Building Scalable and Robust
Wireless Mesh Networks

A dissertation submitted to the
ETH ZURICH

for the degree of
Doctor of Science

presented by
RAINER BAUMANN
Master of Science ETH in Computer Science
born March 30, 1979
citizen of Bubikon (ZH) and Langnau am Albis (ZH)

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Ralf Steinmetz, co-examiner

2007
Abstract

City-wide wireless mesh networks are a promising technology for offering ubiquitous Internet connectivity. These multi-hop wireless access networks consist of fixed and mobile nodes, which help each other relaying packets toward Internet gateways and back. The major challenge with these networks is to make them robust and scalable to their size. In this thesis, we therefore develop a scalable routing protocol for robust routing in wireless mesh networks.

The first contribution is the design, implementation, and evaluation of HEAT, a scalable and robust field-based anycast routing protocol for routing packets toward the gateways. The fundamental idea of HEAT is based on an analogy from thermal physics. This analogy consists of routing packets along the steepest gradient of a temperature field that is created by the gateways. Furthermore, this temperature field accounts for the link diversity and redundancy toward the gateways. Using simulations, we compare HEAT with two established routing protocols, one reactive and one proactive, in terms of scalability and robustness.

The second contribution is the investigation of protocols for routing packets back into wireless mesh networks. Using simulations, we compare three different protocols where each of these protocols represents a routing protocol family. We show that the proactive field-based routing protocol outperforms all
others with respect to the packet delivery ratio but is not scalable to the network size. A good compromise is provided by the gateway source routing protocol, which is the most scalable to the network size and still achieves a high packet delivery ratio.

In addition, we design and evaluate a path metric to select robust paths in wireless networks. This path metric selects paths that offer a good trade-off between long routes consisting of many reliable links and short routes of only few, but unreliable links. Our path metric estimates the reliability of a link based on the average measured received signal strength indicator and assigns each neighbor a link cost depending on how close to a preferred strength its received signal strength indicator is. Using simulations, we show for nodes moving at car speed that our path metric improves the performance considerably compared to the minimum hop count metric.

The next contribution is a mobility detection and notification protocol for large-scale wireless mesh networks. In such networks, wireless nodes have to communicate with the Internet via multiple gateways belonging to multiple access networks. Our protocol informs nodes about the gateways over which their data is sent toward the Internet. Thus, these nodes can use common mobility management protocols, independent of the routing service used, to deal with their macro mobility and multihoming.

The last contribution is the development of two mobility models for realistic simulations of wireless networks. The first mobility model uses vectorized street information including speed data from the Swiss geographic information system. The second model is based on traces gained from a leading microscopic, multi-agent traffic simulator.

Summarizing, this thesis provides the bases for building city-wide wireless mesh networks. The proposed protocols distinguish themselves by their performance as well as scalability and thereby allow robust Internet access for mobile as well as fixed users of the mesh network.
Kurzfassung


Im Zentrum dieser Dissertation steht HEAT, ein feld-basiertes anycast Protokoll, welches sehr gute Eigenschaften in Bezug auf Robustheit und Skalierbarkeit besitzt. HEAT leitet sich aus einer Analogie zur Thermodynamik ab; die Grundidee sieht dabei vor, Pakete entlang des steilsten Gradienten eines Temperaturfeldes zu den Gateways weiterzuleiten. Dieses Temperaturfeld wird durch die Gateways gespiesen und berücksichtigt die Vielfalt und Redundanz von Verbindungen zu den Gateways - je höher die Redundanz, desto höher die Temperatur der Knoten. Mittels Simulationen werden die Vorteile von HEAT gegenüber einem reaktiven und einem proaktiven Routingprotokoll in Bezug
auf die Skalierbarkeit zur Netzgrösse und der Robustheit zur Mobilität der Knoten gezeigt.


Des Weiteren wird ein Protokoll zur Erkennung und Meldung der Mobilität von Knoten in grossen drahtlosen Netzwerken spezifiziert. In sehr grossen Netzwerken kommunizieren die Knoten mit dem Internet über viele verschiedene Gateways,
die zu diversen Zugangsnetzwerken gehören. Falls es zu einem Netzwerkwechsel kommt, informiert das Protokoll die drahtlosen Knoten über das aktuell verwendete Internet Gateway. Dank dieser Benachrichtigungen können die Knoten selbst die Mechanismen wählen, die ihre Makromobilität sowie ihr Multihoming unabhängig von verwendeten Routing-Diensten verwalten.


Im Ganzen betrachtet, liefert diese Dissertation die Grundlagen für den Einsatz von stadtweiten drahtlosen Mesh Netzwerken. Die vorgeschlagenen Protokolle zeichnen sich durch ihre Performance sowie Skalierbarkeit aus und ermöglichen dadurch den robusten Internetzugang von mobilen sowie stationären Benutzern über das Mesh Netzwerk.
# Contents

Abstract iii

Kurzfassung v

Contents ix

List of Figures xiii

List of Tables xvii

1 Introduction 1
   1.1 Motivation .......................... 1
   1.2 Research Problems ................... 3
   1.3 Contributions ........................ 3
   1.4 Outline ............................ 5

2 Related Work 7
   2.1 Unicast Routing Protocols for Ad Hoc Networks . 7
   2.2 Routing Protocols for Wireless Mesh Networks ... 8
   2.3 Field- and Gradient-based Routing .............. 11
      2.3.1 TORA ................................ 11
      2.3.2 AntNet ............................ 12
      2.3.3 Directed Diffusion ................. 12
2.3.4 Routing using Potentials .......................... 13
2.3.5 Packetostatics ................................. 13
2.3.6 Field-based anycasting .......................... 13

2.4 Anycast Routing Protocols ......................... 15
  2.4.1 In the Internet ............................... 15
  2.4.2 In Wireless Networks .......................... 15

2.5 Robust Path Construction .......................... 16

2.6 Mobility Management Protocols ..................... 18

2.7 Mobility Models .................................. 19

3 Scalable and Robust Routing ......................... 23
  3.1 Introduction ..................................... 23
  3.2 Forward Path: Routing Packets Toward the
     Gateways ........................................ 26
     3.2.1 Concept and Implementation of the HEAT
         Protocol ....................................... 27
     3.2.2 Simulation Study .............................. 35
     3.2.3 Extended AODV ............................... 36
     3.2.4 OLSR ........................................ 36
     3.2.5 Summary ..................................... 45
  3.3 A Robust Path Metric ................................ 47
     3.3.1 Metric Design ................................ 49
     3.3.2 Implementation .............................. 57
     3.3.3 Simulation Study ............................. 60
     3.3.4 Summary ..................................... 66
  3.4 Reverse Path: Routing Packets into Wireless Mesh
     Networks .......................................... 67
     3.4.1 AODV-CGA (Reactive Hop-by-hop Routing) .... 70
     3.4.2 FBR (Proactive Field-based Routing) ........ 71
     3.4.3 GSR (Gateway Source Routing) .............. 72
     3.4.4 Enhancements ................................ 73
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.5 Simulation Study</td>
<td>74</td>
</tr>
<tr>
<td>3.4.6 Discussion</td>
<td>82</td>
</tr>
<tr>
<td>3.4.7 Summary</td>
<td>83</td>
</tr>
<tr>
<td>3.5 Simulation Study</td>
<td>84</td>
</tr>
<tr>
<td>3.5.1 Set-up</td>
<td>85</td>
</tr>
<tr>
<td>3.5.2 Results</td>
<td>85</td>
</tr>
<tr>
<td>3.6 Demonstrator</td>
<td>87</td>
</tr>
<tr>
<td>3.6.1 User Space Daemon</td>
<td>87</td>
</tr>
<tr>
<td>3.6.2 Kernel Module for Mesh Nodes</td>
<td>88</td>
</tr>
<tr>
<td>3.6.3 Kernel Module for Gateways</td>
<td>88</td>
</tr>
<tr>
<td>3.6.4 Validation</td>
<td>90</td>
</tr>
<tr>
<td>3.7 Summary</td>
<td>90</td>
</tr>
<tr>
<td>4 Mobility Management</td>
<td>93</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>93</td>
</tr>
<tr>
<td>4.2 Micro Mobility</td>
<td>95</td>
</tr>
<tr>
<td>4.3 Macro Mobility and Multihoming</td>
<td>96</td>
</tr>
<tr>
<td>4.3.1 Protocol Overview</td>
<td>96</td>
</tr>
<tr>
<td>4.3.2 Mobility Notification Messages (MNM)</td>
<td>97</td>
</tr>
<tr>
<td>4.3.3 Macro Mobility Detection and Multihoming Support at the Gateways</td>
<td>98</td>
</tr>
<tr>
<td>4.3.4 Handling Mobility Notification Messages at the Mobile Nodes</td>
<td>99</td>
</tr>
<tr>
<td>4.3.5 Joining of a Mobile Node</td>
<td>101</td>
</tr>
<tr>
<td>4.4 Discussion</td>
<td>102</td>
</tr>
<tr>
<td>4.5 Summary</td>
<td>103</td>
</tr>
<tr>
<td>5 Realistic Mobility Models</td>
<td>105</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>105</td>
</tr>
<tr>
<td>5.2 Mobility Model using GIS-data</td>
<td>106</td>
</tr>
<tr>
<td>5.3 Mobility Model using Multi-agent Microscopic Traffic Simulator</td>
<td>107</td>
</tr>
</tbody>
</table>
5.4 Simulation Study ........................................ 111
5.4.1 Simulation Set-up ................................. 112
5.4.2 Set-up of Mobility Models ...................... 113
5.4.3 Results ............................................ 113
5.5 Summary ............................................. 117

