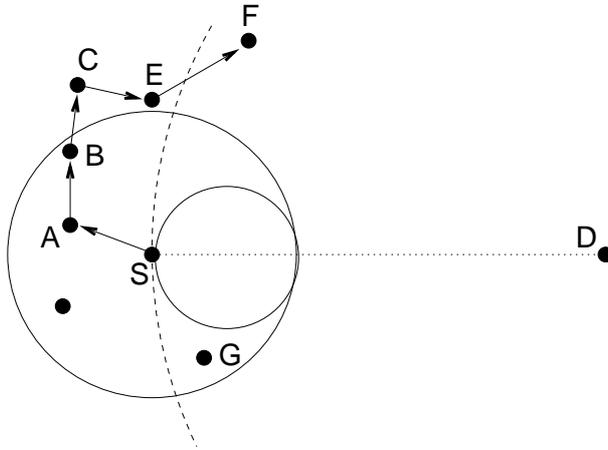


Figure 5.4: Routing with BLR in backup mode

Apart from the higher delay, a further drawback to operate in backup mode is the power consumption. Depending on the number of neighbors n , the additional number of transmissions compared to greedy mode is $n + 1$, namely the n broadcasted beacons and the beacon request packet.

BLR with greedy and backup mode is a fully functional protocol and guarantees delivery of packets in any kind of networks, as long as a path exists between the source and the destination nodes. Obviously, this guarantee is only for theoretical network graphs. Like all other protocols, the delivery is not guaranteed in real networks where packets can be dropped due to collision, loops, etc.

5.2.3 Unicast Mode

Routing in greedy mode has some drawbacks. First, BLR is susceptible to packet duplication as data packets are broadcasted over all hops. Packet duplication occurs for each node in the forwarding area, which does not receive the passive acknowledgement, whether this is the previous transmitting node or any potential forwarder. As we have isotropic transmission ranges with a fixed radius in our theoretical analysis, nodes reliably detect the subsequent forwarding and are suppressed. However in realistic environments, several factors may prevent this suppressing such as unidirectional links and the error prone wireless medium. Secondly, broadcasted data packets need to be passed to the protocol stack at receiving nodes and cannot be dropped at the network interface. And third, greedy mode introduces an additional delay at each node. Even though we will see in Section 5.3 that the additional delay *Add_Delay* is quite small and significantly shorter than the *Max_Delay*.

BLR supports also the transmission of unicast packets. After a node S has received a passive acknowledgment, it is aware of the position of the node B that forwarded the packet as shown in Fig. 5.5(a). Thus, it sends subsequent data packets for the same destination by unicast to node B . Node B relays the unicast packets immediately without any additional delay. Each time node S overhears B 's acknowledgment, node B 's position is updated. Due to mobility, another node A may move to a better position but remain undetected as nodes

do not compete to forward unicast packets Fig. 5.5(b). BLR switches back to greedy mode and broadcasts a data packet periodically to detect new neighbors and prevent from suboptimal routing Fig. 5.5(c1). A packet is also broadcasted as soon as node B has either left the transmission range B'' , or does no longer have forward progress B' to avoid loops Fig. 5.5(c2). The broadcasting of data packets in greedy mode always again yields the best-located neighbor. Consequently, the transmission of a data packet in unicast mode from the source to the destination is a sequence of unicast and broadcast transmissions.

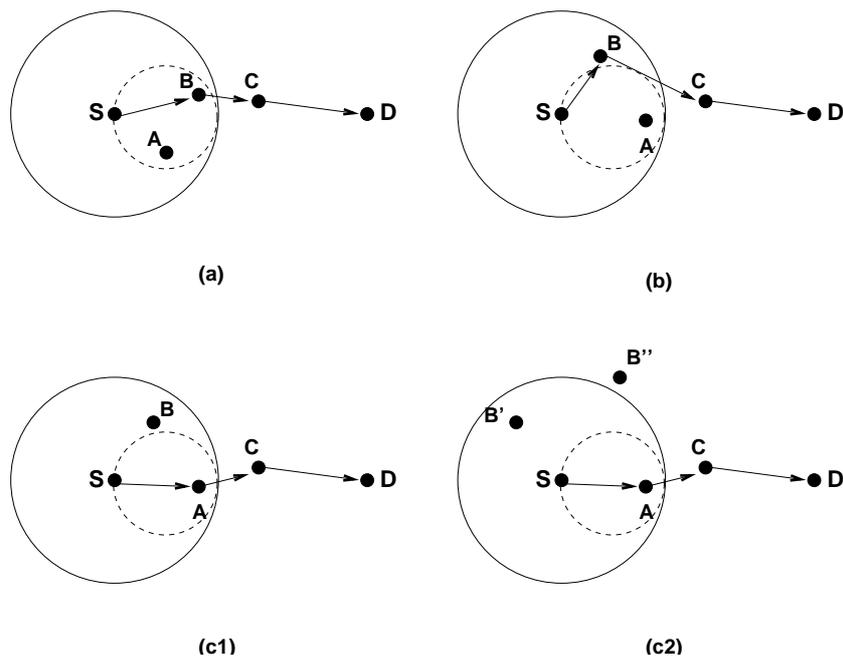


Figure 5.5: Routing with BLR in unicast mode

Unicast mode has several drawbacks apart from the temporarily suboptimal routing because neighbors remains undetected. In greedy mode, the subsequent forwarding passively acknowledges the reception. The explicit acknowledgement of unicast packets introduces an additional transmission. Like in other position-based routing protocols, the caching of the position of neighbors is stateful and information may become stale quickly, which causes suboptimal or false forwarding decisions analogously as discussed in Section 4.2. However, if we assume that several data packets are transmitted per second, the inaccuracy can be kept small as node B 's position is updated frequently. When there are no data packets transmitted for a while, the first packet is again broadcasted in greedy mode such that new neighbors are detected. Furthermore, we could use the same optimizations, e.g., prediction of future nodes' locations, as proposed in Section 4.4 for conventional position-based routing algorithms to minimize the risk of outdated neighbors in unicast mode of BLR. There is always trade-off whether to use unicast or not and the choice depends on the scenario and the requirements.

5.2.4 Reactive Local Routing (RLR)

Basically, it may happen that the position of the destination node does not correspond to the destination position as indicated in the packet header due to several reasons. The position information provided by GPS is sometimes not very accurate as the accuracy depends on the number of satellites in the line-of-sight. If the positions of the destinations are provided by a location service, the positions are often not very accurate because the nodes' positions are updated only from time to time. A node requesting the position of a destination may thus obtain the position of the destination node some time ago.

An intermediate node within transmission range of the destination position, as indicated in the packet header, initiates reactive local routing RLR, if it cannot deliver the packet to the destination and there is also no neighbor closer to the destination position. RLR is based on similar ideas as in [13]. The node A , which initiates RLR, transmits six position requests packets (PREQs) containing its own position. The PREQ packets are sent in six different directions separated each by 60° and with destination coordinates D_1, \dots, D_6 at a distance of twice the transmission range r from the destination position D Fig. 5.6. The fixed number of PREQs and the limited propagation of the PREQs to a specific area around the original destination coordinates help to control the possible number of transmissions.

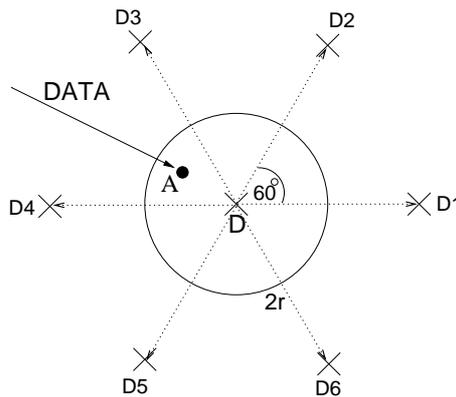


Figure 5.6: Destination position of PREQs

The PREQ packets are routed solely in greedy mode towards their respective destination coordinates. A PREQ is dropped if a node cannot further forward the packet in greedy mode. If the destination node overhears any of the PREQs, it sends a position reply packet (PREP) with its current coordinates back to the position of node A . Node A replaces the destination position in the data packet header with the new position of the destination as indicated in the PREP. The data packets are then routed again as normal by BLR to these new destination coordinates. Node A also intercepts any subsequent data packets for the same destination and replaces the destination coordinates in the packet header.

5.2.5 Options

In the following, we briefly discuss two other possible options that could improve the performance of BLR. They address the problem of packet duplication, which was found to be significant in realistic network scenarios.

Aggregation of Paths

If unicast is applied and a node A is aware of a neighbor B , this information may not only be used for one specific data flow to a destination. All packets with similar destination coordinates can be relayed over the same neighboring node B as long as it has forward progress towards their respective destinations. For example, consider the scenario depicted in Fig. 5.7. Node A transmits packets to destination D_1 over node B . When a packet arrives for destination D_2 , this packet is also sent by unicast to node B . The aggregation of paths allows even further reducing the risk of packet duplication but also bears the risk that a probably better located node C remains undetected.

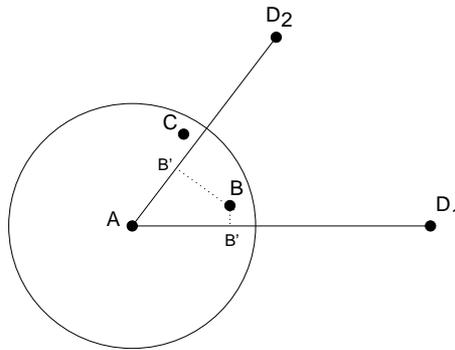


Figure 5.7: Using known nodes for unicast

Integration with RTS/CTS of IEEE 802.11

An other solution to the problem of packet duplication is the integration of BLR with the RTS/CTS dialog of IEEE 802.11. The idea is almost identically to the concept used in the backup mode. A node broadcasts an RTS and all receiving neighbors reply with a CTS, which are delayed according to the DFD in (5.3). Thus, the node, which initiated the RTS/CTS dialog, receives the CTS from the nodes with the most progress first and subsequently forwards the data packet only to the best located node. A further advantage of this integration is, that unlike in greedy mode, all nodes within transmission range are potential forwarders and are not restricted to the forwarding area. However, this integration also has several drawbacks. First, it requires IEEE 802.11 or a similar protocol on the MAC layer that uses an RTS/CTS dialog. Furthermore, unlike in the greedy mode, where only one packet transmission is required per hop, this integration causes four transmissions per hop, namely CTS/RTS/DATA/ACK.

5.3 Analytical Evaluation

For our analytical evaluations, we use the unit disk graph network model, i.e., transmission ranges are isotropic and have a fixed transmission radius of 1. We also assume an unbounded simulation area where the nodes are distributed according to a two-dimensional homogenous Poisson point process of constant spatial intensity.

5.3.1 Probability for at Least One Potential Forwarder

BLR operates in greedy mode as long as there is at least one node in the forwarding area. We can calculate the probability p to have at least one potential forwarder as follows. A characteristics of the Poisson point process is that the number of nodes k in a unit area follows a Poisson distribution. Let A denote the size of a forwarding area, e.g., $\frac{\pi}{4}$ for the circle, and N the total number of neighbors. The size of the whole transmission area is π . We have for the probability that there are k nodes in the forwarding area that

$$P(X = k) = e^{-\frac{N \cdot A}{\pi}} \frac{\left(\frac{N \cdot A}{\pi}\right)^k}{k!} \quad (5.4)$$

Consequently, the probability p for at least one potential forwarder is given by

$$p = 1 - P(X = 0) = 1 - e^{-\frac{N \cdot A}{\pi}} \quad (5.5)$$

This probability function (5.5) is plotted in Fig. 5.8. The circle has the largest

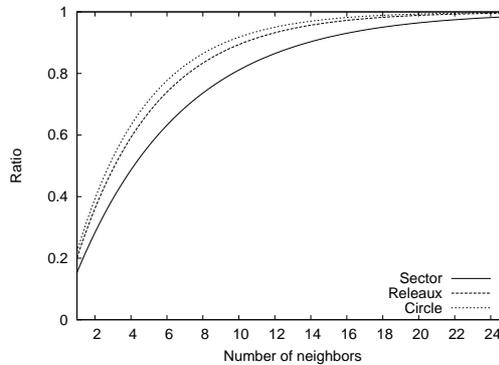


Figure 5.8: Probability for at least one potential forwarder

probability p because its size A is larger than of the Reuleaux triangle and the sector.

5.3.2 Expected Number of Hops before Greedy Mode Fails

Let Y be a random variable that indicates the number of hops before BLR fails in greedy mode, i.e., no node is located within the forwarding area. Y has a geometrical distribution with

$$P(Y = k) = (1 - p)p^k$$

where k is the number of successful hops. The expected value of a random value Y , which is geometrically distributed, is given by

$$E(Y) = \frac{p}{1-p}$$

With (5.5), the expected value $E(Y)$ for the number of successful hops before greedy routing fails is given by

$$E(Y) = \frac{1 - P(X=0)}{P(X=0)} = \frac{1 - e^{-\frac{N \cdot A}{\pi}}}{e^{-\frac{N \cdot A}{\pi}}} \quad (5.6)$$

In Fig. 5.9(a), the expected number of hops in greedy mode is depicted for the three proposed forwarding areas as a function of the number of neighbors. For comparison, we show also the expected number of hops before forwarding fails in greedy mode for a conventional position-based routing protocol where nodes are aware of their neighbors, such as MFR cf. Section 3.3.3. The forwarding area of the MFR protocol is exactly 50% of the transmission range, opposed to the 17% to 25% for the forwarding areas of BLR. This at least twice as large forwarding area for greedy routing results in a significantly higher hop count until greedy routing fails. Despite the rather small differences between the forwarding areas of BLR, the results vary strongly. With an average of 15 neighbors, BLR fails in greedy mode after approximately 40 hops with the circle and within less than 30 and 20 hops for the Reuleaux triangle and the sector. We do not really need to consider node densities of less than approximately seven neighbors, as results in [59, 265] indicate that an ad-hoc network is not connected for lower node densities. The ratios of the expected number of hops for the sector and the Reuleaux triangle to the circle are depicted separately in Fig. 5.9(b). We can see that the ratios are continuously decreasing for higher node densities. Thus, the higher the node density, the more advantageous it is to use the circle as the forwarding area. With 25 neighbors, the number of successful hops with the circle as forwarding area is about twice and five times as high as for the Reuleaux triangle and sector, respectively.

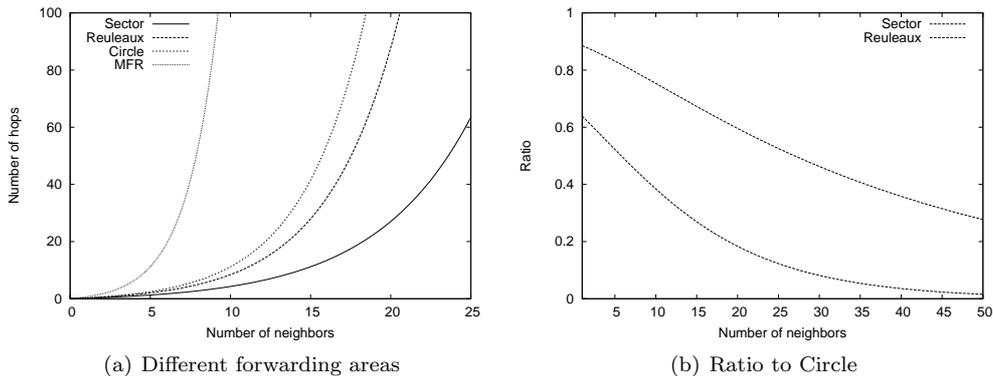


Figure 5.9: Number of hops until greedy mode fails

However, this result does not yet allow to take a conclusive decision which forwarding area is most appropriate. We also have to account for the progress

that is achieved by routing with the different forwarding areas. In Section 5.3.1, we derived a Poisson distribution for the number of nodes located within a forwarding area. If this number is larger than 1, only the node with the most progress, i.e., the node that computes the shortest DFD, relays the packet any further and suppresses the other potential forwarders. Consequently, in order to be able to calculate, e.g., the average progress or the average delay per hop introduced by BLR, we have to take into account the distribution of the maximal progress of the nodes within the forwarding area.

5.3.3 Density Function for the Progress of One Node

In a first step, we derive the cumulative distribution function for a random variable P describing the progress when exactly one node is located within the forwarding area. Since all forwarding areas are symmetrical along the line in the direction of the destination, we may only consider the upper half of the respective forwarding areas as shown in Fig. 5.10.

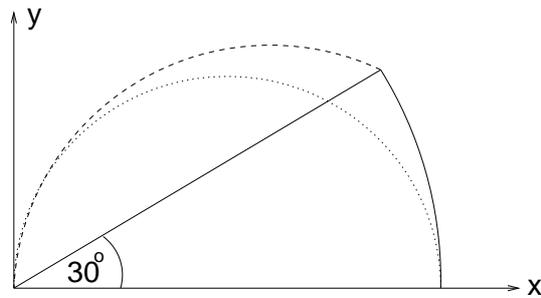


Figure 5.10: Upper half of forwarding areas

We only give explicitly the derivation of the distribution function of the sector. The distribution functions of the Reuleaux triangle and the circle can be derived completely analogously.

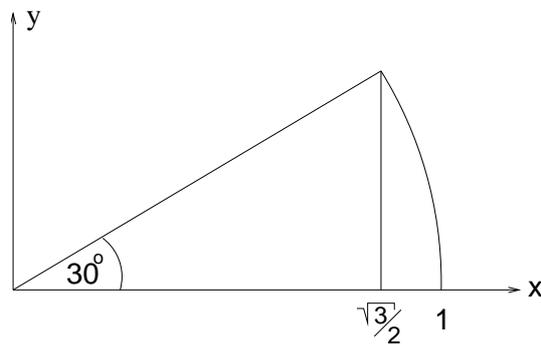


Figure 5.11: Upper half of sector

The sector as depicted in Fig. 5.11 can be described analytically as follows.

$$f(x) = \begin{cases} \frac{1}{\sqrt{3}}x & : 0 \leq x \leq \frac{\sqrt{3}}{2} \\ \sqrt{1-x^2} & : \frac{\sqrt{3}}{2} < x \leq 1 \\ 0 & : \text{otherwise} \end{cases} \quad (5.7)$$

We can calculate the cumulative distribution function $F_P(t)$ for the progress by integrating $f(x)$ from $[0, t]$ with $t \in [0, 1]$ and divide the result by the size of the whole sector. The division is required because $f(x)$ is not the probability density function of P , but only proportional to it as the size of the half forwarding area is not 1.

The size A of the upper half of the sector is given by

$$A = \int_0^{\frac{\sqrt{3}}{2}} \frac{1}{\sqrt{3}}x dx + \int_{\frac{\sqrt{3}}{2}}^1 \sqrt{1-x^2} dx = \frac{\pi}{12}$$

Thus, we have for the distribution function $F_P(t)$ that

$$F_P(t) = \begin{cases} 0 & : t < 0 \\ \left(\int_0^t \frac{1}{\sqrt{3}}x dx \right) \div \frac{\pi}{12} & : 0 \leq t \leq \frac{\sqrt{3}}{2} \\ \left(\int_0^{\frac{\sqrt{3}}{2}} \frac{1}{\sqrt{3}}x dx + \int_{\frac{\sqrt{3}}{2}}^t \sqrt{1-x^2} dx \right) \div \frac{\pi}{12} & : \frac{\sqrt{3}}{2} < t \leq 1 \\ 1 & : t > 1 \end{cases} \quad (5.8)$$

which yields

$$F_P(t) = \begin{cases} 0 & : t < 0 \\ \frac{2\sqrt{3}}{\pi}t^2 & : 0 \leq t \leq \frac{\sqrt{3}}{2} \\ \frac{6t}{\pi}\sqrt{1-t^2} + \frac{6}{\pi}\arcsin(t) - 2 & : \frac{\sqrt{3}}{2} < t \leq 1 \\ 1 & : t > 1 \end{cases} \quad (5.9)$$

This distribution function $F_P(t)$ describe the progress when there is exactly one potential forwarder. In a next step, we derive the cumulative distribution function for the node with the largest progress if there are more than one node in the forwarding area.

5.3.4 Expected Progress

As all the nodes are randomly and independently distributed, we are interested in the distribution of the maximum function of n independent and identically distributed random variables X_i ($i \leq n$), where the distribution function of each X_i is given by the respective $F_P(t)$ as derived in the previous section.

We obtain the distribution of the maximum of n independent and identically distributed random variables X_i with $i \leq n$ by

$$\begin{aligned} F_{\max_{i \leq n} X_i}(t) &= P(\max_{i \leq n} X_i \leq t) \\ &= P(X_i \leq t, \forall i \leq n) \\ &= P(X_1 \leq t, \dots, X_n \leq t) \end{aligned}$$

$$\begin{aligned}
&= [P(X_1 \leq t)]^n \\
&= [F_{X_1}(t)]^n
\end{aligned} \tag{5.10}$$

It is well known that for any given random variable Z and its distribution function F_Z , the expected value $E(Z)$ can be calculated as follows.

$$E(Z) = \int_0^\infty (1 - F_Z(x)) dx - \int_{-\infty}^0 F_Z(x) dx$$

Together with (5.9) and (5.10) this yields the expected progress for the sector

$$\begin{aligned}
E(\max_{i \leq n} X_i) &= \int_0^\infty [1 - (F_{X_1}(x))^n] dx \\
&= \int_0^1 [1 - (F_{X_1}(x))^n] dx \\
&= 1 - \int_0^{\frac{\sqrt{3}}{2}} \left(\frac{2\sqrt{3}}{\pi} x^2 \right)^n dx - \\
&\quad \int_{\frac{\sqrt{3}}{2}}^1 \left(\frac{6t}{\pi} \sqrt{1-t^2} + \frac{6}{\pi} \arcsin(t) - 2 \right)^n dx \tag{5.11}
\end{aligned}$$

Unfortunately, we were not able to integrate analytically these functions for the expected progress for none of the forwarding areas. Thus, the values plotted in the following figures are obtained by numerical integration. In Fig. 5.12, the expected progress $E(\max_{i \leq n} X_i)$ is shown depending on the number of nodes located within the forwarding area. The center of gravity is located farther away from the point of origin for the sector than for the Reuleaux triangle and circle, respectively. Consequently, the expected progress for the sector is the largest followed by the Reuleaux triangle and the circle. For more nodes in the respective forwarding areas, the difference among the forwarding areas gets smaller as all of them asymptotically approach 1.

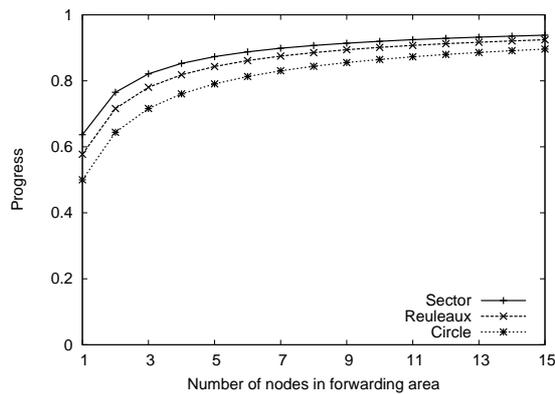


Figure 5.12: Expected progress for the forwarding areas with a given number of neighbors

Until now, we assumed a given number of nodes located within the forward-

ing areas. However, the probability for this given number depends on the node density and the size of the forwarding area, as derived in Section 5.3.1. Given a certain number of neighbors, the circle has the higher probability that there are $n \geq 1$ potential forwarders. The actual expected progress E_P per hop can be derived from (5.4) and (5.11).

$$E_P = \sum_{k=1}^{\infty} e^{-\frac{N \cdot A}{\pi}} \frac{\left(\frac{N \cdot A}{\pi}\right)^k}{k!} E(\max_{i \leq k} X_i) \quad (5.12)$$

The expected progress E'_P under the condition that there is at least one node located within the forwarding area is actually of more relevance than E_P . The reason is that if there are no potential forwarders at all, greedy routing fails anyway and BLR switches to backup mode. The conditional probability for two events A and B is given by

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

For the probabilities of the Poisson distribution from (5.4) given that at least one node is in the forwarding area, we obtain with (5.5) that

$$P(X = k | X > 0) = \frac{e^{-\frac{N \cdot A}{\pi}} \frac{\left(\frac{N \cdot A}{\pi}\right)^k}{k!}}{1 - e^{-\frac{N \cdot A}{\pi}}}$$

because, we have for $k > 0$ that $P(X = k \cap X > 0) = P(X = k)$.

Thus, we obtain for E'_P

$$E'_P = \sum_{k=1}^{\infty} \frac{e^{-\frac{N \cdot A}{\pi}} \frac{\left(\frac{N \cdot A}{\pi}\right)^k}{k!}}{1 - e^{-\frac{N \cdot A}{\pi}}} E(\max_{i \leq k} X_i) \quad (5.13)$$

The graphs for E_P and E'_P are given in Fig. 5.13(a) and 5.13(b). Unlike the previous Fig. 5.12, the differences between the expected progress for the different forwarding areas almost vanish. The reason is that there are fewer nodes in the smaller forwarding areas of the sector and the Reuleaux triangle, which compensates for their advantage in terms of progress. For very low node densities the progress per hops is around 20% Fig. 5.13(a). This is caused by the fact that the probability for no potential forwarder, i.e., an expected progress of 0, is very large. If greedy routing fails, time and resources consuming actions have to be taken anyway in backup mode. Therefore, as explained above, it makes more sense to consider the conditional expected progress E'_P . E'_P is significantly higher than E_P for low node densities. Thus, the progress per hop is still 60% and more, if greedy routing does not fail. For a higher number of neighbors, the probability of having no potential forwarder at all is close to zero and thus the differences between E_P and E'_P become negligible. We cannot compare directly the E'_P among the forwarding areas, because the probabilities for no potential forwarders in the different forwarding areas are also different. Thus, the conditional E'_P is increased disproportionately for the sector and Reuleaux triangle, which explains the larger differences to the circle.

In Fig. 5.13(c), the ratio of the progress in the sector and Reuleaux triangle to the progress in the circle is depicted. The expected progress of the Reuleaux

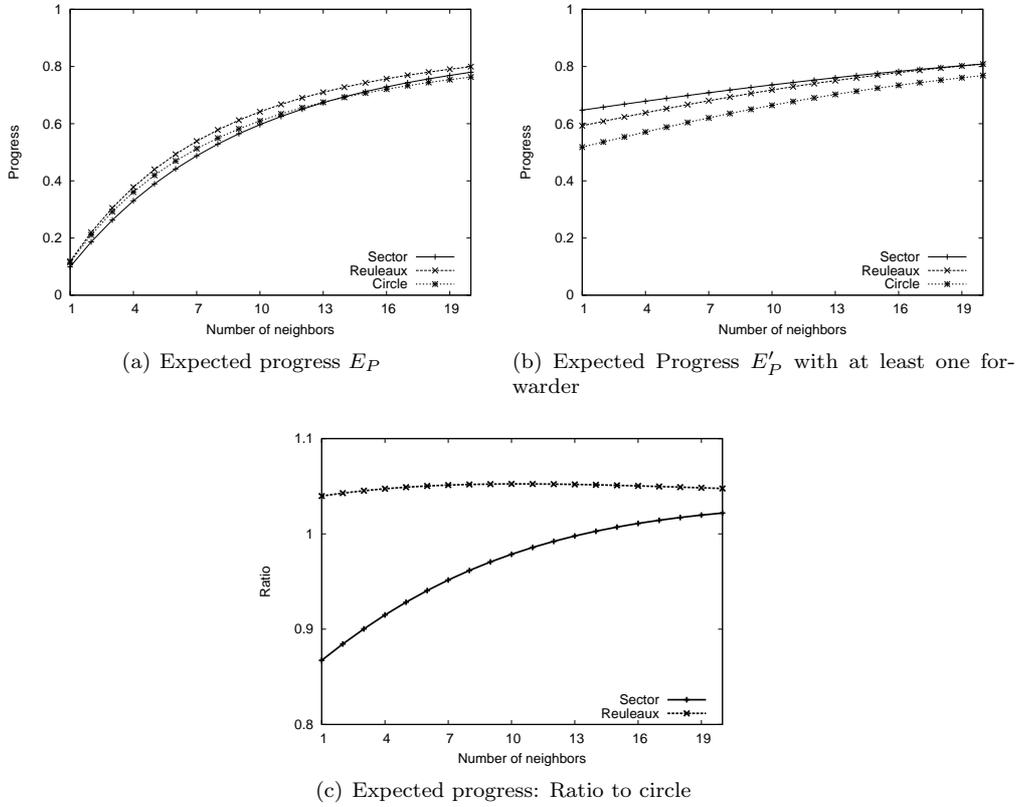


Figure 5.13: Expected progress of forwarding areas

triangle is around 5% higher than of the circle independent of the number of neighbors. Completely unlike the sector, where the progress is less than 90% of the circle for a small number of neighbors. For higher node density, the ratio increases also to over 100% of the circle. Thus, given that greedy routing never fails, it is advantageous to use the Reuleaux triangle as forwarding area in order to minimize the number of hops to the destination.

From these results, we can also estimate the introduced additional delay per hop by the DFD functions. Considering the linear DFD function (5.1), the delay is basically the inverse of the expected progress. Thus, for 10 and more neighbors, the expected progress is more than 70% of the transmission radius, i.e., the expected delay per hop is less than 30% of the maximal delay Max_Delay and can be kept reasonable small.

However, as the probability for greedy routing to fail is not negligible, especially for lower node densities, we have to take the number of hops until routing in greedy mode fails into account as derived in (5.6). In a last step, we now combine the results from the previous sections to derive the Euclidean distance until greedy routing of BLR fails for the different forwarding areas.

5.3.5 Distance Until Greedy Routing Fails

We derive the Euclidean distance a packet can be routed in greedy mode before it fails. This is basically the expected number of hops before greedy mode fails of (5.6) multiplied by the expected progress of (5.12), which yields the expected distance after that greedy routing fails.

$$E_D = \sum_{k=1}^{\infty} \left(1 - e^{-\frac{N \cdot A}{\pi}}\right) \frac{\left(\frac{N \cdot A}{\pi}\right)^k}{k!} E(\max_{i \leq k} X_i)$$

In Fig. 5.14(a), the distance is depicted for the different forwarding areas on a logarithmic y-axis. In Fig. 5.14(b), the ratio of the distance of the sector and Reuleaux triangle to the distance of the circle is again given separately for clarity reasons. The ratio drops continuously for higher node densities, e.g., for 20 neighbors, the ratio drops to around 60% and 20% for the Reuleaux triangle and circle, respectively.

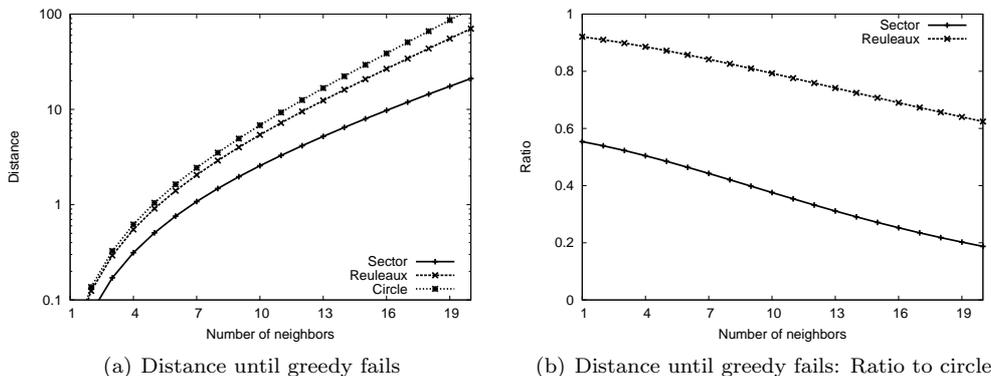


Figure 5.14: Distance until greedy routing fails

Thus, we conclude that the circle is the most appropriate forwarding area as it covers by far the largest Euclidean distance until greedy routing fails, and at the same time the progress per hop is only marginally lower than of the Reuleaux triangle.

5.4 Simulations

5.4.1 Parameters and Scenarios

We implemented and evaluated BLR in the Qualnet network simulator [211]. The results are averaged over 10 simulation runs and given with a 95% confidence interval, which is sometimes very small and barely visible. The payload of the packets is 64 bytes and the interface queue length is set to 1500 bytes. Radio propagation is modeled with the isotropic two-ray ground reflection model. The transmission power and receiver sensitivity are set corresponding to a nominal transmission range of 250m. We use IEEE 802.11b on the physical and MAC layer operating at a rate of 2 Mbps. The simulations last for 900s and data

transmission starts at 180s and ends at 880s such that emitted packets arrive at the destination before the end of the simulation. The simulation area is $6000\text{ m} \times 1200\text{ m}$ and nodes move according to the random waypoint mobility model. We implemented the stationary distribution of the random waypoint model to avoid having an initial warm-up phase. The pause time is set to 0 s because higher pause times lead to pseudo static networks, cf. Section 2.4.2. Instead of using the pause time as the parameter of the simulations, we varied the number of nodes since the node density is a more critical factor for the performance of BLR. (We will see that BLR is basically unaffected by mobility.) The minimal and maximal speeds are set to $\pm 10\%$ of an average speed. The average speed was set to 5 m/s to simulate a slightly dynamic network and to 20 m/s for a highly dynamic network. We consider speed as a proxy for any kind of topology changes, caused by either mobility, sleep cycles, interferences, adjustment of transmission and reception parameters, etc. For BLR, the *Max_Delay* is set to 2 ms and the circle is used as the forwarding area, if not noted otherwise. These choices are determined in a first phase where we simulated BLR over a wide range of parameter values and the unicast and backup mode selectively turned on/off. Afterwards, we compared the performance to GFG/GPSR in several specific scenarios. The parameters of GFG/GPSR are set as suggested in [134], i.e., beacons are transmitted at an interval of 1.5 s and the neighbor time-out interval is set to 6.75 s . We did not use more sophisticated protocols like GOAFR [149] due to two reasons. First, GOAFR is identical to GFG/GPSR for greedy routing and only reduces the hop count if packets are routed in face routing mode to recover from local minima. We are more interested in scenarios where we have high mobility and greedy routing is possible most of the time. And secondly, as already mentioned before, BLR could also implement the optimized face routing mode of GOAFR.

5.4.2 Evaluating Different Parameter Values

To determine the impact of the different parameter choices of BLR such as the forwarding area and the *Max_Delay*, we simulated BLR without the unicast and backup mode if not noted otherwise. The reason is that we are mainly interested in greedy mode routing and the effects of routing packets also in backup or unicast mode are difficult to eliminate a posteriori. Routing packets without the backup and unicast mode means that packets are always broadcasted over each hop and if routing fails in greedy mode, as there is no neighbor in the forwarding area, packets are simply dropped.

Forwarding Area

In a first step, we wanted to determine the effect of the different forwarding areas. These simulations were conducted with a large *Max_Delay* of 10 ms such that the contention among the potential forwarders is minimized and only few packets are dropped due to collisions. In Fig. 5.15, the delivery ratio of the three forwarding areas is depicted for an average speed of 5 m/s and 20 m/s . The number of nodes was varied from 250 to 1000 nodes, i.e., from a very sparse network with approximately seven neighbors to a dense network with 27 neighbors. For low node densities, greedy routing fails and many packets are dropped for all forwarding areas. Only for 750 nodes or more, the node density

is high enough that greedy routing is able to deliver reliably packets. This is the same node density also used in other papers about position-based routing protocols, e.g. [134]. As we were interested in a performance comparison of the three forwarding areas with greedy routing under lower and higher contention among the potential forwarders, we also conducted simulations with lower node densities. The circle and the Reuleaux triangle are almost equal with respect to the delivery ratio. As expected from the analytical assessment in the section 5.3, the delivery ratio with the sector is significantly lower. Furthermore, we can observe that the results are identical for both average speeds. The performance of BLR does not suffer under mobility and changing topology because it is stateless and does not store neighbor positions, which can become outdated. Thus, we only will use the highly mobile network scenario with an average speed of 20 m/s and the circle as the forwarding area from now on. The reasons to use the circle are that the simulation results are almost identical and the analytical results even marginally better for the circle than for the Reuleaux triangle. Considering the fact that transmission ranges are irregular in realistic scenarios, the circle may also be more appropriate than the Reuleaux triangle because more area is centered at the middle and farther from the boundaries of the forwarding area.

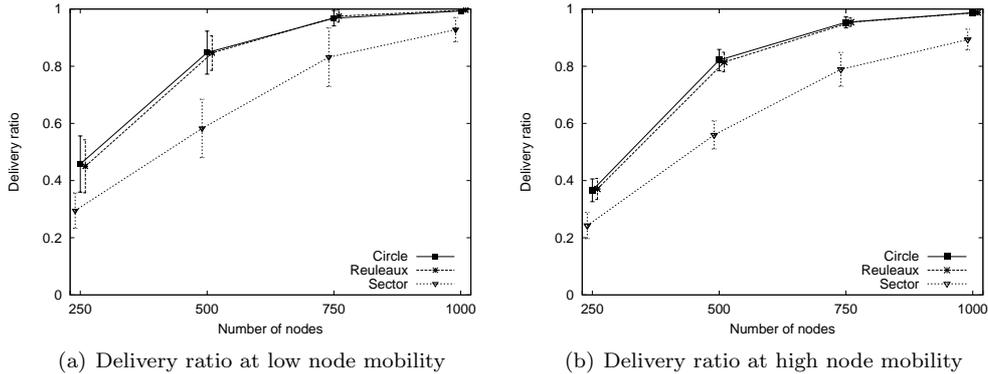


Figure 5.15: Comparison of different forwarding areas

Max_Delay

A very critical parameter for the performance of BLR is *Max_Delay*. A short *Max_Delay* would reduce the introduced delay per hop but would also increase contention among potential forwarders, such that the overall delay may be even higher than for a higher *Max_Delay*. The higher contention may also increase the packet losses. In Fig. 5.16, the results are shown with a varying *Max_Delay*. BLR again is simulated without the backup mode, because the impacts of routing packets in the backup mode on the overall performance are difficult to assess. We only simulated with an average speed of 20 m/s because results from the previous section have shown that the performance of BLR is invariant to network mobility. The delivery ratio is almost unaffected by different values for *Max_Delay*. However, we can observe that the end-to-end delay is increased from 10 ms to about 20 ms for a *Max_Delay* of 2 ms and 10 ms , respectively.