6 Conclusions and Future Work 119
6.1 Summary and Conclusions ......................... 119
6.2 Weaknesses and Shortcomings .................... 121
6.3 Future Work and Outlook .......................... 122
6.3.1 Link-Diversity Routing ......................... 122
6.3.2 Toward Realistic Simulations ................. 122
6.3.3 Transport Protocols for Wireless Mesh Networks .......................... 123
6.3.4 Other Issues .................................... 123

A Appendix 127
A.1 Terminology of Field-based Routing ............ 127
A.1.1 Field Construction ............................. 128
A.1.2 Packet Routing ................................ 129
A.2 Support for different Routing Metrics for HEAT . 129
A.3 Visualization ..................................... 130
A.4 Selection of Parameters for HEAT .............. 131

Bibliography 137

Acknowledgements 159

Curriculum Vitae 163
## List of Figures

1.1 A wireless mesh network consisting of mesh nodes and gateways. .......................... 2

3.1 Example of a temperature field with a conductivity value of $\kappa = 1/4$ and areas of different link redundancies. The packets of the node with temperature value 0.051 are forwarded to the right, across an area of high redundancy instead of to the closest gateway at the left. .................. 29

3.2 Example of the temperature field calculation with a conductivity value of $\kappa = 1/4$ for node 53: step 1, sort neighbors (nbr) by temperature value; step 2-5 iterate down the table until the given temperature value of the node is higher or equal to the next neighbor; node 71 and 17 do not contribute to the temperature value of node 53: they will increase their values after the next *HEAT* beacon message of node 53 (node 77: 0.313 and node 17: 0.118) .......................... 33

3.3 Scalability by the network size. .................. 39

3.4 Scalability by the node density. .................. 40

3.5 Mobile scenario at pedestrian speeds. ............. 42

3.6 Mobile scenario at car speeds. .................. 43
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>The effect of the number of gateways (mobile scenario at pedestrian speeds)</td>
<td>46</td>
</tr>
<tr>
<td>3.8</td>
<td>Dependence of the link cost metric on the received signal strength indicator. Note: for simplicity, we depict the communication range (rxThresh) of a node and the power level of the preferred received signal strength indicator as circles, but our algorithm by no means assumes that the range is indeed circular.</td>
<td>52</td>
</tr>
<tr>
<td>3.9</td>
<td>Path costs for a route from A to D using our link cost metric.</td>
<td>55</td>
</tr>
<tr>
<td>3.10</td>
<td>Comparison of route selection; Source A, Destination E.</td>
<td>56</td>
</tr>
<tr>
<td>3.11</td>
<td>Packet loss ratio.</td>
<td>63</td>
</tr>
<tr>
<td>3.12</td>
<td>Packet loss ratio vs. node density with nodes moving at car speed.</td>
<td>64</td>
</tr>
<tr>
<td>3.13</td>
<td>Path length.</td>
<td>65</td>
</tr>
<tr>
<td>3.14</td>
<td>Route break probability.</td>
<td>66</td>
</tr>
<tr>
<td>3.15</td>
<td>Hybrid wireless mesh network with one default route out of the mesh and a single route to every mesh node.</td>
<td>68</td>
</tr>
<tr>
<td>3.16</td>
<td>Packet delivery ratio in the pedestrian scenario.</td>
<td>77</td>
</tr>
<tr>
<td>3.17</td>
<td>Packet delivery ratio in the car scenario.</td>
<td>78</td>
</tr>
<tr>
<td>3.18</td>
<td>Packet delivery ratio with an interval of packets to the Internet of 5 seconds.</td>
<td>79</td>
</tr>
<tr>
<td>3.19</td>
<td>Average number of routing control packets sent per second and per node.</td>
<td>80</td>
</tr>
<tr>
<td>3.20</td>
<td>Average size of routing packets.</td>
<td>81</td>
</tr>
<tr>
<td>3.21</td>
<td>Average routing overhead encapsulated.</td>
<td>82</td>
</tr>
<tr>
<td>3.22</td>
<td>Packet delivery ratio.</td>
<td>86</td>
</tr>
<tr>
<td>3.23</td>
<td>Demonstrator: User space daemon.</td>
<td>88</td>
</tr>
<tr>
<td>3.24</td>
<td>Demonstrator: Kernel space module for nodes.</td>
<td>89</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.25</td>
<td>Demonstrator: Kernel space module for gateways.</td>
<td>89</td>
</tr>
<tr>
<td>3.26</td>
<td>Validation of demonstrator in a static scenario with node failure.</td>
<td>90</td>
</tr>
<tr>
<td>3.27</td>
<td>Validation of demonstrator in a dynamic scenario.</td>
<td>90</td>
</tr>
<tr>
<td>4.1</td>
<td>A wireless mesh network connected to the Internet through different access networks.</td>
<td>94</td>
</tr>
<tr>
<td>4.2</td>
<td>The gateways inform a mobile node about its macro mobility or multihoming using Mobility Notification Messages (MNM).</td>
<td>97</td>
</tr>
<tr>
<td>4.3</td>
<td>Mobility notification message format.</td>
<td>98</td>
</tr>
<tr>
<td>4.4</td>
<td>A gateway detects a packet with a topologically incorrect routing prefix (1). It sends a mobility notification message to the sending node (2). This node then updates its address (3).</td>
<td>100</td>
</tr>
<tr>
<td>4.5</td>
<td>Algorithm for processing packets destined to the Internet at the relaying gateway.</td>
<td>101</td>
</tr>
<tr>
<td>4.6</td>
<td>Algorithm for handling mobility notification messages at a mobile node.</td>
<td>102</td>
</tr>
<tr>
<td>5.1</td>
<td>Vectorized street maps of four Swiss cities.</td>
<td>108</td>
</tr>
<tr>
<td>5.2</td>
<td>Road network and car traffic in north-eastern Switzerland.</td>
<td>110</td>
</tr>
<tr>
<td>5.3</td>
<td>Visualization examples of different inner city regions of Zurich.</td>
<td>111</td>
</tr>
<tr>
<td>5.4</td>
<td>Superposition of mobility pattern on a map of the center of Zurich.</td>
<td>112</td>
</tr>
<tr>
<td>5.5</td>
<td>Packet delivery ratio of AODV.</td>
<td>116</td>
</tr>
<tr>
<td>5.6</td>
<td>Packet delivery ratio of GPSR.</td>
<td>116</td>
</tr>
<tr>
<td>A.1</td>
<td>Overview of the relation between field theory and routing.</td>
<td>128</td>
</tr>
</tbody>
</table>
A.2 Screen shot of the visualization tool for field-based routing protocols. ........................................ 131
A.3 Parameter selection: Convergence time versus hello interval. .................................................. 133
A.4 Parameter selection: Per node convergence time versus hello interval. ................................. 133
A.5 Parameter selection: Convergence time versus delay of early hellos in backoff intervals. .......... 134
A.6 Parameter selection: Hellos sent per second versus delay of early hellos in backoff intervals. .... 134
List of Tables

3.1 Parameters of the robust path metric for the evaluation. ........................................ 60
3.2 Average route lifetime with nodes moving at car speed. ........................................ 66
3.3 Route set-up time of the AODV-CGA protocol. ........................................ 81
3.4 Qualitative comparison of evaluated routing schemes. ........................................ 83

5.1 Packet delivery ratio of AODV. ........................................ 115
5.2 Packet delivery ratio of GPSR. ........................................ 115
Chapter 1

Introduction

1.1 Motivation

Over the past years, wireless Internet access in cafés, hotels, and public places has become a commodity in urban environments. More recently, even mobile users such as pedestrians or car drivers demand ubiquitous network connectivity. However, permanent Internet access is not yet affordable for mobile devices. This indeed is the main motivation of our work: to enable inexpensive, city-wide Internet access. Today wireless Internet access is mainly provided either by cellular networks like GSM, UMTS, or by networks based on WiMAX or wireless LAN like 802.11. Unfortunately, these technologies are not suitable for providing inexpensive, city-wide Internet access: cellular networks cover large areas, but are prohibitively expensive; wirelesses LANs are inexpensive but their communication range is very limited.

Our envisioned scenario is an inexpensive, city-wide wireless mesh network based on wireless LAN technology. By wireless mesh network, we denote a multi-hop wireless access network. As shown in Fig. 1.1, such a network consists of fixed access points (providing Internet connectivity) and wireless mesh nodes
(fixed or mobile). In the remainder of this thesis, we call the access points Internet mesh gateways.

In this thesis, we specifically focus on mesh networks without a reliable, static backbone. The mesh nodes serve as relays, forwarding traffic for other mesh nodes to (and also from) the gateways and thus maintaining network-wide Internet connectivity. This approach makes optimal use of all mesh nodes and gateways. Given a high density of mesh nodes, ubiquitous Internet access can be provided by a small number of gateways.

To turn such a vision into reality the city-wide wireless mesh network has to fulfill two major requirements: (i) it must be scalable in size - with the number of participating nodes as well as with the number of gateways, and (ii) to support mobile users, the routing protocol must be robust with regards to node mobility and the network has to be able to handle node mobility.

**Figure 1.1:** A wireless mesh network consisting of mesh nodes and gateways.
1.2 Research Problems

In this thesis, we tackle the following research problems.

- **How to build scalable and robust wireless mesh networks?** When building large scale, city-wide wireless mesh networks, scalability to the number of nodes and gateways is a major concern. In addition, these networks have to be robust to node mobility for supporting mobile devices and ubiquitous communication.

- **How to manage node mobility in large wireless mesh networks?** Nodes want to be permanently reachable from the Internet and never loose their connections to peers. In large-scale wireless mesh networks, this demands for specific support because they consist of multiple access networks.

- **How to generate realistic mobility models for simulations of wireless mesh networks?** Today, simulations are the primary tool for assessing network solutions for large scale wireless mesh networks. To increase the credibility and predictability of these simulations, realistic mobility models have to be used.

1.3 Contributions

The main contribution of our work is the design, implementation and evaluation of a scalable and robust routing solution for wireless mesh networks. The detailed contributions are listed below. We also list the conferences where each contribution has been published.

- **HEAT**, a scalable field-based routing protocol for routing in wireless mesh networks [BHLM07b, BHLM07c]. HEAT
considers the availability of relaying nodes to the gateways for constructing more robust routes. The comparison of HEAT with two established routing protocols, one reactive and one proactive, shows that in city-wide wireless mesh networks with nodes moving at car speed HEAT outperforms these protocols.

- A quantitative comparison of three routing protocols as well as two extensions to route packets into wireless mesh networks [BHLM07c]. The comparison show that gateway source routing scales to the number of nodes and achieves high performance.

- A path metric to construct robust routes by preferring paths consisting of neighbors close to a preferred signal strength indicator [NBG06, BHL07a]. Compared to shortest paths, the paths selected by this metric exhibit a lower break probability especially for nodes moving at car speed.

- A fully functional demonstrator of the proposed scalable and robust routing solution for wireless mesh networks on Linux.

- A detection and notification protocol for managing macro mobility and multihoming in wireless mesh networks [BBH+06, BBHM07]. It enables mobile nodes to use mobility management protocols like MobileIP or HIP independent of the deployed routing service.

- Realistic mobility models for simulations of wireless mesh networks using the Swiss geographic information system as well as a leading multi-agent microscopic traffic simulator [NBG06, BHM07]. The evaluation shows that performance results of wireless mesh networks depend highly on the
underlying mobility model and that simple random waypoint mobility models tends to considerably overestimate the performance of routing protocols.

1.4 Outline

This thesis is structured as following:

- **Chapter 2** reviews related work and highlights the novel aspects of this thesis compared to previous work.

- **Chapter 3** introduces our scalable and robust routing solution for routing packets toward the gateways and back. In addition, we describe a robust path metric for wireless networks. We present detailed evaluations and comparisons of our proposals.

- **Chapter 4** describes our detection and notification protocol for macro mobility and multihoming.

- **Chapter 5** presents two mobility models that we have developed for realistic evaluation by simulation of our proposals, a GIS-data and a traffic simulator based model.

- **Chapter 6** concludes this thesis and provides an outlook on future work.
Chapter 2

Related Work

In this chapter, we review the related work as well as background, that has influenced our work. In addition, we highlight the differences to our work. First we have a brief look at unicast routing in wireless ad hoc and mesh networks. Then, we present approaches that use the concept of fields or gradients for routing. Afterward, we summarize previous work on anycasting. Following, we survey related work on robust path metrics; especially on signal strength-based routing algorithms. Finally, we give an overview on mobility management protocols and mobility models.

2.1 Unicast Routing Protocols for Ad Hoc Networks

In recent years, countless different ad hoc routing protocols have been proposed, as for example DSDV [PB94], OSLR [CJ03], TBRPF [OLT03], AODV [PBRD02], GPSR [KK00], DSR [JMHJ02], GPSR [KK00], and ZRP [HPS02] to name just a few. These routing protocols have been surveyed and classified in many publications as for example [Uni06, ZRM02,
Lan03, HDL⁺02, MTE04]. Instead of presenting yet another survey on ad hoc routing protocols, we briefly list three primarily characteristics of these protocols related to our work.

1. *Initiation of route determination*: reactive or proactive.

2. *Scope of topology information exchange*: locally with neighbors (distance-vector) or globally with all nodes (link-state).


## 2.2 Routing Protocols for Wireless Mesh Networks

By wireless mesh networks, we denote multi-hop wireless access networks. Specifically in this thesis, we focus on mesh networks without a reliable, static backbone. Specific to these networks is that the data traffic is between a large number of wireless mesh nodes and a few gateways with Internet connectivity. For assessing such networks many different metrics can be used. But in contrast to common wireless networks, the scalability and robustness is a major concern. Thus, in this thesis we focus on the following metrics.