Considering the fact that the average hop count is around 10, the delay per hop is in the order of 1 ms and 2 ms , which is much shorter than the maximal possible additional delay between 2 ms and 10 ms . For a higher Max_Delay , we can observe that the end-to-end delay decreases for a higher node density. We may expect that the potential forwarder with the largest progress is closer to the transmission range boundary for higher node densities and, thus, calculates a shorter Add_Delay . The effect is hardly observable for $Max_Delay = 2\text{ ms}$ as the increased contention on the MAC layer negates the shorter delay.

The reason that the shortest delay was measured with a node density of 250 is simply because only packets arriving at the destination contribute to the average delay. For 250 nodes, the risk of a packet to be dropped because there are no potential forwarders is higher for distant source and destination nodes. Therefore, if packets arrive, the two nodes are rather close, which can be seen from the smaller hop count in Fig. 5.16(c). Consequently, packets also have a shorter delay. Because the results are very definite, we use a Max_Delay of 2 ms in the following simulations.

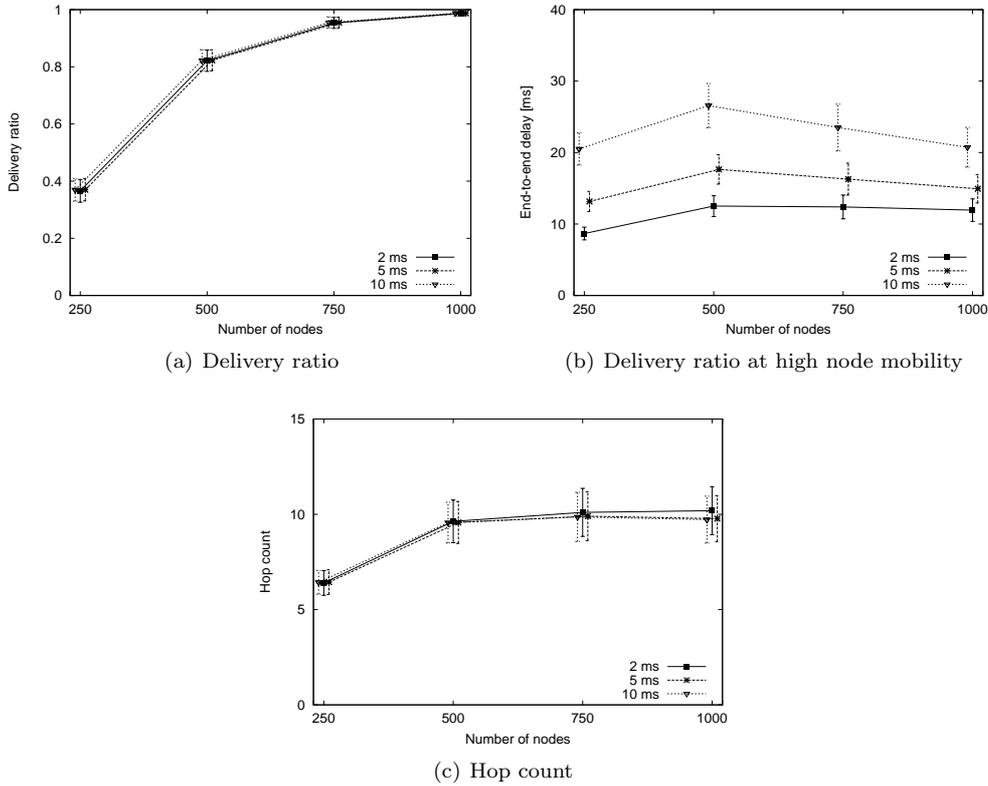


Figure 5.16: Comparison of different Max_Delay values

DFD Functions

We also compared the performance of BLR with the two proposed DFD functions in Section 5.2.1, namely with the linear and the exponential mapping from

the progress to the $[0, Max_Delay]$ time interval. The delivery ratio for the exponential DFD function is marginally higher as depicted in Fig. 5.17(a). The potential forwarders with a large progress, which eventually relay the packets, are distributed over a larger time interval for the exponential DFD function which reduces the risk of collisions. As the backup mode was not enabled in these simulations, a packet may be easily lost if two potential forwarders broadcast simultaneously as none of the nodes in the two overlapping forwarding areas successfully receive the packet. Potential forwarders close to the previous transmitting node have a low probability that there is an other potential forwarder that calculated a shorter *Add_Delay*. Normally these close nodes will not transmit the packet anyway and thus the timing of their retransmissions is not important, i.e., they can be scheduled very close to each other without having an effect. On the other hand, the delay was increased significantly with the exponential DFD function Fig. 5.17(b). The linear function showed a more than two times shorter end-to-end delay. The steep increase of the exponential function yields much longer delays for the distant nodes. For example, if a potential forwarder with a progress of 80% of the transmission radius transmits the packet, the *Add_Delay* is approximately three times longer with the exponential function than with the linear function, cf. (5.1) and (5.2). Due to the significantly shorter delay of the linear function, we apply the linear function in the following simulations. We have to note however that the exponential DFD function is not tailored to the shape of the circle as forwarding area. Therefore, a more appropriate function may help to reduce the delay without affecting the delivery ratio.

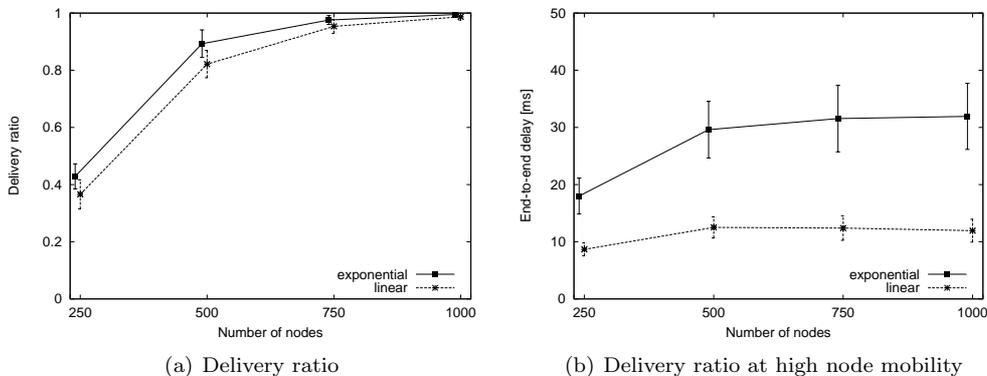


Figure 5.17: Comparison of DFD functions

Varying Packet Size

Until now, we simulated BLR with a rather small packet size of 64 bytes. The overhead of the UDP, IP, TCP, and MAC header sums up to approximately 100 bytes of control information per packet. Nevertheless, the size of the packet remains small. In this section, we simulated BLR with varying payload lengths of 64, 512, and 1024 bytes. The simulations were conducted with 500 nodes. BLR only operates in greedy mode such that packets are dropped if a node can no longer forward the packets in greedy mode and would need to switch

to backup mode. We again did not use the backup mode as its influence on the delay and delivery ratio is hard to predict and distorts the results. The simulation results are depicted in Fig. 5.18 for the three different packet sizes. We observe that the packet size does not have any influence on the delivery ratio. The low delivery ratio of approximately 80% is because greedy mode of BLR fails frequently and also simply because the rather sparse network is not always connected. On the other hand, the average end-to-end delay increases constantly from 11 ms to 27 ms and 48 ms . In order to assess whether BLR really has an influence on the delay, we have to subtract the actual transmission delay from the measured end-to-end delay. Assuming a total of 100 bytes of header information, the transmitted bits on the physical layer per hop are approximately 1300, 4900, and 9000 for the three packet sizes. If we multiply these numbers by the average hop count, which is roughly nine for all packet sizes, and divide the result by the data rate of 2 Mbps, we obtain a pure transmission delay of 6 ms , 22 ms , and 41 ms . Therefore, we can conclude that the measured end-to-end delay is due to this transmission delay and BLR only adds approximately 5 ms to the delay, independent of the packet size.

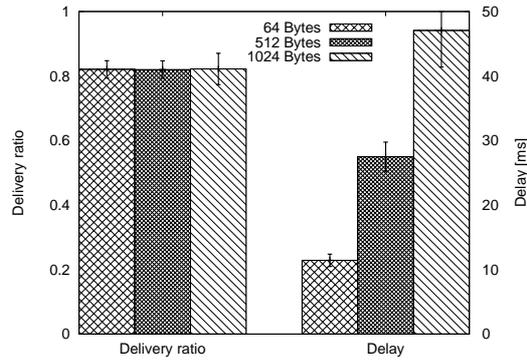


Figure 5.18: Comparison of different packet sizes

Overhead

In this section, we evaluate the overhead induced by BLR. We are especially interested in the number of duplicated packets. Packet duplication is especially critical for BLR as data packets are broadcasted over all hops. Therefore, these simulations were conducted without the unicast and backup mode. As packets are only routed in greedy mode, many packets are dropped in sparse networks because there are no potential forwarders. We had to approximate the number of transmissions induced by packets that were not delivered to the destination. Thus, for higher delivery ratios, i.e., with higher node densities, the calculated values become more accurate. The values for the ratio of duplicated packets to the total number of received packets at the destination are given in Fig. 5.19. The ratio is close to 1, when we use the two-ray ground reflection model with an isotropic transmission range. For sparser networks the ratio increases, but also the confidence interval, which are both caused by the way we approximated the overhead. Considering the error introduced by the approximation, the number of duplicated packets is basically independent of the network density. The

number of duplicated packets is less than 10% for a longer *Max_Delay* of 5 *ms* and 10 *ms*. We can observe here now that a shorter *Max_Delay* has also a certain negative impact. The number of duplicated packets is approximately 10% higher caused by the increased contention among potential forwarders. More potential forwarder will transmit simultaneously as the time is too short to suppress efficiently other potential forwarders with a similar *Add_Delay*. This results in a duplicated packet per potential forwarder that is not suppressed. The situation looks different in case of irregular transmission ranges. We conducted the same simulations with the RIM propagation model with *DOI* = 0.01 and *VSP* = 0.5, cf. Section 2.4.3. A node recalculates its transmission range with the RIM whenever it has moved 50 *m*, which may result in a completely different transmission range. We introduced this behavior to reflect the changes in the transmission range of a node caused by changing environmental factors when the node moves. We can observe that the high inaccuracy of the approximation remains for low node densities. Furthermore, the number duplicated packets is significantly higher than in the two-ray ground reflection model as nodes do no longer suppress each other reliably. We observed between 20% and 60% of duplicated packet depending on the parameters. The number of duplicated packets increases for higher node densities. The reason is that for higher node densities more neighbors are not suppressed by the previous transmitting node, especially if its transmission range is small and does not cover the whole forwarding area. If each packet is acknowledged on the MAC layer as it is done in unicast mode of BLR and basically all other position-based routing protocols where packets are sent by unicast, the overhead is at least 100%. Considering this fact, the number of duplicated packets in greedy mode of BLR with irregular transmission ranges still remains low.

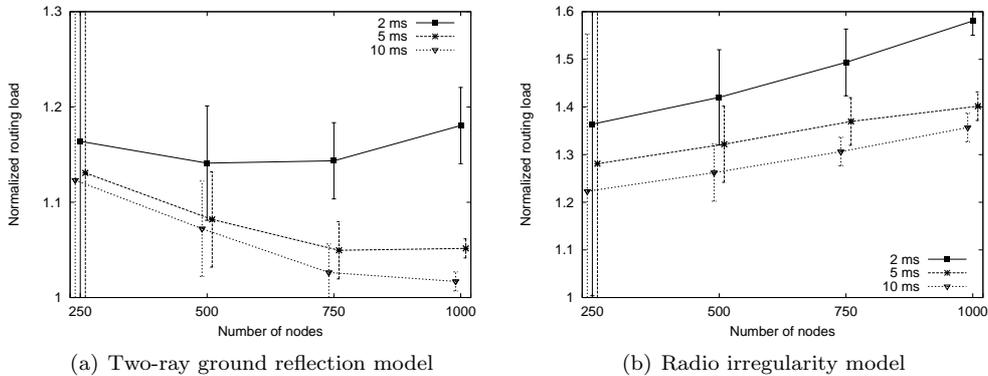


Figure 5.19: Overhead with isotropic and irregular transmission ranges

5.4.3 Impact of Backup Mode and Unicast Option

Backup Mode

Until now, we only evaluated BLR without the backup mode as we were interested solely in the performance of routing in greedy mode and, thus, results should not be distorted by routing in backup modes. In this section, we eval-

uate now the performance of BLR with the backup mode such that a packet can also be delivered if there are no potential forwarders. We first compare the performance to BLR without the backup mode. The delivery ratio of BLR with backup mode is increased significantly especially in sparse networks, where greedy routing fails frequently Fig. 5.20. However, for 250 nodes, the delivery ratio is still only around 70%. But this is not a characteristic of BLR, but can be observed in other protocols as well, cf. to Section 5.4.4. The node density is approximately the required minimum for a connected network. However, due to mobility it may still happen that the network is temporarily disconnected and no paths between source and destination exist. The increased delivery ratio causes the higher end-to-end delay for sparse networks. Many packets are delivered that have been routed in backup mode, causing additional delay per hop and at the same time increasing the hop count on average. For higher node densities, fewer packets are routed in backup mode and the delay approaches the delay of BLR without backup mode.

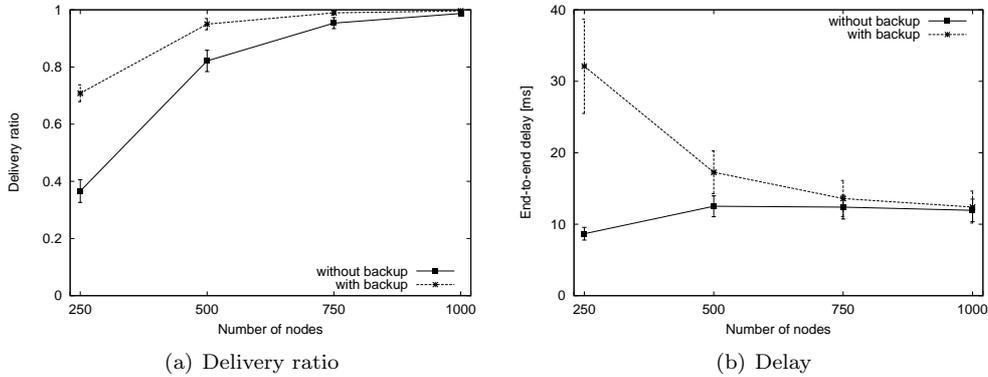


Figure 5.20: Impact of backup mode

Unicast Mode

For these simulations, the backup mode was again disabled to isolate the effects of the unicast mode. As the unicast mode makes BLR somewhat stateful, the unicast mode is especially critical to use in highly dynamic networks. In Fig. 5.21, we can see that the delivery ratio and the average hop count are almost identical for all node densities. As expected, the unicast mode shows a slightly higher hop count. The reason is that due to mobility a suboptimal neighbor may temporarily forward the packet, even if there are better located nodes. Only when the unicast mode periodically switches back to greedy mode to broadcast one packet, the “best” node is detected again. We set the interval at which a node broadcasts a packet again in greedy mode to 2 s.

The average end-to-end delay is increased by the unicast mode. The explanation is that the statefulness introduced by the unicast mode has a major impact for this frequently changing topology. Neighbor information may become again outdated such that unicast packets cannot be delivered to the next hop, which requires up to seven retransmissions on the MAC layer with IEEE 802.11 and adds on average 30 ms per unreachable next hop, cf. Section 4.2.2.

Considering the actually observed delay of approximately 40 ms , this however does not happen frequently, otherwise the delay would be much higher. A further reason for the increased delay may be the required acknowledgments on the MAC layer if unicast packets are transmitted. An ongoing transmission of an acknowledgment may block the transmission of a data packet because IEEE 802.11 applies a CSMA (Carrier Sense Multiple Access) mechanism. These two factors more than negate the saved time by the elimination of the additional delay introduced by the DFD function at each node. Although, unicast mode did not prove to be useful in these simulations, we have to keep in mind that unicast mode was not introduced for such idealistic scenarios. Unicast mode is expected to show its advantages in scenario where transmission ranges are irregular, the packet error rate is high, etc.

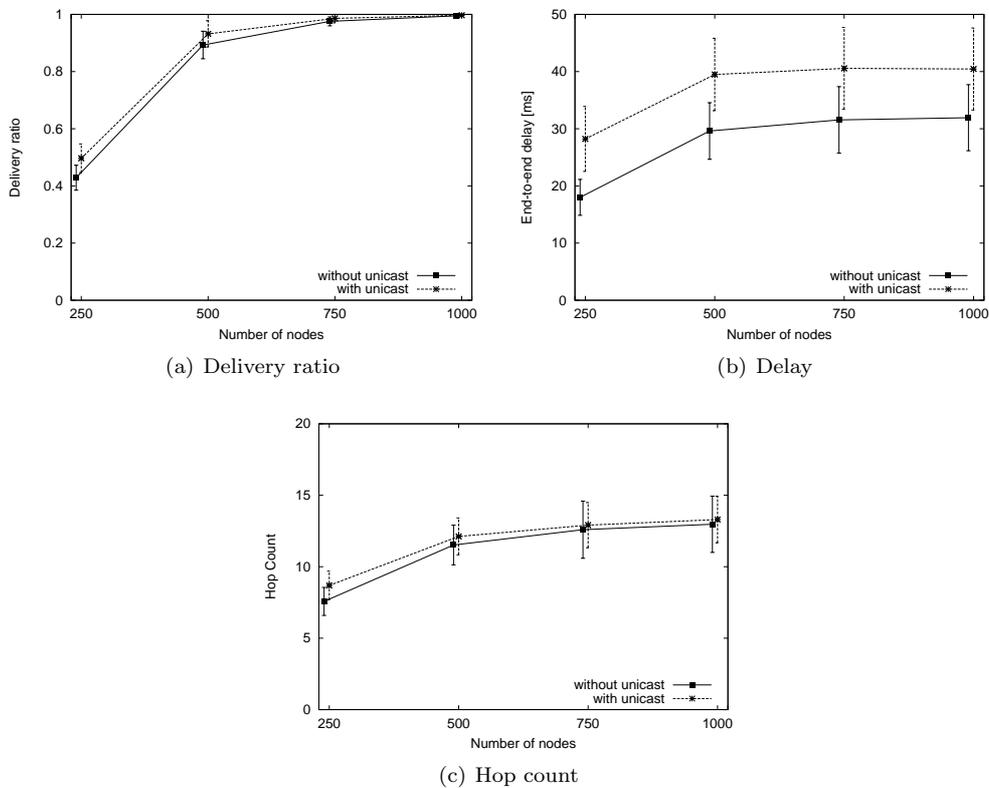


Figure 5.21: Impact of unicast mode

5.4.4 Comparison with GFG/GPSR

After having elaborated on the different components and parameters, we compare BLR with a standard position-based routing protocol, namely GFG/GPSR. We first compare their performance in the same scenario as already used before and also in case of irregular transmission ranges. Then we also evaluate their performance in some highly dynamic network for which BLR is particularly tailored.

Varying Node Densities

We simulated the protocols with an average node speed of 20 m/s . In Fig. 5.22, the delivery ratio and the end-to-end delay are shown for different network densities. Additionally to GFG/GPSR, we also used again the BL and BNU protocols introduced in Section 4.4.2 for comparison. They use perfect neighborhood information provided by the global data of the simulator, which allows to give a “theoretical” bound for the performance of position-based routing protocols. For low-density networks, the delivery ratio of BLR and GFG/GPSR are almost equal because packets are routed frequently in backup mode. The backup mode of BLR is similar to face routing, cf. Section 3.3.3, except for the fact that it is reactive. As already discussed before, the low delivery ratio of both protocols is due to temporarily partition of the network. This is confirmed by the delivery ratio of BL and BNU, which were also not able to deliver all packets to the destination for lower node densities. For denser networks, the delivery ratio increases for BLR to almost 100% whereas GFG/GPSR is not able to deliver more than 90%. The delivery ratio reaches a peak at 500 nodes and decreases again for higher node densities. BLR outperforms GFG/GPSR especially in terms of end-to-end delay. The delay remains unaffected by the node density and below 30 ms . GFG/GPSR on the other hand has a delay of at least 200 ms , which is even increasing for higher node densities. The performance of BLR was close to the end-to-end delay of the optimal BL and BNU protocols. For clarity reasons, the end-to-end delays from Fig. 5.22(b) are given on a different scale in Fig. 5.23. We can see that for higher node densities, the delay of BLR is very close to the delay of BL and BNU because packets can be routed in greedy mode all the time. For sparser networks, the delay of BLR is marginally higher, which is caused by the transmissions of the beacon request packets and the beacons in the backup mode. BL performs better than BNU because it does not transmit beacons, which may collide with data packets.

As already explained in Section 4.4, the reasons for the much longer delays of GFG/GPSR are mainly threefold. First, nodes broadcast beacons periodically and for a dense network this may congest the network. Secondly, for a higher node densities the chosen next hop is closer to the transmission range boundary and has a higher probability of not being available, even though still listed in the neighbor table. And third, due to the high mobility, we observed that packets loop between nodes as the stored position about neighbors does not correspond to the actual physical location of that node. Thus, not only face routing of the recovery mode does not guarantee delivery in mobile networks, but already also the greedy mode may fail.

Irregular Transmission Ranges

Routing protocols that use beacons to determine the one-hop neighbors are less susceptible to irregular transmission ranges. There is a link to a certain neighbor whenever a beacon is received from that neighbor independent whether the transmission ranges are irregular. Beacons are used to determine this connectivity in the local neighborhood. On the other hand, BLR assumes that nodes at a distance less than a fixed transmission radius r always are able to receive each others’ transmissions. This is however not true for realistic scenarios, where transmission ranges may be highly irregular due to non-isotropic path losses,

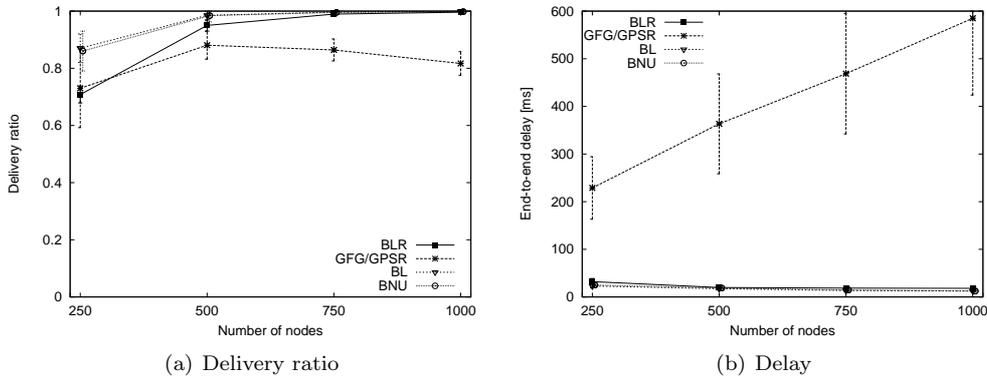


Figure 5.22: Comparison of BLR, GFG/GPSR, BL, and BNU

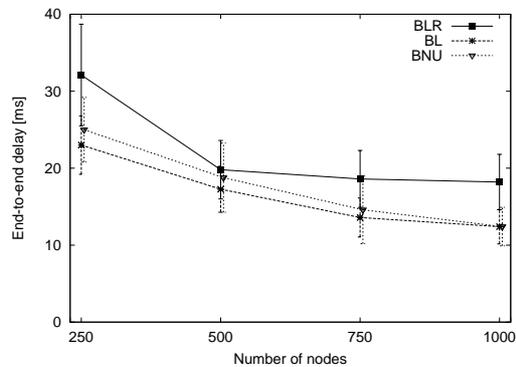


Figure 5.23: Delay of BLR, BL, and BNU

heterogeneous signal sending power and alike. Therefore, we are especially interested in the performance and behavior of BLR in such scenarios. We used the RIM to model radio propagation and set the two defining parameters of the model as follows; $DOI = 0.01$ and $VSP = 0.5$, cf. Section 2.4.3. Furthermore, a node recalculates its transmission range after it has moved 50 m to reflect the changes in its environment. The simulation results are given in Fig. 5.24. Compared to the values of BLR with an isotropic transmission range in the previous section, the delivery ratio decreased by approximately 10% in sparse networks. The transmission range is often much smaller than in the isotropic case. Thus, the size of the actual forwarding area is further reduced, i.e., also the number of potential forwarders. For denser networks, the delivery ratio and the delay remained basically unaffected for BLR even though the hop count is increased by approximately 50% (not shown). GPSR suffers also slightly in terms of delay and delivery ratio. The reason is that the changing transmission ranges calculated by the RIM propagation model whenever a node has move a certain distance further increases the inaccuracy of the neighbor tables.

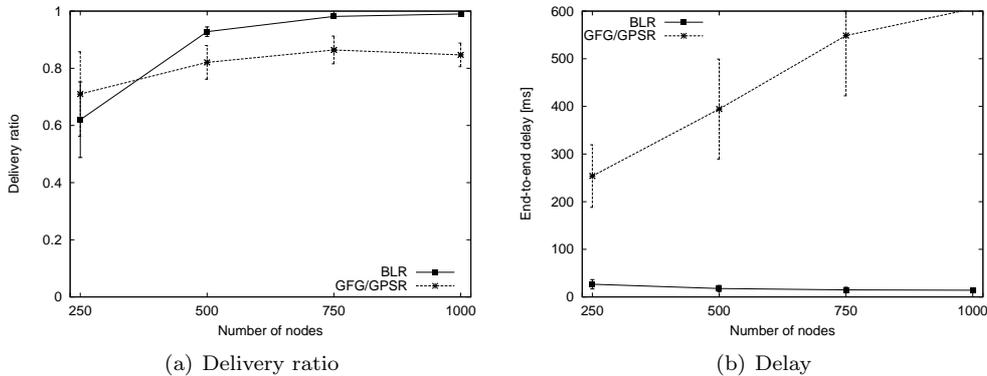


Figure 5.24: Comparison of BLR and GFG/GPSR with irregular transmission ranges

Highly Dynamic Networks

Highway We conducted simulations in an area of $20000\text{ m} \times 50\text{ m}$ where 1000 nodes move according to the random waypoint mobility model with an average speed of 30 m/s , i.e., the minimal and maximal speed are set to 27 m/s and 33 m/s , respectively. Although, nodes on a highway do not move according to the random waypoint model, the chosen values approximate a typical highway scenario. The large ratio between length and width of the simulation area causes nodes to move either in the same direction or against each other. The relative speed of two nodes is either very small if they move in the same direction or up to 60 m/s if they move in opposite directions. The results for these simulations are given in Table 5.1. Especially in such scenarios, BLR is able to show its advantages over a stateful position-based protocol. Even highest node speeds do not affect the delivery ratio and BLR is still able to deliver almost 97% of the packets. The few packet losses are rather due to temporary network partition than wrong routing decisions. BLR almost never makes any wrong routing decision and packets can be delivered within a fraction of a second over 22 hops. On the other hand, GFG/GPSR suffers drastically. Although it is still able to deliver 60% of the packet, the average end-to-end delay is two order of magnitudes higher than of BLR. The reasons are as illustrated in Section 4.2 that an unreachable neighbor introduces approximately 30 ms , if IEEE 802.11 is used on the MAC layer. The high mobility may result in the selection of several unreachable neighbors before the packet can be delivered to the next hop. The three times higher hop count of GFG/GPSR is caused by the fact that the positions as stored in the neighbor table do not correspond to the actual nodes' positions. This results in suboptimal selection of the neighbors, sometimes located even in the opposite direction of the destination.

Limitations of BLR As BLR performed still very good in the highway scenario, we also ran simulations with even higher nodes speed. Therefore, we simulated again 1000 nodes on $6000\text{ m} \times 1200\text{ m}$ where they move constantly with 100 m/s , i.e., minimal and maximal speed are both set to 100 m/s . In Table 5.2, the performance results are given for BLR and GFG/GPSR. The

	BLR	GFG/GPSR
Delivery ratio	0.9669	0.6009
Conf. Inter. Ratio	0.0130	0.0619
End-to-end delay [s]	0.02487	2.70966
Conf. Inter. Delay [s]	0.00252	0.64082
Hop count	21.86	60.32
Conf. Inter. Hop count	2.18	0.42

Table 5.1: Comparison of BLR and GFG/GPSR in highway scenario

	BLR	GFG/GPSR
Delivery ratio	0.9772	0.2169
Conf. Inter. Ratio	0.0039	0.0269
End-to-end delay [s]	0.00996	1.50114
Conf. Inter. Delay [s]	0.45	0.26868
Hop count	8.45	61.69
Conf. Inter. Hop count	0.36	0.18

Table 5.2: Comparison of BLR and GFG/GPSR in highest speed scenario

delivery ratio of BLR is even higher than in the highway scenario before, which supports the explanation that packet loss in the highway scenario is due to network partition rather than the inability of BLR to cope with the dynamic topology. The average end-to-end delay is also shorter than in the highway scenario simply because the path between the source and destination is shorter. GFG/GPSR is no longer able to deliver packets in an ordered and controlled fashion to the destination. The reason that some packets are delivered is rather because the destination has moved coincidentally into the range of the source, or any other node which currently holds the packet. We mention again that the chosen scenarios are especially tailored for BLR. In both, the highway and this scenario, packets can be forwarded in greedy mode most of the time due to the high node density. The performance of BLR may suffer in scenarios where packets are routed in backup mode, which is stateful. However, considering the results of Section 4.4.7 where the performance of GFG/GPSR with a reactive beaconing mechanism still showed superior or at least equivalent performance compared to the standard GFG/GPSR protocol, BLR should also not perform worse than GFG/GPSR when packets are often routed in backup mode. Basically because the backup mode of BLR is similar to a reactive GFG/GPSR. This conclusion is also supported by the results in Fig. 5.22, where packets are routed frequently in backup mode for the sparsest networks with 250 nodes. However, still if packets are routed frequently in backup mode, BLR and GFG/GPSR showed the same delivery ratio and BLR clearly outperformed GFG/GPSR in the end-to-end delay.

5.5 Implementation in a Linux Testbed

5.5.1 Implementation

Overview

The target platform of the implementation is GNU/Linux. We used Gentoo Linux [78], although any other GNU/Linux distribution based on Linux 2.6 will work for our implementation. We integrated BLR logically correct within the protocol stack as depicted in Fig. 5.25 such that it is transparent to the upper layers and applications. Consequently, any application such as HTTP, ssh, ping and also ICMP can be run unmodified. The BLR protocol was however implemented in the user space of Linux due to simplicity reasons. Therefore, outgoing packets (solid line) have to be intercepted and processed accordingly before being passed to the physical wireless network adapter. More specifically, we introduced a virtual interface `tun0` provided by the `tuntap` [240] device. A new route that redirects all traffic to the BLR network (private destination IP-Addresses 10.0.1.0/24) through `tun0` is added to the system routing table. Consequently, Internet traffic is not affected by the BLR application and routed as normal directly to the 802.11 interface. By listening on `tun0`, the BLR application can catch all traffic sent to the BLR network and inserts the BLR header and updates the IP header. Afterwards, packets are sent via the `pf_packet` facility, which allows the sending of Ethernet and IP packets directly to the 802.11 network adapter. Incoming packets (dashed line) are passed over `pf_packet` to

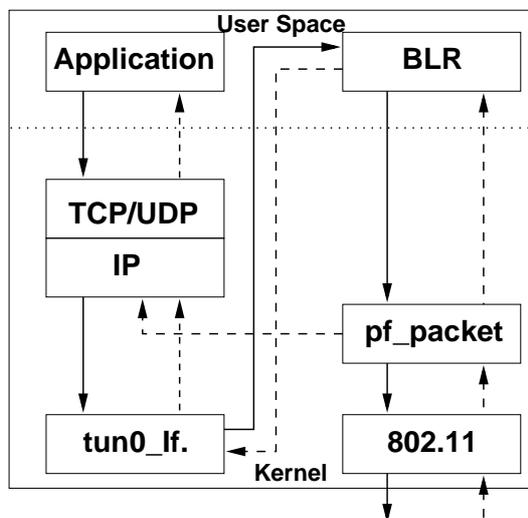


Figure 5.25: Implementation of BLR in the protocol stack

the BLR application and are either forwarded to the next hop or passed to localhost, depending on the destination address in the IP header. When packets are forwarded, the BLR application only updates the BLR header and additionally delays the packets by the newly calculated *Add_Delay* before the packets are passed again via `pf_packet` to the network adapter. On the other hand, when the packet is destined for this host, the BLR header is stripped off and

the changes done by the sending BLR application in the IP header are reversed. Afterwards, the packet is forwarded through the `tun0` to the application.

A problem occurs because `pf_packet` actually creates a copy of all incoming packets. One copy is passed to the BLR application, while the original packet is passed to the kernel and from there to the application. The original packet has to be blocked somehow. This is achieved by deploying the IPtables [122] packet filter right after the `pf_packet` facility. This filter blocks all incoming traffic that has the protocol number of BLR set in the IP header. For broadcast packets, this blocking would not be necessary since the kernel simply drops broadcast traffic with a protocol number for which there is no open socket. However, when the kernel receives unicast traffic with an unknown protocol number, it sends an ICMP destination unreachable message back to the sender, which has to be avoided.

Another option when implementing a routing protocol prototype would be to implement it on the application layer as it is commonly done. However that means that other applications have to tunnel data packets through the routing protocol, i.e., they have to perform a kind of application layer tunneling. Although, the implementation and management of a routing protocol on the application layer is simpler, it has several drawbacks. The most severe is that each application has to use a specific API, which requires the adaptation of each and every used application. Furthermore, Transport and IP layer headers are added to the routing protocol packets and, thus, increase the overhead.

BLR Application

The BLR application is split into three separated processes as depicted in Fig. 5.26. The main process receives/sends the packet from/to the localhost/network, transforms and updates headers, calculates the *Add_Delay*, and manages packet timeouts, unicast route information, as well as a list of duplicate packet IDs. The GPS process is connected to an external GPS device and provides location information. The sendqueue process receives outgoing packets together with the calculated *Add_Delay* from the main process and sends the packet after the indicated delay. If the main processes receives a packet from `pf_packet`, it calls the sendqueue process in order to determine if the packet is queued for transmission. If so, another node forwarded the packet first and the sendqueue process can remove the packet from the queue.

The size of the BLR header is 32 bytes and has the following fields.

- Packet type (1 byte): DATA, LOCATION-REQ, LOCATION-REPLY, etc.
- Original protocol (1 byte): Protocol number of TCP, UDP, ICMP, etc. The BLR header is inserted between the IP and transport layer header.
- Sequence number (2 bytes): To unambiguously identify a packet together with the source address.
- Backup distance (4 bytes): Distance to destination where greedy routing failed.
- Position information (8 bytes each): Position of previous node, the source and destination node. The previous node and destination node positions

are required to calculate *Add_Delay*. The source node's position is used to update location information at peers.

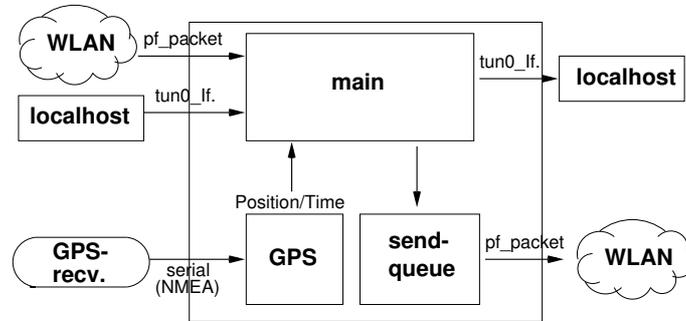


Figure 5.26: Running processes in the BLR application

In the following, we describe in more detail the processes and how packets are handled.

GPS Process The GPS process is connected to an external GPS device, which it polls for changes in location information. It parses the GPS data and passes position updates to the main process. The connection is established through an RS-232 interface and the GPS information is transferred with the NMEA-0183 protocol [185] from the GPS receiver to the laptops.

Sendqueue Process The sendqueue process is responsible for queuing the packets according to their respective dynamic forwarding delay. It receives packet/delay tuples from the main process and maintains an ordered list of all pending packets. When the associated timer expires, the packet will be sent to `pf_packet`. The queue is implemented separately from the main process, because packet delays are independent of each other and must not be accumulated. The sendqueue further handles the deletion of packets from the queue, whenever the main process detects that another node has already forwarded a pending packet.

Main Process This is by far the most complex process and is responsible for switching packets between components, management and coordination of the other components, execution of the BLR functions, etc. When the main process receives an IP packet through `tun0`, it inserts the BLR header and updates the IP header. Changes are necessary in four IP header fields.

- The source address needs to be changed from the IP address of the `tun0` interface to the IP address of the outgoing interface.
- The packet length fields needs to be increased by the size of the BLR header.
- The protocol field is changed to 254, which we used for the BLR protocol. The original protocol number is stored in the BLR header.

- Finally, the header checksum needs to be recalculated.

Furthermore, it calculates *Add_Delay* and forwards this along with the packet to the sendqueue process. It also maintains a `hosttable` to store information about known destinations, namely their most recent positions and the next hop to reach them if unicast mode is used. The `packetlist` caches packets that have been sent and have not yet been acknowledged, together with a timeout value for each packet. `packetlist` also handles retransmissions in case of timeouts. Whenever BLR has to switch to backup mode, it takes some time until the next hop is determined due to the sending of the beacon request packet and the time until the beacons from the neighbors are received. The `backupqueue` caches outgoing packets that have to wait to be forwarded until the backup mode setup is completed.

5.5.2 Challenges

In this section, we briefly review the main challenges we faced during the implementation in the Linux testbed as opposed to the previous implementation in the simulator.

Localization

The localization of the destination is a research aspect in itself and several solutions have already been proposed (see [167] for an overview). Therefore, a common assumption of most position-based routing protocols is that the position of the destination is somehow known. In the network simulator, it can be implicitly assumed that this position information is available. In reality, we have to implement a mechanism that provides the position. As it was not our aim to implement a fully functional location service and it would not be appropriate for a small testbed with few laptops, we chose to implement a simple request-reply mechanism based on flooding. In case of unidirectional traffic, position information is invalidated periodically, and the source broadcasts a new location request. In case of bidirectional traffic, or simply if TCP is used, destination locations are not invalidated, but the position can be simply extracted from packets returning from the destination, namely from the source field in the BLR header. Thus, the overhead can be reduced to the initial flooding of one location request packet.

Duplicated Packets

The objective of the unicast mode is to reduce the number of duplicated packets. However, still transmissions over intermediate hops are broadcast. In ideal conditions of a network simulator, radio propagation is modeled by simple models, which typically yield isotropic transmission ranges. In reality however, we observed many duplicated packets due to irregular transmission ranges. Therefore, we additionally implemented a filtering mechanism. Each node matches the uniquely identifying source address and sequence number of a packet against a table containing the recently received and also overheard packets. If a duplicate is detected, the node broadcasts a control packet suppressing the further forwarding of that packet by its neighbors.

IP Fragmentation

The BLR header is part of the IP payload in the current implementation. Thus, if fragmentation occurs, only the first IP fragment will contain the BLR header. The header is however required to route packets by BLR. Subsequent fragments will not contain the BLR header and will simply be dropped, because nodes do not know how to process them. Therefore, IP fragmentation has to be avoided. To achieve this, the MTU of the virtual tunnel interface is decreased by the size of the BLR header, which is inserted before Transport layer header, in order to avoid fragmentation at the source node. Additionally the DF (Don't Fragment) bit is set in the IP header such that intermediate nodes do not fragment the packet. PMTU (Path MTU) discovery is used to handle links where the standard MTU is too large.

MAC Layer Control

If a unicast packet is not acknowledged, the MAC layer retransmits a packet up to seven times before giving up. In the network simulator implementation, the MAC layer can signal a failed transmission to the upper layer, which in turn selects another next hop and passes the packet again to the MAC layer. This mechanism is also applied by BLR [101] and GPSR [134]. Without this optimization, many unicast packets would be dropped due to unreachable neighbors, and recovery is left to TCP or the application. This severely decreases network performance as retransmissions are end-to-end and not link retransmissions. In a Linux implementation of BLR with WLAN cards however, the MAC protocol is largely implemented in the firmware of the 802.11b card, which makes accessing the mentioned functions virtually impossible.

Interrupt Granularity

The *Max_Delay* can be chosen in the order of some milliseconds based on the results from the network simulator. Basically, *Max_Delay* indicates the range over which potential forwarder schedule their retransmissions. The Linux kernel has a limitation that severely affects the possible value of *Max_Delay*, namely the granularity of the timer interrupts. This granularity is defined by a compile-time kernel constant called HZ. On Linux kernels 2.6 or newer, this constant is set to 1000 resulting in timer events every 1 millisecond. (In kernel 2.4 and older, the HZ was set to 100). This means that the `select()` system call, the heart of network programming, returns at 1 millisecond intervals only. Consequently the granularity of *Add_Delay* is also only 1 millisecond. Therefore, a rather long *Max_Delay* has to be chosen to reduce the risk that all nodes transmit simultaneously and limit the usefulness of the DFD concept. (In [101], it was proposed to set $Max_Delay = 2ms$ based on simulation results, which is definitely too short for the Linux implementation.) However, the longer *Max_Delay* also increases the end-to-end delay. While possible in theory, a further increase of the HZ value is not yet completely supported by the Linux kernel. Even if possible, the raising of the HZ also increases the overall timer overhead as more timer interrupts are generated. This may not be an issue for our testbed where no other applications are running, but definitely it will be for small mobile devices with limited computation resources.

5.5.3 Experiments

Equipment and Configuration

The testbed consists of 5 laptop computers running Linux 2.6. Each laptop is equipped with an IEEE 802.11b WLAN cards. The cards are configured to run in ad-hoc mode without RTS/CTS, i.e., the DCF of 802.11b is used, and the data rate is set to 2 Mbps. The hardware equipment is heterogeneous, i.e., the laptops are from different manufacturers. The same applies to the WLAN cards, some laptops have built-in cards, while other use Orinoco WLAN cards plugged in the PCMCIA-slot. Each laptop also has a GPS receiver connected via the serial RS-232 line. The GPS devices are not only used for providing positioning information to the nodes, but we also use GPS for timing information. This information is provided once per second. The timing information is actually not required for the BLR protocol, but only for performance measurements. The accuracy of the information is as low as 5 *m* and 200 *ns* for the positioning and timing information, respectively.

In this section, we present the results from experiments that were conducted in the laboratory in order to validate the implementation and for reference purposes, which allow a comparison with future outdoor experimental results. As the GPS receivers do not work in indoor environments, the position of the laptops had to be hard coded to yield a virtual topology. Therefore, the positions and the distances between nodes in this virtual topology do not match the actual physical location of the laptops. Furthermore, all laptops are placed on a table within a few meters of each other and thus could physically receive all transmissions of all nodes. To ensure that a laptop only processes the packets from laptops within the transmission range in the virtual topology, a filter based on MAC-Addresses has been implemented. This filter operates directly on the `pf_packet` socket and simply drops packets from out-of-range nodes in order to match the physical and the virtual topology such that the BLR application never sees packets from virtually out of range nodes. This approach saves processing work on the side of the BLR application since the kernel does all the necessary filtering. The implementation of the MAC filter is done by means of the Berkeley Packet Filter (BPF) language [168]. The GNU/Linux implementation is called Linux Socket Filter (LSF) and is compatible with the BPF language.

Traffic is sent by the ping utility, which yields the round trip time RTT, also simply referred to as delay in the following. For each measurement, 2000 ICMP echo requests were sent, which together with the echo replies result in 4000 total data packets. The transmission rate had to be limited to 10 echo request per second, because all nodes are physically within each other transmission range. Thus, a transmission of a node blocks all other nodes on the MAC layer, and not only the neighbors in the virtual topology. The default packet size was set to 56 bytes, including the ICMP, IP, BLR, and MAC header this yields 180 transmitted bytes. The experiments were conducted with a rather long *Max_Delay* of 5 *ms* and 25 *ms* to reduce the risk that nodes transmit simultaneously due to the low interrupt granularity as explained before in Section 5.5.2. The transmission range for calculating the *Add_Delay* was set to a 250 *m* and the unicast mode was switched off by default. We used four topologies for the laboratory experiments as depicted in Fig. 5.27, called chain, pairs, contention, and backup topology.

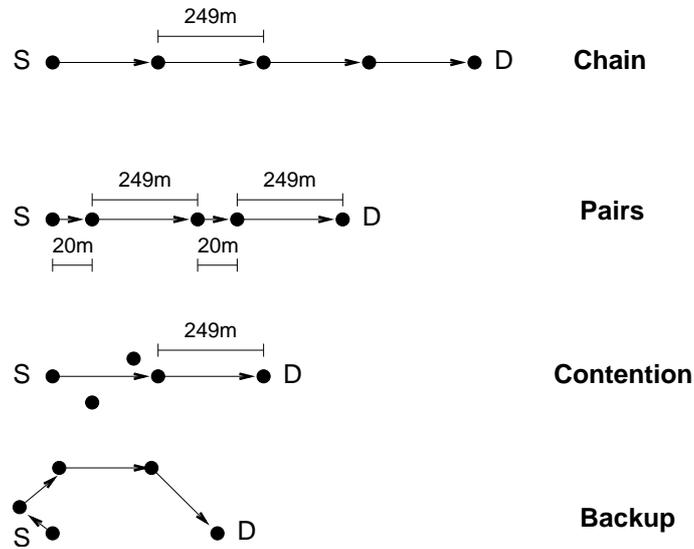


Figure 5.27: Topologies for the experiments

Chain Topology

The chain topology is the most simply topology as no contention occurs and only one node always will forward the packet. The forwarding node is located at the boundary of the transmission range and almost immediately forwards the packet without introducing *Add_Delay*. Thus, the RTT is basically independent of the *Max_Delay* as shown in the histogram in Fig. 5.28, which shows the distribution of the measured RTTs. The average is in both cases 17.4ms and the delivery ratio was always 100%. Considering the fact that a packet is transmitted over eight hops (four hops from the source to the destination for the echo request and four hops back to the source for the echo reply), the measured RTT is approximately only 2ms per hop.

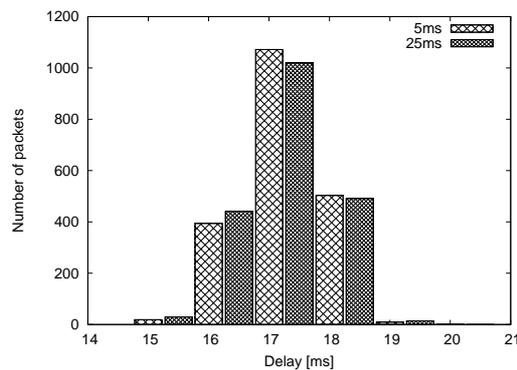


Figure 5.28: Chain topology with $Max_Delay = 5\text{ms}$ and $Max_Delay = 25\text{ms}$

Pairs Topology

The results from the pairs topology are given in Fig. 5.29. In this topology, a packet is again transmitted over eight hops. However, the delays now vary strongly for the two different Max_Delay as expected. The two transmissions from the nodes with only 20 m progress are delayed significantly as they calculate a long Add_Delay , which is close to Max_Delay according to (5.1). This yields in the pairs topology delays of approximately 30 ms and 80 ms for $Max_Delay = 5\ ms$ and $Max_Delay = 25\ ms$, respectively.

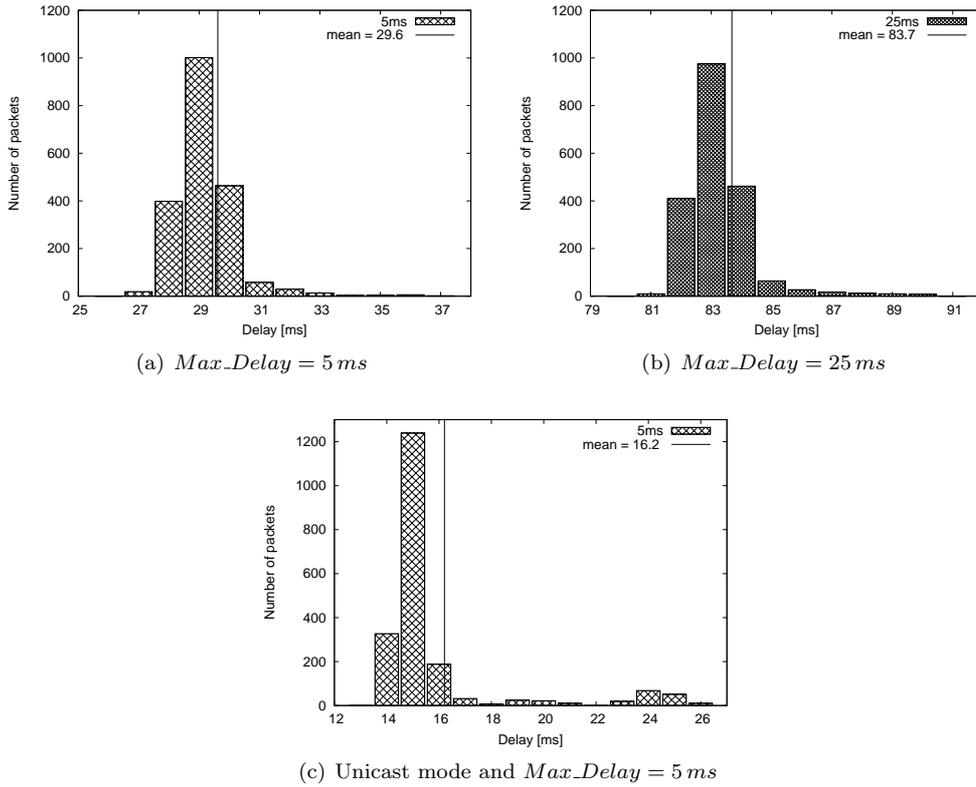


Figure 5.29: Pairs topology

In the pairs topology, we also evaluated the impact of the unicast mode. Although packet duplication is not an issue as only one potential forwarder exists, the delay is affected. Recall that in unicast mode, the packets are forwarded without introducing Add_Delay if the next hop is known. In Fig. 5.29(c), the histogram of the measured delays with $Max_Delay = 5\ ms$ is shown. The delay is significantly shorter than when packets are always broadcasted in greedy mode and is reduced from 29 ms to 16 ms . We can also see that there are some packets with longer delays around 25 ms . The reason is that the unicast mode switches to greedy mode every 5 s in order to detect possibly better located neighbors. Packets transmitted in greedy mode are again dynamically delayed at each node and not immediately forwarded as in unicast mode.

Contention Topology

In the contention topology, three nodes receive the transmitted packet from the source node and schedule the packet for forwarding as they are all within the forwarding area. They calculate different *Add_Delay* however and the first transmitting node suppresses the others accordingly. In Fig. 5.30, the distribution of the delays is shown. We can observe that the delay is quite short around 8 ms compared to the previous investigated topologies because a packet is only transmitted over four hops (two hops to the destination and two hops back to the source).

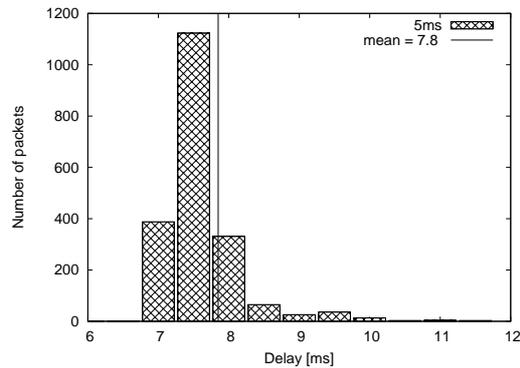


Figure 5.30: Contention topology with *Max_Delay* = 5 ms

Backup Topology

In a last experiment, we validated the backup mode of BLR. There is no node located in the forwarding area and the packets are routed in backup mode for three hops until arriving at the node closer to the destination than the source. When packets are generated at 10 packets per second, we measured two different RTTs of approximately 40 ms and 60 ms as depicted in Fig. 5.31(a). The reason is that while the backup mode acquires neighbor information if greedy forwarding failed, i.e., during the beacon request reply dialog, other arriving packets are queued in the `backupqueue`. When the backup mode setup is completed, all queued packets are sent immediately to the next hop, thus, some packets in the queue encounter shorter delays. The first following packet again has then to wait until the request reply dialog is completed. On the other hand if packets are generated at 2 packets per second, all packets have a delay of approximately 60 ms . The reason is that there are no packets arriving at a node before the backup setup was completed and the previous packet was already forwarded to the next hop. However, the delay is still quite short considering the fact that nodes have to transmit a request for beacon packet and wait until neighbors have replied.

General Observations

Until now, we only considered the measured delays in the experiments. We can conclude that the delays are short, are as expected, and vary only slightly

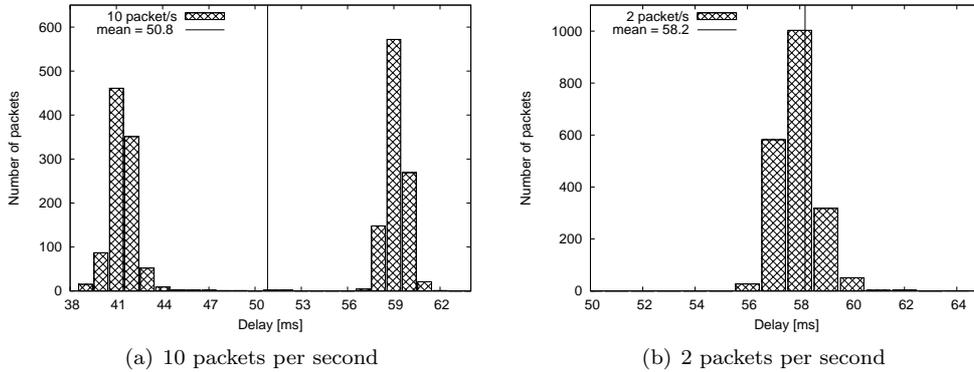


Figure 5.31: Backup topology with $Max_Delay = 5\ ms$

around the mean. The other quantitative performance metrics were simply not suited to be depicted graphically and are briefly discussed in the following. In all four topologies, the delivery ratio was always 100%. This is not surprising considering the fact that the nodes are physically close to each other, even if they are distant in the virtual topology. Furthermore, we observed that in the chain, pairs, and contention topology, there was approximately one packet per thousand transmitted unexpectedly in backup mode. The reason was that very rarely some packets showed a higher delay and collided with subsequent transmitted packets, which caused the required retransmissions in backup mode. This effect is especially obvious in the laboratory where all nodes are within transmission range. Furthermore, we observed very infrequently duplicated packets, again in the order of some few per thousand. However, they could be successfully suppressed at the next node by transmitting a control packet as described in Section 5.5.2. Thus, no duplicated packet arrived at the destination node. Especially for the contention topology, the few duplicated packet indicate that the first transmitting node is able to suppresses successfully the other potential forwarders.

Thus, we can summarize that the forwarding of the packets in the greedy, unicast, and backup mode of BLR was as expected. The results also indicate that BLR is able to deliver packet over multiple hops in a short time. Packets are forwarded reliably and the delivery ratio was always 100%. Furthermore, the forwarding nodes successfully suppressed the other potential forwarders and acknowledges also the previous node reliably, because basically no duplicated packets were observed.

5.6 Conclusions

In this chapter, we proposed and evaluated the Beacon-Less Routing protocol (BLR). BLR is based on a new routing paradigm where forwarding decisions are taken at the receivers of a packet instead at the sender, which allows to route packet without any neighbor information. The sending node broadcasts a data packet that is delayed dynamically at the receiving nodes before rebroadcasting. The delay is calculated depending on the nodes' positions. The first

node that retransmits the packet suppress the others. If routing in greedy mode fails, a more time- and energy-consuming backup mode is applied. We first evaluated BLR analytically and derived some general properties about the expected behavior. Then, we compared the performance of BLR to a standard position-based routing protocol GFG/GPSR by simulations. The delivery ratios were similar for low-density networks where BLR forwards packets often in backup mode. The delay remains significantly below that of GFG/GPSR because of the statelessness of BLR, which almost completely eliminates wrong routing decisions. In dense networks, the performance of BLR was even close to the “theoretical” optimum in obtained by the BL protocol, which uses perfect neighborhood information. In backup mode of BLR, the forwarding decisions are again taken no longer at the receivers but at the sender of a packet like conventional position-based routing protocols. However, as the backup mode is reactive, where beacons are only sent when requested, is also superior to a simple periodical beaconing as it was shown in Section 4.4.7. The introduced delay of the reactive approach more than compensates for the fewer wrong routing decisions. As long as BLR is able to operate in greedy mode, the performance of BLR significantly better than GFG/GPSR. The performance was even close to the “theoretical” optimum in terms of delay. Furthermore, its statelessness enables BLR to be almost completely unaffected by mobility. Even for most highly dynamic networks with nodes moving at 100 *m/s*, neither the delivery ratio nor the end-to-end delay suffered and still 97% and more of the packet could be delivered to the destination. Although, the MAC protocol may have a major impact on the performance and the behavior of the routing protocol and the results may look different with other MAC protocols, the general observations and conclusions should hold as well.

However, the advantages of BLR in dense and highly mobile networks over position-based routing protocols with neighbor knowledge come at a certain cost. The main problem of BLR is packet duplication as forwarding decisions are taken in a distributed manner at the receivers such that we have to ensure that potential forwarders are able to overhear each other. Even with a simple network model with isotropic transmission ranges, greedy routing is limited to a limited area. More precisely, the area which allows greedy routing is about half the size of other position-based routing algorithms. Consequently, greedy mode will fail much more often and BLR requires a rather high node density in the direction of the destination. For realistic scenarios, there are several reasons that prevent potential forwarders nevertheless to detect transmissions from other potential forwarders such as obstacles, irregular transmission ranges, bit error rates, etc., so that we face a large number of packet duplications. Assuming that the network does not get congested due to these packet duplications, neither the end-to-end delay nor the delivery ratio suffers. They can even be improved due to these redundant packets. Thus, the drawbacks of packet duplication are restricted in such scenarios to the additionally consumed network resources such as bandwidth and battery power however. BLR also provides some mechanisms to limit the duplicated packets so that they are often only transmitted over one hop and suppressed again at the following receivers.

A further important finding of this chapter is that GFG/GPSR and, thus, all position-based routing protocols that require neighbor knowledge and transmit beacons, are not able to operate in highly dynamic scenarios where neighbor information is outdated quickly. Thus, this is an indication that stateful protocol

may also not be appropriate for sensor networks where network topology changes frequently due to sleep cycles of nodes.

We can summarize that BLR has its advantages in the timely delivery of packets, which is not affected by topology changes and is much shorter than of other positions-based protocols if node density along the routed path is high. However, even if BLR does not require nodes to transmit beacons, it is not sure how total energy consumption is affected due to the duplicated packets and the required operation of nodes in promiscuous mode. Considering the facts however that other position-based protocols transmit beacons that need to be processed at all receiving nodes and that many retransmissions occur if a next hop is unavailable, the total power consumption of BLR should normally remain below that of other position-based routing protocols. A detailed analysis would be required in order to have further evidence, which is unfortunately out of scope of this thesis. A possible application area could be vehicular ad-hoc networks where messages are routed along a highway where power consumption is not an issue.

As the concept of dynamic forwarding delay DFD showed promising results for a unicast routing protocols, it was just a logical step to use the same concept also for other operations in wireless multihop networks. In Chapter 6, we consider broadcasting with the objective to deliver a packet reliably and efficiently to all other nodes in the network. Unlike for BLR, the proposed broadcasting protocol DDB has the advantage that no forwarding zones and unicast modes have to be implemented to alleviate the problem of packet duplication as packets are anyway broadcasted and duplication is required. In DDB, a node simply determines when to rebroadcast a packet with a dynamic forwarding delay DFD function according to its probability for reaching new neighbors or any other metric. A threshold is used to suppress nodes that have no, or little probability for reaching new nodes.

Chapter 6

Dynamic Delayed Broadcasting (DDB)

6.1 Introduction

In this chapter, we present a simple and stateless broadcasting protocol called Dynamic Delayed Broadcasting (DDB) that allows locally optimal broadcasting without any prior knowledge of the neighborhood. This is achieved by the use of the dynamic forwarding delay (DFD) concept, which was also already used in the stateless BLR position-based routing protocols in Chapter 5. DFD delays the transmissions dynamically and in a completely distributed way. Nodes compute a short delay by applying a DFD function in order to determine when to rebroadcast a message based solely on the information available at the node itself and the information given in the broadcast packet. The same information is also used to decide whether a packet is rebroadcasted at all. The concept of DFD supports the optimization for different metrics such as the number of retransmitting nodes, end-to-end delay, network lifetime, etc., and can take different parameters as input such as distance to other nodes, incoming signal strength, etc.

As DDB does not require any transmissions of control messages, it does not cause any control traffic overhead and conserves critical network resources such as battery power and bandwidth. Furthermore, it is highly scalable in dynamic networks. Due all these characteristics, DDB is especially suited for networks with highly dynamic topologies or strict power consumption constraints such as vehicular ad-hoc and sensor networks.

We first describe the details of DDB in section 6.2. Some characteristics and properties of DDB are then studied analytically in Section 6.3. In Section 6.4, we evaluate DDB by simulations over a wide range of network scenarios. Finally, we conclude the chapter about DDB in Section 6.5. Further information can also be found in the publications about DDB [106], [107].

6.2 Description of DDB

6.2.1 Introduction

We assume that nodes are aware of their absolute geographical location by any means. Many applications in sensor and vehicular ad-hoc networks already require per se location information. This location information available for free can be used to optimize lower network operations such as routing and broadcasting. However, it is not required that a node has any information about its neighborhood. Thus, no hello messages have to be transmitted periodically which saves scarce resources like bandwidth and battery power. The last broadcasting node only stores its current position in the header of the packet. This is the only external information required by other nodes in order to calculate when and whether to rebroadcast. Location information may not always be available however. DDB can also operate without location information and use incoming signal strength to approximate the distance to other transmitting nodes. As usual, packets are uniquely identified by their source node ID and a monotonically increasing sequence number. Like in BLR, there is only parameters taken by the algorithm called *Max_Delay*, which indicates the maximum delay a packet can perceive per hop and is used to calculate the delay at the nodes.

As mentioned before, the concept of DFD in DDB supports the optimization for different metrics. We explicitly propose and evaluate in more detail DDB with four different DFD functions. The first two DFD functions aim to reduce the number of overall transmissions to deliver the packet to all nodes in the network. The first uses the distance to the previous transmitting node, which allows estimating the additionally covered area. The second uses the distance itself. The third DFD function has the same objective of minimizing the number of rebroadcasting nodes, but assumes that no location information is available and instead uses the power level of the incoming signal to approximate the distances between nodes. The fourth function addresses the problem of power consumption and aims at extending the network lifetime by favoring nodes with more residual battery energy. We refer to DDB with one of these four specific DFD functions as DDB_{AC} , DDB_{DB} , DDB_{SS} , and DDB_{RB} , respectively. (AC, DB, SS, and RB stand for “Additional Coverage”, “Distance-Based”, “Signal Strength”, and “Residual Battery”, respectively.) DDB without subscript refers to the general DDB protocol without any explicit DFD function.

6.2.2 Minimizing the Number of Transmissions

The objective of the first scheme DDB_{AC} is to minimize the number of transmissions and at the same time to deliver the packet reliably to all nodes. Nodes that receive the broadcast packet use the concept of dynamic forwarding delay (DFD) to schedule the rebroadcasting and do not forward the packet immediately. From the position of the last visited node stored in the packet header and the node’s current position, a node can calculate the estimated additional area that it would cover with its transmission. Depending on the size of this additionally covered area, the node introduces a delay before relaying the packet, where the delay is longer for a smaller additional area. In this way, nodes that have a higher probability of reaching additional nodes broadcast the packet first. Note

that this is achieved without nodes having knowledge of their neighborhood. Unlike stateful broadcast algorithms, the “best” nodes for rebroadcasting are chosen in a completely distributed way at the receiving nodes and not at the senders. If a node receives another copy of the same packet and did not yet transmit its scheduled packet, i.e., the calculated DFD timer did not yet expire, the node recalculates the additional coverage of its transmission considering the previously received transmissions. From the remaining additional area, the DFD is recalculated which is reduced by the time the node already delayed the packet, i.e., the time between the reception of the first and the second packet. For the reception of any additional copy of the packet, the DFD is recalculated likewise. A node does not rebroadcast a packet if the estimated additional area it can cover with its transmission is less than a *rebroadcasting threshold*, denoted as RT , which also may be zero. Obviously, DDB_{AC} can “only” take locally optimal rebroadcasting decisions as nodes receive only transmissions from their immediate one-hop neighbors and thus have no knowledge of other more distant nodes which possibly already partially cover the same area.

To illustrate the complete procedure of the algorithm, consider the example given in Fig. 6.1, where we assume a rebroadcasting threshold $RT = 0$. Furthermore, we do not account for propagation and processing delay. They are typically in the order of μs and negligible compared to the transmission delay and the delay introduced by DFD which are several orders of magnitude higher.

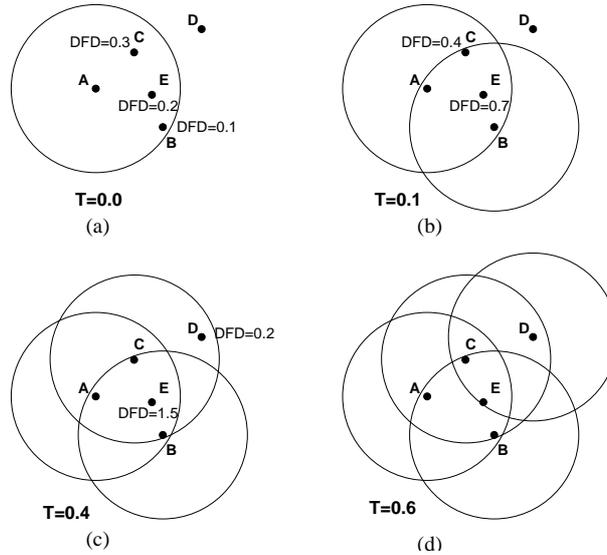


Figure 6.1: Example of the broadcast algorithm

Node A broadcasts a packet at time $T = 0.0 ms$. The packet is received at neighbors B, E, C Fig. 6.1(a). These nodes determine the size of the additional area they cover and introduce the additional delay accordingly. Let us assume, nodes B, E, C calculate a DFD of $0.1 ms, 0.2 ms$ and $0.3 ms$, respectively. Note that node C has no knowledge that there are two other neighbors which are located at a better position, i.e., calculate a smaller DFD. Similarly, nodes B or E are not aware of their neighbors as well. As node B introduces the

shortest additional delay and consequently rebroadcasts the packet first after 0.1 ms which is also overheard at nodes E and C Fig. 6.1(b). Upon the detection of this transmission, they determine a new DFD depending on the remaining additional coverage. Unlike before the transmission of node B , C calculates now a smaller delay than E . Assume that node E and C calculate a new DFD of 0.7 ms and 0.4 ms minus the 0.1 ms they have already delayed the transmission. Consequently, node C will rebroadcast the packet 0.3 ms later Fig. 6.1(c) already at time $T = 0.4\text{ ms}$. Nodes D and E receive the packet and calculate the DFD as 0.2 ms and 1.5 ms , respectively. Node D received the packet for the first time only now, but it still schedules the rebroadcasting much earlier than node E . Node D sends after 2 ms while node E is delayed 1.5 ms minus 0.4 ms from the initial reception of the first copy of this packet. After node D transmits the packet in Fig 6.1(d), node E drops the packet because it cannot cover any additional area. The dynamic calculation and recalculation of the DFD always assures that nodes that have a higher probability of reaching new neighbors transmit first. As these nodes are located close to the transmission boundary, the calculated delay is short and the packet should be disseminated quickly within the network. In Section 6.3, we will give some analytical results about the expected delay and additionally covered area by DDB_{AC} .

DFD Functions

The explicit DFD function is crucial to the performance of DDB_{AC} and should fulfill certain requirements in order to operate efficiently. The function should yield larger delays for smaller additional coverage and vice versa, if the objective is to minimize the number of transmissions. We assume the unit disk graph as the network model and thus a transmission range scaled to 1.

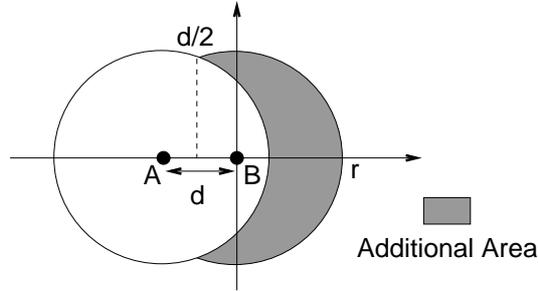


Figure 6.2: Additional covered area

Considering Fig. 6.2, we can determine the size of the additionally covered area AC of node B 's transmission if it is at a distance $d \in [0, 1]$ from the previous transmitting node A as follows.

$$AC(d) = 2 \cdot \left(\int_{-\frac{d}{2}}^1 \sqrt{1-x^2} dx - \int_{-\frac{d}{2}}^{-d+1} \sqrt{1-(x+d)^2} dx \right)$$

which immediately yields

$$AC(d) = \frac{d}{2}\sqrt{4-d^2} + 2 \arcsin\left(\frac{d}{2}\right) \quad (6.1)$$

The size of the additional covered area is maximal if node B is located just at the boundary of the transmission range of node A , i.e., if $d = 1$.

$$AC_{MAX} = \left(\frac{\sqrt{3}}{2} + \frac{\pi}{3}\right) \simeq 1.91$$

Consequently, one transmission can cover a maximum of $\frac{AC_{MAX}}{\pi} \simeq 61\%$ additional area which was not yet covered by the transmission of other nodes, i.e., at least already 39% were covered by other nodes' transmissions.

Taking into account this maximal AC_{MAX} , we propose a DFD function which is exponential in the size of the the additionally covered area, as it was shown in [186], that exponentially distributed random timers can reduce the number of responses. Let AC denote the size of the additionally covered area, i.e., $AC \in [0, 1.91]$,

$$Add_Delay = Max_Delay \cdot \sqrt{\frac{e - e^{\left(\frac{AC}{1.91}\right)}}{e - 1}} \quad (6.2)$$

where Max_Delay is the maximum delay a packet can experience at each node. We also evaluated the linear and exponential DFD functions of BLR, cf. Section 5.2.1. However, these functions showed an inferior performance. The reason is that the potential forwarders of BLR are restricted to a rather small forwarding area, while with DDB all nodes which receive the packet compete to forward, thus, increasing contention and collisions. The proposed function for DDB (6.2) is similar to the exponential of BLR, but the square root distributes farther nodes at the transmission range boundary over even a larger time interval and, thus, reducing contention.

The function is depicted graphically in Fig. 6.3 for a $Max_Delay = 1$. We see that when nodes have a higher AC , the calculated DFD timers are distributed over a larger interval, e.g., d_1 for two nodes with 0.2 difference in the additional coverage. Thus, the probability that a collision occurs at the first transmitting nodes, i.e., the ones close to the transmission boundary, is lower. The timers of nodes with only a small AC are closer to each other and e.g., only differ by d_2 for the same difference of 0.2 in the additional coverage. However, as they transmit much later, they have received multiple transmission of other nodes and may not require to retransmit at all because $AC < RT$.

Calculation of Additional Coverage The derivation of the additional area AC a node can cover with its transmission is easy to calculate for just one received packet. However, it gets more and more complicate when the node has to calculate AC after having overheard several copies which requires to determine the intersection of several circles. We approximate AC in the following way. The transmission range is covered with a grid of square cells as depicted in Fig. 6.4(a).

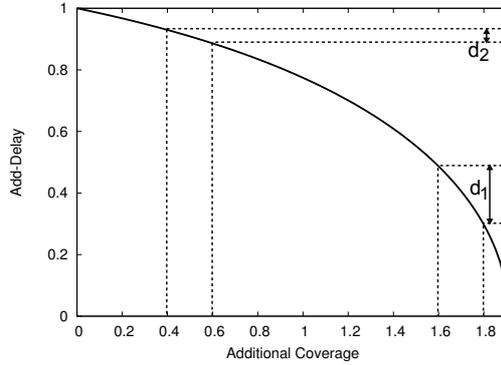


Figure 6.3: Delay introduced by the DFD function

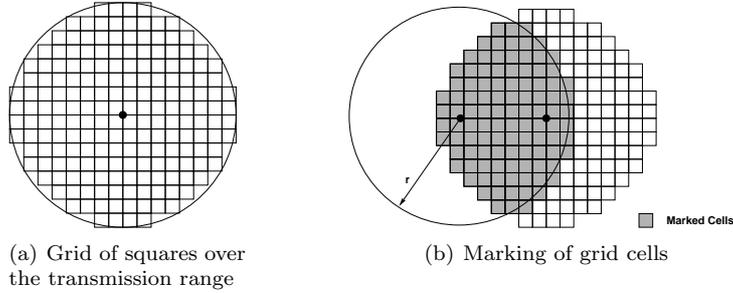


Figure 6.4: Grid cells

The size of the square cells determines the accuracy of the approximation. Each node considers itself located at the origin of the coordinate system. When a node A receives a packet from a node B , it calculates that node's position relative to its position (x_r, y_r) and uses the circle disk inequality given in (6.3) to determine which of its grid cells are covered with B 's transmission and marks the corresponding cells as shown in Fig.6.4(b).

$$(x - x_r)^2 + (y - y_r)^2 \leq r^2 \quad (6.3)$$

A node proceeds analogously for each subsequent received copy of the same packet and marks the unmarked cells which are covered by that transmission. Thus, each node can now easily determine AC by dividing the number of marked cells by the total number of cells. With a typical transmission radius of 250 m and a grid square sizes of $5 \times 5\text{ m}$, the divergence is in the order of 1%.

Minimizing the Number of Transmissions based on Distances

Instead of using the additional covered area, which can be computationally expensive, the distance d between the transmitting nodes is used as an approximation of the likelihood to cover additional area. Each node keeps track of the minimal distance $d_{min} \in [0, 1]$ to all nodes from which it received a copy of the broadcast packet. After the reception of the first copy, d_{min} is simply the

distance d to the transmitting node. Like DDB_{AC} where AC was used to derive the Add_Delay , DDB_{DB} uses d_{min} to calculate the additional delay in (6.4).

$$Add_Delay = Max_Delay \cdot \sqrt{\frac{e - e^{\frac{d_{min}}{r}}}{e - 1}} \quad (6.4)$$

Unlike the area-based variant of DDB_{AC} , a node only recalculates the DFD if the packet is received from a node which is closer than the currently stored d_{min} . The *rebroadcasting threshold* RT is accordingly based on distances. A node with a d_{min} smaller than the rebroadcasting threshold RT_{DB} does not rebroadcast the packet. The distance does not have the additive property of the additional coverage and cannot be summed up so that the number of rebroadcasting nodes is higher. For example, a node may have received many packets, but as none was transmitted by a node closer than RT , the node rebroadcasts the packet anyway.

Minimizing the Number of Transmissions based on Signal Strength

Dynamic Forwarding Delay (DFD) may also be applied to optimize broadcasting in sensor and ad-hoc networks where nodes are not location aware. Instead of using the distance to the previous transmitting node as the input to the DFD function, nodes use the incoming signal strength. Packets received at higher power levels are delayed more as one may assume that the sender is located close by, i.e., for a higher signal strength, the DFD should calculate a larger additional delay as we may assume that we are close to the transmitting node, i.e., only cover little additional area. Signals can only be decoded if they are received above the receiver sensitivity. If the signal strength just equals the receiver sensitivity, the transmitting node is at the boundary of the transmission range. Thus, we may assume that it has a large additional coverage area and should retransmit quickly. For an attenuation factor a , a receiver sensitivity S_r , and a received power of P_r measured in dBm , we propose the following DFD function.

$$Add_Delay = Max_Delay \cdot \sqrt{\frac{e - e^{\frac{A}{\sqrt{10}} \left(\frac{S_r - P_r}{10} \right)}}{e - 1}} \quad (6.5)$$

Basically, (6.5) corresponds to (6.4) of the distance based DDB_{DB} , respectively. A typical IEEE 802.11b WLAN card have a transmission power P_t of about 15 dBm and a receiver sensitivity S_r of -81 dBm . These values are just exemplary and are not fixed. The transmission power is normally subject to regulatory limitations and may vary in different countries. The receiver sensitivity depends on the modulation scheme, i.e., on the data rate used, where lower data rates normally use more robust modulation schemes which can still be decoded at lower power levels, i.e., at higher distances.