- Scalability to the number of nodes and gateways
- Robustness to node mobility and link failures
- Packet delivery or loss ratio
- Routing overhead
- Route set-up and convergence time
- Route break probability
- Length of routes
Most routing protocols for wireless mesh networks are extended unicast ad hoc routing protocols [CLCH05, HWLC05, AZ03, RK03, SBRP02, MB05, BWJ06, Bau04]. They have been enhanced with gateway discovery functionality to allow their use for wireless mesh networks. Thus, they do not take advantage of the fact that all gateways offer access to the Internet and establish for every mesh node a unicast route to one specific gateway. As a consequence, these extended protocols do neither scale to the number of nodes nor to the number of gateways. For example Nordström et al. proposed to use source routing based on DSR to route packets between wireless mesh nodes and multiple gateways [NGT04]. There, the wireless mesh gateways specify the routes using previously gained knowledge on the network. In OLSR [CJ03], it is specified that inter operation with other networks should be possible by injecting external route information into the OLSR network. Since all of these protocols provide unicast routes, for every mesh node an individual route must be maintained to one of the gateways. Therefore, these protocols scale poorly to the number of nodes as well as gateways in the mesh network.

In [CYK06], scalability to the number of mesh nodes is improved with the use of location information; however, this kind of information is typically not available in our target scenario where the mobile nodes in the mesh network are commodity laptops or hand-held devices.

The authors of [JR06] propose a single-hop client network where mobile clients connect directly to a mesh backbone of gateways. Compared to our approach, where mobile users also relay packets on behalf of their neighbors, this requires a much higher gateway density for a comparable wireless coverage. There are also some efforts ongoing to put these capabilities into the MAC Layer [AHR04].

The task group for mesh networking of the IEEE 802.11 working group goes in the same direction. In its first draft
[IEE06, Bah06], they propose to implement routing on the MAC Layer. According to [HBC04], their target size of an IEEE 802.11s WLAN mesh network is up to 32 static mesh points. In particular, the 802.11s task group specifies a default mandatory routing protocol named hybrid wireless mesh protocol that is inspired by a combination of AODV and tree-based routing. In addition, the draft allows further standardized or vendor specific path selection protocols. Up to now, the only alternative protocol described in the draft is the radio aware optimized link state routing protocol.

A problem that is common to most mesh routing protocols is that gateway announcements or gateway requests are prone to vanish due to route breaks, and the recovery procedure is often as expensive as establishing a new route. In contrast, [MGLA06] proposes an efficient mechanism to fix broken routes locally. We also go into this direction. With our approach, all control messages are local and routes can easily be repaired locally. Moreover, since routing is based on scalar fields and not on route entries, alternative routes can most of the time be determined without additional overhead when the primary route fails.

Mosko et al. [MGLA05], propose to establish multiple non-disjoint paths for better performance, but again the established routes are unicast and this protocol is not scalable to the number of mesh nodes as well as gateways.

Summarizing, there is a lack of a specialized routing protocol for wireless mesh networks. This routing protocol should take advantage of the specific setting that all traffic is to or from the gateways. Especially, the protocol should be scalable to the number of nodes as well as to the number of gateways and be robust to node mobility.
2.3 Field- and Gradient-based Routing

Most routing protocols fall into two main categories: distance vector routing or link state routing. In distance vector protocols, neighbors periodically exchange information about topology changes and determine best paths based on a distance metric, as for example BGP [RL95], RIP [Mal98], DSDV [PB94], or AODV [PBRD02]. In link state routing protocols, all nodes exchange information about the complete set of links that exist in the network, as for example OSPF [Moy98], OSLR [CJ03]. Thus every node has a complete view of the network, which enables them to compute the optimal path locally.

In this thesis, we present HEAT, a routing protocol that is related to a specific group of distance-vector routing protocols based on the concept of fields, so called field-based or gradient-based routing protocols. Following, we present other works that use fields or gradients for routing in networks, and explain how these works differ from our proposals.

2.3.1 TORA

Temporally Ordered Routing Algorithm (TORA) [PC97] is one of the most popular routing protocols for mobile wireless ad hoc networks. It has been proposed by Vincent Park and Scott Corson in 1997. TORA does not uses shortest path routing but tries find a somehow optimal path. It establishes a directed acyclic graph (DAG) rooted at the destination node. This directed acyclic graph can be viewed as a field. The graph is constructed by assigning a "height" (or a field value as we call it in our work) to each node and then determining the steepest ascent by looking at the relative height between neighboring nodes. The protocol tries to minimize routing overhead, and for this purpose, applies a technique called link reversal to repair broken links in the DAG as nodes move. Link reversal consists of reversing
individual links in the directed acyclic graph until it is routed at the destination again.

TORA is designed for unicast routing and operates only on directed acyclic graphs (or fields) with a single destination node. But for taking advantage of the fact that there are multiple gateways offering access to the Internet multiple destinations would be required. But the authors do not show how to extend the model for multiple destinations and it remains unclear if the proposed protocol as such can be extended for this purpose.

2.3.2 AntNet

In 1997, Gianni Di Caro and Marco Dorigo has proposed AntNet [CD97], a routing algorithm inspired from the behavior of ants. The algorithm mimics the way real ants find shortest paths from their nest to food by following a pheromone trail (or gradient) deposited by other ants. At the beginning, when no pheromone trails are available, the ants move randomly. The algorithm starts to converge as soon as one ant has discovered a path to the desired destination. Ant-based routing belongs to the class of statistical unicast routing algorithm since it introduces randomness to find routes. In our work, fields and gradients are established in a deterministic way and used for anycast routing.

2.3.3 Directed Diffusion

Directed Diffusion [IGE00] has been proposed in 2000 by Chalermek Intanagonwiwat et. al. as a novel communication paradigm for sensor networks. Sensor data is tagged and ”diffuses” according to pre-established interest gradients toward the sinks (nodes that are interested in the data). Directed Diffusion supports multiple destinations but the goal of the system is to deliver the data to all of the interested nodes (multicast). In our
work, we focus on delivering data to the best destination node (anycast).

2.3.4 Routing using Potentials

Routing using Potentials has been proposed in 2003 by Anindya Basu et. al. [BLR03]. They aim on developing a dynamic, traffic-aware routing algorithm for the Internet. Their main idea is to avoid congested areas in the network by modeling the queue length of routers as an input for a potential value of a potential field. In their case, steepest gradient routing results in routing through routers that are less congested. However, the work of Basu et. al. focuses on unicast in fixed wired networks. We focus on anycast routing in wireless mesh networks and use different field establishment and calculation methods.

2.3.5 Packetostatics

Packetostatics [TT05] uses electrostatic fields to route flows in sensor networks. They show that the optimal distribution of nodes induces a traffic flow identical to the electrostatic field that would exist if the sources and sinks of traffic were substituted with an appropriate distribution of electric charge. The authors of this work assume a massively dense sensor network that justifies their analysis using the continuous electrostatic field. Their work is focused on finding the optimal node placement given a set of flows, where our work is really about routing packets given a dynamic network topology.

2.3.6 Field-based anycasting

Lenders et al. propose a routing model based on the concept of potential fields [LMP05], more specifically the electrostatic potential field. This model initially aimed at service discovery and
has later been generalized for anycast routing. In their model, the services (or destinations) flood announcements through the network, including a hop counter. Using these announcements, a node calculates its potential value for a specific service by a linear superposition of the potential fields from all nodes offering this service.

In [LMP06], the authors show how to use the proposed anycast model to route packets toward a network area with a high density of group members (density-based routing), in contrast to existing routing techniques, that only consider the distance information (proximity-based routing).