Analogously, the *rebroadcasting threshold* is set to some signal strength value and a node only transmits a packet if it has not received any packet at a power level above this threshold. As the attenuation factor is normally not known, it has to be estimated. The more accurate the estimation of the attenuation factor is, the better the performance will be. An advantage of DDB_{SS} based on signal strength is that it is less sensitive to non-isotropic transmission ranges. If

a node very close to the transmitting node receives a packet at a very low power level, we may nevertheless assume that it is at the boundary of the transmission range, e.g., due to a very high attenuation factor or a very power limited sender. Furthermore, nodes do not need to store their position in the packet header. This reduces not only the size of the packet and consequently the energy to transmit and receive it, but also allows faster processing as packets remain unaltered through the whole broadcasting. Thus, no overhead and external information is required at all.

6.2.3 Maximizing Network Lifetime

The objective of extending the network lifetime can be complementary to the objective of minimizing the number of transmissions to reach all nodes, cf. Section 3.5.2. It may be beneficial that more nodes with a lot of residual battery energy broadcast a packet instead of fewer nodes with an almost depleted battery. In scenarios, where the source of the broadcast message is almost uniformly distributed over all nodes in the network or mobility is high and movement patterns are random, we may expect that the traffic load is also uniformly distributed over all nodes, and thus the battery will deplete roughly at the same time at all nodes. However, in many network environments, nodes rarely move and traffic flows are highly directed. This especially applies to sensor networks where all traffic is normally originating from or directed to one or a few designated sinks and the mobility is rather low. If a deterministic algorithm is applied in such a scenario, which does not take into account the battery level at the nodes, the same nodes will always rebroadcast the packet. Consequently, some nodes will deplete much quicker than others will.

In DDB_{RB} , the calculated delay by DFD depends solely on the residual battery level of a node and does not take into account the additionally covered area and the signal strength. They are only used to determine whether to rebroadcast a packet, i.e., whether they are smaller than RT . Nodes with an almost depleted battery schedule the rebroadcasting of the packet with a large delay whereas nodes with a lot of remaining battery power forward the packet almost immediately. Consequently, energy is conserved at almost depleted nodes, which increases their lifetime and in turn extends the connectivity of the network. Therefore, we simply adapt the DFD function to favor nodes with a lot of residual battery energy for rebroadcasting of packets. The DFD function introduces a small delay for nodes with a lot of battery energy whereas nodes with an almost depleted battery add a large delay. This is again done similar as in (6.2).

$$Add_Delay = Max_Delay \cdot \sqrt{\frac{e - e^{E_B}}{e - 1}} \quad (6.6)$$

E_B is the remaining battery power of a node as a percentage of the total battery capacity. The possible benefit of such an energy-based scheme is highly depending on the MAC protocol and the ratio between the energy consumption of sending/receiving/idle listening. If idle listening consumes a substantial amount of energy compared to actual sending and receiving, all nodes spend their energy almost independently whether they forward packets or not. In scenarios, where either the MAC protocol puts a node into sleep mode to save

energy or sending/receiving consume substantial more energy than idle listening, it is essential that the task of forwarding packets is fairly distributed among the nodes to maximize network lifetime even if traffic flows are spatially constant.

6.2.4 Optimizations

We explicitly discuss these optimization for DDB_{AC} . They may be applied completely analogously for other DDB versions such as DDB_{DB} , DDB_{SS} , and DDB_{RB} .

“First Always” Forwarding Policy

A common problem of broadcast protocols based on fixed parameter values is that they are not able to cope with varying network conditions such as node density and traffic load, cf. Section 3.5.3. DDB also uses a rebroadcasting threshold and thus would be susceptible to the same problem. However, only to a reduced extent as the DFD concept already enables DDB to cope better with such scenarios. A minor modification to the forwarding policy eliminates the problem almost completely. Nodes always forward a packet, which is received exactly once after the DFD expires *independent* of the additional coverage, i.e., even if $AC < RT$. That means that the rebroadcasting threshold is only applied from the second received packet onwards. Especially in sparse networks, even a node with only very little additional covered area, may still be the only one to connect to other nodes and serve as the bridge to other node clusters. With this “first always” forwarding policy of DDB, the packet will almost always be forwarded in such scenarios thus reducing the risk of packets dying out. At the same time there is only a small increase in the number of “unnecessary” transmissions compared to the case when the threshold is applied to all packets, including the first received packet. Particularly in dense networks, nodes overhear more than one copy such that the “first always” forwarding policy is not applied anyway. Thus, nodes apply the threshold criterion as normal, which prevents packets from being rebroadcasted.

Cross-Layer Information

Only DDB on the network layer is able to interpret the payload of the packet such as source ID and sequence number and, thus, detects that a newly received packet is a redundant packet. As long as the packet has not yet been passed down to the MAC layer, this does not create a problem. The node simply either drops the packet if the threshold RT is exceeded or recalculates a new DFD for that packet. However, it may frequently happen that the packet has already been forwarded to the MAC layer. Two neighboring nodes normally receive the same broadcast packet almost simultaneously and may calculate nearly the same additional delay before rebroadcasting, i.e., because they have the same additional coverage. Thus, the packet is handed down to the MAC layer at about the same time and both nodes try to send the packet. The MAC layer is responsible to serialize the two transmissions. In this situation, a network layer protocol normally has no longer control over the further processing and cannot prevent the second “unnecessary” of the two transmissions. DDB is able to access packets on the MAC layer, more precisely in the queue of the wireless

interface, and to reprocess them accordingly by either dropping the packets or scheduling their transmissions for a later time.

Directional Antennas

As we have seen in Section 6.2.2, at least 39% of the transmission range of a node was covered by previous transmissions, often much more. Consequently, a transmission with an omnidirectional antenna radiates a lot of power unnecessarily in directions where no additional area can be covered. Directional antennas may mitigate this drawback by forming the beam only in directions of uncovered areas. Furthermore, for certain scenarios, the packet does not need to be broadcasted to all nodes in the network but only in some specific directions. In sensor networks, a request is sent into the network to collect some data from a specific region, thus, nodes distant from the target region broadcast the packet only to nodes in the corresponding direction and not to all neighbors. DDB could be further improved, if nodes are equipped with directional antennas. Implementing DDB with directional antennas and a comparison with broadcast protocols, which make use of directional antennas, are outside the scope of this thesis and left for future work.

6.3 Analytical Evaluation

For our analytical evaluation, we again use the unit disk graph network model, i.e., transmission ranges are isotropic and have a fixed radius of 1. We also assume an unbounded simulation area where the nodes are distributed according to a two-dimensional homogenous Poisson point process of constant spatial intensity. In a first phase, we analyze the DFD function used to calculate the additional delay at rebroadcasting nodes. Afterwards, we try to estimate the size of the additionally covered area by each node's transmission.

6.3.1 DFD Function

We assume a potential rebroadcasting node at a distance $t \in [0, 1]$ from the previous transmitting node. For a random variable X describing the distance t , the cumulative distribution function $F_X(t)$ is given simply by dividing the area of a circle with radius t by the size of the whole transmission range π .

$$F_X(t) = P(X \leq t) = \frac{t^2\pi}{\pi} = t^2$$

We obtain the corresponding probability density function $f_X(t) = 2t$ by derivation. Thus, informally speaking, the probability is twice as high for a rebroadcasting node to be located at a distance t_0 than at a distance $\frac{t_0}{2}$. A DFD function that maps the distances linearly to the range $[0, Max_Delay]$ would not account for this circumstance such that more distant nodes have a higher probability that their packets will collide. This is in contrast to the objective that they rebroadcast first in order to reduce the total number of transmissions and thus should not have a higher risk of collisions.

In order to have the same probability of two rebroadcasting nodes transmitting simultaneously, i.e., of collisions, independent of their distances to the

previous transmitting node, a delay function $f(t)$ has to fulfill the following requirement

$$2(f(t) - f(t + \epsilon)) = f(2t) - f(2t + \epsilon)$$

because the probability that $t \in [x, x + \epsilon]$ is twice as small as $t \in [2x, 2x + \epsilon]$ for any neighbor. This requirement is depicted graphically in Fig. 6.5. Let $\epsilon \rightarrow 0$

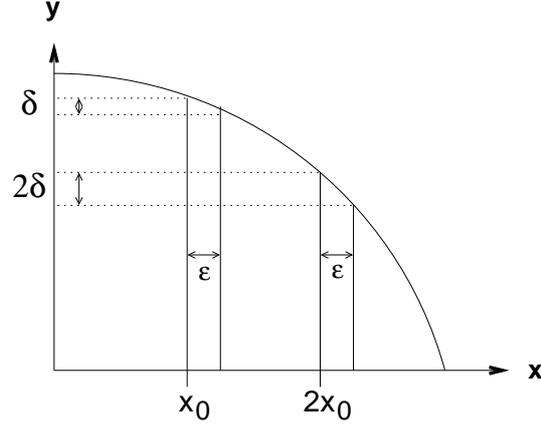


Figure 6.5: Condition for the DFD function

and we immediately obtain

$$2f'(t) = f'(2t) \quad (6.7)$$

We solve this equation by guessing. It is easy to see that the equation does not hold for any exponential function like $f(t) = e^t$. For a polynomial $f(t) = at^n + c$, we have $f'(t) = ant^{n-1}$ and together with (6.7)

$$2ant^{n-1} = an(2t)^{n-1}$$

which yields $n = 2$ and we obtain

$$f(t) = at^2 + c$$

As we want to minimize the number of transmissions and thus want distant nodes to transmit first, we have the additional requirement for the DFD function that $f(0) = 1$ and $f(1) = 0$, which yields

$$f(t) = -t^2 + 1$$

Thus, the DFD function should be at least quadratic such that all rebroadcasting nodes have the same probability for collisions independent of their distance t to the transmitting node. Actually, the DFD function should not only assign delays such that all nodes have the same probability for collisions, but even favor more distant nodes as they rebroadcast first and suppress the others anyway. In Fig. 6.6, a linear and the exponential DFD functions of Section 6.2.2 are plotted together with the quadratic analytical DFD function. We can see that the exponential DFD function fulfills this requirement and the *Add_Delay* of distant nodes is distributed even over a larger range, which minimizes the risk of collisions. On the other hand, a linear function increases the probability for

distant nodes to transmit simultaneously as they are distributed over a smaller delay range.

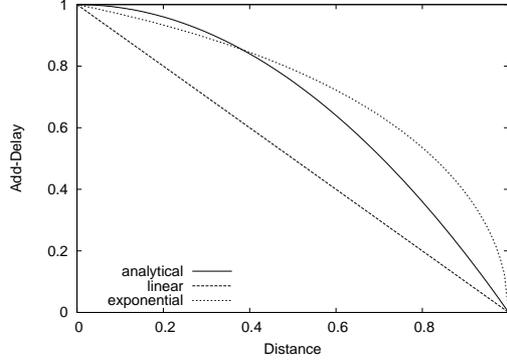


Figure 6.6: Analytical, exponential, and linear DFD functions

6.3.2 Expected Size of Covered Area

We want to calculate the expected size of the additional area AC that is covered by a node's transmission with DDB_{AC} , i.e., nodes which cover a larger additional area broadcast the packet first to minimize the number of transmissions. In order to simplify the calculation, we compute the Taylor series expansion of the additional coverage $AC(d)$ as given in (6.1) with respect to the variable d about the point 0. The Taylor series expansion of a function $f(x)$ about a point $x = a$ is given by

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(n)}(a)}{3!}(x-a)^3 + \dots$$

Thus, we obtain for $AC(d)$

$$AC(d) = d - \frac{1}{8}d^3 + \dots + d + \frac{1}{24} + \dots \simeq 2d \quad (6.8)$$

Let n indicate the number of neighbors and $X_i \in [0, 1]$ be a random variable indicating the Euclidean distance of a neighbor $i \leq n$. We assume that nodes are independently and randomly distributed according to a two dimensional Poisson point process with constant spatial intensity. Thus, the X_i are identically and independently distributed and have the same cumulative distribution function (cdf) and probability density function (pdf). The cdf of the X_i can simply be derived by dividing the area of the circle with radius x by the size of the whole transmission range, which is π . Thus, we obtain for the cdf F_X and pdf f_X with $0 \leq x \leq 1$.

$$F_X(x) = P(X \leq x) = x^2 \quad f_X(x) = 2x$$

From probability theory, we know that for a random variable $V = g(U)$ as a function of a random variable U , the pdf f_V of V can be derived from g and

the pdf f_U of U as follows

$$f_V(x) = f_U[g^{-1}(x)] \frac{d}{dx} g^{-1}(x)$$

Thus, for a random variable Y , which indicates the additional area covered by a node's transmission, given as $Y = g(X) = 2X$ by the approximation of the distance (6.8), the pdf f_Y of Y can be calculated as follows.

$$f_Y(x) = f_X[g^{-1}(x)] \frac{d}{dx} g^{-1}(x) = \frac{x}{2} \quad \text{with } 0 \leq x \leq 2 \quad (6.9)$$

Thus, the cdf of the additional coverage of a node's transmission is simply

$$F_Y(x) = \frac{x^2}{4} \quad \text{with } 0 \leq x \leq 2 \quad (6.10)$$

In order to derive the expected additional coverage of each of the n neighbors, we sort their additional coverage Y_i such that $Y_{(1)} \leq Y_{(2)} \leq \dots \leq Y_{(n)}$. Thus Y_i is only the same as $Y_{(i)}$ with probability $\frac{1}{n}$ and the sample maximum and minimum are $Y_{(n)}$ and $Y_{(1)}$, respectively. Obviously, the k -most distant neighbor has also the k -largest expected additionally covered area. The general cumulative distribution function cdf $F_{Y_{(k)}}(x)$ for all $Y_{(k)}$ is given by

$$\begin{aligned} F_{Y_{(k)}}(x) &= P(Y_{(k)} \leq x) \\ &= \sum_{j=k}^n P(\text{Exactly } j \text{ of the } Y_i \leq x) \\ &= \sum_{j=k}^n \binom{n}{j} [F_Y(x)]^j [1 - F_Y(x)]^{n-j} \end{aligned}$$

where $F_Y(x)$ is the cdf of the Y_i as given in (6.10).

The derivation $f_{Y_{(k)}}$ of $F_{Y_{(k)}}$ with respect to x can be calculated straightforwardly.

$$\begin{aligned} f_{Y_{(k)}}(x) &= \frac{d}{dx} F_{Y_{(k)}}(x) = \frac{d}{dx} \left[\sum_{j=k}^n \binom{n}{j} [F_Y(x)]^j [1 - F_Y(x)]^{n-j} \right] = \\ &= \frac{d}{dx} \left[\binom{n}{k} [F_Y(x)]^k [1 - F_Y(x)]^{n-k} + \binom{n}{k+1} [F_Y(x)]^{k+1} [1 - F_Y(x)]^{n-k-1} \right. \\ &\quad \left. + \dots + \binom{n}{n-1} [F_Y(x)]^{n-1} [1 - F_Y(x)]^1 + \binom{n}{n} [F_Y(x)]^n [1 - F_Y(x)]^0 \right] \end{aligned}$$

and thus

$$\begin{aligned} f_{Y_{(k)}}(x) &= \\ &= \binom{n}{k} \left[k F_Y(x)^{k-1} f_Y(x) (1 - F_Y(x))^{n-k} \right. \\ &\quad \left. - (n-k) F_Y(x)^k (1 - F_Y(x))^{n-k-1} f_Y(x) \right] + \end{aligned}$$

$$\begin{aligned}
& \binom{n}{k+1} \left[(k+1)F_Y(x)^k f_Y(x) (1-F_Y(x))^{n-k-1} \right. \\
& \quad \left. - (n-k-1)F_Y(x)^{k+1} (1-F_Y(x))^{n-k-2} f_Y(x) \right] + \dots + \\
& \binom{n}{n-1} \left[(n-1)F_Y(x)^{n-2} f_Y(x) (1-F_Y(x)) - F_Y(x)^{n-1} f_Y(x) \right] + \\
& \binom{n}{n} \left[nF_Y(x)^{n-1} f_Y(x) \right]
\end{aligned}$$

By expanding the terms we obtain

$$\begin{aligned}
f_{Y_{(k)}}(x) &= \frac{n!}{k!(n-k)!} k F_Y(x)^{k-1} f_Y(x) (1-F_Y(x))^{n-k} - \\
& \frac{n!}{k!(n-k-1)!} F_Y(x)^k (1-F_Y(x))^{n-k-1} f_Y(x) + \\
& \frac{n!}{(k+1)!(n-k-1)!} (k+1) F_Y(x)^k (1-F_Y(x))^{n-k-1} f_Y(x) - \dots \\
& + n(n-1) F_Y(x)^{n-2} f_Y(x) (1-F_Y(x)) - \\
& n F_Y(x)^{n-1} f_Y(x) + \\
& n F_Y(x)^{n-1} f_Y(x)
\end{aligned}$$

what eventually simply yields

$$f_{Y_{(k)}}(x) = \binom{n}{k} k F_Y(x)^{k-1} f_Y(x) (1-F_Y(x))^{n-k}$$

From (6.10), we have $F_Y(x) = \frac{x^2}{4}$ and obtain

$$f_{Y_{(k)}}(x) = \binom{n}{k} k \left(\frac{x^2}{4}\right)^{k-1} \frac{x}{2} \left(1 - \frac{x^2}{4}\right)^{n-k} \quad \text{with } 0 \leq x \leq 2$$

It is well-known that the expected value of a random variable Z can be calculated from its pdf f_Z by

$$E_Z = \int_{-\infty}^{\infty} x f_Z(x) dx \quad (6.11)$$

Therefore, we obtain the expected value $E_{AC}^{Y_{(k)}}$ for the additional coverage for the k -most distant neighbor solely depending on the number of neighbors n as follows.

$$\begin{aligned}
E_{AC}^{Y_{(k)}} &= \int_0^2 \binom{n}{k} k \frac{x}{2} x \left(\frac{x^2}{4}\right)^{k-1} \left(1 - \frac{x^2}{4}\right)^{n-k} dx \\
&= 2 \binom{n}{k} k \int_0^2 \left(\frac{x^2}{4}\right)^k \left(1 - \frac{x^2}{4}\right)^{n-k} dx \quad (6.12)
\end{aligned}$$

In order to calculate this integral, we use the beta function $B(p, q)$, which is

defined by

$$B(p, q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$$

and can be expressed as

$$B(p, q) = \int_0^1 u^{p-1}(1-u)^{q-1} du$$

To put it in the form we need, let $u = \frac{x^2}{4}$ and $du = \frac{1}{2}x dx$, and

$$\begin{aligned} \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} &= \int_0^1 u^{p-1}(1-u)^{q-1} du \\ &= \frac{1}{2} \int_0^2 \left(\frac{x^2}{4}\right)^{p-1} \left(1 - \frac{x^2}{4}\right)^{q-1} x dx \\ &= \int_0^2 \left(\frac{x^2}{4}\right)^{p-\frac{1}{2}} \left(1 - \frac{x^2}{4}\right)^{q-1} dx \end{aligned}$$

Together with (6.12), this yields

$$E_{AC}^{Y^{(k)}} = 2 \binom{n}{k} k \frac{\Gamma(n-k+1)\Gamma(k+\frac{1}{2})}{\Gamma(n+\frac{3}{2})}$$

and by using $\Gamma(n) = (n-1)!$, we finally obtain

$$E_{AC}^{Y^{(k)}} = \frac{2\Gamma(n+1)\Gamma(k+\frac{1}{2})}{\Gamma(k)\Gamma(n+\frac{3}{2})} \quad (6.13)$$

We compare this result with the expected additional coverage E_{AC}^* of other stateless broadcasting schemes where the sequence of neighbors' transmission is independent of their additional coverage, e.g., as in the location-based and probability-based schemes, cf. Section 3.5.2. Clearly, the pdf f_Y of the additional coverage for a single node is the same as derived before in (6.9). However, the expected additional coverage is independent of the number of neighbors n and the same for all neighbors $k \leq n$ and therefore is constant. Again with (6.11), we obtain

$$E_{AC}^* = \int_0^2 x \frac{x}{2} dx = \frac{4}{3}$$

In Fig. 6.7, the graph is plotted for $E_{AC}^{Y^{(k)}}$ of DDB_{AC} and E_{AC}^* of other stateless broadcasting algorithms depending on the number of neighbors for $n = 1 \dots 30$. Again, $k \leq n$ denotes the k -most distant neighbor, i.e., the node with the k -largest additional coverage. E_{AC}^* is simply the plane at $\frac{4}{3}$. Already for very few neighbors, the “best” node, i.e., $k = n$, already covers almost the maximum size of additional area of 1.91. Furthermore, the next $k \leq n$ -best nodes cover normally more than $\frac{4}{3}$ what would be covered by a node's transmission with other stateless broadcasting schemes. Assuming the same rebroadcasting threshold RT for DDB_{AC} and the other location- and probability-based schemes, we can conclude that we might expect an improved performance up to $43\% = \frac{1.91}{4/3}$ in terms of transmissions. However, the advantage of DDB_{AC} is

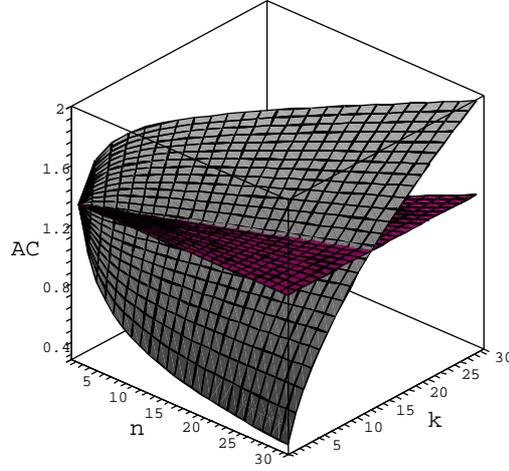


Figure 6.7: Expected additional coverage

not only the reduction in number of transmissions, but also that the delay can be reduced as distant nodes which transmit first add almost no delay. From the expected additional coverage in (6.13) and the DFD function (6.2), we can easily determine the expected additional delay introduced at the nodes.

$$E_{AD}^{Y(k)} = Max_Delay \cdot \sqrt{\frac{e - e^{-\frac{1.91 \cdot 2 \Gamma(n+1) \Gamma(k+\frac{1}{2})}{\Gamma(k) \Gamma(n+\frac{3}{2})}}}{e - 1}}$$

The results are depicted in Fig. 6.8 for a *Max_Delay* of 1. Most nodes, which broadcast, i.e., where k is in the order of n , delay the transmission only by a small fraction of the total *Max_Delay*.

Furthermore with DDB_{AC} we know that nodes which cover a larger additional area broadcast first and thus can design the DFD accordingly, which allows reducing the number of collisions. In other stateless schemes, the delay has to be much larger to have the same number of collisions than in DDB_{AC} as neighbors transmit randomly. As it is difficult to assess the exact influence of the MAC layer and to take into account the dependencies between neighboring nodes when their transmission ranges overlap, these analyses only provide a rough kind of boundary for the performance.

Obviously, the values are only correct when the transmission ranges of rebroadcasting neighboring nodes do not overlap. There is a maximum of three non-overlapping transmission ranges of neighboring nodes, namely when the three nodes are located at the boundary of the source node's transmission range forming a regular triangle. Thus, for more than three rebroadcasting neighbors, the transmission ranges inevitably overlap. Often already the second rebroadcasting node's transmission range may partially overlap with the first one. Thus, only the expected additional coverage for the first transmitting node is ex-

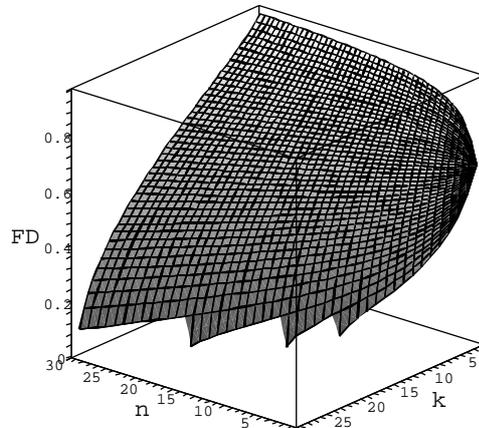


Figure 6.8: Expected delay through DFD

act. For the following rebroadcasting nodes, the derived values are slightly too high. The values are more and more overestimated when more nodes transmit and more transmission ranges overlap. Thus, the general conclusions of the analytical results hold and will be validated in the Section 6.4. Especially if considering that the dynamic recalculation of the DFD ensures that maximally distant nodes rebroadcast first and, thus, only overlap partially.

6.4 Simulations

6.4.1 Parameters and Scenarios

We implemented and evaluated the protocol in the Qualnet network simulator [211]. The results are averaged over 10 simulation runs and given with a 95% confidence interval, which is sometimes very small and barely visible. The payload of the packets is 64 bytes and the interface queue length is set to 1500 bytes. Radio propagation is modeled with the isotropic two-ray ground reflection model. The transmission power and receiver sensitivity are set corresponding to a nominal transmission range of 250m. We use IEEE 802.11b on the physical and MAC layer operating at a rate of 2 Mbps. The simulations last for 900s and data transmission starts at 180s and ends at 880s such that emitted packets arrive at the destination before the end of the simulation.

The DDB protocol was implemented with two optimizations proposed in Section 6.2.4, namely the “first always forwarding policy” and the “cross-layer information”. However, we did not use directional antennas for DDB as the other protocols were not optimized for use with directional antennas. The performance of DDB is compared to the location-based broadcasting protocol, cf. Section 3.5.2, which is abbreviated by LBP in the following. Furthermore, we also implemented the multipoint relay MPR, cf. also to Section 3.5.2, and simple

flooding which is the most simple broadcasting protocol. LBP and MPR were chosen as representatives for the categories of stateless and stateful broadcast protocols, respectively. However, we did not use any energy-efficient and directional antenna-based algorithms for comparison as they use adjustable transmission power and transmission directions, respectively.

The parameters of LBP and MPR are set as suggested in [258] and in RFC 3626 [44], respectively. Specifically, the random delay at each node for LBP is set to 10 ms and the rebroadcasting threshold to 40% of the maximal additional covered area. The hello message interval and neighbor hold time are 2 s and 6 s respectively for MPR. With flooding, the packets are jittered 2 ms to avoid that all neighbors transmit simultaneously.

6.4.2 Evaluating Different Versions of DDB

We first simulated DDB in various scenarios to study the effect of the different components and also to determine appropriate values for the rebroadcasting threshold RT and the Max_Delay . These simulations had three main purposes, namely to optimize the parameters, verify the assumptions, and find weaknesses. DDB_{RB} was not evaluated in this section as it has a different objective and we only used it in the simulations where we consider network lifetime.

Versions to Minimize the Number of Transmissions

We proposed three versions of DDB all of which have the same objective of reducing the number of rebroadcasting nodes, namely DDB_{AC} , DDB_{DB} , DDB_{SS} . In this subsection, we compare their relative performance. The rebroadcasting thresholds RT are set to 40% of the maximum used for the respective DFD functions. The selection of such a high value will be clarified in the Section 6.4.2. The thresholds for the area- and distance-based versions are easy to determine. For DDB_{AC} and DDB_{DB} , RT was set to 40% of the maximal area a node can cover and 40% of the maximal transmission radius, respectively. For a transmission radius of 250 m this yields $0.4 \cdot 1.91 \cdot (250\text{ m})^2 \simeq 47750\text{ m}^2$ for DDB_{AC} and $0.4 \cdot 250\text{ m} = 100\text{ m}$ for DDB_{DB} . The threshold for the signal strength version requires some calculations and further assumptions about the typical attenuation factor. The attenuation factor in real physical environments is about 2 for free space and may raise up to 6 for indoor environment. We choose an average attenuation factor of $a = 3$ to roughly estimate the distance between nodes in the signal strength version DDB_{SS} . We set the threshold RT to $S_r + 12\text{ dBm}$. The value is motivated by the fact that nodes with 40% additional coverage are at a distance of approximately 100 m . This is 2.5 times closer to the source than a node at 250 m , which receives a packet just at S_r and has the maximal additional covered area. Assuming an average attenuation factor of 3, this immediately yields a signal strength at a distance of 100 m which is $10 \cdot \log_{10}(2.5^3) = 12\text{ dBm}$ stronger than S_r . Obviously, we could derive exact distances from signal strengths as the underlying propagation model is known. Thus, the performance of the DDB_{DB} and DDB_{SS} would be the same. In reality, the attenuation factor is not known and can only be estimated. Therefore, we did also not use the exact attenuation factor used in the two-ray ground reflection propagation model, which is 2 until a certain distance and 4 thereafter.

The simulations were conducted in a static network without any congestion as we wanted to compare the efficiency of the core algorithms and excluded any external influences. Thus, only one source broadcasts one packet per second. We placed 1000 nodes randomly over a square area with side lengths of 1414, 2000, 2828, 4000, 5656 m to obtain different node densities. The density is always doubled for the next smaller area size and equals approximately 6, 12, 24, 49, and 98 neighbors per node. The minimum node density of five neighbors was chosen as results from percolation theory have shown [60] that five neighbors is just about the minimal required density for a completely connected network. For lower node density, the network is almost always disconnected. However, it may still happen that the network is not completely connected with only five neighbors and that a packet cannot be delivered to all nodes. To eliminate this bias of the results, we implemented an algorithm to determine the size of the maximal connected cluster which includes the source node. The delivery ratio and the number of rebroadcasting nodes are calculated relatively to the size of that cluster.

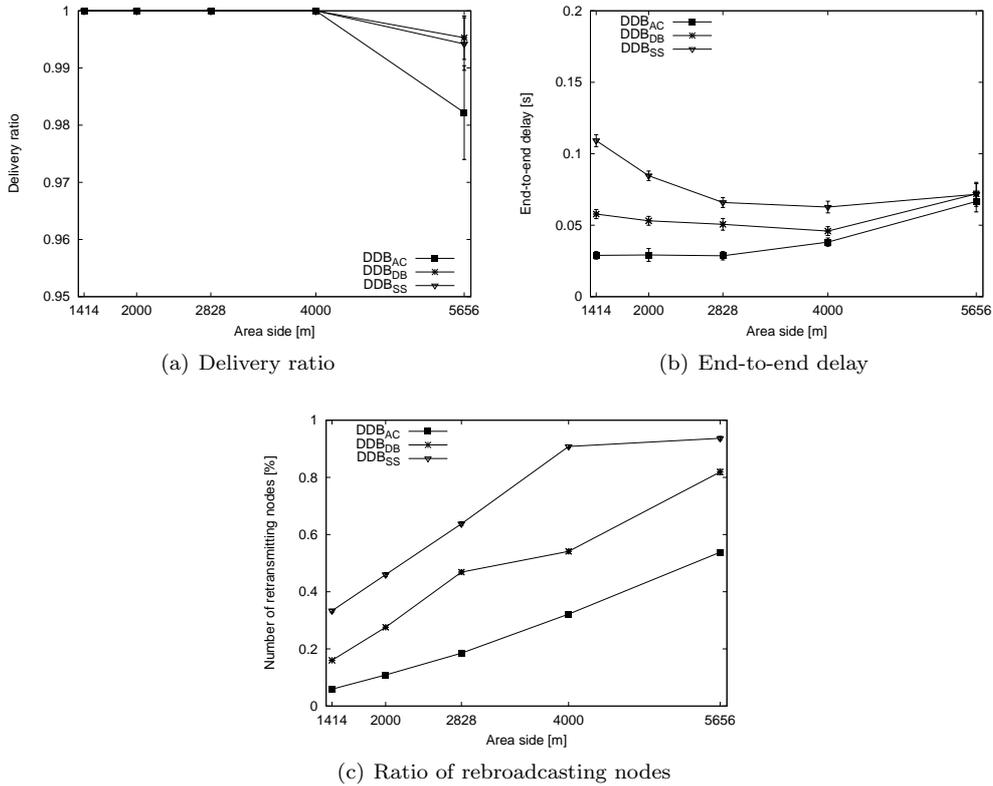


Figure 6.9: Comparison of different DFD functions

The delivery ratio is almost always 100% Fig. 6.9(a), except for very sparse networks where all protocols suffer slightly. DDB_{AC} has the lowest delivery ratio, even though it is still higher than 98%. This is due to the fact that the metric of the additionally covered area is additive. Although none of the neighbors covers by itself more than the threshold of 40%, all together may cover

more than the threshold and suppress some nodes from rebroadcasting. On the other hand, DDB_{DB} and DDB_{SS} are not additive. As long as no node is below the 40% threshold, the node will rebroadcast, independent on how many nodes already cover 39%. Due to the same reason, the number of rebroadcasting nodes for the DDB_{AC} is smaller than that for DDB_{DB} however Fig. 6.9(c). DDB_{SS} performed not as well as the DDB_{DB} because signal strengths only allow to approximate distances. We used a path loss factor of three, which does not correspond to the path loss factor of the used two-ray ground propagation model, and therefore the estimated distances between nodes are not very accurate. Furthermore, the transmission ranges are perfectly circular so that distances can be mapped accurately to covered area. The situation may completely look different in case of irregular transmission ranges, where distances may no longer correspond to the additionally covered area. For example, if the signal strength just equals the receiver sensitivity, the transmitting node is at the boundary of the transmission range, although the node may be very close due to a high path loss factor. DDB_{AC} outperforms the other two versions also with respect to end-to-end delay Fig. 6.9(b). This is to due the reduced number of transmitting nodes. Thus, in the following we will only evaluate DDB_{AC} in more detail. The general observations should however still hold for the other two versions as well.

Impact of *Max_Delay*

The delivery ratio was similar to the results in the previous subsection, almost 100% for all scenarios, and independent of the *Max_Delay* and, thus, is not depicted. In Fig. 6.10(b), the delay for *Max_Delay* of 2, 5, and 10 *ms* is given. A smaller *Max_Delay* has a significant smaller delay in sparse networks. The difference is reduced for denser networks. On the other hand, we can see that the number of rebroadcasting nodes is basically not affect of different values for *Max_Delay* Fig. 6.10(a). As we will seen in Section 6.4.2, the reason that a shorter *Max_Delay* does not increase the rebroadcasting nodes is due to the “cross-layer information” optimization.

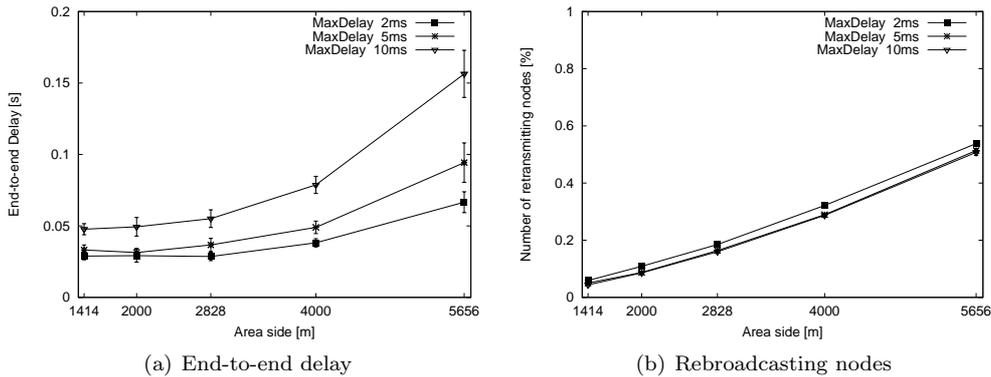


Figure 6.10: Impact of *Max_Delay*

Impact of Rebroadcasting Threshold RT

The values of RT are given as ratio of the maximal additionally covered area, i.e., 0% signifies that no area must be left uncovered. The results show that the delivery ratio does not suffer significantly from a higher rebroadcasting threshold RT even in sparse networks Fig. 6.11(a). The reason is the “first always” forwarding policy, which ensures that in dense network where nodes only may receive one packet, the packets are rebroadcasted independent of the additional coverage. On the other hand, we observe that a higher RT has a major impact on the delay and the rebroadcasting nodes Fig. 6.11(b) and Fig. 6.11(c). Most notably, the rise from 0 to 10% of the maximal additional area yields much better values, whereas a greater increase only marginally improves the results. That indicates that most neighbors cover less than 10% of additional area after having received some transmission from their neighbors and do not contribute significantly if they transmit.

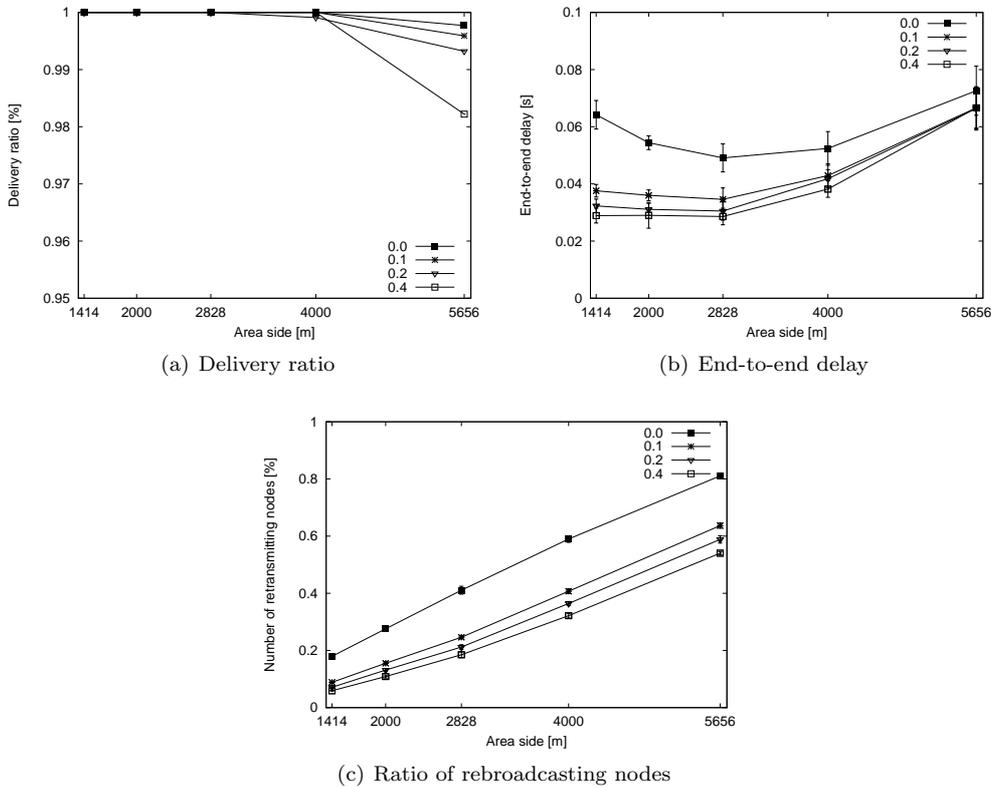


Figure 6.11: Impact of rebroadcasting threshold RT

Impact of the Different Components

In this section we evaluate the impact of the two different optimizations proposed in Section 6.2.4, namely of the “first always” forwarding policy and the “cross-layer information”. We compare the performance of the DDB_{AC} with

both optimizations to two slimmed versions, each one only comprising one of the optimizations. In the DDB_{AC} version without the “first always” optimization, the rebroadcasting threshold is applied if only one packet is received and not only if two or more redundant packets are received. If the “cross layer information” is not enabled, DDB_{AC} does not have the ability to access packets on the MAC layer. Thus, as soon as DDB_{AC} passes the packet down to the MAC layer, the packet will be sent and can no longer be cancelled. In Fig. 6.12(b) and Fig. 6.12(c), we can observe that the delay remains unaffected by the “first always” forwarding policy and that the number of rebroadcasting nodes is increased marginally. On the other hand, in Fig. 6.12(a) the delivery ratio sharply drops in sparse networks, if the “first always” option is not enabled. This optimization allows DDB to cope efficiently with varying node densities. These results correspond to our prior considerations in Section 6.2.4.

The performance of DDB_{AC} without the “cross-layer information” suffers drastically, especially in numbers of rebroadcasting nodes Fig. 6.12(c). The ratio to the DDB_{AC} is about two for sparse networks, but then increases to more than 10 for denser networks. As more nodes transmit almost simultaneously, the ability to access packets on the MAC layer is more beneficial in denser networks. The increased delay in Fig. 6.12(b) is a consequence of the higher number of transmitting nodes. However, if we simply increase the *Max_Delay* to 10 ms, the performance without the “cross-layer information” optimization almost equals again the “original” DDB Fig. 6.12(d). With this longer *Max_Delay*, nodes keep the packet longer before passing to the MAC layer, this in turn increases the probability of receiving redundant packets to the point of exceeding the rebroadcasting threshold. We may conclude that this optimization allows us to have a short *Max_Delay* which decreases the end-to-end delay.

Conclusions

As the simulations showed a superior performance of DDB_{AC} in most scenarios, we exclusively used DDB_{AC} for the comparison with other broadcast protocols in the following sections. The two parameters are set to values which were found to have the best average performance in those scenarios, i.e., *Max_Delay* = 2 ms and the rebroadcasting threshold to 40% of the maximal additionally covered area. Interestingly, this is the same rebroadcasting threshold as proposed for LBP in [258]. For lower values, LBP was not able to reduce significantly the number of retransmitting node. However as we will see later from the simulation results, the performance of LBP suffers in sparse networks and not all packets could be delivered. This inability to adapt to different networking conditions is typical for stateless algorithms as discussed earlier in Section 3.5.3.

However, we have to keep in mind that the situation may look different, if transmission ranges are highly irregular. In such scenarios, DDB_{SS} may show superior performance as the signal strength naturally accounts for those effects while distances do not.

6.4.3 Efficiency of Packet Delivery

To compare the performance in terms of rebroadcasting nodes of the different protocols with a theoretical optimum, we additionally implemented an algorithm that constructs the minimal connected dominating set (MCDS), which

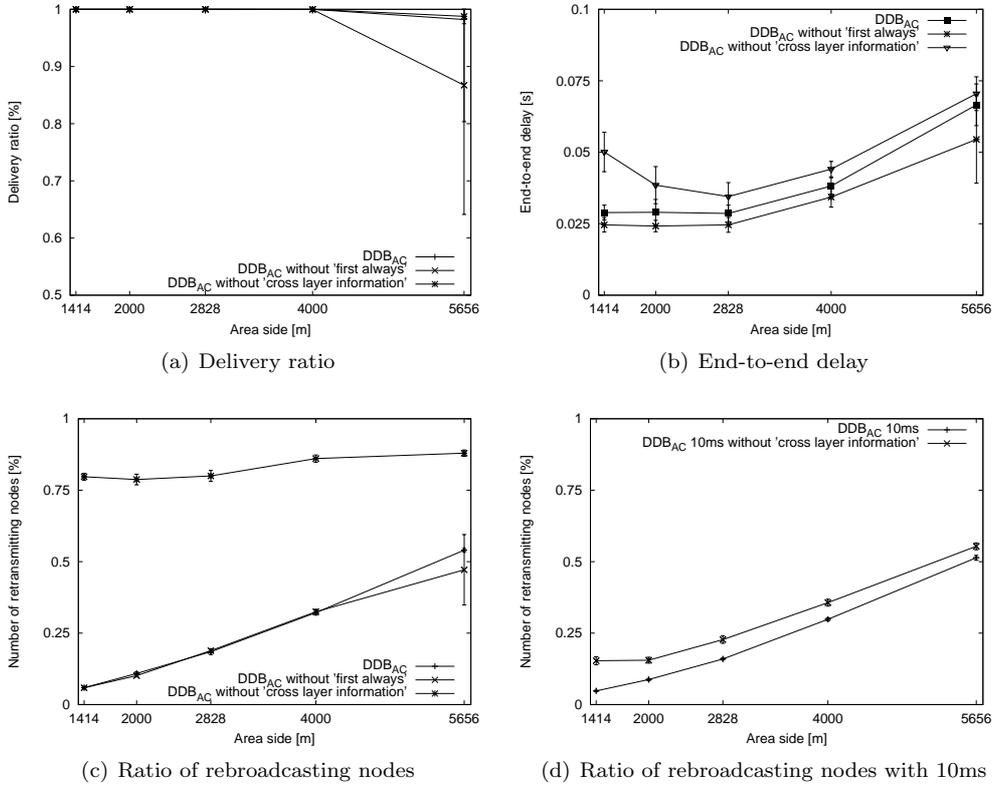


Figure 6.12: Impact of the different components

provides a lower theoretical bound for the number of rebroadcasting nodes. In Fig. 6.13(a), the number of transmissions of DDB_{AC} is about twice as high as for the MCDS for all network densities. As expected from the analytical results in Section 6.3, the number constantly decreases for DDB_{AC} with higher node densities, whereas LBP remains around 45%. This is due to the fact that the expected additional coverage of LBP is constant and increases for DDB_{AC} for higher node densities. Thus, the more neighbors a node has, the more additional coverage the rebroadcasting nodes have and the less transmissions are required. MPR performs significantly better than LBP. This is in accordance with [258], which observed that stateful protocols perform better than stateless protocols in dense networks. However, due to the locally optimal and dynamic rebroadcasting decisions, the stateless DDB_{AC} outperforms even MPR. Although the ratio of MPR also decreases for higher node densities, it always remains significantly above the ratio of DDB_{AC}. The results in Fig. 6.13(b) show that the delay of DDB_{AC} first drops and then remains almost constant. For low node densities, a node has few neighbors that often do not cover a substantial additional area, but need to transmit anyway as no other neighbors rebroadcast. These nodes add a non-negligible delay through the DFD function (6.2). For higher node densities, the delay is much shorter as the “best” nodes are close to the transmission range boundary and therefore calculate a short DFD. DDB_{AC} always performs much better than LBP for two reasons. Nodes delay packets indepen-

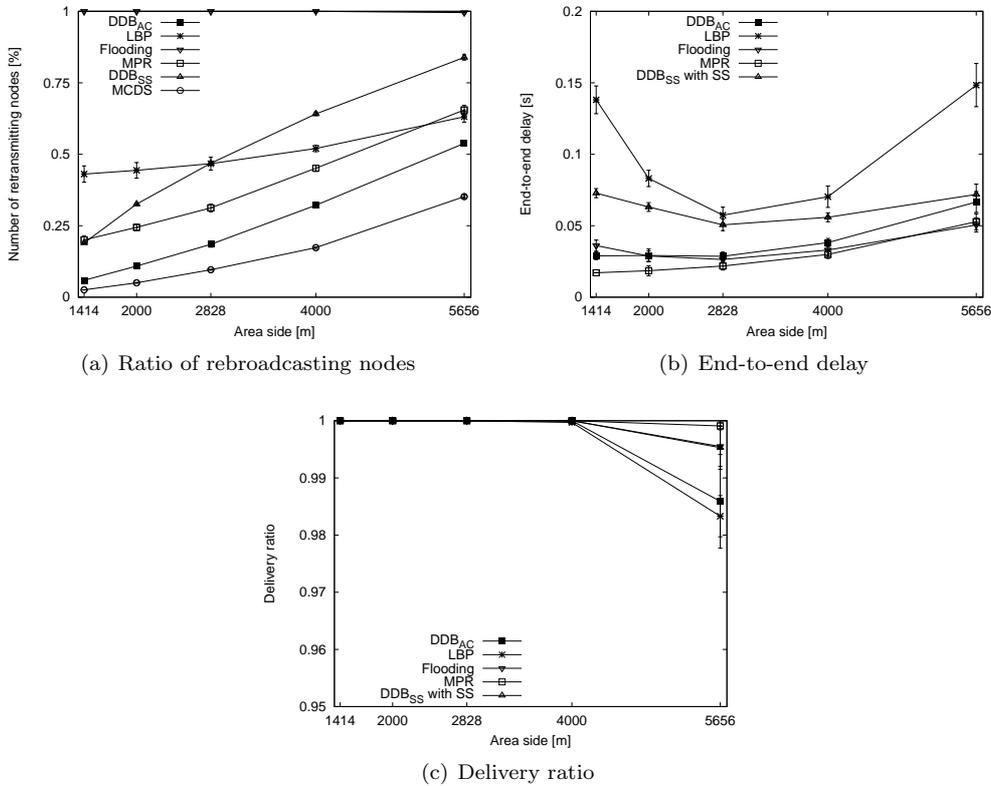


Figure 6.13: Algorithm efficiency

dently of the additional coverage in LBP and the delay has to be much higher to avoid collisions. These facts are again supported by the analytical results. The delay for LBP increases because the number of retransmitting nodes is not reduced for higher node densities, which causes more and more collisions. Thus, nodes may not receive the actual first packet due to these collisions and have to “wait” for another copy which increases the delay. Even though, MPR relays packets immediately, the delay was only slightly lower than that of DDB_{AC}, especially in denser networks. Again this is because the “best” nodes in DDB_{AC} rebroadcast first and add lower delays for higher node densities. As expected, DDB_{SS} performs satisfactory, but not as well as DDB_{AC} based on the additionally covered area because signal strength only allows a rough estimation of the distance. The delivery ratio Fig. 6.13(c) is always 100%, except for the case of very sparse networks with about five neighbors per node where the ratio drops marginally to approximately 99%.

6.4.4 Mobile Networks

These simulations were computationally expensive, especially for MPR, and required a lot of memory. Therefore, we could only run simulations with 80 nodes. The size of the simulation areas were adapted accordingly to yield on average 9, 19, and 49 neighbors per node similar to the node densities used in

the previous subsection, but omitting the sparsest and densest networks with six and 98 neighbors respectively. We did not conduct simulations with 98 neighbors as then all nodes could be covered just by one transmission and the results are no longer meaningful. The reason for excluding six neighbors is that it is hardly possible to determine reliably the size of the maximal cluster in a mobile network for every point in time when a packet is transmitted and received. To obtain results without network partition, the minimal node density was increased to nine neighbors. Packets are generated at a rate of 10 packets per second and nodes move according to the random waypoint mobility model. As the stationary distribution of the random waypoint mobility model is not a uniform distribution [179], the number of neighbors is higher for nodes in the center and lower for nodes at the border of the simulation area. The pause time is set to 0 s because higher pause times can lead to pseudo static networks, cf. Section 2.4.2. The minimal and maximal speeds are set to $\pm 10\%$ of an average speed that was varied over 1, 5, 10, 20, and 40 m/s. We also ran the simulation with the rather high speed values of 20 and 40 m/s as we consider speed as a proxy for any kind of topology changes, caused either by mobility, sleep cycles, interferences, adjustment of transmission and reception parameters, etc.

The delivery ratio is depicted in Fig. 6.14(a) for an average network density of nine neighbors. The three stateless protocols are not affected and the performance remains constant independent of the mobility. The reason for their delivery ratio being slightly below 100% is due to the temporary partition of the network caused by mobility even for an average of nine neighbors. As expected, only the performance of the stateful MPR suffers under mobility because its view on the network topology may be inconsistent, i.e., the known one- and two-hop neighbors do not correspond to the actual physical neighbors. This also causes an incorrect calculation of the forwarding nodes, i.e., either nodes which should rebroadcast the packet based on the physical network topology do not, or which should not broadcast do. If the network density is low, a few wrong rebroadcast decisions may prevent the packets from being delivered to all nodes in the network. Obviously, the number of rebroadcasting nodes drops for the MPR for higher node densities as shown in Fig. 6.14(c) because the delivery ratio decreases significantly and because the percentage of retransmitting nodes is calculated relative to the total number of nodes and not the number of nodes which received the packet. The delays of MPR and DDB are about twice as low as for flooding and LBP and are stable for all protocols over all average speeds as shown in Fig. 6.14(b).

For a higher node density with 19 neighbors as shown in Fig. 6.15, the results are basically the same as for the sparser networks with only two notable differences. First, the performance of MPR did not decrease that significantly for a higher average speed. The reason is that the inconsistent view has a smaller impact as packets can be still delivered due to the high connectivity, even if the “wrong” nodes rebroadcast the packets. And secondly, the delivery ratio of the stateless protocols increased to 1 over all simulations. The high node density keeps the network connected all the time. The simulations could not be conducted for MPR with the highest node density of 49 neighbors as each topology change requires additional computation to determine the forwarding nodes, which turned out to be too resource consuming. Thus, the results are only given for the three stateless protocols in Fig. 6.16. The results are basically the same as before only that the number of rebroadcasting nodes is further

decreased. Again due to the same reasons as already mentioned before, the DDB_{AC} yields the shortest delay among the three stateless protocols followed by LBP and flooding because of the higher number of rebroadcasting nodes.

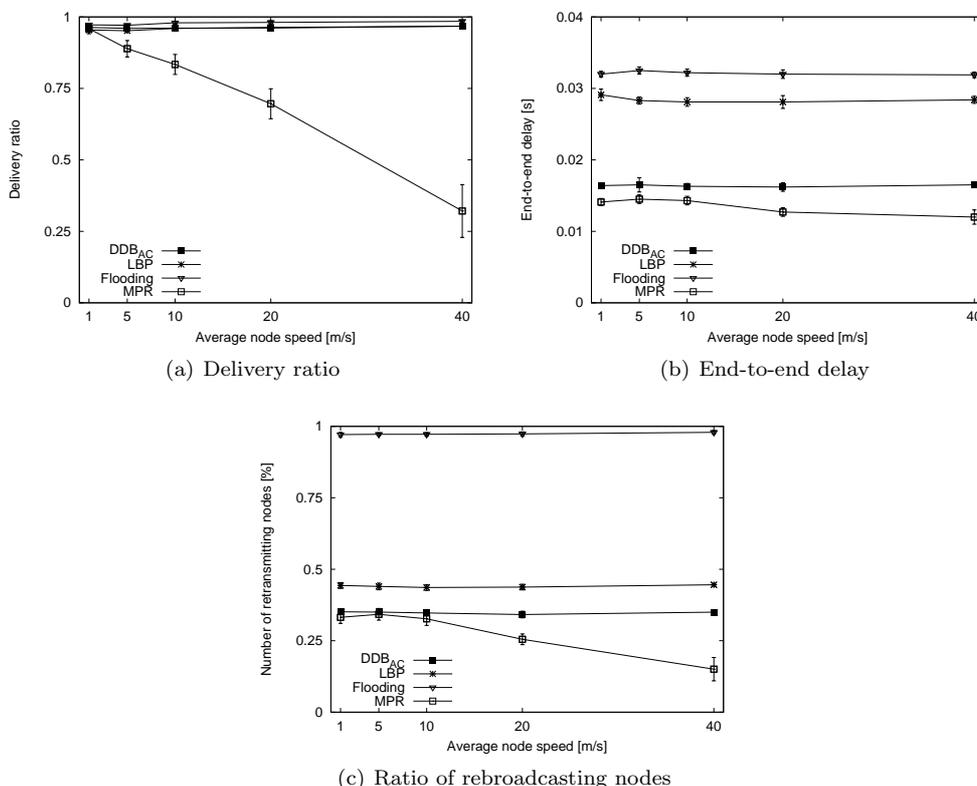


Figure 6.14: Mobility with 9 neighbors

6.4.5 Congested Networks

The simulation parameters are the same as in the mobile network in the previous subsection, without mobility however, as it is the objective of these simulations to evaluate solely the effect of congestion. One randomly chosen node broadcasts packets at different rates from 20 to 100 packets per second.

For an average density of 19 neighbors, the delay and the delivery ratio of all protocols suffer in congested networks due to collisions and queue overflows Fig. 6.17(a). MPR outperforms the other protocols in these scenarios yielding almost always 100% delivery ratio and very short delays. Two facts contribute to this superior performance. First, packets are rebroadcasted at the nodes immediately and, secondly, nodes only have to forward packets received from specific nodes, namely the ones which selected them as forwarding nodes. Thus, the queues do not fill up too quickly. The stateless protocols add delay to each packet and also first have to buffer all packets received from any neighbor. Among the stateless protocols, DDB_{AC} performs by far the best and lags behind only MPR for the highest chosen level of congestion. The delay of DDB_{AC}

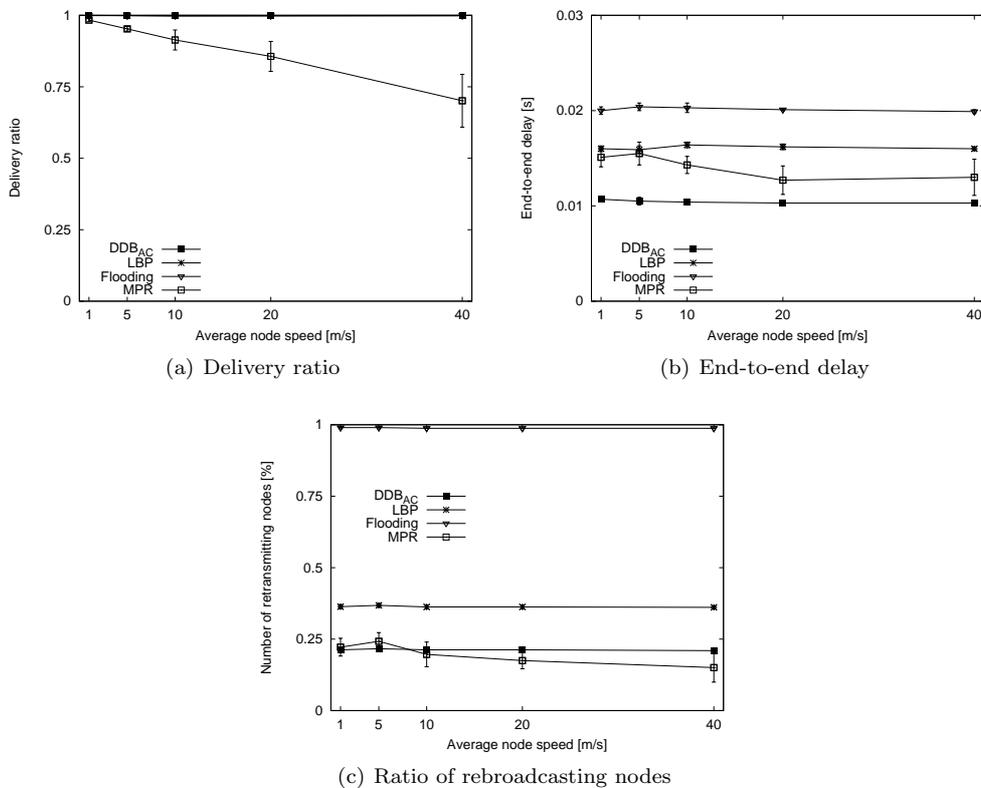


Figure 6.15: Mobility with 19 neighbors

remains very short and only increases for the highest traffic load. It is by a factor of five times or more lower for highly congested networks than the other stateless protocol LBP and flooding. They show an increased delay already for lightly loaded networks. With DDB, only few packets need to be buffered at the nodes because of the short DFD in dense networks, the fewer rebroadcasting nodes, and the high RT threshold that allows the dropping many packets quickly. Flooding suffers from its inability, and LBP from its limited ability, to reduce the number of retransmitting nodes. LBP performs worse than simple flooding because of the required long buffering time of 10 m_s , which causes more queue overflows. The number of rebroadcasting nodes are depicted in Fig. 6.17(b). Only MPR and DDB_{AC} remain unaffected by the packet generation rate, except that DDB_{AC} increases slightly for the highest rate. This is reflected by the increased delay and decreased delivery ratio Fig. 6.17(a). Clearly, the number of retransmitting nodes of LBP and flooding decreases at least with the delivery ratio.

The results are similar for other simulated node densities in Fig. 6.18(a) and Fig.6.18(b). Only three significant differences can be observed. First, in a rather sparse network with only nine neighbors, none of the protocols were able to deliver all the packets. Nodes are connected only over a few links and, thus, if packets are dropped at some nodes due to congestion, the packet can no longer be delivered to all nodes. Secondly, the flooding improved in terms

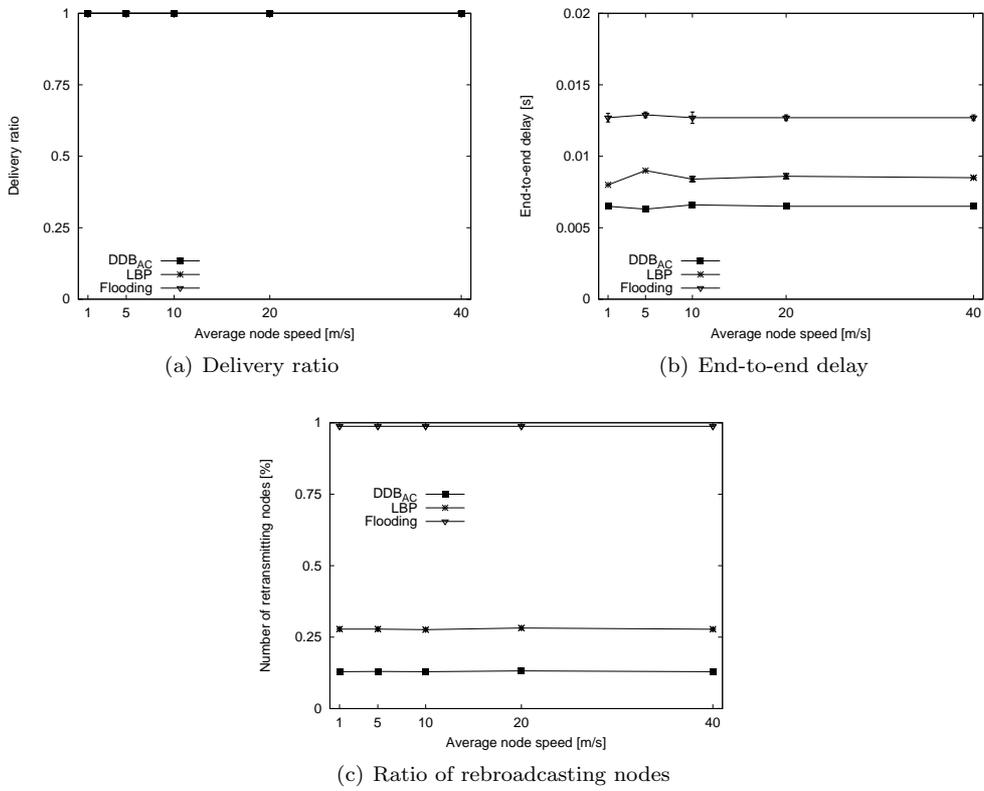


Figure 6.16: Mobility with 49 neighbors

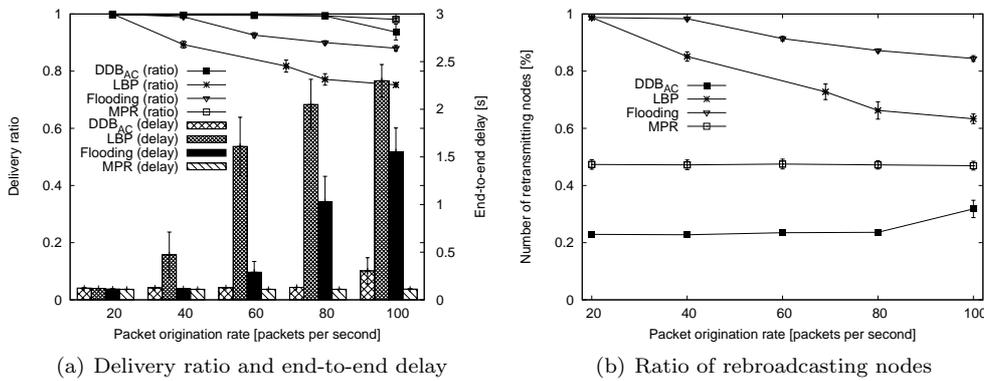
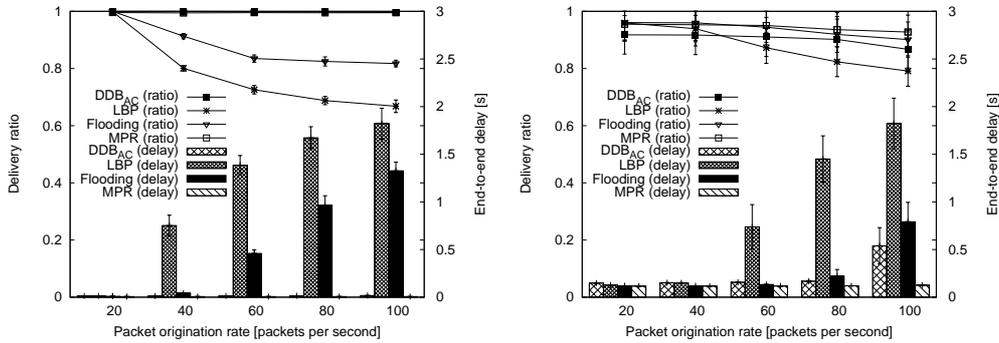


Figure 6.17: Congested network with 19 neighbors

of delay and delivery ratio and was similar to DDB_{AC} in the sparse network because the smaller number of neighbors also reduces the number of collisions of flooding. Finally, the delivery ratio of DDB_{AC} raises to almost 1 in denser networks with 44 neighbors over all levels of congestion. At the same time, the delay was reduced to similar values as for MPR.



(a) Delivery ratio and end-to-end delay for 44 neighbors (b) Delivery ratio and End-to-end delay for nine neighbors

Figure 6.18: Congestion in sparse and dense networks

Obviously, the CSMA-based IEEE 802.11 MAC protocol has a major influence and results may differ for other MAC protocols. The MAC protocol definitely has also an impact on the lightly loaded static network in the previous subsection, however should be very small and almost negligible.

6.4.6 Irregular Transmission Range

Basically all papers on broadcasting have conducted simulations using only isotropic propagation models which do not accurately reflect real radio propagation characteristics. Especially for position-based broadcasting protocols, the irregularity of transmission ranges may have a strong impact on the performance. We use the radio irregularity model (RIM) to evaluate the performance under non-isotropic transmission ranges, cf. Section 2.4.3. The two parameters that are used to control the degree of irregularity (DOI) of the transmission range and the variance of sending power (VSP) are set in our simulations to 0.1 and 0.5, as also suggested in [270]. As in Section 5.4, the transmission ranges are recalculated after a node has moved 50 m to reflect the changes in its environment. The rest of the simulation parameters are set as in Section 6.4.3 where algorithm efficiency was evaluated, i.e., 1000 static nodes over different simulation areas. In Fig. 6.19, we can see that the performance of DDB_{AC} suffers under irregular transmission ranges if we compare the results with Fig. 6.13. Specifically, the delivery ratio for sparse networks drops quite a bit. Unfortunately, it is not clear whether this is due to the protocol's inability to deliver the packet or due to the partition of the network. This partition cannot be detected reliably due to irregular transmission ranges. Therefore, it is not possible to determine accurately the maximal connected cluster as before. Most probably, both contribute to the decrease of the delivery ratio. On the other hand, in dense networks the number of rebroadcasting nodes can no longer be reduced efficiently with DDB_{AC} as it assumes isotropic transmission ranges to calculate the additionally covered area. Thus, with irregular transmission ranges the calculation of the DFD and the application of the rebroadcasting threshold RT is suboptimal. However, the performance is still very good when compared to the other protocols.

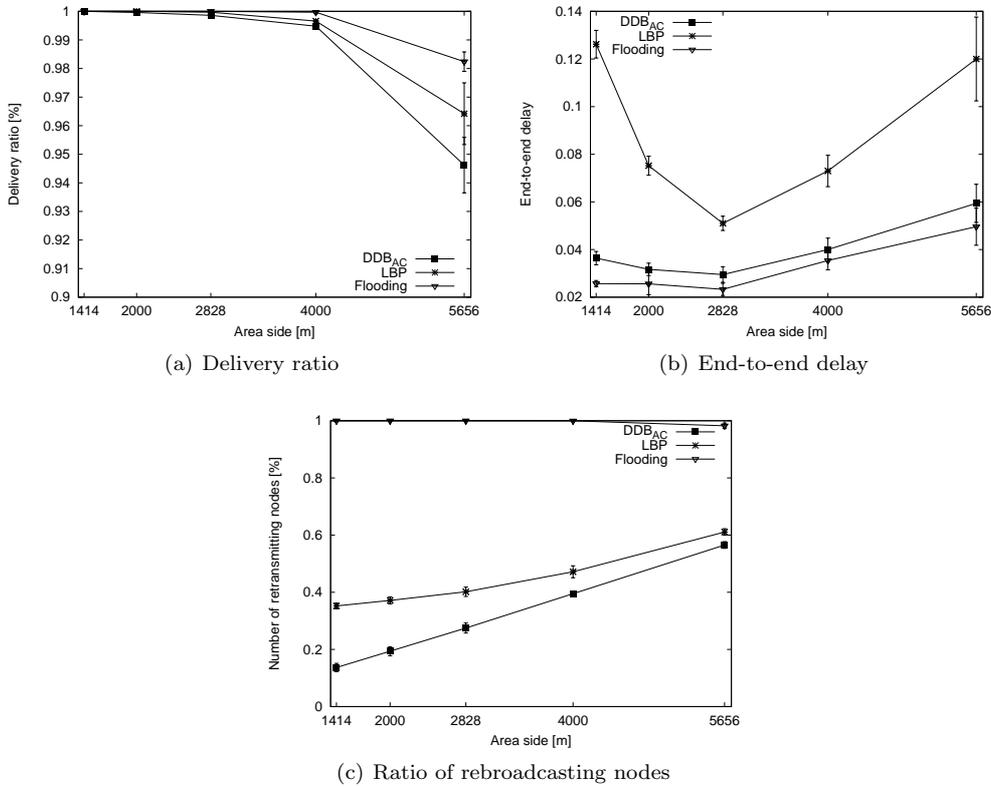


Figure 6.19: Impact of irregular transmission range

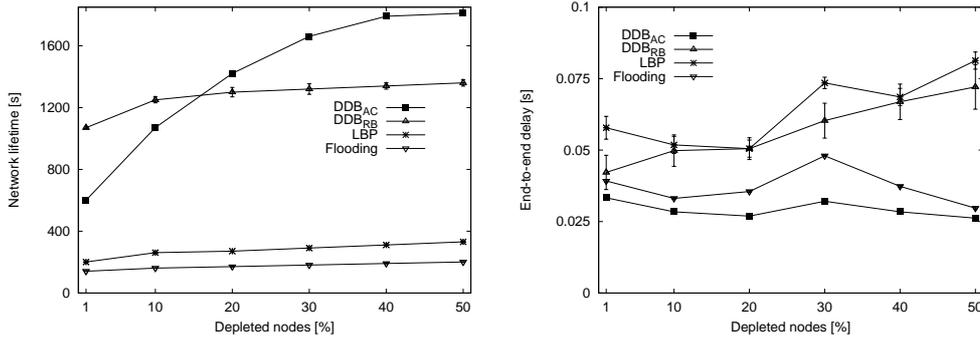
6.4.7 Network Lifetime

In many network scenarios, where batteries of nodes cannot be recharged or replaced, the network lifetime may be of higher importance than other performance metrics. We define the network lifetime as the time until a certain number of nodes fail due to battery depletion similar to [247] and [264]. Other definitions of network lifetime are used in [222, 41], which measure the mean expiration time and the time of the first node failure respectively.

The network lifetime strongly depends on the consumed energy during sending, receiving, and idle listening. If the ratio between these three modes is small, then obviously, which and how many nodes broadcast does not have any effect, and almost all nodes deplete at the same time. The interesting scenarios occur if the ratio is large enough that we may then expect nodes that transmit more frequently deplete sooner. For our simulations, the ratio of sending/receiving/idle listening was set to 10/1/0.01. These values are justified by recent technology advances, cp. e.g. [64], which also allow even higher ratios. The transmission delay of one packet is approximately $\frac{\simeq 2000 \text{ bit}}{2 \text{ Mbps}} \simeq 1 \text{ ms}$ such that most of the energy would still be consumed for idle listening as a node is idle for $\simeq 99.9\%$ of the time. However, if we broadcast packets at such a high rate that nodes are transmitting most of the time, i.e., more than 500 packet per second, we would encounter severe congestion and the results would be misleading. Thus, packets

are still broadcasted at only a rate of one packet per second, but we modified the energy model in a way that the energy consumption in idle mode is no longer time dependant, but is simply decreased by 0.01 between two broadcast packets. This would equal a situation where a node is idle 50% of the time and transmits and receives the other 50% of the time, but without congestion. We place 1000 nodes over an area of $2000 \times 2000 m$. Furthermore, we assume that nodes have equal battery level at the beginning, and all nodes consume the same amount of energy for transmission. As DDB_{RB} always favors nodes with more residual power, the power level of all nodes are kept more or less at the same level which in turn also increases the probability of simultaneous transmissions. Similar to simple flooding, we also jitter the transmissions at node for $2 ms$ to reduce collisions. Assuming that sending and receiving of a hello message consumes about the same energy than for a data packet, the lifetime of MPR will only be a very small fraction of the other stateless protocols. In our scenario with 1000 nodes and a hello message interval of $2 s$, 500 hello messages are broadcasted per second which will deplete the nodes' batteries very quickly. Thus, the MPR protocol is not depicted.

As shown in Fig. 6.20(a), the second scheme DDB_{RB} where rebroadcasting decisions are based solely on residual battery power exhibits by far the longest time until the first nodes fail and outperforms significantly LBP and DDB_{AC} . This is achieved even after the fact that the number of rebroadcasting nodes is about the same for DDB_{RB} as for LBP, because the rebroadcast decision is independent of the additional covered area and, thus, much higher than that of DDB_{AC} . However, the initially longer lifetime of DDB_{RB} comes at the cost of a longer delay as depicted in Fig. 6.20(b). For a higher percentage of depleted nodes, DDB_{AC} shows longer network lifetimes than DDB_{RB} due to the smaller number of rebroadcasting nodes leading to a smaller total amount of energy consumed for each packet. With DDB_{RB} , the remaining nodes deplete quickly after the first one fails because nodes with more residual power normally rebroadcast packets. Thus, all nodes always have similar residual energy levels.



(a) Network lifetime until a certain percentage of nodes fail (b) End-to-end delay after a certain percentage of nodes failed

Figure 6.20: Network lifetime

6.5 Conclusions

In this chapter, we presented the simple stateless broadcasting protocol DDB, which uses the dynamic forwarding delay (DFD) concept to optimize broadcasting in wireless multi-hop networks. With DFD, nodes are able to take locally optimal rebroadcasting decisions without any neighbor knowledge. Therefore, a node delays the rebroadcast of a previously received packet depending on its probability of reaching new neighbors, or any other metric such as residual battery power.

We compared the performance of DDB to another stateless broadcasting protocol LBP and a state-of-the-art stateful protocol MPR, which uses neighbor knowledge obtained through hello messages. LBP was not able to perform well over a wide range of network conditions, namely the performance degrades under heavy traffic load and high node density, as also observed in [258]. However, DDB did not suffer from these drawbacks of other stateless protocols such as LBP. Actually, quite the contrary is true. The performance of DDB even improved for those scenarios of high traffic load and high node density. MPR performed well in most scenarios, except in highly dynamic networks where the delivery ratio collapsed. The delay of MPR was the shortest in all simulated scenarios closely followed by DDB whose delay was approximately 10% longer, except in the case of highly congested networks. On the other hand, DDB outperformed MPR significantly with respect to the efficiency of the algorithm. DDB only required about half of the transmissions to deliver the packet reliably to all nodes compared to MPR. Furthermore, as DDB is stateless, its performance was completely unaffected in highly dynamic networks. However, the biggest advantage of DDB over MPR is its simplicity and economical use of network resource because no control messages are transmitted. These costs of proactively transmitting hello messages in MPR, which occur even if no data packets are broadcasted, makes their use in certain kind of networks with strict resource constraints even more inappropriate, e.g., sensor networks.

We can summarize that the dynamic forwarding delay concept allows DDB to operate completely localized without any neighbor knowledge. Therefore, it does not suffer from the drawbacks of stateful broadcast protocols, such as control traffic overhead and outdated neighbor information. However, it neither suffers from the drawbacks of other stateless broadcast algorithms such as inability to adapt to changing network conditions. These characteristics make DDB a valuable broadcast protocol for wireless multi-hop networks with either frequently changing topology and/or very strict power limitations such as vehicular and sensor networks.