In our work, we use the same basic idea of field-based anycasting and apply it to wireless mesh networks. We model the gateways offering Internet access as an anycast group. In contrast to the previous work of Lenders et al., we propose a novel way of field propagation and calculation which is based on an analogy from thermal physics. Lenders et al. propagate the field by periodically flooding announcements of every anycast group member through the network and thus is not scalable to the size of the anycast group. In contrast, our field propagation mechanism is based on a local exchange mechanism that only propagates the actual field intensity of a node and thus is scalable to the size of the anycast group as well as to the number of nodes in the wireless mesh network. For calculating the field intensity of a node, Lenders et al. calculate a potential value at the node for every known member of an anycast group based on the distance in hops accumulated in the announcements. These per group member potential values are then accumulated by a linear superposition resulting to the node’s field intensity. This superposition may lead to unintended local maxima. In contrast in our approach, a node calculates its field intensity solely based on the field intensities of its neighbors. By design, our field calculation algorithm guarantees that there are no local maxima. In addition, the algorithm takes into account the degree of node
redundancy. Thus, it enables to prefer routes through network areas with high redundancies. We call this routing strategy conductivity-based routing.

2.4 Anycast Routing Protocols

Anycast is a packet delivery scheme whereby a packet is forwarded to one of a group of hosts according to some routing protocol estimations. In this section, we first discuss research efforts around anycast routing for the Internet and for wireless networks.

2.4.1 In the Internet

In 1993, IP anycast for the Internet has been proposed as an RFC [PMM93] to simplify the task of finding an appropriate server when several servers support similar services. They propose to assign an unicast IP address to multiple hosts, and advertising it into the routing infrastructure from all the hosts that share the same address. Later, it was incorporated into the IPv6 addressing architecture [DH98].

The reasons why IP anycast is not widely deployed nowadays in the Internet is mainly due to two problems. IP anycast, as originally proposed, does not scale well and is hard to deploy in the core of the Internet at a large scale. As an attempt, Katabi et. al. proposed GIA [KW00], a more scalable IP architecture. However, this approach requires a change in the core Internet routers and puts a severe dent on the practical appeal of the approach. For this reason, various research groups proposed application-layer solutions [ZAFB00, SAZ+02, BF05].

2.4.2 In Wireless Networks

In [PM99a] and [PM99b], Vincent Park et. al. describe how to enable anycasting in wireless networks by extending well known
unicast routing techniques such as link state and distance vector routing. Jianxin Wang et al. propose in [WZJ03b] and [WZJ03a] to extend AODV [PBRD02] and DSR [JMHJ02] respectively, to support anycast delivery in mobile ad hoc networks. All of these anycast proposals use unicast routing as an underlining protocol, and thus are not scalable to the size of the anycast group because for every member proper routes have to be maintained. In Subsection 2.3.6, we discuss an anycast routing protocol based on electrostatic potential fields. But also this protocol is not scalable to the size of the anycast group because every member of the group periodically floods an announcement through the network.

In this thesis, we introduce a new anycast routing protocol that scales completely to the size of the network and the size of the anycast group. In addition to the anycast routing protocol presented in Subsection 2.3.6, we propose a new protocol based on the path resilience, conductivity-based routing.

2.5 Robust Path Construction

Even in fixed wireless networks, the bad performance of shortest path routing has been reported in different contexts. For example, Aguayo et al. [ABB04] observe that in Roofnet, a fixed wireless mesh network, the packet delivery ratios of links are typically well distributed between 0 and 1. As a consequence, a minimum hop count metric often chooses lossy links that cause a lot of retransmissions and degrade the end-to-end performance. Yarvis et al. [YCK02] observe that hop count performs poorly as a routing metric for sensor networks.

One way to discriminate links is to consider the packet loss ratio. De Couto et al. proposed ETX [CABM03], a metric that maximizes the expected transmission count instead of minimizing the hop count. Draves et al. [DPZ04] compared this metric in a
static wireless indoor network and confirmed an increased end-to-end throughput. Others have proposed to completely avoid links with a loss ratio above a desired threshold [LWM06].

Aside from these metrics based on measured packet loss ratio, several signal strength-based routing protocols have been proposed. Examples are ABR [Toh97], SSR [DRWT97], and RABR [AASS00]. The common idea of these protocols is to prefer stable links over transient ones. Therefore, these protocols select paths that are based on links that had a strong signal strength for a long time in the past. In contrast, our link cost metric assigns lowest cost to links with a received signal strength indicator that is closest to a certain value.

Cheng and Heinzelman [CH04] explore different design trade-offs to maximize the route lifetime in ad hoc networks. However, they do not propose specific mechanisms or metrics that can be applied by a routing protocol.

An alternative way to cope with frequent path failures is to maintain alternative routes such that end-to-end connectivity can be re-established quickly. Preemptive routing [GAGPK01] and pro-active route maintenance [QK02] have been proposed to determine an alternative route when a link is expected to break (e.g., indicated by low received signal strength indicator) on an active route. Multi-path routing protocols [LG01, YKT03, MD01] establish multiple routes to a destination and thus have immediately an alternative route when the active route breaks. All these approaches are complementary to the metric we propose in this thesis because also these routing protocols require a metric to determine path costs. Thus they might well benefit from using our metric instead of the common shortest path metric.

In [NBG06], we propose to use the received signal strength indicator for a discrete rating scheme of preferential links. This idea was employed for enhancing the flooding of routing messages. In contrast, we propose in this thesis a continuous link rating
scheme and incorporate this scheme into a routing metric to increase route robustness in mobile wireless networks.

2.6 Mobility Management Protocols

IP mobility management protocols allow mobile nodes to move from one network to another without changing their IP addresses. This enables them to maintain transport and higher-layer connections while moving around.

There are two IP mobility management protocols proposed by the IETF for enabling macro mobility in IPv6: MobileIPv6 [JPA04] and the Host Identification Protocol (HIP) [MN06]. Both protocols maintain a fixed proxy (home agent / rendezvous server), a host that is aware of the current location and address of a node. This architecture enables permanent reachability even with mobile nodes. MobileIPv6 and HIP also offer an address change notification mechanism to preserve established transport sessions in the presence of macro mobility.

The main difference between them is that HIP decouples the dual roles of IP address while MobileIP does not. In HIP networks, the host identifiers take over the naming role, while IP addresses become pure locators. As a consequence, the transport-layer mechanisms operate on host identities instead of using IP addresses as endpoint names. In addition with HIP, several rendezvous servers may be used in parallel and there are more advanced security mechanisms for the handshakes.

For HIP and MobileIP, extensions to enable multihoming are proposed, that allow to attach nodes simultaneously to multiple networks [Hen06,FT03,EN06]. Note however that these two IP mobility management protocols and their extensions for multihoming require that a node explicitly knows the access networks over which its packets are forwarded to the Internet.
This knowledge allows a node to deal with its macro mobility or to maintain its multihoming.

To deal with macro mobility, a moving node updates its address to topologically fit to the access network relaying its packets and notifies its fixed proxy as well as its communication peers about its address change. Also for multihoming, a node has to inform its communication peer about additional or outdated locators. For this purpose, each node has to maintain a list of access networks currently used.

2.7 Mobility Models

The analysis of related literature reveals that the results of performance studies of ad hoc networks heavily depend on the mobility model chosen [YLN03, BRS03, CBD02, CaEB05]. We distinguish two categories of mobility models. First, there are simple mobility models like random waypoint [BMJ+98], random directions [RMSM01], reference point group [HGPC99], Manhattan grid [3GP98], and other models where the nodes change their speed and direction randomly [CBD02]. Second, there are more complex models where the movement and behavior of nodes are modeled closer to reality [SJ04, LHT+03, CaEB05, RCV+03, RVC+02, NBG06]. Results obtained with models from different category often differ significantly. This discrepancy raises the question whether the simple models are consistent enough with reality.

One realistic modeling approach would be to use mobility traces. However, no traces are available for pedestrians or vehicles moving around in cities. There are traces for buses of public transport systems [JHP+03], but these do not necessarily match the mobility of pedestrians or cars. As a result, the mobility of vehicles is often approximated by random mobility models configured with higher maximum node speed [MJK+00].
Sawant et al. [STY05] describe vehicular movement on a highway by simulating a set of nodes moving at a constant speed along the longer side of a rectangular area of $1000\text{m} \times 100\text{m}$. A slightly more advanced model is used in [CKV01], where the decisions of the car drivers are included in the mobility model.

Saha and Johnson proposed in [SJ04] to use street maps from the TIGER database to generate quite realistic mobility patterns. In [MHd+07] Morge et al. extend this models with more sophisticated node density and speed distributions. However, their model neither follows the exact geometry of the streets nor does it incorporate actual speed limits. In Section 5.2 we present a new mobility model that is based on exact vectorized street maps of major Swiss cities with complete speed limit information.

Lochert et al. [LHT+03] use the traffic flow simulator Videlio developed by Daimler Chrysler AG to create a movement pattern for a small part ($6.25\text{km} \times 3.45\text{km}$) of the city of Berlin. No information is given on how the density/distribution of vehicles were chosen. The authors consider only one area of the city and 10 data sources that are active for only 5 seconds. Choffnes et al. proposed STRAW [CaEB05], a simple microscopic traffic simulator based on the TIGER Street maps. In Section 5.3, we describe a source of realistic vehicular mobility traces that we proposed in [NBG06]. These traces are obtained from a multi-agent microscopic traffic simulator that was developed by Kai Nagel (at ETH Zurich, now with the Technical University in Berlin, Germany). In contrast to STRAW, this simulator is capable of simulating public and private vehicular traffic over real regional road maps of Switzerland with a high level of realism [RCV+03, RVC+02]. The multi-agent core enables accurate reproduction of the behavior of the individuals, such as the choice of means of transportation. Moreover, the start time of a trip and its destination is considered for every single person in the simulation.
Chapter 3

Scalable and Robust Routing

3.1 Introduction

In this chapter we address the problem of routing in city-wide wireless mesh networks. Specific to these networks is that the data traffic is between a large number of wireless mesh nodes and a few gateways with Internet connectivity. For building such large networks, a routing protocol must be capable to deal with a huge number of mesh nodes distributed over a wide area with many gateways. Thus it must be absolutely scalable to the number of mesh nodes as well as to the number of gateways. In addition, most of these mesh nodes are mobile and move around. Thus, a routing protocol should be able to deal with the mobility of the nodes and be highly robust to it. Existing unicast routing protocols like for example AODV, OLSR or TORA are not well suited for such wireless mesh networks because these protocols are designed for node to node communication. They maintain proper routes to every gateway and mesh node. As a consequence they
do neither scale to the number of mesh nodes nor to the number of Internet gateways.

When analyzing the problem of routing between mesh nodes and gateways, we found that it is best to distinguish between the problem of routing packets toward the gateways and back to the mesh nodes.

Routing from mesh nodes to a gateway is similar to an anycast routing problem with a single service, the bridging to the Internet. In Section 3.2, we present HEAT, an anycast routing protocol for wireless mesh networks particularly scalable for this type of communication. HEAT relies on a temperature field to route data packets to available Internet gateways. That is, every node is assigned a temperature value and packets are routed along increasing temperatures until they reach a source of heat, which corresponds to finding any Internet gateway. The distinguishing feature of our protocol is that it does not require flooding of control messages. Rather, every node in the network determines its temperature considering only the temperature of its direct neighbors, which renders our protocol particularly scalable to the network size. Using simulations, we analyze our approach and compare its performance with AODV and OLSR. Our results clearly show the benefit of HEAT versus AODV and OLSR in terms of scalability to the number of nodes and robustness to node mobility. The packet delivery ratio with HEAT is more than two times higher than with AODV or OLSR in large-scale mobile scenarios.

Routing packets into a wireless mesh network is different from routing packets out, since a proper route has to be maintained for every single node in the mesh. In Section 3.4, we compare three protocols for routing packets into wireless mesh networks using simulations. Each of these protocols represents a routing protocol family: (i) AODV with an extension for mesh networks, a reactive routing protocol, (ii) FBR, a proactive routing protocol, and (iii) GSR, a source routing protocol. Our results indicate that
3.1 Introduction

FBR has the highest packet delivery ratio but is not scalable to the network size. The extended AODV seems to be neither scalable nor does it achieve a high packet delivery ratio. A good compromise is provided by GSR, which is the most scalable to the network size and still achieves a high packet delivery ratio.

A common problem in wireless mesh networks are route breaks due to node mobility. In order to allow the selection of robust paths, we develop a path metric that considers the robustness of a path. This path metric is presented in Section 3.3. The goal of this path metric is to select paths that offer a good trade-off between long routes consisting of many reliable links and short routes consisting of only few but unreliable links. In contrast to the widely used minimum hop count metric, which chooses arbitrarily among those paths with minimum length, our path metric estimates the reliability of a link based on the measured Received Signal Strength Indication (RSSI). Every node determines a preferred signal strength indicator and assigns each neighbor a link cost depending on how close to this strength its RSSI value is.

In each section, we present a detailed simulation study to show the advantages of our proposals. In Section 3.5, we present a simulation study for the complete routing solution using realistic mobility patterns described in Section 5.2. Our results clearly show the benefit of HEAT versus AODV and OLSR in terms of scalability and robustness for city-wide wireless mesh networks. In Section 3.6, we present a proof of concept implementation of our routing solution in a demonstrator for Linux.

In the appendix, we provide additional information about the terminology of field-based routing and the support of HEAT for different routing metrics. In addition, we present a visualization tool for field-based routing.
3.2 Forward Path: Routing Packets Toward the Gateways

The particular problem we address in this section is how to route data packets from the mesh network to the Internet. The routing includes traffic from the mobile nodes to any gateway. This routing problem is different from what MANET routing protocols like AODV [PBRD02], OLSR [CJ03], DSR [JMHJ02], DSDV [PB94], or ZRP [HPS02] were originally designed for. These protocols were designed to provide end-to-end paths between two communicating hosts (unicast-type of communication). However, in our work, routing is from any mobile node in the network to any Internet gateway (anycast). In principle, unicast routing protocols can be applied, however, they scale poorly in terms of communication overhead since they establish an individual path per node-gateway pair.

In this section we present HEAT, an anycast routing protocol for wireless mesh networks. As its name suggests, HEAT is inspired by the heat conduction in physics. That is, we model the gateways as heat sources which create a temperature field in the network. The higher the temperature of a node, the closer it is to a gateway. Using these fields, packet forwarding is fairly simple: packets are forwarded along the nodes with the highest temperature until they eventually reach any heat source (an Internet gateway). Our protocol to establish temperature fields in the network is purely based on local information. It means that every node calculates its own temperature by only evaluating the temperature of its direct neighbors. This makes our protocol particularly scalable since no flooding of messages is required.

We compare using simulations with Glomosim the performance of HEAT with AODV and OLSR. AODV is a representative reactive unicast MANET routing protocol. OLSR is a popular proactive protocol. Our simulations show that HEAT
scales much better than AODV and better than OLSR to the network size, the node density, the number of gateways, and the node mobility. For example in large or/and dense networks HEAT outperforms AODV and OLSR by more than a factor of two in terms of successful packet delivery. Furthermore, the packet delivery ratio of HEAT remains above 95% even for large and dense networks with nodes moving at typical pedestrian speeds.

The rest of this section is organized as follows. In the next subsection, we explain the concept and implementation of HEAT. Then, we present our evaluation study and finally, we give a conclusion.

3.2.1 Concept and Implementation of the HEAT Protocol

We propose to apply anycast to provide scalable routing in wireless mesh networks. In this subsection, we first present our general anycast routing concept based on temperature fields. Then, we describe our protocol implementation.

Concept of Routing using Temperature Fields

In this subsection, we present HEAT, a novel method to construct the scalar field for field-based routing. HEAT has two distinguishing features. Firstly, it considers both the length and the robustness of paths in the routing decision. Secondly, the field construction and maintenance mechanism of HEAT scales to the number of nodes and the number of gateways since it only requires communication among neighboring nodes.