In a last chapter, we address the second class of problems as mentioned in the introduction, cf. Chapter 1, caused by the fact that routing protocols do not keep track of the overall network distribution and solely route packets based on geographical positions. Particularly for large ad-hoc networks, where routing along a line-of-sight to the destination is not possible because of voids, it may be beneficial to build a global view of the network at the nodes so that unsuccessful paths are avoided. AMRA as proposed in Chapter 7 creates an abstract and static topology of the network and tries to find short paths on this topology on a large scale.

Chapter 7

Ant-Based Mobile Routing Architecture (AMRA)

7.1 Introduction

In this chapter, we introduce the Ant-based Mobile Routing Architecture AMRA that combines position-based routing, topology abstraction, and swarm intelligence. AMRA is tailored for large networks of possibly tens of thousands of nodes with irregular topologies where routing along the line-of-sight towards the destination is not possible due to obstacles like lakes and mountains.

Most position-based routing protocols assume that greedy routing along a roughly straight line between source and destination node is possible, i.e., along the shortest Euclidean path. If this is not the case and greedy routing fails, e.g., due to lake or mountains, most of them switch to a recovery mode, where face routing is applied, which has however several significant drawbacks, cf. Section 3.3.5. In such scenarios, AMRA tries to circumvent voids in the network topology in the first place and to route packets along paths with a high node density such that packets can be forwarded in greedy mode most of the time, which avoids the shortcomings of face routing in recovery mode. Topology abstraction is used to provide in a transparent manner an aggregated and static topology. On this topology, a routing protocol based on ant colony optimization determines good paths on a large scale. Topology abstraction is also the key to make ant-based routing scalable. Position-based routing is then applied to forward the packets physically along the selected paths. This may be performed by any position-based protocol such as BLR proposed in Chapter 5.

The remainder of this chapter is organized as follows. In a first Section 7.2, the protocol architecture AMRA is presented in detail. The protocols are evaluated and simulation results are given in Section 7.3. Finally, Section 7.4 concludes the chapter. Further information can also be found in the publications about AMRA [98], [102], [103].

7.2 Ant-based Mobile Routing Architecture (AMRA)

7.2.1 Introduction

In this section, we first give an overview of the whole AMRA architecture consisting of three independent protocols. Then, we describe each of the three protocols separately. Finally, we show how they interact to route packets efficiently to the destination in large-scale ad-hoc networks with irregular topologies.

We model large-scale ad hoc networks as a set of wireless nodes distributed over a two-dimensional area with an isotropic transmission range with radius r . As usual, packets are uniquely identified by their source node ID and a monotonically increasing sequence number. Nodes are aware of their absolute geographical position by any means and are able to determine destination nodes position. Whether, nodes are aware of their immediate one-hop neighbors as in other position-based routing protocols, which is achieved by the periodical broadcasting of beacons, depends on the protocol used to physically forward the packets based on nodes' positions. We assume that the overall node distribution in the network remains quite static and only varies slowly over time. This is necessary in order to find useful paths around voids. If the overall distribution changed too quickly, no such paths could be reliably found. For most realistic scenarios, this is reasonable assumption, as nodes are typically located in towns and on/along streets. We will also study the performance of AMRA if this is not the case and the node distribution changes abruptly.

7.2.2 Overview of Protocol Operation

AMRA is a two-layered framework with three independent protocols rather than a single routing protocol. The two protocols used on the upper layer are called Topology Abstraction Protocol (TAP) and Mobile Ant-Based Routing Protocol (MABR). StPF (Straight Packet Forwarding) is situated on the lower layer and acts as an interface for MABR to the physical network. An overview of the architecture with the interactions between the protocols is depicted in Fig. 7.1. TAP is the key to make routing scalable and provides in a transparent manner an aggregated and static topology with fixed “logical routers” and fixed “logical links” to MABR. A logical router represents a fixed geographical area. Thus, mobile nodes within a logical router are situated close together sharing similar routing information and have a similar view on the network topology on a large scale. A logical link represents a path along a straight line to another logical router over possibly multiple physical hops. The actual routing protocol MABR operates on top of this abstract topology and thus does not have to cope with changing topologies. MABR maintains probabilistic routing tables and is responsible for determining logical paths on this abstract topology. Data packets are routed based on these probabilistic routing tables between logical routers over logical links. They increase the probability of the followed path depending on the encountered network conditions. Furthermore, “artificial ants” packets are transmitted periodically to explore new paths. Unlike data packets, these packets are routed purely position-based directly towards their randomly chosen destination. Eventually, the best paths will emerge and MABR is able to circumvent areas with bad or no connectivity, i.e., data packets will always

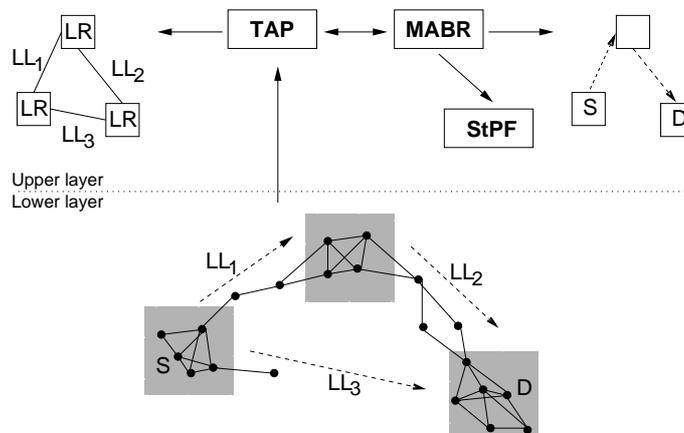


Figure 7.1: Architecture of AMRA

be routed over logical links with high connectivity such that greedy routing is possible. StPF is a position-based routing protocol and then responsible to physically forward packets over the logical link determined by MABR to the next logical router. We will use the term “logical” in general to indicate any object in the abstracted topology in the upper layer. For example, a logical paths consists of logical links, which in turn may require the transmission over multiple physical links. In Fig. 7.1, a source node S located in the logical router LR_1 routes a packet first towards the logical LR_2 over the logical link LL_1 instead of forwarding it directly towards logical router LR_3 , which will fail because there are no nodes located in between.

A number of new features have been added and in order to improve the performance over the original description in [98].

7.2.3 Topology Abstraction Protocol (TAP)

TAP is used to supply in a transparent manner an aggregated and static topology with fixed “logical routers” and fixed “logical links”. The objective for this abstraction of the actual network topology is to provide a rather static topology such that the routing protocols do not have to cope with frequent changing topologies. Furthermore, it was observed that ant-based algorithm take some time until good paths emerge, cf. Section 3.6.4. Logical routers are fixed geographical areas of equal size arranged in a grid to cover the whole area. Unlike in a cellular network where regular hexagons are typically used, we use squares for simplicity reasons. Depending on its current position, each node is part of one specific logical router. A node can easily detect based on its position, when it crosses the border of the current logical router and then automatically becomes a member of the new logical router. All nodes located within a logical router have the same logical view on the network. Nodes within a logical router cooperate in specific routing control tasks such as the emitting of ants. However, each node maintains its own routing table and does never share with, or transmit any routing information to its neighbors.

In order to scale to large networks, each logical router groups other logical

routers into zones $Z_{i,j}$ as shown in Fig. 7.2. The zone size increases exponentially with the distance to the center router and allows covering large areas with few zones. In the most inner ring of zones, i.e., $i = 1$, each zone corresponds just to one logical router. In the second ring, a zone covers already nine logical routers and so on. In general, a zone $Z_{i,j}$ in the i -th ring covers $3^{2(i-1)}$ logical routers. Thus, a node has a fisheye-like view on the network, i.e., the network resolution is higher in the vicinity and drops for more distant parts of the network. The reason is that in the view of a fixed node, close-by nodes that move some distance may be located in an entirely different direction, whereas the same movement of a node far away only marginally affects the direction. It is important to notice that the view of zones is relative. Each router resides in the center of its own zone model. That means that the view of a node changes when it moves to another logical router. A specific fixed geographical position may belong to a certain zone at a given moment and belong to another zone when the node has moved. To simplify addressing, each logical router is identified by the geographical coordinates of its center. This geographical identification simplifies routing with StPF, which uses position information for routing over logical links.

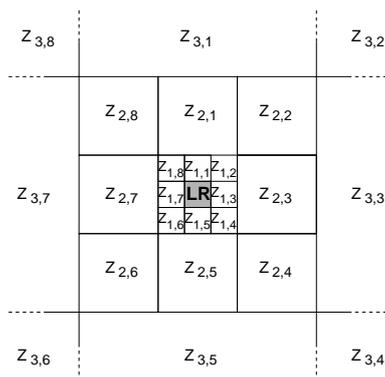


Figure 7.2: Zones relative to logical router in the center

Each logical router maintains a set of outgoing logical links to all its adjacent logical routers as depicted in Fig 7.3. A logical link represents a path along a roughly straight line to a distant logical router over possibly multiple physical hops. TAP is the key to make routing scalable and find good paths on a large scale in the network, i.e., by routing over logical links and not directly towards the destination.

7.2.4 Mobile Ant-Based Routing (MABR)

The actual routing protocol MABR operates in the upper layer on top of the abstract topology provided by TAP and thus does not have to cope with frequent changing topologies inevitable in mobile networks. It determines over which logical links, i.e., intermediate anchor positions, packets should be forwarded to circumvent voids in the network topology. These logical links may just lead in the opposite direction of the final destination, e.g., in cases the routing along a

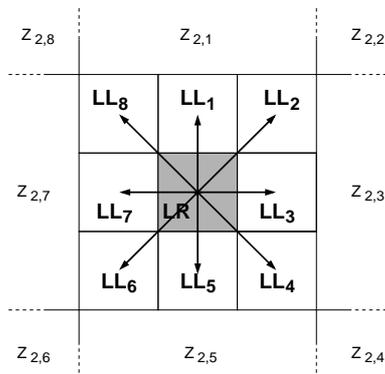


Figure 7.3: Logical router and its outgoing links

line of sight towards the destination is not possible. The important point is that these logical links should be chosen by MABR in such a way that the packet can be routed always greedily to the next logical router along this logical link.

Each node maintains a probabilistic routing table, which depends on its current view on the network, its past locations, and overheard packets. Consequently, routing tables are generally also slightly different for nodes within the same logical router. The zones and the logical links are organized in rows and columns respectively. More precisely, there is a column for the eight unidirectional outgoing logical links and a row for each zone. The entries $P_k^{i,j}$ indicate probabilities to choose the logical link LL_k for destination coordinates in zone $Z_{i,j}$. Instead of maintaining pheromone values for the logical links, which are then mapped to probabilities, the entries are organized as probabilities in the first place, i.e., the sum of all eight entries in a row is 1. If location information is provided by GPS nodes are normally also synchronized. Thus, we can determine the estimated average delay $\mu_d^{i,j}$ of packets received from zone $Z_{i,j}$. The average delay is stored to judge the goodness of the paths taken of incoming packets from the respective zone. If nodes are not synchronized, the estimated average $\mu_d^{i,j}$ can also be calculated based on hop counts of packets or we can sum the encountered delays in the intermediate nodes to estimate the end-to-end delay. At the beginning, all entries are initialized to 0.125 and μ_d to 0. An exemplary routing table is given in Table 7.1 with the entries $P_k^{i,j}$. The most probable logical link to route to a close by zone is typically the direct link, e.g., for zone $Z_{1,2}$ this is LL_2 . For more distant zone, it may however not be possible to route directly towards the destination, and thus a logical link in another direction may have the highest probability, e.g., LL_2 for zone $Z_{3,1}$. The size of the routing table is in the order of some hundred bytes even for very large networks due to the exponential growing size for farther zones. It requires nine columns to store the eight logical links and the average delay and a multiple of eight rows for the zones. Assuming a typical logical router size of $250 m \times 250 m$, $11 \cdot 8 = 88$ rows would be sufficient to cover the whole globe. A further advantage is that these routing tables never have to be transmitted and only kept in a node's memory.

The packet header of both packets, data and ants, contain fields for the

Table 7.1: Routing table

	LL ₁	LL ₂	...	LL ₈	μ_d
Z _{1,1}	0.9	0.1	...	0	257
Z _{1,2}	0.05	0.9	...	0	504
Z _{1,3}					
...					
Z _{2,1}	0.8	0.1	...	0.1	1045
Z _{2,2}	0.125	0.125	...	0.125	0
...					
Z _{3,1}	0.15	0.8	...	0	5348
...					

source node coordinates, the last visited logical router, and a time stamp, which are used by MABR to update the routing table. Furthermore, nodes operate in promiscuous mode such that the routing table is not only updated when a packet is received, but also is also updated for all overheard packets to expedite the dissemination of routing information. Each packets is however only used once to update the routing table at a given node, even if the same packet is overheard several times. The last visited logical router is stored in the packet header. This entry is updated each time at the first node in a new logical router. When a node receives a packet, it first determines from which zone $Z_{i,j}$ the packet originates with respect to its own current view on the network. That means that packets update the routing tables at the nodes in the opposite direction that they travel, i.e., towards their sources. A node determines the delay d of a packet, i.e., the time difference between the reception of the packet and the time the packet was sent by the source node. The measured delay d is then used to update the estimated average delay $\mu_d^{i,j}$ for the respective zone in the following way similar to the estimated RTT of TCP [197].

$$\mu_d^{i,j} \leftarrow \mu_d^{i,j} + n(d - \mu_d^{i,j})$$

$\mu_d^{i,j}$ is increased if the current sample delay d is longer than the previously estimated average delay $\mu_d^{i,j}$. On the other hand, it is decreased if the packet arrived quicker than previous packets. The factor n is used to smooth the effect of the last sample delay d . If set to 1, the $\mu_d^{i,j}$ would always equal the last measured delay and previous samples would not have any impact. As the delay d of packets may vary strongly within a short time interval, the factor n should be chosen rather small to avoid a too high fluctuation of the average delay $\mu_d^{i,j}$. Thus, this updating formula yields an exponential weighted moving average where the weight of a past sample decreases exponentially fast with n .

Based on the goodness of the trip taken of the current packet with respect to the estimated average, a factor r is calculated which affects the amount of pheromones laid down on the followed logical link, i.e., the ratio of $\mu_d^{i,j}$ to d

determines by how much the probability $P_k^{i,j}$ is increased.

$$r = \begin{cases} \frac{\mu_d^{i,j}}{3 \cdot d} & : \frac{\mu_d^{i,j}}{3 \cdot d} < r_C \\ r_C & : \text{otherwise} \end{cases} \quad (7.1)$$

The worse the delay d of the current packet is compared to the average $\mu_d^{i,j}$, the smaller the factor r and the less the probability will be increased. $0 < r_C \leq 1$ is the ceiling of r and limits the amount of pheromones one packet can deposit on a logical link. The factor 3 in the dominator limits the impact of packets that have a delay higher than, or close to the estimated average, e.g., it avoids that a large r is calculated for only marginally shorter trip times than the estimated average. Thus, only really short trip times compared to the estimated average yield a large r and increase the probability significantly. For example, only trips three times shorter than the average yield the maximal value of r , i.e., r_C .

The last logical link LL_k over which the packet was forwarded is now updated, i.e., the logical link to the previously visited logical router of the packet, with respect to zone $Z_{i,j}$. The important idea here is to increase in this way only the probability for logical links along which packets could be routed in greedy mode. $P_k^{i,j}$ is recalculated as follows.

$$P_k^{i,j} \leftarrow P_k^{i,j} + (1 - P_k^{i,j}) \cdot r^2 \quad (7.2)$$

In this way, small probabilities are increased quicker than already large probabilities for a given r . The probability of the other seven logical links $l \neq k$ for that zone $Z_{i,j}$ are decreased by

$$P_l^{i,j} \leftarrow P_l^{i,j} - P_l^{i,j} \cdot r^2$$

such that the sum of all logical links in a row to a certain zone remains 1. Metaphorically speaking, the pheromone value for a traveled path is increased depending on the quality of the path.

Unlike other ant-based routing algorithms, pheromones do not necessarily need to decay with time as the overall distribution of the nodes remains rather static and only changes slowly with time, even if the node's neighborhood changes rapidly. The slow change of the overall distribution allows AMRA to adapt to the new situation even if pheromones do not decay. The reason is that the pheromones also decay indirectly for every received packet, which increases the probability for a link and simultaneously decreases the probabilities for the other links. This however presumes that there is always traffic in the network while the overall distribution changes. If this is not the case, the routing of packets may be very suboptimal. The mechanisms of AMRA described later in Sec. 7.2.8 ensure that the packet is delivered eventually nevertheless to the destination. On the other hand, if pheromones decayed with time, it would be difficult to ensure a high enough pheromone concentration to find short paths. Consequently, there is a tradeoff whether pheromones should decay with time or not. To cope with this circumstance, the decay of pheromones could have been made adaptive to the distance a node moves and the packet it receives.

The pheromones decay in AMRA if a node moves to a new logical router and a node's view on the network changes. In order to reflect the change in its view on the network, a node adapts the pheromone values in its routing table

as follows.

$$P_k^{i,j} \leftarrow P_k^{i,j} + \left(0.125 - P_k^{i,j}\right) \cdot \frac{1}{3^i} \quad (7.3)$$

All entries asymptotically approach a probability of 0.125. A uniform distribution of 0.125 for all links indicates that links do no longer have pheromone trails and no link is favored over another for a given destination. The factor $\frac{1}{3^i}$ is larger for smaller i , thus, the probabilities $P_k^{i,j}$ for closer zones $Z_{i,j}$ approach 0.125 faster. The dominator is the distance to the zones which increases exponentially with 3^i , i.e., the pheromone values decay more rapidly for closer zones. The reason is as already discussed before that if a node moves a fixed distance, a close-by destination may turn out in a completely different direction, which requires to decrease the pheromones more quickly than for a distant destination. MABR updates the routing tables identically for ants and data packets. Ant and data packet differ only in their forwarding policy as described below.

7.2.5 Straight Packet Forwarding (StPF)

Finally, the physical forwarding process along the logical links selected by MABR is accomplished by StPF, which can be basically any position-based routing protocol. Because many such position-based routing protocols have already been proposed and analyzed in the literature, we did not design a new protocol for StPF. We instead use the perhaps best known position-based protocol GFG/GPSR as StPF, cf. Section 3.3.4. Basically, any other position-based routing protocol may be applied as well, such as GOAFR [149], GRA [124], BLR [101]. If BLR is used, nodes would not be required to transmit beacons as MABR and TAP do not require neighbor knowledge.

7.2.6 Routing of Data Packets and Ants

Data packets are routed based on these probabilistic routing tables between logical routers over logical links. They are used to reinforce and maintain existing paths and do not discover new paths. The first node within a logical router that receives the packet determines the next logical hop. Therefore, it determines to which zone a packet should be routed from the destination coordinates as given in the packet header. The node then selects the logical link with the highest probability to this zone to forward further the packet. As only logical links over which packets arrived at the current logical router receive pheromones, routing between logical routers should always be possible in greedy mode solely. The packet is then routed with StPF towards the center of the logical router at the end of the selected logical link. Furthermore, if desired load balancing can be achieved easily by selecting a logical link proportionally among all possible logical links. Thus, data packets are routed logical-hop by logical-hop over the logical links, i.e., from one logical router to one of its adjacent logical routers and so on. If none of the eight logical links for the destination zone has a probability above a threshold *Prob_Thres*, the data packet is routed purely geographically directly towards the final destination. This is a reasonable heuristic decision if no useful routing information is available.

Ants are used to explore new emerging paths and find shorter paths. Unlike data packets, ants are solely routed by StPF, i.e., they always head directly towards the destination and are only diverted if voids in the routing topology

cause them to be routed in recovery mode. The ants update the routing tables at all intermediate nodes in the opposite direction towards their source nodes. In order to control the ant generation rate, nodes cooperate in emitting ants and detecting new paths. A node only emits an ant packet, when it did not detect that an ant was created from any node within its current logical router for $t_{Emit.Ants}$, i.e., each logical routers generates an ant every $t_{Emit.Ants}$. Thus, nodes within a logical router cooperate to minimize the number of ants in the network, which makes AMRA more adaptive to the encountered network conditions. E.g. in cities with a high node density and a large number of nodes in a logical router area, a node only rarely has to transmit an ant packet. The destination of the ant is chosen uniformly randomly over the whole simulation area. The chosen destination logical router may not be reachable because there are no nodes in this area. To minimize the risk that packets spin around the indicated position and distort the entries in the routing tables, an ant is solely routed in greedy mode, when it entered the zone of its destination logical router. The zone is determined by the source node and indicated in the packet header. A node in the destination zone that cannot relay the packet further in greedy mode simply drops the packet. Ants are either routed according to the left or right-hand rule if face routing is applied in the recovery mode, i.e., if they reached a dead end and greedy routing failed. The reason for routing ants with both rules is that ants otherwise could miss shortest paths as shown in Fig. 7.4. If ants always used solely the right-hand rule, the shorter path from S to D over A would not be discovered. Consequently, D only has a routing table entry for this longer path and always forwards packets to S along the path over B

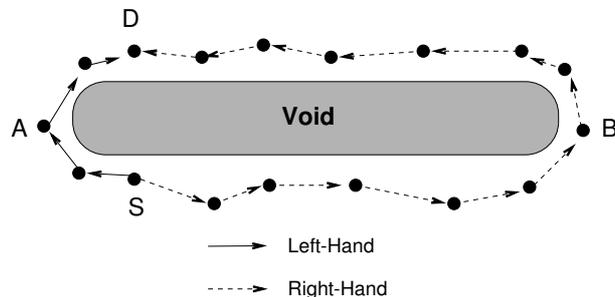


Figure 7.4: Ants routed with right- and left- hand rule

7.2.7 Example of Routing with AMRA

Until now, we only discussed the protocols separately and only roughly described their interaction. In this section, we study how they cooperate by means of an example depicted in Fig. 7.5. The zones of the network are sketched in the view of node S . Nodes in $Z_{3,3}$ have previously transmitted ants, or also data packets, which either were destined for the logical router in which S is located or just pass through S for a more distant destination. Exemplarily, the path taken of ants emitted by D are shown. Ants routed by the right-hand and left-hand rule enter the logical router of S over the logical links LL_1 and LL_6 , respectively. S determines that these ants origin from a source in zone $Z_{3,3}$ according to its

current view. The delay of ants arriving over LL_1 is much shorter than of ants LL_6 due to the shorter path. Consequently, the probability of LL_1 is higher than of LL_6 for zone $Z_{3,3}$ in the routing table at node S . Furthermore, as no packets enter over one of the other six logical links, their probability is close to zero. Not only node S but all intermediate nodes which forward or overhear the ants update their routing table according to their view on the network.

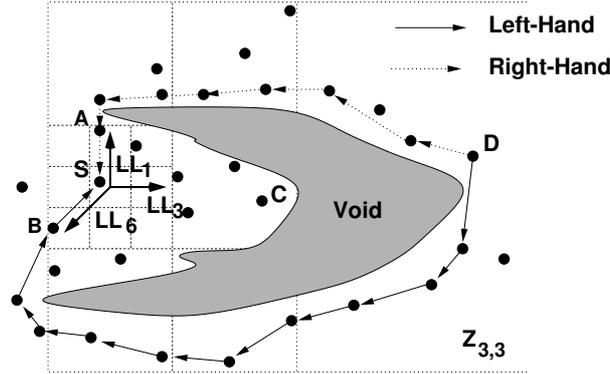


Figure 7.5: Ants laying trails over LL_1 and LL_2 to $Z_{3,3}$

When S now has to send packets to any node located in zone $Z_{3,3}$, it routes the packets over LL_1 , i.e., to node A in Fig. 7.5. Thereby, it is irrelevant whether S is the source of the packet or just any intermediate node which forwards the packet. A in turn forwards the packets then over one of its logical links with the strongest pheromones and the packets travel the indicated path to D . S does not forward packets for D towards C over LL_3 . The probability of LL_3 is close to zero because no packets originating from zone $Z_{3,3}$ have been received from this direction. In this way, AMRA avoids to forward packets into directions which have no, or only a very small, probability for the packets to arrive directly. AMRA routes packets in directions from which it received traffic, which also indicates that there are nodes along the respective links. Thus, packets can almost always be routed in greedy mode because the routing over logical links should not fail in greedy mode.

On the other hand, if a purely position-based routing protocol is used such as GFG/GPSR, packets for D are routed first towards C where greedy routing fails as no neighbor is located closer to D as already depicted in Fig. 3.8. The hop count is not only increased by the longer path but also through a property of the recovery mode, where face routing is applied together with a local extraction of a planar subgraph, more precisely by the Gabriel Graph, which yields shorter links than potentially available. The reason is that links only belong to the Gabriel Graph if no other node is located within the defining circle, cf. Section 2.4.1, which obviously retains shorter links with a higher probability than longer links. As GFG/GPSR is memoryless, subsequent packets are routed exactly over the same longer suboptimal paths. Even if S did not have any useful entry in its routing table, AMRA would simply forward the packet without any delay and the next node which has an entry for the destination zone forwards the packet along the best logical links. The probability that nodes do not have entries

is very small because very distant zones are proportional large to ensure that packets from this zone are overheard.

7.2.8 Looping Packets

Packets are routed based on the routing tables encountered at the nodes. The routing table gives an estimation for the direction in which the packet should be routed advantageously, but they are not guaranteed to be consistent among different nodes, which is definitely a severe drawback of AMRA. Therefore, it is not surprising that in scenarios with high mobility and where a lot of nodes have not very accurate routing tables because the time is often too short for the best paths to emerge, we encountered that packets may be loop temporarily in the network, sometimes forward and backward between adjacent logical routers or sometimes even over several logical routers.

To mitigate the effect that packets are routed back and forth, a packet must never be sent back to the last visited logical router or one of its two adjacent logical routers. The packet is routed over the logical link with the highest probability among the remaining five possible logical links.

Furthermore, logical links may form a loop in which packet can get trapped. Even though this was observed rarely, AMRA implements a mechanism to cope with such situations. In order to prevent such loops, a packet is sent purely position-based if it does not arrive within five times the expected average for this zone.

7.3 Simulations

We implemented and simulated AMRA in the Qualnet network simulator [211] and compare its performance with two other position-based protocols, namely GFG/GPSR and Terminode routing. We used GFG/GPSR as a representative for position-based protocols that always try to route greedily towards the destination. We also used Terminode routing for comparison because, to the best of our knowledge, it is the only protocol which tries to avoid routing in recovery mode with face routing in the first place by introducing anchor points. (The code of the Terminode routing protocol is a courtesy of the authors of the Terminodes protocol [15].) The problem of suboptimal forwarding of face routing was also addressed previously in GOAFR [149]. The objective of GOAFR is however different from that of AMRA. In GOAFR packets are still routed directly towards the destination and if greedy routing fails, it reduces the number of hops that packets are forwarded according to the face routing algorithm in recovery mode. This is unlike AMRA that tries to avoid forwarding packets with face routing in in recovery mode at all by routing over intermediate positions.

We first describe the general simulation scenario. In a next step, we briefly present results from simulations in small networks with uniform node distribution. Even though we are not particularly interested in simple network topologies, AMRA should perform nevertheless comparable to other position-based protocols. We conducted several simulations with large and irregular network over a wide range of conditions and present the obtained results afterwards. Many simulation parameters are the same for the small and the large network. These parameter are set to the values as given in Table 7.2, if not noted other-

wise. The parameters of AMRA are set as follows. The logical router size was

Table 7.2: Parameters

Parameter	Value
Simulation Time	900 <i>s</i>
Traffic Start	120 <i>s</i>
Traffic End	880 <i>s</i>
Traffic Type	Constant Bit Rate
Traffic Rate	4 Packets/ <i>s</i>
Packet Size	64 Byte
MAC Protocol	802.11
Bandwidth	2 Mbps
Transmission Range	250 <i>m</i>
Confidence Interval	95%

set to the transmission range, i.e., 250 *m* x 250 *m*. In this way, nodes within the same logical router overhear the same packets and thus have similar routing tables. Furthermore, we set the probability threshold *Prob_Thres* for a link to 0.2, i.e., links with a lower probability are not considered as a valid alternative. The time $t_{EmitAnts}$ after which a node creates an ant, when it did not detect that an ant was emitted from any node within its current logical, was set to 5 *s*. The parameters for the pheromone laying function are set to $r_C = 0.8$ *s* and $n = 10$, i.e., approximately the last ten samples contribute to the average $\mu_d^{i,j}$.

7.3.1 Small Network with Uniform Node Distribution

For small networks, we simulated 50 nodes over an area of 1500 *m* x 300 *m*. The nodes move according to the random waypoint mobility model. They move to a randomly chosen destination with a speed uniformly chosen between a minimum and maximum speed set to 1 *m/s* and 20 *m/s* respectively. After they arrive at the destination position, they pause for a certain time and then select a new speed and destination position. The pause time is used to control the mobility of the nodes and, thus, the rate of topology changes. We conducted simulations with 1 and 20 CBR traffic flows between randomly chosen source and destination nodes.

The delivery ratio is 90% or more even for highest mobility, i.e., no pause time, and highest traffic load for all protocols as depicted in Fig. 7.6. As expected, GFG/GPSR performs slightly better than AMRA because packets can usually be routed greedy, i.e., along a roughly straight line between source and destination. In high mobility scenarios, pheromone concentration varies in the probabilistic routing table and AMRA may temporarily take “wrong” forwarding decisions, which causes the slight encountered performance decrease and the large confidence interval. For lower mobility, the pheromone paths more and more stabilize such that the delivery ratio increases to almost 100%. Only Terminode routing lags a little behind the other protocols, especially for high traffic load and mobility.

The end-to-end delay is very similar for GFG/GPSR and AMRA in the low

traffic scenario. With 20 CBR sources, the delay increases almost an order of magnitude for both protocols. Thereby, the delay of GFG/GPSR remains almost unaffected by the mobility. On the other hand, AMRA suffers a little under high mobility and results fluctuate strongly as can be seen from the large confidence interval. Again this is due to inconsistent pheromone trails in the routing tables, which disappear for lower mobility and the delay drops to 100ms . Terminode routing performs worse and yields much higher delays already for only one source. The delay for 20 sources is around one second with confidence intervals of half a second and, thus, the graph is not depicted in the figure. The reason for this behavior is found in the large control traffic overhead of Terminode routing. Terminode local routing requires nodes to add all neighbors in the periodical broadcasted hello messages. Thus, in high density network these packets easily grow to the size of several hundred bytes, which congests the network causing the observed higher delays.

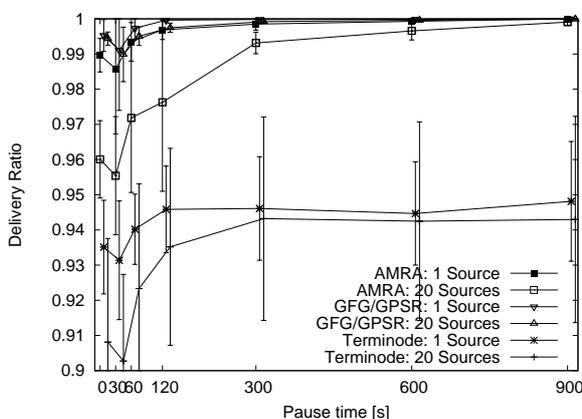


Figure 7.6: Delivery ratio

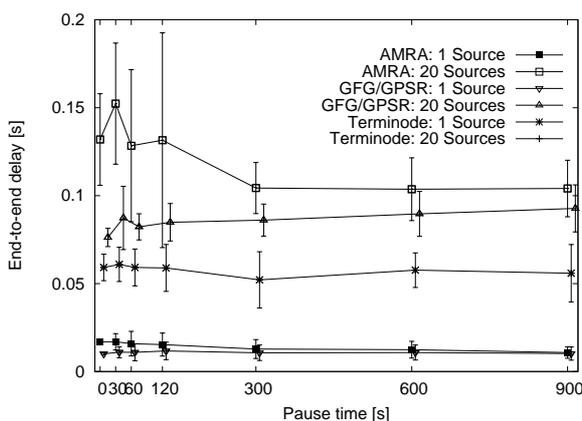


Figure 7.7: End-to-end delay

7.3.2 Large Network with Irregular Topology

To simulate realistically large networks with irregular topologies, we use the restricted random waypoint mobility model, cf. Section 2.4.2. The parameter values used for this restricted mobility model are given in Table 7.3.

Table 7.3: Parameters for large networks

Parameter	Value
Number of Nodes	500 (Commuters: 300 Ordinary nodes:200)
Prob to change city	Commuters: 0.9 Ordinary nodes: 0.1
Size of Area	3000 <i>m</i> x 2500 <i>m</i>
Number of Cities	4
Size of Cities	1000 <i>m</i> x 1000
Number of Highways	3
Min Speed in City	1 <i>m/s</i>
Max Speed in City	15 <i>m/s</i>
Pause Time	Commuters: 1 <i>s</i> Ordinary nodes: 30 <i>s</i>
Min Speed on Highway	10 <i>m/s</i>
Max Speed on Highway	30 <i>m/s</i>

Simulations with Mobility

The results of the simulations with mobility were found to be very disappointing. Even when varying, the number of ants, the size of the logical router, etc. AMRA was only able to deliver around 10% of the packets. The reason for the poor performance was found when we simulated GFG/GPSR in the same scenarios. Surprisingly, also GFG/GPSR achieved only a delivery ratio of around 15%. The same experiments without mobility yielded a delivery ratio of almost 100% as expected, because GFG/GPSR guarantees delivery in static networks. Further analysis revealed that the reason is the high mobility on the highway that causes packets to loop frequently as also already stated in Section 5.4.4. An exemplary path of a packet routed by GFG/GPSR in one of the simulations is shown in Fig. 7.8. Nodes keep track of their neighbor positions obtained by hello messages. If mobility is high, these stored positions do not correspond to the actual position and wrong forwarding decisions are taken. Some packets may recover from the loop, others do not. Consequently, the queues of the nodes get filled up and start to drop packets. Furthermore, packets are also dropped because the TTL-field expires if they are caught in a loop for a while.

Therefore, it is not surprising that AMRA performs poorly as it also uses GFG/GPSR to physically forward packets. Terminode routing suffers from the same problem as packets are also routed by GFG/GPSR between the anchor positions. The reason why in [15] Terminode was able to deliver much more packets in a similar scenario is because there are stationary nodes distributed all over the area and second the minimum speed for nodes on the highway was set to 1 *m/s*. Thus, there were always slow moving or stationary nodes on the highway which build a backbone for routing.

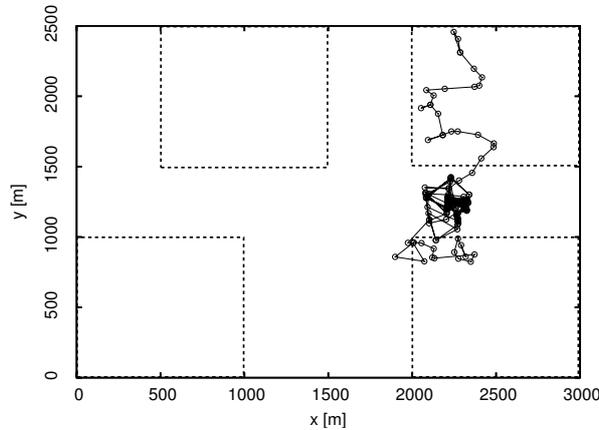


Figure 7.8: Looping packet on the highway

Due to these observations, we conclude that position-based routing protocols which rely on neighbor information for forwarding are not able to operate in highly dynamic networks. The problem may also exist in networks with frequently changing topology due to sleep cycles of nodes such as in sensor networks. The outdated neighbor information leads to many wrong routing decisions. This is a typical behavior of GFG/GPSR and all stateful position-based routing protocol as already observed in Section 5.4.4. Thus, if AMRA used BLR as StPF on the lower layer to forward packets over physical hops, we may hope to not encounter this behavior.

Simulations without Mobility

As it was not possible to obtain meaningful simulation results with mobility, we evaluated the performance of the protocols in static irregular networks. Nodes only move at the beginning in order to obtain a typical node distribution. After a certain time, nodes are stopped and remain static during the rest of the simulation. It is uninteresting to have source and destination nodes close to each other during the data transmission, which may happen if they are randomly chosen. Therefore, we choose the source and destination nodes from city 1 and 4 only. The results of these simulations are given in Fig. 7.9. The hop counts for AMRA and Terminode routing are in the order of 40 whereas GFG/GPSR required almost 80 hops. Unlike GFG/GPSR, AMRA and Terminodes routing forward packets not directly towards the other city but along the intermediate anchor positions, which indicate the path over cities 2 and 3 instead. Along this path, packets can be routed most of the time in greedy mode. An example for the path over which packets are routed with GFG/GPSR and AMRA between city 1 and 4 is depicted in Fig. 7.10.

As they are able to deliver data packets with significant less hops, the delay is accordingly also shorter. The delay of AMRA is approximately 0.18 s and is about half of GFG/GPSR 0.35 s . The reason for the longer delay of 0.28 s for Terminode is the large size of the headers as already mentioned before. The Terminode local routing requires adding all known positions of neighbors in

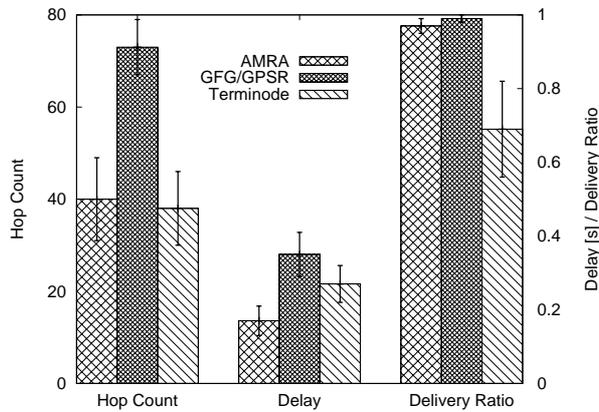


Figure 7.9: Results in a large static network

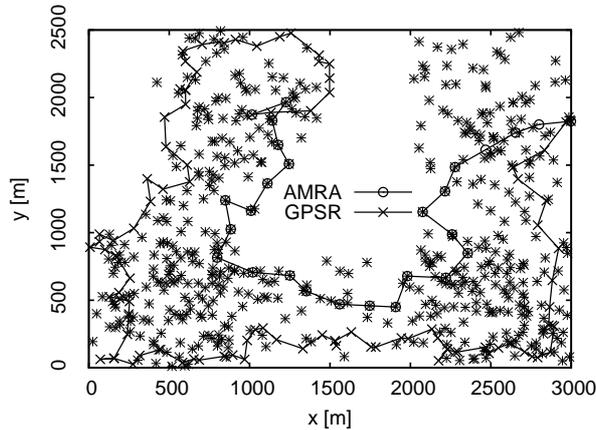


Figure 7.10: Path of AMRA and GFG/GPSR in irregular topology

the periodically transmitted hello messages by each node. The hello messages can easily grow to a size of several hundred bytes in dense networks leading to network congestion.