The HEAT algorithm is a fully distributed, proactive anycast routing algorithm. It is inspired by the properties of temperature fields, as discussed in the next subsection. In brief, our algorithm assigns a temperature value to every node in the mesh network.
New nodes are assigned a value of zero; gateway nodes are assigned the maximum value. In contrast to strict shortest-path routing, HEAT determines the temperature value of a node based on (i) its distances to available gateways but also based on (ii) the robustness of the paths toward these gateways. That is, a path providing multiple alternative delivery opportunities along its way is preferred to a path over which packets cannot be naturally re-routed to alternative nodes toward the destination. An illustrative example is depicted in Fig. 3.1. The part of the network leading to the gateway on the right hand side has more links than the left part of the network leading to the gateway on the left. As a result, the temperature gradient toward the gateway on the right is steeper and packets are routed in this direction, even if the network distance to the left gateway is shorter (measured in the number of hops).

Nodes calculate their temperature based solely on the temperature values of their neighbors, which they learn through periodically broadcast messages. Data packets are then routed along the steepest gradient and finally reach a gateway.

**Analogy to Temperature Fields**

As mentioned, HEAT is inspired from temperature fields in thermal physics. A temperature field assigns a single scalar value to every particle in space. The values of the temperature field are higher in the vicinity of heat sources and decrease with the distance to the source.

In a solid, heat is transferred by conduction. On a microscopic scale, conduction presents itself as hot, rapidly moving or vibrating atoms and molecules. By interactions among neighboring atoms and molecules, heat is transferred. The physical parameter *thermal conductivity*, $\kappa$ indicates its ability to conduct heat. The conduction of heat is governed by *Fourier’s Law*. In essence, this law demands that the temperature of the field always decrease
3.2 Forward Path: Routing Packets Toward the Gateways

![Diagram of mesh network with field intensity and forwarding direction]

**Figure 3.1:** Example of a temperature field with a conductivity value of $\kappa = 1/4$ and areas of different link redundancies. The packets of the node with temperature value 0.051 are forwarded to the right, across an area of high redundancy instead of to the closest gateway at the left.

...away from sources, resulting in a temperature gradient whose maxima are at the source.

In order to map the properties of temperature fields to a given network topology, we consider nodes in the mesh network as particles and gateways as heat sources. In [LMP06], Lenders et al. show that under the assumption that there are no local maxima in the field, following the path defined by the steepest gradient always leads to a gateway; and that there are no loops in this path. However, not all policies for assigning scalars to nodes guarantee that there are no local maxima in the potential field. In our approach, we avoid local maxima by adhering to the following policy: For every node, only neighbors with a higher temperature...
may contribute to the own temperature. This policy guarantees monotonicity of the field and thus ensures that there are no local maxima.

**HEAT Protocol**

We describe in this subsection our protocol for routing using temperature fields. According to the concept described before, the gateways supply a temperature field that enables routing from the mesh nodes to the Internet gateways. The temperature values of neighbors, which are required by *HEAT* for constructing the temperature field, are periodically exchanged between neighboring mesh nodes by *HEAT beacon* messages. Based on these messages, every mesh node calculates its own temperature using the same function.

Once the field is constructed, routing packets from the mesh nodes to the gateways is straightforward and implemented on a hop-by-hop basis: A packet is always forwarded to the neighbor with the highest temperature, resulting in steepest-gradient routing. Routing of packets back from the gateways to the mesh nodes is achieved by recording the paths that packets have taken when following the steepest gradient and using the reverse path.

**Temperature Field Construction and Maintenance**

As mentioned before, the sources of the temperature field are the gateways. Therefore, each gateway initializes its temperature with a certain maximum value. As the gateway receives more and more traffic, it may adjust its temperature value according to its current load level. This allows the gateway to avoid congestion by controlling, how much traffic it receives. The dynamic adjustment of gateway temperature values requires special attention as oscillations of the temperature field must be avoided under all circumstances. For the heat propagation,
every node (including the gateway nodes) periodically broadcasts its temperature value to its neighbors with a given HEAT beacon time interval. Based on these messages, all nodes build and maintain a data structure called neighbor table, which contains an entry for every known neighbor. Neighbor entries comprise the address, the last reported temperature, and a timestamp value of the corresponding node. Whenever an entry is added, removed, or changed, the temperature value is re-computed. In essence, we have to differentiate among three cases:

1. **New neighbor**: If a beacon from an unknown neighbor is received, a corresponding entry is added to the neighbor table.

2. **Maintain neighbor**: If the reported temperature value of a known neighbor changes, the node re-calculates its temperature value.

3. **Missing neighbor**: If no beacon is received from a neighbor for a certain period, its entry is removed and the temperature value is re-computed.

The key idea of HEAT is to provide scalability (with regard to protocol overhead) and robustness (with regard to link and node failures). Due to the local message exchanges, our method scales with the number of neighbors per node (scalability and convergence of the implementation are evaluated in Section 3.2.2). Robustness is achieved by assigning the temperature values such that routes through network areas with high redundancy (in terms of node and link redundancy) are preferred. The more neighbors with high temperatures, the higher is the temperature of a given node. The detailed algorithm is described in Alg. 1. The algorithm calculates the temperature $t_{\text{final}}$ of a node as follows: In a first step, the node sorts its neighbors based on their temperatures $\theta_i, i \in \{0, \ldots, n\}$ in ascending order (line 1) into an array $a$. Then, it iterates over $a$ accumulating the temperature of
the next neighbor to the sum of the temperatures of the previous neighbors $t_j$ until the temperature of the next neighbor is less than the accumulated temperature (line 4). In each step $j$, the value $t_{j+1}$ is calculated as follows (line 5): The difference between the temperature of the currently considered neighbor, denoted by $a[j]$, and the temperature accumulated so far, $t_j$, is calculated. Then, this difference is multiplied by the conductivity parameter $\kappa$, and the result is added to the temperature accumulated so far, denoted by $t_j$.

\textbf{Algorithm 1} Temperature field calculation function.
1: $a = \text{sort}_{\text{ascending}}(\theta_0, \ldots, \theta_n)$
2: $j = 0$
3: $t_j = 0$
4: while $t_j < a[j]$ do
5: \hspace{1em} $t_{j+1} = t_j + (a[j] - t_j) \cdot \kappa$
6: \hspace{1em} $j = j + 1$
7: end while
8: $t_{\text{final}} = t_j$

For completeness, a closed formula of Alg. 1 is given in Eqn. 3.1 and 3.2 with $\varepsilon$ being an arbitrarily small positive number.

$$t_l = \kappa \cdot \sum_{j=0}^{l} (1 - \kappa)^j \cdot a[l - j], \quad t_{\text{final}} = t_m$$ (3.1)

$$m = \arg\min_{l \in \{0, \ldots, n-1\}} (\max\{a[l+1] - t_l, l \cdot \varepsilon\}), \quad \varepsilon > 0 \text{ small}$$ (3.2)

As a result, nodes which have many neighbors toward the gateways obtain higher temperatures than links with only a small number of neighbors toward gateways. This effect is more pronounced the smaller the parameter $\kappa$ is chosen. Figure 3.1 illustrates an example temperature field with $\kappa = 1/4$. As $\kappa$ is set to a rather small value, one recognizes that the high link redundancy in the right half of the network is taken into account.
A step-by-step example of the field calculation function is given in Fig. 3.2 for \( \kappa = 1/4 \).

![Diagram showing Step 1 to Step 5 calculations]

**Figure 3.2:** Example of the temperature field calculation with a conductivity value of \( \kappa = 1/4 \) for node 53: step 1, sort neighbors (nbr) by temperature value; step 2-5 iterate down the table until the given temperature value of the node is higher or equal to the next neighbor; node 71 and 17 do not contribute to the temperature value of node 53: they will increase their values after the next HEAT beacon message of node 53 (node 77: 0.313 and node 17: 0.118)
Improvements for Better Convergence

New node arrivals are easy to handle in our approach and do not require a long time until the temperature field has converged to its steady state: A node joining the network sets its temperature value to 0 and learns a route to the Internet with the first arriving *HEAT beacon*. As more *beacons* arrive, the temperature is adjusted until it converges to its final value. In the meantime, the new node does not interfere with the temperature field.