Terminode routing performs poorly and only delivers 70% of the packets. This is again due to increasing control traffic, which congests links. GFG/GPSR and AMRA on the other hand are able to deliver almost 100% of the packets. Only few packets drops were observed due to collisions with hello messages and other data packets. The reason that AMRA has a slightly lower delivery ratio is because of the probabilistic pheromone trails. These may cause packets to loop sometimes over several logical routers and packets are dropped because of an expired TTL-field.

Radical Topology Changes

Until now, we assumed that the overall node distribution in the network is approximately constant even though individual nodes may be highly mobile. Considering a realistic scenario, it may happen that the distribution also changes, e.g., there are only few cars on a highway during the night, but there is a very high node density in the morning during the rush hour. Therefore, we also want to assess the ability of the investigated protocols to adapt to radical topology changes. We evaluated two scenarios where a highway was inserted and removed respectively between city 1 and 4 after 240 seconds of simulation time. In a static network, this was accomplished by simply placing 30 nodes along the highway between city 1 and 4 after 240 seconds. The corresponding results are depicted in Fig. 7.11(a) and 7.11(b) where each graph refers to a specific simulation run. GFG/GPSR has a high hop count before the highway insertion because packets are routed for many hops with face routing in recovery mode over cities 2 and 3. After the insertion of the highway, GFG/GPSR almost immediately forwards packets over the newly available path and the hop count drops from over 100 to below 20 as depicted in Fig. 7.11(a).

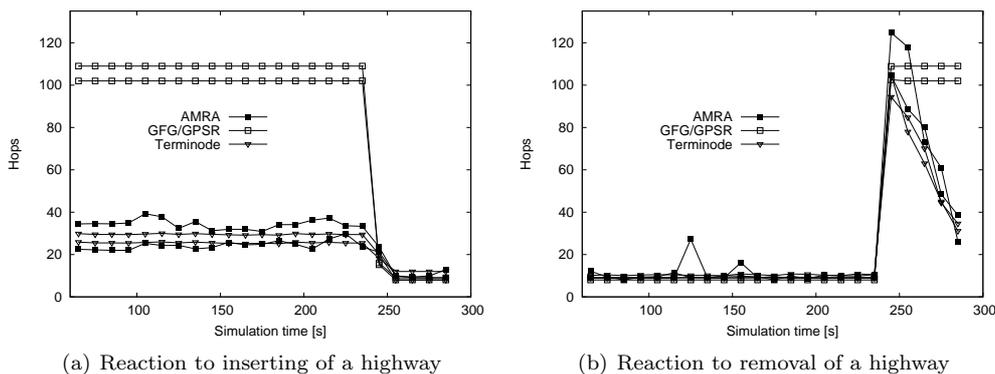


Figure 7.11: Reaction to radical topology changes

The reason is that GFG/GPSR always tries to route packets directly towards the destination, which is possible after the insertion of the highway, and packets can be delivered purely in greedy mode. AMRA and Terminode yield much shorter hop counts than GFG/GPSR when the new highway is not yet inserted. After the insertion of the highway, both protocols are also able to adapt quickly to the new topology. First, data packets still use the longer path over cities 2 and 3 with AMRA as there are no routing table entries at the nodes for logical links in the direction over the new highway. Ants transmitted from nodes in city 1 and 4 to the respective other city, or more precisely to any logical router in the direction of the other city, detect the new available path between the two cities quickly. Ants are routed purely position-based and thus are not diverted from existing pheromone trails. These ants have traveled a path much shorter than the average indicated in the routing table of nodes and, thus, strongly increases the pheromone value for the respective logical links such that already few ants are sufficient to redirect the data packets. The other longer existing pheromone trails will vanish quickly. After a while, the performance of AMRA stabilizes

at approximately the same values as GFG/GPSR. The slight fluctuation of the hop count of AMRA at the beginning is caused by ants packets which change the probability of the logical links. Thus, packets are not necessarily routed over the same logical links all the times.

On the other hand, if the highway is removed after a while, the hop count sharply increases for GFG/GPSR in Fig. 7.11(b) and remains high as the protocol has no way to adapt to the new situation and learn shorter paths. The hop count of AMRA and Terminode routing also increases for a short time to approximately the same values as GFG/GPSR. However, within few ten seconds of simulation time, the values decrease again when AMRA learned the new topology and pheromone trails were established between cities 1 and 4 over cities 2 and 3. Similarly, Terminode routing needs some time to adapt to the new network topology and find shorter paths.

7.3.3 Results Obtained from a Java-Simulator

Unfortunately, we were not able to conduct simulations, not even static, with thousands of node and a highly complex network topology with the Qualnet network simulator because the resource consumption was too high. Thus, we also implemented and simulated AMRA in a simple Java simulator that allows for such simulations also under mobility. (The code that served as the basis for Java simulator is a courtesy of Aaron Zollinger and Regina O'Dell of the Distributed Computing Group at ETH Zürich.) The Java-simulator implements functionality such as CBR traffic and the restricted random waypoint mobility. However, it does not account for any physical propagation medium properties or MAC layer functionality. Therefore, packets cannot be dropped due to collisions or congestion and packets do not experience delay. In order to assess the performance of AMRA, the hop count was used. We think that a hop count metric is an appropriate representative for the delay as CSMA based MAC protocols such as IEEE 802.11 have high costs for acquiring the medium. The faster Java-simulator also allowed us to simulate the 500 nodes network with irregular topology from Section 7.3.2 with mobility and multiple traffic sources. The Terminode routing protocol is a highly complex protocol, especially the FAPD part of it, cf. Section 3.3.4. Therefore, we did not implement Terminode routing but used additionally to GFG/GPSR a shortest path algorithm for comparison and to assess the goodness of paths chosen by AMRA.

Large Network with Mobility and Multiple Traffic Sources

The configuration was like for the simulations in the Qualnet simulator, cf. Table 7.2 and Table 7.3, except for the MAC layer protocol that was not implemented and the bandwidth which is practically infinite. AMRA was simulated with unidirectional and bidirectional traffic between the source and the destination. The reason is that AMRA can use traffic flowing in the opposite direction to update the routing tables towards the destination. On the other hand, GFG/GPSR and the shortest path algorithm are not affected by bidirectional traffic and thus they were only simulated with unidirectional traffic. If we have bidirectional traffic between two communication peers, we consider that as having two individual traffic sources, e.g., 5 bidirectional traffic flows indicate 10 traffic sources.

Varying Number of Ants We first conducted simulations where the number of transmitted ants was varied and we had a fixed number of traffic sources set to 10. In Fig. 7.12, we can see that GFG/GPSR has on average an about 2.5 times higher hop count than the shortest possible path. Considering the fact that often the traffic flow is between nodes in the same city or one of the adjacent cities, we may conclude that the hop count for traffic flows between non-adjacent cities is much more than 2.5 times the shortest path. If nodes are in adjacent cities, routing along a straight line between them is possible and the performance of GFG/GPSR is almost identical to the shortest path.

AMRA with only unidirectional traffic and no ants performs even worse than GFG/GPSR. However, as soon as few ants are transmitted the hop count drops sharply. With only 50 ants per second in the whole network, i.e., with 500 nodes, each node transmits an ant every 10 seconds, the hop count is about 15 compared to 10 of the shortest path and 25 of GFG/GPSR. The further increase of ants does not further reduce the hop count however. For a certain number of ants, the performance of the unidirectional and bidirectional traffic scenario is even equal within the confidence interval. For bidirectional traffic, the hop count is almost constant over all rates of ant generation. The data packets in the opposite direction are sufficient to establish high probability entries in the routing tables.

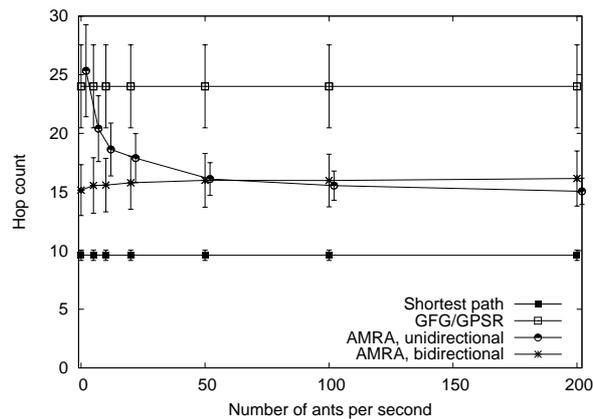


Figure 7.12: Irregular network with varying number of ants

Varying Number of Traffic Sources In a next step, we simulated a scenario where no ants are transmitted at all and only the number of traffic flows was varied Fig. 7.13. Again, the performance of GFG/GPSR shows an about 2.5 times higher hop count than the shortest path. Unlike before, the graphs for GFG/GPSR and the shortest path are no longer exactly constant, but only statistically constant within the confidence intervals. The reason is that, unlike the number of ants, a varying number of sources may yield slightly different results among the different simulation runs. AMRA with bidirectional traffic remains almost unaffected by the number of traffic flows, i.e., traffic flowing in different directions does not distort the entries in the routing tables for traffic flows to other destinations. As before where we had 10 traffic flows, AMRA

with unidirectional traffic suffers if we have no ants and only few traffic flows. The chance that a node has overheard a lot of traffic to a given destination zone is low and, thus, the risk when it has to forward a packet to that zone is high that it forwards the packet in a wrong direction. However, as more traffic flows there are in the network, the performance of AMRA with unidirectional traffic approaches the performance of AMRA with bidirectional traffic. If we have sufficient traffic, the entries in the routing tables are updated accurately by the data packet themselves. The reason is that if there are no useful entries in the routing tables, data packets are routed purely position-based and thus adopt the role of ants.

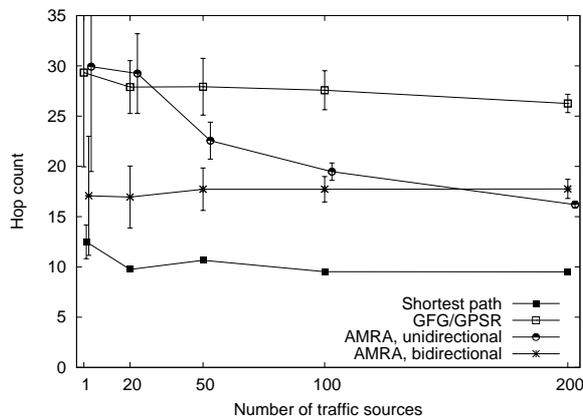


Figure 7.13: Irregular network with varying data traffic and no explorer packet

Very Large Network

We simulated 10000 mobile nodes in an area of $10000\text{ m} \times 12000\text{ m}$ with 19 cities and 19 interconnecting highways. The nodes move again according to the restricted random waypoint mobility model with the same parameters as before except that the size of the cities varies between $1000\text{ m} \times 1000\text{ m}$ and $2000\text{ m} \times 2000\text{ m}$. A snapshot of this network is depicted in Fig. 7.14.

There are 200 traffic sources in this simulation. We wanted to study the effect of a varying number of ants and also if traffic is sent unidirectional and bidirectional. It is common for a lot of applications to have bidirectional traffic flows or sometimes simply because TCP is used as Transport protocol. Thus, for the bidirectional simulations, we had 100 pairs of nodes and nodes transmit data to their respective peer. Once AMRA was simulated with no additional ants transmitted to explorer shortest path and once with 500 ants transmitted per second in the whole network. In Fig. 7.15, we see the average hop count of AMRA, GFG/GPSR, and a shortest path algorithm. Obviously, the hop count of GFG/GPSR and the shortest path algorithm does not change for uni- and bidirectional traffic as the traffic in the opposite direction has no influence. On the other hand, AMRA benefits from bidirectional traffic and the hop count drops about 10% compared to pure unicast transmissions, independent whether ants are transmitted or not. The reason is that packets update the routing

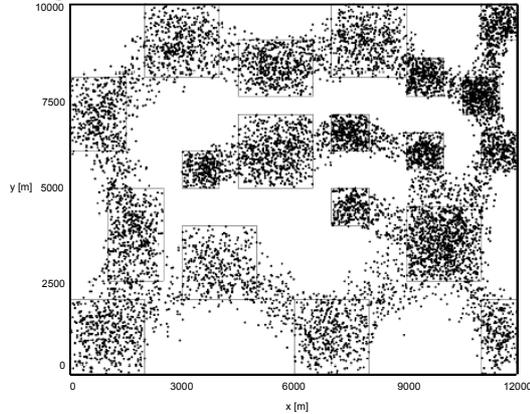


Figure 7.14: Complex network with 10000 nodes and 19 cities

tables in the opposite direction of their trip, i.e., towards their source. Thus, if data packets flow in both directions, they can help each other directly to find shorter paths. The ratio of AMRA to GFG/GPSR is approximately 250 to 150 hops in this scenario. Furthermore, we can observe that AMRA without ants performs only marginally worse than with ants. The 200 sources generate already enough traffic in the network such that node can maintain useful path information on the large scale to distant areas. Source and destination nodes may be temporarily close to each other such that greedy routing of GFG/GPSR is able to deliver packets. During this time, GFG/GPSR will perform equal or even slightly better than AMRA as seen in Section 7.3.1. Considering this fact, one might expect an even higher performance gain of AMRA over GFG/GPSR when source and destination nodes are far apart such that simple greedy routing will not succeed.

7.4 Conclusions

In this chapter, we proposed the AMRA protocol architecture, which makes use of topology abstraction, ant colony optimization, and position-based routing. Topology abstraction is used to hide the dynamic topology of an ad-hoc network to the actual routing protocol. The routing protocol on top of this abstract topology finds short paths over intermediate anchor points. A conventional position-based routing protocol is used to route the packets between the anchor points. The forwarding of packets in greedy mode only between the anchor points allows to circumvent effectively areas with no or low connectivity.

AMRA shows almost equal performance as GFG/GPSR in simple networks where the packets can be delivered directly to the destination node and do not need to be routed around voids. Designed for large and irregular topologies, AMRA performs superior in such scenarios and routes packets over paths that

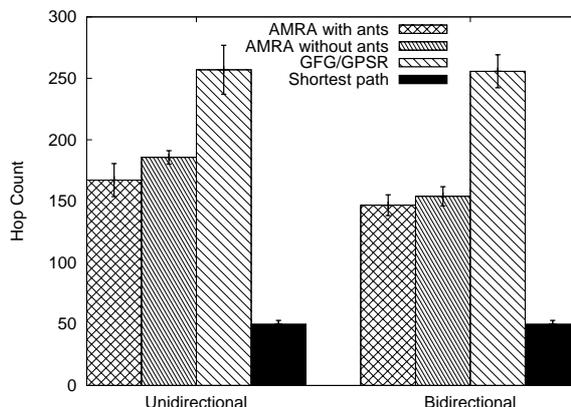


Figure 7.15: Hop count in a very large network

are around 50% shorter than of GFG/GPSR. This performance comes at a certain cost, but which can be kept reasonable small. Compared to conventional position-based routing, AMRA has to maintain a small routing table at the nodes limited to some hundred bytes only and little additional control packets are transmitted to detect new paths. We also conducted simulation without any ant packets, which revealed that AMRA can operate efficiently without ants. The performance suffers only slightly in low traffic scenarios and not at all in network with high traffic load because data packets adopt the role of ants. The advantages of AMRA compared to Terminode routing are that AMRA uses hop-by-hop and not source routing and, thus, has smaller headers. Unlike Terminodes, AMRA does not introduce latency before forwarding if no path is available. Packets are simply transmitted and the next node which has an entry for the destination forwards the packet over the correct logical links. The mechanism of Terminode to determine the anchored path requires communication with possibly distant nodes, causes a lot of transmissions, and long delays. The simulation results also showed that the Terminode local routing degrades the performance significantly in dense networks. The transmitted hello messages with all known neighbors may become large and congest the network. The AMRA architecture is designed such that the individual protocols could be replaced with relative small costs. Instead of using MABR based on ant colony optimization, e.g., DSR [129] could be used to find paths on the abstract topology. Similarly, the more sophisticated GOAFR [149] protocol could be used instead of GFG/GPSR for the physical forwarding. The gain may be however limited as GOAFR is superior to GFG/GPSR for scenarios where face routing has to be applied, i.e., if packets are routed in recovery mode, what is exactly avoided by the use of AMRA. Furthermore, the presented TAP protocol could also be replaced with another protocol, which, e.g., allows to aggregate nodes in a different way and provide another logical topology to MABR.

BLR as proposed in Chapter 5 may also be an appropriate protocol for StPF mainly for two reasons. First, AMRA selects paths along areas of high node densities where greedy routing hardly ever fails. BLR is suited for such scenarios because BLR shows its advantages especially if packets can be routed in greedy

mode most of the time. Secondly, BLR could be beneficial because GFG/GPSR and, thus, all position-based routing protocols that require neighbor knowledge and transmit beacons, are not able to operate in highly dynamic scenarios such as on a highway where neighbor information is outdated quickly.

However, AMRA has several drawbacks caused by the non-deterministic routing of packets. The fluctuation in the chosen path is quite high and it is difficult to predict which path a given packet will take and even more difficult to reason about the behavior of the protocol. Therefore, it is also not possible to provide hard results. We observed also that the non-deterministic forwarding of packets causes many packets to loop in the network. Even though AMRA addresses the problem by introducing logical links and also some countermeasures to reduce the risk of looping packets, the problem is not solved satisfactory.

Thus, we can summarize that unlike conventional position-based routing protocols such as GFG/GPSR, AMRA does not route packets greedily towards the destination if not possible but over intermediate anchor positions which yields more optimal paths as forwarding packets with face routing in recovery mode can be avoided. However, these advantages are partially negated by looping packets. The actual forwarding of the packets between these intermediate anchor positions is still accomplished by a position-based routing protocol such that AMRA retains their advantages such as low control traffic and resilience to frequently changing topologies.

Chapter 8

Conclusions

In this thesis, we studied broadcast and routing protocols for ad-hoc networks that make use of location information. We proposed and designed two routing protocols and one broadcast protocol. Their performance and behavior was evaluated analytically as well as by simulations. Furthermore, one routing protocol was also implemented in a testbed of Linux laptop computers.

In Chapter 4, we first investigated in more depth the impact of beaconing and inaccurate neighbor information on position-based protocols. We saw that packets are frequently forwarded to unreachable neighbors in highly mobile networks. The attempts to deliver packets to unavailable neighbors introduce significant additional delays and consumes a substantial amount of scarce network resources such as bandwidth and battery. The proposed simple enhancements to a standard position-based routing protocol made it more robust to changing network topologies. However, the main flaw of these protocols remain, namely their statefulness about the neighborhood. The network topology may change too frequently in certain kinds of ad-hoc networks so that information about neighbors is never accurate. Consequently, forwarding nodes have to select a next hop based on outdated neighbor information.

In Chapter 5, we proposed a routing protocol BLR based on a novel routing paradigm that addresses the problem that source nodes explicitly have to select a next hop. Forwarding decisions are no longer taken at the sender of a packet, but in a completely distributed way at all the receivers. Optimized forwarding among the receiving neighbors is achieved by a concept of dynamic forwarding delay DFD. Each node schedules the forwarding of the packet depending on its suitability so that the most appropriate neighbor forwards the packet first and suppresses the other potential forwarders. This new routing paradigm has two major advantages. First, nodes are no longer required to transmit proactively hello messages to announce their presences, which saves network resources. The second advantage is that BLR is completely stateless as forwarding decisions are taken at the receivers of a packet. A sending node does not have to route packets on an outdated network topology and BLR proves to be almost unaffected even by highest rate of topology changes. These two characteristics make BLR especially suited for any kind of ad-hoc networks with frequently changing topology and/or for networks with very strict power and bandwidth constraints. The main drawbacks of BLR are that this new routing paradigm requires a rather high node density to route packets efficiently,

and that packet duplications occur frequently in realistic scenarios if not all nodes within transmission range receive a packet. BLR incorporates features to cope with such scenarios of lower node densities and not reliable transmissions among neighbors, but then BLR basically reduces to a standard position-based routing protocol and loses many of its advantages. Furthermore, even if we have indications that the power consumption of BLR is reduced compared to other position-based routing protocols because no beacons are transmitted, a more detailed analysis is required to come to a validated conclusion, which is unfortunately out of scope of this thesis.

In Chapter 6, the same paradigm was used in the proposed broadcast protocol DDB, where receiving nodes determine in a completely distributed way whether and when to rebroadcast a packet, without any knowledge about their neighborhood. In analogy to BLR, we achieve locally optimal rebroadcasting decisions by using the concept of dynamic forwarding delay DFD so that the most appropriate neighbor rebroadcasts the packet first. The dynamic forwarding delay can be calculated based on various metrics such as the probability for reaching new neighbors and the residual battery power in order to minimize the number of transmissions to deliver the packet to all nodes in the network and to maximize network lifetime respectively. Again similar as for BLR, the performance of DDB for certain metrics may suffer significantly in realistic scenarios where transmission ranges can be highly irregular. For both DDB and BLR, and most other protocols proposed in the literature, the impact of real world effects have to be investigated in more depth to come to a final conclusion about their suitability in practice.

In Chapter 7, we proposed a protocol architecture AMRA that is topology aware on a large scale by memorizing overheard traffic. This allows AMRA to find more optimal paths in irregular networks where routing along the line-of-sight towards the destination is not feasible. Packets are routed over intermediate anchor positions to circumvent voids and avoid dead ends. AMRA makes use of topology abstraction, ant colony optimization, and position-based routing. The advantages that nodes do not route packets directly towards the destination are a reduced average path length and hop count because packets are not routed over infeasible paths. We could show that these advantages can be achieved almost with no overhead. Only very little memory is required to store an approximation of the overall network distribution and, depending on the data traffic, no additional control traffic is required to retain these advantages. However, we also encountered a severe problem in the design of AMRA, namely the non-deterministic routing of packets, which caused many packets to loop in the network. Even though, the basic idea of determining a rough approximation of the overall network topology proved its potential, a careful redesign of the protocols is required.

We can summarize the main conclusions from the work performed in this thesis as follows. Existing position-based routing protocols may be appropriate for many scenarios but have also significant shortcomings in others. First, these protocols need to keep track of their local neighborhood, which requires resource consuming transmissions of hello packets. In dynamic networks with frequently changing topologies, it is also hardly possible to maintain accurate information about the neighborhood. Routing protocols have to operate on outdated neighbors' positions and the network performance degrades significantly. As second drawback, we identified that existing position-based routing protocol

forward packet solely on local neighbor information. Especially in large networks, the local optimal choice to forward a packet may be highly suboptimal on the global scale. We proposed two routing protocols, BLR and AMRA, and a broadcast protocol, DDB, which are specifically designed for the aforementioned two scenarios and address the respective drawbacks. AMRA tries to obtain an aggregated view on the global network topology by overheard packets and uses this knowledge to optimize routing to areas where no direct path along a line-of-sight exists. BLR and DDB are stateless about the local neighborhood and, thus, are immune to topology changes and save scarce network resources by the disposal of any proactive communication traffic. Therefore, these protocols may especially be appropriate in sensor and/or vehicular ad-hoc networks.

Chapter 9

Future Work

In this chapter, we briefly elaborate on possible future work in the field of routing in ad-hoc networks as studied in this thesis. We start by considering more general aspects of future work such as the adaptation of the protocols to new application areas, the design of new protocols based on proposed ideas, or the integration with new technologies. In a second step, we have a closer look at some specific problems encountered in the proposed protocols and possible ways to solve or deal with them.

Although, several properties of BLR and DDB make them especially appropriate for sensor networks, a lot of work remains to be done. For example most researchers agree that we need a more integrated way of looking at the protocols in sensor networks, and ad-hoc networks in general, due to the strict constraints which pose strong challenges. The strict layered protocol stack does often not allow making use of the available information in the most efficient way. Protocols should be designed with cross layer objective in mind and not be restricted to one specific layer. We have seen in this thesis that BLR and DDB already partially cooperate with the MAC layer. However, an overall integrated design of BLR and DDB with protocols on the other layers would raise many further possibilities. Already, an appropriate MAC protocol that is designed specifically for the cooperation with BLR and DDB could alleviate some of the observed drawbacks. For example, a multichannel access scheme could lower the contention among forwarding nodes. The adaptation of network layer protocols such as BLR and DDB to be fully integrated with other layers and the study of the interactions with the other layers are a large area of future research.

In sensor networks, protocols based on the concept of dynamic forwarding delay DFD, which are immune to topology changes and do not require control packet overhead, may be very appropriate. In this thesis, we proposed a unicast routing BLR and a broadcast protocol DDB based on the DFD concept. Similar to DDB, BLR can also be used to route packets in sensor networks and save energy at almost depleted nodes, e.g., by making the delay at each node dependent on the residual battery energy. Furthermore, the same concept of DFD may also be used for the design of a robust multicast or geocast protocol for sensor networks. Especially geocasting is a common task in sensor networks. Typically, information is requested about a certain area. Therefore, a request is forwarded to this specific area and only sensors in this area transmit their collected data back to the sink.

New technologies can also be used to design more sophisticated schemes, or the offered possibilities can be incorporated in existing protocols. For example in BLR and DDB, the potential receivers of a packet are located only in a fraction of the whole transmission range area. In the greedy mode of BLR, this area is limited to the forwarding area. In DDB, we have seen that the previously broadcasting node already covers about half of the neighbors. Therefore, an interesting approach would be the integration with directional antennas. They could be used to transmit the packets only in the required directions in order to reduce energy consumption and interferences with other transmissions. However, the use of directional antennas raises also many new problems, e.g., passive acknowledgement of BLR is no longer possible and in DDB the determination of the already covered neighbors is difficult if radio transmissions are not omnidirectional.

AMRA as proposed in this thesis focused on the optimization of end-to-end performance. However, the concept of determining more appropriate paths between source and destination is not limited to the end-to-end objective. Considering large sensor networks, the network lifetime may be the most important factor and, thus, data packets should circumvent areas with almost depleted nodes. AMRA could be used in such scenarios to identify areas which should be avoided and to route packets along other routes. Therefore, packets keep track of the encountered battery power levels of visited nodes, which allows nodes to send packets along paths where nodes have a lot of remaining energy.

After having briefly elaborated on possible future directions of research in a more general way, we also want to pick up some of the encountered problems of the proposed protocols and discuss possible optimizations to cope with them.

Regarding the routing protocol BLR, we envision to adjust dynamically the parameters and to selectively turn on and off the proposed options. In certain scenarios, they may be advantageous while in others they only introduce unnecessary overhead. This would allow coping more efficiently with the encountered network conditions. For example, we have seen that packet duplication is a major problem of BLR in many realistic scenarios where not all potential forwarders detect each others' transmissions. Therefore, if it was possible to determine the packet duplication rate by any means, e.g., by simply counting the arrived duplicated packets at the destination, BLR should dynamically switch to unicast mode to lower the risk of duplicated packet. A further example for a possible dynamic adaptation in BLR is the value of *Max_Delay*. For scenarios with high traffic volumes and high node densities, *Max_Delay* should be longer to lower contention among forwarding nodes while for sparse networks a long *Max_Delay* introduces unnecessary long delays. As nodes are unaware of their neighborhood, a simple approach to estimate roughly the node density is for example from the *Add_Delay* introduced at each hop. As the broadcasting protocol DDB is based on the same concept of dynamic forwarding delay and uses the same parameter *Max_Delay*, the same dynamic adaptation as proposed BLR can be incorporated into DDB as well. Furthermore, the Linux implementation of BLR was only validated in the laboratory with some basic experiments. For future work, it is definitely interesting to conduct also outdoor experiments with moving nodes and position information provided by GPS.

For AMRA, the problem of looping packets is not solved satisfactory. One possibility to reduce the risk of loops could be the introduction of longer logical links as already proposed in the original approach in [98]. If a packet is routed

over a long logical link, the packet is significantly closer to the final destination and therefore the probability for routing again in the opposite direction is lowered. We can also think of many other enhancements to AMRA. First, the static formation of the logical routers and the zones often does not really match to the actual network topology. Therefore, we may apply more sophisticated methods to group nodes to logical routers in a dynamic way, e.g., by clustering algorithms, and similarly group also logical routers to zones. Secondly, we may also consider exchanging routing tables, or parts of it, between nodes within a logical router. This could help to improve routing if nodes move rapidly and a single node does not have reliable information in its routing tables. The small size of the routing table would consume only little resources and bandwidth for transmissions. Furthermore, packet could keep track of their followed path and updates routing tables not only with respect to their source, but to all visited intermediate logical routers as well. This would disseminate information about good paths more quickly in the network.

For future work, we also plan the integration of BLR and AMRA, i.e., the use of the BLR protocol as StPF within the AMRA framework. This integration could show many advantages as BLR and AMRA are complementary in the sense that BLR addresses the local neighborhood and how packets are forwarded to the immediate next hops while AMRA is concerned with the routing on a large scale.

In this chapter, we briefly discussed possible future directions of research and some more or less obvious possible extensions and optimization to the proposed protocols. We may think of many further optimizations for all three protocols. They may improve the performance and reliability in certain scenarios, but may perform poorly in others and cause new problems. It is an inherent property of the complex simulations of ad-hoc networks that they are very sensitive to the parameter values of the simulations, and also of the protocols. Therefore, simulation results can solely give an indication about the possible potential of a proposed protocol but cannot provide hard evidence for its superior, or inferior, performance. This statement also holds for the simulations conducted in this thesis.

Bibliography

- [1] I. Abraham, D. Dolev, and D. Malkhi, “LLS: a locality aware location service for mobile ad hoc networks,” in *Proceedings of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing (DIALM-POMC 2004)*, Philadelphia, Pennsylvania, USA, Oct. 2004.
- [2] A. Agarwal and S. R. Das, “Dead reckoning in mobile ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, New Orleans, Louisiana, USA, Mar. 2003, pp. 1838–1843.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
- [4] S. Appleby and S. Steward, “Mobile software agents for control in telecommunication networks,” *British Telecom Technologie Journal*, vol. 12, no. 2, pp. 104–113, Apr. 1994.
- [5] S. Bansal, R. Shorey, and A. Misra, “Comparing the routing energy overheads of ad-hoc routing protocols,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, New Orleans, Louisiana, USA, Mar. 2003, pp. 1155–1161.
- [6] J. S. Baras and H. Mehta, “A probabilistic emergent routing algorithm for mobile ad hoc networks,” in *Proceedings of IEEE Workshop of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '03)*, INRIA Sophia-Antipolis, France, Mar. 2003.
- [7] L. Barri re, P. Fraigniaud, and L. Narayanan, “Robust position-based routing in wireless ad hoc networks with unstable transmission ranges,” in *Proceedings of the 5th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '01)*, Rome, Italy, July 2001, pp. 19–27.
- [8] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, “Geographic messaging in wireless ad hoc networks,” in *Proceedings of IEEE Vehicular Technology Conference (VTC'99)*, Houston, Texas, USA, May 1999, pp. 1957–1961.
- [9] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A distance routing effect algorithm for mobility (DREAM),” in *Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, Dallas, Texas, USA, Oct. 1998, pp. 76–84.

- [10] R. Beckers, J. L. Deneuborg, and S. Goss, "Trails and u-turns in the selection of a path by the ant *lasius niger*," *Journal of Theoretical Biology*, vol. 159, pp. 397–415, 1992.
- [11] B. Bellur and R. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, New York, USA, Mar. 1999, pp. 178–186.
- [12] R. Bischoff and R. Wattenhofer, "Analyzing connectivity-based multihop ad-hoc positioning," in *Proceedings of the Second Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '04)*, Orlando, USA, Mar. 2004.
- [13] L. Blazevic, "Scalable routing protocols with applications to mobility," Ph.D. dissertation, EPFL, Lausanne, Switzerland, Feb. 2002.
- [14] L. Blazevic, S. Giordano, and J.-Y. Le Boudec, "Self organized terminode routing," *Cluster Computing Journal*, vol. 5, no. 2, pp. 205–218, Apr. 2002.
- [15] L. Blazevic, J.-Y. Le Boudec, and S. Giordano, "A location based routing method for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, Mar. 2005.
- [16] Bluetooth core specification v1.2. Bluetooth Special Interest Group (SIG). [Online]. Available: <https://www.bluetooth.org/spec/>
- [17] B. Blum, T. He, S. Son, and J. A. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," Department of Computer Science, University of Virginia, USA, Tech. Rep. CS-2003-11, 2003.
- [18] J. Boleng, "Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks," in *Proceedings of the 11th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN'01)*, Boulder, CO, USA, Mar. 2001, pp. 9–12.
- [19] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, July 2000.
- [20] E. Bonabeau, F. Henaux, S. Guérin, D. Snyers, P. Kuntz, and G. Theraulaz, "Routing in telecommunication networks with "smart" ant-like agents," in *Proceedings of the Second International Workshop on Agents in Telecommunications Applications (IATA '98)*, Paris France, July 1998.
- [21] P. Bose and P. Morin, "Online routing in triangulations," in *Proceedings of the 10th Annual International Symposium on Algorithms and Computation (ISAAC '99)*, Chennai, India, Dec. 1999, pp. 113–122.

- [22] P. Bose, P. Morin, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, J. I. Munro, and A. Lopez-Ortiz, "Online routing in convex subdivisions," in *Proceedings of the 11th Annual International Symposium on Algorithms and Computation (ISAAC '00)*, Taipei, Taiwan, Dec. 2000, pp. 47–59.
- [23] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99)*, Seattle, USA, Aug. 1999, pp. 48 – 55.
- [24] —, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM/Baltzer Wireless Networks*, vol. 7, no. 6, pp. 609–616, Nov. 2001.
- [25] E. Bretz, "Precision navigation in european skies," *IEEE Spectrum*, vol. 40, no. 9, p. 16, Sept. 2003.
- [26] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, Mar. 2005.
- [27] B. Bullnheimer, R. Hartl, and C. Strauss, "Applying the ant system to the vehicle routing problem," in *MetaHeuristics: Advances and Trends in Local Search Paradigms for Optimization*. Boston, USA: Kluwer Academic Publishers, 1999, pp. 285–296.
- [28] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, Oct. 2000.
- [29] J. J. Caffery and G. L. Stüber, "Overview of radiolocation in CDMA cellular systems," *IEEE Communications Magazine*, vol. 36, pp. 38–45, Apr. 1998.
- [30] M. Cagalj, J.-P. Hubaux, and C. Enz, "Energy-efficient broadcasting in all-wireless networks," *ACM/Baltzer Mobile Networks and Applications*, 2004, to appear.
- [31] D. Câmara and A. A. F. Loureiro, "A novel routing algorithm for hoc networks," *Baltzer Journal of Telecommunications Systems*, vol. 18, no. 1-3, pp. 85–100, 2001.
- [32] T. Camp, "Location information services in mobile ad hoc networks," The Colorado School of Mines, Golden, CO, USA, Tech. Rep. MCS-03-15, Oct. 2003.
- [33] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, vol. 2, no. 5, pp. 483–502, Aug. 2002.
- [34] T. Camp, J. Boleng, B. Williams, W. Navidi, and L. Wilcox, "Performance comparison of two location based routing protocols for ad hoc networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer*

- and Communications Societies (INFOCOM '02), New York, USA, June 2002, pp. 1678–1687.
- [35] S. Capkun, M. Hamdi, and J.-P. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” *Cluster Computing Journal*, vol. 5, no. 2, pp. 118–124, Apr. 2002.
- [36] J. Cartigny and D. Simplot, “Border node retransmission based probabilistic broadcast protocols in ad-hoc networks,” *Telecommunication Systems*, vol. 22, pp. 189–204, Apr. 2003.
- [37] J. Cartigny, D. Simplot, and I. Stojmenovic, “Localized energy efficient broadcast for wireless networks with directional antennas,” in *Proceedings of the Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET'2002)*, Sardegna, Italy, Sept. 2002.
- [38] —, “Localized minimum-energy broadcasting in ad-hoc networks,” in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, CA, USA, Mar. 2003, pp. 2210–2217.
- [39] I. D. Chakeres and E. M. Belding-Royer, “The utility of hello messages for determining link connectivity,” in *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC '02)*, Honolulu, Hawaii, Oct. 2002, pp. 504–508.
- [40] S. Chameron, G. Beugnon, B. Schatz, and T. Collett, “The use of path integration to guide route learning in ants,” *Nature*, vol. 399, no. 6738, pp. 769–777, June 1999.
- [41] J.-H. Chang and L. Tassiulas, “Energy conserving routing in wireless ad-hoc networks,” in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, Tel Aviv, Israel, Mar. 2000, pp. 22–31.
- [42] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, “Routing in clustered multihop, mobile wireless networks,” in *Proceedings of IEEE Singapore International Conference on Networks (SICON '97)*, Singapore, Apr. 1997, pp. 197–211.
- [43] C.-H. Chu, J. Gu, X. D. Hou, and Q. Gu, “A heuristic ant algorithm for solving qos multicast routing problem,” in *Proceedings of 2002 Congress on Evolutionary Computation (CEC 2002)*, Honolulu, Hawaii, USA, May 2002, pp. 1630–1635.
- [44] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR),” RFC 3626, Internet Engineering Task Force IETF, Oct. 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3626.txt>
- [45] T. Clausen, P. Jacquet, P. Muhlethaler, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol,” in *Proceedings of IEEE International Multi-topic Conference (INMIC '01)*, Lahore, Pakistan, Dec. 2001.

- [46] (2005, May) Environmental/GPS Sensor Module (MTS420). Crossbow Technology Inc. [Online]. Available: <http://xbow.com/Products/productsdetails.aspx?sid=76>
- [47] F. Dai and J. Wu, "Performance analysis of broadcast protocols in ad hoc networks based on self-pruning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, Nov. 2004.
- [48] A. K. Das, R. J. M. II, M. El-Sharkawi, P. Arabshahi, and A. Gray, "The minimum power broadcast problem in wireless networks: An ant colony system approach," in *Proceedings of IEEE CAS Workshop on Wireless Communications and Networking*, Pasadena, CA, USA, Sept. 2002.
- [49] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," *Cluster Computing Journal, special issue on Mobile Ad-hoc Networking*, vol. 5, no. 2, pp. 169–178, Apr. 2002.
- [50] J.-L. Deneubourg, S. Aron, S. Goss, and J. Pasteels, "The selforganizing exploratory patterns of the argentine ant," *Journal of Insect Behaviour*, vol. 3, pp. 159–168, 1990.
- [51] G. Di Caro, F. Ducatelle, and L. Gambardella, "AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications, Special Issue on Self-Organization in Mobile Networking*, vol. 16, no. 2, 2005.
- [52] G. Di Caro and M. Dorigo, "An adaptive multi-agent routing algorithm inspired by ants behavior," in *Proceedings of the Fifth Annual Australasian Conference on Parallel and Real-Time Systems (PART'98)*, Adelaide, Australia, Sept. 1998.
- [53] —, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, Dec. 1998.
- [54] —, "Two ant colony algorithms for best-effort routing in datagram networks," in *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, Las Vegas, Nevada, USA, Oct. 1998, pp. 541–546.
- [55] —, "AntNet: a mobile agents approach to adaptive routing," Université Libre de Bruxelles, Belgium, Tech. Rep. IRIDIA 97-12, 2003.
- [56] L. Doherty, K. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, USA, Apr. 2001, pp. 201–212.
- [57] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for distributed discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.

- [58] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
- [59] O. Dousse and P. Thiran, “Connectivity vs capacity in dense ad hoc networks,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, Hong Kong, China, Mar. 2004, pp. 1079–1088.
- [60] O. Dousse, P. Thiran, and M. Hasler, “Connectivity in ad-hoc and hybrid networks,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, USA, June 2002.
- [61] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, “Signal stability-based adaptive routing (SSA) for ad hoc mobile networks,” *IEEE Personal Communications Magazine*, vol. 4, no. 1, pp. 36–45, Feb. 1997.
- [62] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, “Age matters: efficient route discovery in mobile ad hoc networks using encounter ages,” in *Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003, pp. 257 – 266.
- [63] F. Ducatelle, G. Di Caro, and L. Gambardella, “Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks,” *International Journal on Computational Intelligence and Applications (IJCIA), Special Issue on Nature-Inspired Approaches to Networks and Telecommunications*, 2005, to appear.
- [64] A. El-Hoiydi and J.-D. Decotignie, “WiseMAC: An ultra low power mac protocol for multi-hop wireless sensor networks,” in *First International Workshop on Algorithmic Aspects of Wireless Sensor Networks ALGO-SENSORS 2004*, Turku, Finland, July 2004.
- [65] A. Ephremides, “Energy concerns in wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 46–59, Aug. 2002.
- [66] L. M. Feeney, “An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks,” *ACM/Baltzer Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, June 2001.
- [67] L. M. Feeney, B. Ahlgren, and A. Westerlund, “Spontaneous networking: An application oriented approach to ad hoc networking,” *IEEE Communications Magazine*, vol. 39, no. 6, pp. 176–181, June 2001.
- [68] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, USA, Apr. 2001.

- [69] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Information Sciences Institute, University of Southern California, USA, Tech. Rep. ISI/RR-87-180, Mar. 1987.
- [70] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 122–136, Apr. 1994.
- [71] H. Frey, "Scalable geographic routing algorithms for wireless ad hoc networks," *IEEE Network*, vol. 18, no. 4, pp. 18–22, July 2004.
- [72] H. Friis, "A note on a simple transmission formula," in *Proceedings of the IRE*, Vol. 41, May 1946, pp. 254–256.
- [73] H. Füssler, M. Mauve, H. Hartenstein, M. Käsemann, and D. Vollmer, "A comparison of routing strategies for vehicular ad-hoc networks," in *Proceedings of the 8th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '02)*, Atlanta, Georgia, USA, Sept. 2002.
- [74] H. Füssler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad-hoc networks," *Elsevier's Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, Nov. 2003.
- [75] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003, pp. 222–232.
- [76] J. Gao, L. J. Guibas, J. Hershbarger, L. Zhang, and A. Zhu, "Discrete mobile centers," in *Proceedings of the 17th ACM Symposium on Computational Geometry*, Boston, MA, USA, June 2001, pp. 188–196.
- [77] J. Gao, L. J. Guibas, J. Hershbarger, L. Zhang, and A. Zhu, "Geometric spanners for routing in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 174–185, Jan. 2005.
- [78] (2005, May) Gentoo linux website. Gentoo Foundation, Inc. [Online]. Available: <http://www.gentoo.org>
- [79] M. Gerharz, C. de Waal, M. Frank, and P. Martini, "Link stability in mobile wireless ad hoc networks," in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN '02)*, Tampa, Florida, USA, Nov. 2002, pp. 30–39.
- [80] M. Gerharz, C. de Waal, P. Martini, and P. James, "Strategies for finding stable paths in mobile wireless ad hoc networks," in *Proceedings of the 28th Annual IEEE Conference on Local Computer Networks (LCN '03)*, Königswinter, Germany, Oct. 2003, pp. 130–139.
- [81] S. Giordano and M. Hamdi, "Mobility management: The virtual home region," EPFL, Lausanne, Switzerland, Tech. Rep. SSC/1999/037, Oct. 1999.

- [82] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position based routing algorithms for ad hoc networks: A taxonomy," in *Ad Hoc Wireless Networking*. Kluwer, 2003, to appear.
- [83] A. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Communications Magazine*, vol. 9, no. 4, pp. 8–25, Aug. 2002.
- [84] P. P. Grassé, "La reconstruction du nid et les coordinations inter-individuelles chez bellicositermes natalensis et cubitermes sp. la theorie de la stigmergie: Essai d'interpretation des termites constructeurs," *Insectes Sociaux*, vol. 6, pp. 41–81, 1959.
- [85] M. Grossglauser and M. Vetterli, "Locating nodes with EASE: Mobility diffusion of last encounters in ad hoc networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, USA, Mar. 2003.
- [86] M. Güneş, M. Kähler, and I. Bouazizi, "Ant-routing-algorithm (ARA) for mobile multi-hop ad-hoc networks - new features and results," in *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003)*, Mahdia, Tunisia, June 2003.
- [87] M. Güneş, U. Sorges, and I. Bouazizi, "ARA - the ant-colony based routing algorithm for MANETs," in *Proceedings of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN '02)*, Vancouver, Canada, Aug. 2002, pp. 79–85.
- [88] M. Güneş and O. Spaniol, "Routing algorithms for mobile multi-hop ad-hoc networks," in *Proceedings of International Workshop on Next Generation Network Technologies*, European Comission Central Laboratory for Parallel Processings - Bulgarian Academy of Sciences, Oct. 2002.
- [89] Z. Haas, M. Pearlman, and P. Samar, "Bordercasting resolution protocol (BRP)," IETF Internet Draft, Internet Engineering Task Force IETF, July 2002. [Online]. Available: draft-ietf-manet-brp-02.txt
- [90] —, "Interzone routing protocol (IERP)," IETF Internet Draft, Internet Engineering Task Force IETF, July 2002. [Online]. Available: draft-ietf-manet-ierp-02.txt
- [91] —, "Intrazone routing protocol (IARP)," IETF Internet Draft, Internet Engineering Task Force IETF, July 2002. [Online]. Available: draft-ietf-manet-iarp-02.txt
- [92] —, "The zone routing protocol (ZRP) for ad hoc networks," IETF Internet Draft, Internet Engineering Task Force IETF, July 2002. [Online]. Available: draft-ietf-manet-zone-zrp-04.txt
- [93] Z. J. Haas, J. Deng, P. Papadimitratos, and S. Sajama, "Wireless ad hoc networks," in *Encyclopedia of Telecommunications*. John Wiley & Sons, Inc., 2002.

- [94] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, USA, June 2002, pp. 1707 – 1716.
- [95] Z. J. Haas and B. Liang, "Ad hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 228–240, Apr. 1999.
- [96] Z. J. Haas and M. R. Pearlman, "A hybrid framework for routing in ad hoc networks," in *Ad Hoc Networking*. Addison-Wesley, 2002, pp. 221 – 253.
- [97] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer, "Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project," in *Proceedings of the 2nd ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '01)*, Long Beach, CA, USA, Oct. 2001, pp. 259–262.
- [98] M. Heissenbüttel and T. Braun, "Ants-based routing in large-scale mobile ad-hoc networks," in *Proceedings of Kommunikation in verteilten Systemen (KiVS '03)*, Leipzig, Germany, Feb. 2003, pp. 91–99.
- [99] —, "BLR: A beacon-less routing algorithm for mobile ad-hoc networks," Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland, Tech. Rep. IAM-03-001, Mar. 2003.
- [100] —, "A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks," in *Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN' 03)*, Bern, Switzerland, July 2003, pp. 197–210.
- [101] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, "BLR: Beaconless routing algorithm for mobile ad-hoc networks," *Elsevier's Computer Communications Journal (Special Issue)*, vol. 27, no. 11, pp. 1076–1086, July 2004.
- [102] M. Heissenbüttel, T. Braun, T. Huber, and D. Jörg, "Routing in large wireless multihop networks with irregular topologies," in *5th Scandinavian Workshop on Wireless Ad-hoc Networks (ADHOC 05)*, Stockholm, Sweden, May 2005.
- [103] M. Heissenbüttel, T. Braun, D. Jörg, and T. Huber, "A framework for routing in large ad-hoc networks with irregular topologies," in *Fourth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2005)*, Ile de Porquerolles, France, June 2005.
- [104] M. Heissenbüttel, T. Braun, and T. Roth, "GNU/Linux implementation of a position-based routing protocol," in *submitted to IEEE ICPS Workshop on Multi-hop Ad hoc Networks (REALMAN 2005)*, Santorini, Greece, July 2005.

- [105] M. Heissenbüttel, T. Braun, and M. Wälchli, “Optimizing neighbor table accuracy of position-based routing algorithms,” *under submission to Elsevier Ad Hoc Networks*.
- [106] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli, “Optimized stateless broadcasting in sensor and ad-hoc networks,” Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland, Tech. Rep. IAM-04-015, Dec. 2004.
- [107] —, “Optimized stateless broadcasting in wireless multi-hop networks,” in *Submitted to the Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2005)*, Santa Clara, California, USA, sep 2005.
- [108] M. Heusse, S. Guerin, D. Snyers, and P. Kuntz, “Adaptive agent-driven routing and load balancing in communication networks,” Department Intelligence Artificielle et Sciences Cognitives, ENST de Bretagne, Tech. Rep. RR-98001-IASC, 1998.
- [109] M. Heusse, D. Snyers, S. Guerin, and P. Kuntz, “Adaptive agent-driven routing and load balancing in communication network,” in *Proceedings of the First International Workshop on Ant Colony Optimization (ANTS’98)*, Brussels, Belgium, Oct. 1998.
- [110] —, “Adaptive agent-driven routing and load balancing in communication networks,” *Advances in Complex Systems*, vol. 1, no. 2 & 3, pp. 237–254, June & Sept. 1998.
- [111] X. Hong, K. Xu, and M. Gerla, “Scalable routing protocols for mobile ad hoc networks,” *IEEE Network*, vol. 16, no. 4, pp. 11–21, July 2002.
- [112] T.-C. Hou and V. Li, “Transmission range control in multihop packet radio networks,” *IEEE Transactions on Communications*, vol. 34, no. 1, pp. 38–44, Jan. 1986.
- [113] P.-H. Hsiao, “Geographical region summary service for geographical routing,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 25–39, Oct. 2001.
- [114] C. Hu, Y. Hong, and J. Hou, “On mitigating the broadcast storm problem with directional antennas,” in *Proceedings of IEEE International Conference on Communications (ICC 2003)*, Anchorage, Alaska, USA, May 2003, pp. 104–110.
- [115] IEEE 802.11b standard for local and metropolitan area networks. The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, USA. [Online]. Available: <http://standards.ieee.org/getieee802/802.11.html>
- [116] IEEE 802.15 working group for wireless personal area networks (WPAN). The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, USA. [Online]. Available: <http://www.ieee802.org/15/>
- [117] IEEE 802.16 standard for local and metropolitan area networks. The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, USA. [Online]. Available: <http://www.ieee802.org/16/published.html>

- [118] IEEE 802.20 working group for mobile broadband wireless access (MBWA). The Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, USA. [Online]. Available: <http://grouper.ieee.org/groups/802/20/>
- [119] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, Massachusetts, USA, Aug. 2000, pp. 56–67.
- [120] (2005, May) Mobile Ad-hoc Networks (manet) Working Group. Internet Engineering Task Force (IETF). [Online]. Available: <http://www.ietf.org/html.charters/manet-charter.html>
- [121] (2005, May) Ad-hoc Network Systems (ans) Research Subgroup. Internet Research Task Force (IRTF). [Online]. Available: <http://www.flarion.com/ans-research/>
- [122] (2005, May) The Netfilter Website. [Online]. Available: <http://www.netfilter.org>
- [123] J. J. Garcia-Luna-Aceves and S. Roy, "On-demand loop-free routing with link vectors," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 533–546, Mar. 2005.
- [124] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Personal Communications Magazine*, vol. 8, no. 1, pp. 48–57, Feb. 2001.
- [125] A. Jayasuriya, S. Perreau, A. Dadej, and S. Gordon, "Hidden vs. exposed terminal problem in ad hoc networks," in *Proceedings of the Australian Telecommunication Networks and Applications Conference*, Sydney, Australia, Dec. 2004.
- [126] S. Jiang, "An enhanced prediction-based link availability estimation for MANETs," *IEEE Transactions on Communications*, vol. 52, no. 2, pp. 183–186, Feb. 2004.
- [127] S. Jiang, D. He, and J. Rao, "A prediction-based link availability estimation for mobile ad hoc networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, USA, Apr. 2001, pp. 1745–1752.
- [128] J.J.Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *Proceedings of the 7th IEEE International Conference on Network Protocols (ICNP'99)*, Toronto, Canada, Oct. 1999.
- [129] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad Hoc Networking*. Addison-Wesley, 2001, ch. 5, pp. 139–172.
- [130] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 21–32, Jan. 1987.

- [131] I. Kang and R. Poovendran, "A comparison of power-efficient broadcast routing algorithms," in *Proceedings of IEEE Global Telecommunications Conference (Globecom 2003)*, San Francisco, CA, USA, Dec. 2003.
- [132] —, "Maximizing static network lifetime of wireless broadcast adhoc networks," in *Proceedings of IEEE International Conference on Communications (ICC) 2003*, Anchorage, Alaska, USA, May 2003.
- [133] —, "Power-efficient broadcast routing in adhoc networks using directional antennas: technology dependence and convergence issues," University of Washington, Washington, USA, Tech. Rep. UWEETR-2003-0015, July 2003.
- [134] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, USA, Aug. 2000, pp. 243–254.
- [135] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information dissemination in partitionable mobile ad hoc networks," in *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS '99)*, Lausanne, Switzerland, Oct. 1999, pp. 4–13.
- [136] I. Kassabalidis, M. A. El-Sharkawi, R. J. M. II, P. Arabshahi, and A. A. Gray, "Swarm intelligence for routing in communication networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'01)*, San Antonio, Texas, USA, Nov. 2001.
- [137] —, "Adaptive-SDR: adaptive swarm-based distributed routing," in *Proceedings of IEEE World Congress on Computational Intelligence (WCCI'02)*, Honolulu, Hawaii, USA, May 2002.
- [138] W. Kellerer, C. Bettstetter, and C. P. Sties, "(auto) mobile communication in a heterogeneous and converged world," *IEEE Personal Communications Magazine*, vol. 8, no. 6, pp. 41–47, Dec. 2001.
- [139] M. Kirkpatrick and G. Madey, "Using swarm intelligence to broadcast messages in highly mobile ad hoc networks," in *Proceedings of the Seventh Annual Swarm Researchers Meeting (Swarm2003)*, Notre Dame, IN, USA, Apr. 2003.
- [140] Y.-B. Ko and N. H. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," A&M University, Texas, USA, Tech. Rep. TR-98-018, Sept. 1998.
- [141] —, "Location-aided routing (LAR) in mobile ad hoc networks," *ACM/Baltzer Wireless Networks*, vol. 6, no. 4, pp. 307–321, Sept. 2000.
- [142] Y. B. Ko and N. H. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 471 – 480, Dec. 2002.
- [143] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG '99)*, Vancouver, Canada, Aug. 1999, pp. 51–54.

- [144] S. Krco and M. Dupcinov, "Improved neighbor detection algorithm for AODV routing protocol," *IEEE Communications Letters*, no. 12, pp. 584–586, Dec. 2003.
- [145] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing newly deployed ad hoc and sensor networks," in *Proceedings of the 10th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '04)*, Philadelphia, USA, Sept. 2004.
- [146] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC '03)*, Boston, USA, July 2003, pp. 63–72.
- [147] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *Proceedings of the 6th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '02)*, Atlanta, USA, Sept. 2002, pp. 24–33.
- [148] —, "Ad-hoc networks beyond unit disk graphs," in *Proceedings of the 1st ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, San Diego, California, USA, Sept. 2003, pp. 69–78.
- [149] —, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003, pp. 267 – 278.
- [150] A. Laouiti, A. Qayyum, and L. Viennot, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, Hawaii, USA, Jan. 2001.
- [151] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, USA, Aug. 2000, pp. 120–130.
- [152] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 147–159, Feb. 2005.
- [153] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWIM 2003)*, Boston, Massachusetts, United States, Aug. 2000, pp. 61–68.
- [154] X. Lin, M. Lakshdisi, and I. Stojmenovic, "Location based localized alternate, disjoint, multi-path and component routing algorithms for wireless networks," in *Proceedings of the 2nd ACM International Symposium*

- on *Mobile and Ad Hoc Networking and Computing (MobiHoc '01)*, Long Beach, California, USA, Oct. 2001, pp. 287–290.
- [155] X. Lin and I. Stojmenovic, “GEDIR: Loop-free location based routing in wireless networks,” in *Proceedings of International Conference on Parallel and Distributed Computing and Systems (IASTED'99)*, Nov. 1999, pp. 1025–1028.
- [156] (2005, May) LocustWord. [Online]. Available: <http://www.locustworld.com/>
- [157] W. Lou and J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–122, Apr. 2002.
- [158] —, “Double-covered broadcast (dcb): A simple reliable broadcast algorithm in MANETs,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, Hong Kong, China, Mar. 2004.
- [159] G. Lu, Z. Liu, and Z. Zhou, “Multicast routing based on ant algorithm for delay-bounded and load balancing traffic,” in *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN'00)*, Tampa, FL, USA, Nov. 2000, pp. 362–368.
- [160] J. P. Macker and M. S. Corson, “Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations,” RFC 2501, Internet Engineering Task Force IETF, Jan. 1999. [Online]. Available: <http://www.ietf.org/rfc/rfc2501.txt>
- [161] C. Maihöfer, “A survey of geocast routing protocols,” *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 32–42, Apr. 2004.
- [162] C. Maihöfer and R. Eberhardt, “Virtual warning signs: A geocast enabled service for ad hoc networks,” in *Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN' 03)*, Bern, Switzerland, July 2003, pp. 135–147.
- [163] G. S. Malkin, “RIP version 2,” RFC 2453, Internet Engineering Task Force IETF, Nov. 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2453.txt>
- [164] M. Marathe, H. Brey, H. Hunt III, S. Ravi, and D. Rosenkrantz, “Simple heuristics for unit disk graphs,” *Networks*, vol. 25, no. 59-68, 1995.
- [165] S. Marwaha, C. K. Tham, and D. Srinivasan, “Mobile agents based routing protocol for mobile ad hoc networks,” in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'02)*, Taipei, Taiwan, Nov. 2002.
- [166] M. S. C. Matthew Impett and V. Park, “A receiver-oriented approach to reliable broadcast in ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '00)*, Chicago, IL, USA, Sept. 2000, pp. 117–122.

- [167] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad-hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, Nov. 2001.
- [168] S. McCanne and V. Jacobson, "The BSD packet filter: a new architecture for user-level packet capture," in *Proceedings of the 1993 winter USENIX conference*, San Diego, CA, USA, Jan. 1993, pp. 259–269.
- [169] A. B. McDonald and T. F. Znati, "A path availability model for wireless ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '99)*, New Orleans, USA, Sept. 1999, pp. 35–40.
- [170] J. McNeff, "The global positioning system," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 2, pp. 645–652, Mar. 2002.
- [171] (2005, May) MeshNetworks. [Online]. Available: <http://www.meshnetworks.com/>
- [172] S. Milgram, "The small world problem," *Psychology Today*, vol. 61, pp. 60 – 67, May 1967.
- [173] G. Morgan-Owen and G. Johnston, "Differential GPS positioning," *Electronics & Communication Engineering Journal*, vol. 7, no. 1, pp. 11–21, Feb. 1995.
- [174] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, "Virtual coordinates for ad hoc and sensor networks," in *Proceedings of the ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC'04)*, Philadelphia, Pennsylvania, USA, Oct. 2004.
- [175] J. T. Moy, "OSPF Version 2," RFC 2178, Internet Engineering Task Force IETF, Apr. 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2328.txt>
- [176] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, pp. 183–197, Oct. 1996, Special Issue on Routing in Mobile Communication Networks.
- [177] (2005, May) Wireless propagation bibliography. National Institute of Standards and Technology NIST, Wireless Communication Technologies Group. [Online]. Available: http://w3.antd.nist.gov/wctg/manet/wirelesspropagation_bibliog.html
- [178] J. C. Navas and T. Imielinski, "Geocast - geographic addressing and routing," in *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '97)*, Budapest, Hungary, Sept. 1997, pp. 66–76.
- [179] W. Navidi and T. Camp, "Stationary distributions for the random way-point mobility model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, Jan. 2004.

- [180] R. Nelson and L. Kleinrock, "The spatial capacity of a slotted ALOHA multihop packet radio network with capture," *IEEE Transactions on Communications*, vol. 32, no. 6, pp. 684–694, June 1984.
- [181] A. Neskovic, N. Neskovic, and G. Paunovic, "Modern approaches in modeling of mobile radio systems propagation environment," *IEEE Communications Surveys*, vol. 3, no. 3, July 2000.
- [182] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, Seattle, USA, Aug. 1999, pp. 151 – 162.
- [183] D. Niculescu and B. Nath, "Ad-hoc positioning system," in *Proceeding of IEEE Global Communications Conference (Globecom 2001)*, San Antonio, Texas, USA, Nov. 2001, pp. 2926–2931.
- [184] —, "Ad hoc positioning system (APS) using AoA," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, CA, USA, Mar. 2003.
- [185] National Marine Electronics Association (NMEA). [Online]. Available: <http://www.nmea.org/pub/0183/>
- [186] J. Nonnenmacher and E. W. Biersack, "Scalable feedback for large groups," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 375–386, June 1999.
- [187] K. Obraczka, K. Viswanath, and G. Tsodik, "Flooding for reliable multicast in multi-hop ad hoc networks," *Wireless Networks*, vol. 7, no. 6, pp. 627–634, Nov. 2001.
- [188] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," RFC 3684, Internet Engineering Task Force IETF, Feb. 2004. [Online]. Available: <http://www.ietf.org/rfc/rfc3684.txt>
- [189] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97)*, Kobe, Japan, Apr. 1997, pp. 1405–1413.
- [190] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing in mobile ad hoc networks," in *Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS '00)*, Taipei, Taiwan, Apr. 2000, pp. 71–78.
- [191] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility," in *Proceedings of the 1st ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '00)*, Boston, USA, Aug. 2000, pp. 11–18.
- [192] W. Peng and X. Lu, "AHBP: An efficient broadcast protocol for mobile ad hoc networks," *Journal of Science and Technology*, vol. 16, no. 2, Mar. 2001.

- [193] W. Peng and X.-C. Lu, “Efficient broadcast in mobile ad hoc networks using connected dominating sets,” *Journal of Software*, vol. 10, no. 7, Apr. 1999.
- [194] C. Perkins and P. Bhagwat, “Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers,” in *Proceedings of the ACM/SIGCOMM '94 Conference on Communications Architecture, Protocols and Applications*, London, UK, Aug. 1994, pp. 234 – 244.
- [195] C. E. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, New Orleans, USA, Feb. 1999, pp. 90–100.
- [196] —, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, Internet Engineering Task Force IETF, July 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3561.txt>
- [197] J. Postel, “Transmission control protocol,” RFC 793, Internet Engineering Task Force IETF, Sept. 1981. [Online]. Available: <http://www.ietf.org/rfc/rfc793.txt>
- [198] R. A. Powers, “Batteries for low power electronics,” *Proceedings of the IEEE*, vol. 83, no. 4, pp. 687 – 693, Apr. 1995.
- [199] S. Preuss and C. Cap, “Overview of spontaneous networking-evolving concepts and technologies,” in *Proceedings of Future Services For Networked Devices (FuSeNetD 1999)*, Heidelberg, Germany, Nov. 1999.
- [200] N. Priyantha, A. Chakraborty, and H. Padmanabhan, “The cricket location support system,” in *Proceedings of the 1st ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '00)*, Boston, MA, USA, Aug. 2000, pp. 32–42.
- [201] S. Rajagopalan, C. Jaikao, and C. Shen, “Unicast routing for mobile ad hoc networks with swarm intelligence,” University of Delaware, Newark, USA, Tech. Rep. 2003-07, May 2003.
- [202] S. Rajagopalan and C. Shen, “A routing suite for mobile ad hoc networks using swarm intelligence,” University of Delaware, Newark, USA, Tech. Rep. 2004-14, May 2004.
- [203] V. Ramasubramaniam, Z. J. Haas, and E. G. Sirer, “SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks,” in *Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003.
- [204] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic routing without location information,” in *Proceedings of the 9th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '03)*, San Diego, CA, USA, Sept. 2003, pp. 96–108.
- [205] G. G. Richard, “Service advertisement and discovery: enabling universal device cooperation,” *IEEE Internet Computing*, vol. 4, no. 5, pp. 18–26, Sept. 2000.

- [206] M. Roth and S. Wicker, "Termite: Ad-hoc networking with stigmergy," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'03)*, San Francisco, USA, Dec. 2003.
- [207] —, "Termite: Emergent ad-hoc networking," in *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003)*, Mahdia, Tunisia, June 2003.
- [208] P. Samar, M. Pearlman, and Z. Haas, "Independent zone routing: An adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 595–608, Aug. 2004.
- [209] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, New Orleans, Louisiana, USA, Mar. 2003, pp. 1124–1130.
- [210] A. Savvides, C.-C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '01)*, Rome, Italy, July 2001.
- [211] (2005, May) Qualnet. Scalable Network Technologies (SNT). [Online]. Available: <http://www.qualnet.com/>
- [212] C. Schindelhauer, T. Lukovszki, S. Rührup, and K. Volbert, "Worst case mobility in ad hoc networks," in *Proceedings of the 15 annual ACM symposium on Parallel algorithms and architectures (SPAA'03)*, San Diego, California, USA, June 2003, pp. 230–239.
- [213] R. Schoonderwoerd, O. Holland, and J. Bruten, "Ant-like agents for load balancing in telecommunications networks," in *Proceedings of the First International Conference on Autonomous Agents (AA '97)*, Marina del Rey, CA, USA, 1997.
- [214] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based load balancing in telecommunications networks," HP Labs Bristol, UK, Tech. Rep. HPL-96-76, May 1996.
- [215] —, "Ant-based load balancing in telecommunications networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1997.
- [216] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 3550, Internet Engineering Task Force IETF, July 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3550.txt>
- [217] M. Seddigh, J. S. Gonzalez, and I. Stojmenovic, "RNG and internal node based broadcasting algorithms for wireless one-to-one networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 2, pp. 37–44, Apr. 2001.

- [218] C.-C. Shen, Z. Huang, and C. Jaikaeo, “Ant-based distributed topology control algorithms for mobile ad hoc networks,” *Wireless Networks*, vol. 11, no. 3, p. 299317, May 2005.
- [219] C. Shen and C. Srisathapornphat, “Ant-based energy conservation for mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, p. to appear, Mar. 2004.
- [220] K. M. Sim and W. H. Sun, “Ant colony optimization for routing and load-balancing: survey and new directions,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 33, no. 5, pp. 560–572, Sept. 2003.
- [221] —, “Multiple ant colony optimization for load balancing,” *Lecture Notes in Computer Science*, vol. 2690, pp. 467 – 471, Sept. 2003.
- [222] S. Singh, M. Woo, and C. S. Raghavendra, “Power-aware routing in mobile ad hoc networks,” in *Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’98)*, Dallas, USA, Oct. 1998, pp. 181–190.
- [223] D. Son, A. Helmy, and B. Krishnamachari, “The effect of mobility-induced location errors on geographic routing in mobile ad hoc and sensor networks: Analysis and improvement using mobility prediction,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 233–245, Sept. 2004.
- [224] C. Srisathapornphat and C. Shen, “Ant-based energy conservation for ad hoc networks,” in *Proceedings of 12th International Conference on Computer Communications and Networks (ICCCN’03)*, Dallas, Texas, USA, Oct. 2003.
- [225] W. Stark, H. Wang, A. Worthen, S. Lafortune, and D. Teneketzis, “Low-energy wireless communication network design,” *IEEE Wireless Communications Magazine*, vol. 1, no. 4, pp. 60–72, Aug. 2002.
- [226] J. A. Stine and G. de Veciana, “A paradigm for quality-of-service in wireless ad hoc networks using synchronous signaling and node states,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1301–1321, Sept. 2004.
- [227] I. Stojmenovic, “Position based routing in ad hoc networks,” *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–134, 2002.
- [228] I. Stojmenovic and X. Lin, “Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023 – 1032, Oct. 2001.
- [229] —, “Power-aware localized routing in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, Nov. 2001.
- [230] I. Stojmenovic, M. Russell, and B. Vukojevic, “Depth first search and location based localized routing and QoS routing in wireless networks,” *Computers and Informatics*, vol. 21, no. 2, pp. 149–165, 2002.

- [231] I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, Jan. 2002.
- [232] I. Stojmenovic and B. Vukojevic, “A routing strategy and quorum based location update scheme for ad hoc wireless networks,” SITE, University of Ottawa, Ottawa, Canada, Tech. Rep. TR-99-09, Sept. 1999.
- [233] I. Stojmenovic and J. Wu, “Broadcasting and activity-scheduling in ad hoc networks,” in *Ad Hoc Networking*. IEEE Press, 2004, pp. 205–229.
- [234] T. Stützle and M. Dorigo, “The ant colony optimization metaheuristic: Algorithms, applications, and advances,” in *Handbook of Metaheuristics*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.
- [235] W. Su, S.-J. Lee, and M. Gerla, “Mobility prediction and routing in ad hoc wireless networks,” *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, Jan. 2001.
- [236] D. Subramanian, P. Druschel, and J. Chen, “Ants and reinforcement learning: A case study in routing in dynamic networks,” in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, Aug. 1997, pp. 832–838.
- [237] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246–257, Mar. 1984.
- [238] C.-K. Toh, “Associativity based routing for ad hoc mobile networks,” *IEEE Personal Communications Magazine*, vol. 4, no. 2, pp. 103–139, Mar. 1997, Special Issue on Mobile Networking and Computing Systems.
- [239] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih, “Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network,” *IEEE Transactions on Computers*, vol. 52, no. 5, pp. 545–557, May 2003.
- [240] (2005, May) Virtual Point-to-Point (TUN) devices. [Online]. Available: <http://vtun.sourceforge.net/tun/index.html>
- [241] D. Turgut, S. K. Das, and M. Chatterjee, “Longevity of routes in mobile ad hoc networks,” in *Proceedings of the IEEE semiannual Vehicular Technology Conference (VTC '01)*, Rhodes, Greece, May 2001, pp. 2833–2837.
- [242] P. Varaiya, “Smart cars on smart roads: problems of control,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 195–207, Feb. 1993.
- [243] G. N. Varela and M. C. Sinclair, “Ant colony optimisation for virtual-wavelength-path routing and wavelength allocation,” in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, Washington DC, USA, July 1999, pp. 1809–1816.
- [244] I. Wagner, M. Lindenbaum, and A. M. Bruckstein, “ANTS: Agents, networks, trees, and subgraphs,” *Future Generation Computer Systems Journal*, vol. 16, no. 8, pp. 915–926, June 2000.

- [245] P.-J. Wan, G. Calinescu, X.-Y. Li, and O. Frieder, “Minimum-energy broadcast routing in static ad hoc wireless networks,” in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, USA, Apr. 2001, pp. 1162–1171.
- [246] Y. Wang and J. J. Garcia-Luna-Aceves, “Broadcast traffic in ad hoc networks with directional antennas,” in *Proceedings of IEEE Global Telecommunications Conference (Globecom 2003)*, San Francisco, CA, USA, Dec. 2003, pp. 210–215.
- [247] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks,” in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, USA, Apr. 2001, pp. 1388–1397.
- [248] R. Wattenhofer and A. Zollinger, “XTC: A practical topology control algorithm for ad-hoc networks,” in *Proceedings of the 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN '04)*, Santa Fe, USA, Apr. 2004.
- [249] T. White, “SynthECA: A synthetic ecology of chemical agents,” Ph.D. dissertation, Carleton University, Northfield, MN, USA, Aug. 2000.
- [250] T. White and B. Pagurek, “Towards multi-swarm problem solving in networks,” in *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS '98)*, Paris, France, July 1998, pp. 333–340.
- [251] —, “Application oriented routing with biologically-inspired agents,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, Orlando, FL, USA, July 1999.
- [252] —, “Distributed fault location in networks using learning mobile agents,” in *Proceedings of the Second Pacific Rim International Workshop on Multi-Agents (Prima '99)*, Kyoto, Japan, Dec. 1999, pp. 182 – 196.
- [253] —, “Emergent behavior and mobile agents,” in *Proceedings of the workshop on Mobile Agents in the Context of Competition and Cooperation at Autonomous Agents '99*, Seattle, WA, USA, May 1999.
- [254] T. White, B. Pagurek, and D. Deugo, “Collective intelligence and priority routing in networks,” in *Proceedings of the 15th International Conference on Industrial and Engineering, Applications of Artificial Intelligence and Expert Systems: Developments in Applied Artificial Intelligence*, Cairns, Australia, June 2002, pp. 790 – 800.
- [255] T. White, B. Pagurek, and F. Oppacher, “Ant search with genetic algorithms: application to path finding in networks,” in *Combinatorial Optimization '98*, Brussels, Belgium, Apr. 1998.

- [256] —, “ASGA: improving the ant system by integration with genetic algorithms,” in *Proceedings of the Third Genetic Programming Conference (GP’98)*, Madison, Wisconsin, USA, July 1998, pp. 23–31.
- [257] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “On the construction of energy-efficient broadcast and multicast trees in wireless networks,” in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’00)*, Tel Aviv, Israel, Mar. 2000, pp. 585–594.
- [258] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *Proceedings of the 3rd ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc ’02)*, Lausanne, Switzerland, June 2002, pp. 194–2002.
- [259] B. Williams, D. P. Mehta, T. Camp, and W. Navidi, “Predictive models to rebroadcast in mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 295–303, Sept. 2004.
- [260] J. Wu and F. Dai, “Broadcasting in ad hoc networks based on self-pruning,” in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’03)*, San Francisco, USA, Mar. 2003, pp. 2240–2250.
- [261] —, “Mobility management and its applications in efficient broadcasting in mobile ad hoc networks,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’04)*, Hong Kong, China, Mar. 2004.
- [262] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks,” in *Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM ’99)*, Seattle, USA, Aug. 1999, pp. 7–14.
- [263] H. Xia, H. Bertoni, L. Maciel, A. Lindsay-Stewart, and R. Rowe, “Radio propagation characteristics for line-of-sight microcellular and personal communications,” *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 10, pp. 1439–1447, Oct. 1993.
- [264] Y. Xu, J. S. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’01)*, Rome, Italy, July 2001, pp. 70–84.
- [265] F. Xue and P. R. Kumar, “The number of neighbors needed for connectivity of wireless networks,” *Wireless Networks*, vol. 10, no. 2, pp. 169–181, Mar. 2004.
- [266] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, “Security in mobile ad hoc networks: Challenges and solutions,” *IEEE Wireless Communications Magazine*, vol. 11, no. 1, pp. 38–47, Feb. 2004.

- [267] W. Ye, J. S. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, USA, June 2002, pp. 1567–1576.
- [268] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, USA, Mar. 2003.
- [269] J. Zhang, "Location management in cellular networks," in *Handbook of wireless networks and mobile computing*. John Wiley & Sons, Inc., 2002, pp. 27–49.
- [270] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio asymmetry on wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*, Boston, USA, June 2004.
- [271] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349–365, Oct. 2003.
- [272] —, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct. 2003.
- [273] X. Zou, B. Ramamurthy, and S. Magliveras, "Routing techniques in wireless ad hoc networks classification and comparison," in *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics, and Informatics (SCI '02)*, Orlando, USA, July 2002.

Acknowledgement

I would like to thank Dr. Ljubica Blažević, a former member of the Laboratory for computer Communications and Applications at EPF Lausanne, for the code of the Terminode routing protocol. Many thanks go also to Aaron Zollinger and Regina O'Dell of the Distributed Computing Group at ETH Zürich who kindly provided the code that served as the basis for Java simulator used in this thesis.

Curriculum Vitae

1974	Born on August, 5 in Bern, Switzerland
1981 - 1984	Elementary School Hinterkappelen
1984 - 1986	Secondary School Hinterkappelen
1986 - 1994	Gymnasium Bern-Neufeld, Typus C
1995	Tianjin Foreign Language Institute, Tianjin, China
1995 - 2001	University of Bern, Switzerland. Major in Computer Science and Minors in Mathematics and Information Systems
2001	M.Sc. in Computer Science, University of Bern
2001 - 2005	Research Assistant and Ph.D. Student at the Institute for Computer Science and Applied Mathematics, University of Bern