Disappearing nodes (e.g., due to mobility) usually have none or only a local interference with the temperature field. Also if the neighbor with the steepest gradient becomes unavailable, a node immediately switches to the neighbor with the second steepest gradient. But infrequently, disappearing nodes may cause individual gateways to become unreachable or parts of the network partitioned. These type of events might take longer until the temperature fields has converged and in the worst case, some mesh node might not be able to reach any gateways during such period. For this purpose, we propose two additional mechanisms that aim at reducing the convergence time of the temperature field.

The first mechanism consists of an *early HEAT beacon* that is used when nodes detect that a neighbor is no longer reachable and that this neighbor departure has a significant influence on the temperature of that node.

To avoid extensive overhead caused by such *early HEAT beacons*, each node is allowed to delay the forwarding of such *early HEAT beacons* for a short period (e.g., a few broadcast intervals) to ensure that multiple *early HEAT beacon* triggered by the same event are aggregated at the relaying nodes.

If a node leaves unexpectedly, the node departure is detected by one of the neighboring nodes by its maintenance procedure of its neighbor table. If its own temperature value is affected,
the node informs its neighboring nodes again using *early HEAT beacon*.

The second mechanism is to avoid a possible effect we call “swing down” of temperatures that is similar to the count to infinity problem [Hui95]. This effect occurs for instance in the following scenario. We look at three mesh nodes. Two of them, \( \text{node}_a \) and \( \text{node}_c \), are neighbors of the third node, \( \text{node}_b \). Assume that \( \text{node}_a \) is close to a gateway and the temperature value of \( \text{node}_b \) is heavily influenced by \( \text{node}_a \) and the value of \( \text{node}_c \) is influenced by \( \text{node}_b \). In this scenario, a failure of the link between \( \text{node}_a \) and \( \text{node}_b \) should lead to a substantial decrease of the temperature of \( \text{node}_b \) and an even heftier change at \( \text{node}_c \).

However, the following problem arises. Since only \( \text{node}_b \) notices the link failure, it re-calculates its temperature value first and incorporates the high temperature of its neighbor \( \text{node}_c \). In the next step, \( \text{node}_c \) decreases its temperature value also, and this game goes on and on until both nodes have their correct temperature values. In order to avoid such expensive back and forth adaptation, we use the following technique that is similar to poison reverse [Mal98]. We add the identifiers of the contributing neighbors to the temperature value in each *HEAT beacon*. Using these identifiers, the *HEAT* algorithm at a particular node then ignores all temperature values from neighbors that derived their temperature from this node.

### 3.2.2 Simulation Study

To evaluate the performance and scalability of our approach, we performed simulations with Glomosim [ZBG98], a network simulator for wireless networks. We implemented the complete *HEAT* protocol in Glomosim. The relevant parameters of our protocol are described in Appendix A.4. These include the *HELLO beacon* interval, the *HELLO beacon* timeout, the delay of *early HELLO beacons*, and the conductivity value \( \kappa \). As a
reference for the performance of our implementation, we use an extended version of AODV as well as OLSR. The settings and assumptions we used for our simulations are described next.

### 3.2.3 Extended AODV

As first reference for the performance and scalability of our implementation, we extended the standard implementation of AODV [PBRD03] included in Glomosim according to [MB05] to support gateway discovery in mesh networks. As proposed in the cited paper, all gateways are connected to a dedicated router that acts as a proxy to the Internet. This router has two tasks: (i) on the forward path, it sends route replies on behalf of hosts in the Internet; (ii) on the backward path, it initiates route requests for nodes in the wireless mesh network. Thus AODV does not have to distinguish between the different gateways and only a common route to the Internet has to be maintained, the route to the dedicated router.

### 3.2.4 OLSR

As second reference for HEAT, we use the OLSR [CJ03] implementation from the University of Niigata [Sot06]. OLSR allows to redistribute routing information from so-called “Non OLSR Interfaces” as the gateway uplink interface to the Internet. In our experiments, we have found that the performance of OLSR drops quickly with increasing mobility. We assume that this is in part due to the long hello interval of 2 seconds. In order to achieve a fair comparison with HEAT, which has a beacon interval of 1 second, we tried to adjust the hello interval of OLSR also to 1 second. With this adjustment, the performance of OLSR improves by roughly 10% and we use this setting for all experiments presented in this chapter.
Set-up

Our simulations are based on a WiFi network. All nodes are equipped with a 802.11b radio with a bandwidth of 11 Mbps and a nominal range of 250 meters. As MAC layer protocol, we use the 802.11 DFWMAC-DCF w/RTS/CTS [IEE07] and the two-ray ground propagation model [Rap96]. Due to the large network sizes we use, we were unable to model the effect of intermediate buildings in our city scenarios. However, we expect that the trends of our results also hold when such obstacles are present.

The actual node movement is modeled according to the city mobility model presented in Subsection 5.2. If not differently specified, we model 1000 nodes and 5 gateways on an area of 5km by 5km in the city of Zurich.

We expect wireless mesh networks to be used for Internet-type of applications like web browsing, messaging, chatting, etc. Therefore, we rely on an Internet traffic model as used in [FGHW99,Fie03] consisting of a half-half mix of streaming and web-like traffic. Streaming traffic has a bidirectional constant bit rate of 64 kb/s and the duration of streams are exponentially distributed with an average of 480 seconds. Web-like traffic consists of sporadic 1 kB requests according to an exponentially distributed inter-request time with an average of 10 seconds, followed by response messages with a message size that is Pareto II [JKB94] distributed (average 12 kB, minimal 0.1 kB, maximum 1000 kB).

All traffics in our simulations are from nodes within the wireless mesh network to hosts in the Internet and vice versa (there is no communication between the wireless nodes themselves). Note however, that we do not explicitly simulate the connection between the Internet gateway nodes and the hosts in the Internet because we assume the gateway nodes to be connected to the Internet over broadband connections with high bandwidth, low delays, and low packet losses compared to the wireless mesh
network. This means that all results we present are for packets inside the wireless mesh network.

All simulations have a duration of at least 10000 seconds and the results are always an average over at least 20 runs with different random seeds.

Results

We present the following metrics to compare the performance and scalability of HEAT with OLSR and AODV.

- **Packet delivery ratio** - The number of packets that are successfully received to the total number of packets sent. This metric includes all data packets from the mesh nodes to the gateways as well as packets from the gateways back to the mesh nodes.

- **Routing overhead** - The number of routing control messages that every node sends on average per second.

Scalability with the Network Size

In a first experiment, we look at how the performance is affected when increasing the network size while keeping the average node degree constant. The node degree is kept constant by increasing the simulation area (the section of the maps) as we increase the number of nodes. The results for a static scenario with randomly placed nodes, 100 active nodes (half constant-bit rate and half web-like traffic as described in the previous section), 5 Internet gateways, and an approximate average node degree of 3 are shown in Fig 3.3. In Fig. 3.3(a), we see the packet delivery ratio. As the network size increases, this ratio for HEAT remains constant at almost 100% and for OLSR decreases only marginally, while it significantly drops for AODV for network sizes greater than 500. The routing overhead, as shown in Fig. 3.3(b), explains the
3.2 Forward Path: Routing Packets Toward the Gateways

(a) Packet delivery ratio.

(b) Routing overhead.

Figure 3.3: Scalability by the network size.
reason for the performance degradation of AODV. As the network size increases, the average distance between the data sources and the Internet gateways also becomes larger. AODV, which then increases the flooding scope of its extending ring search route discovery algorithm [PBRD02], begins to exhaust the network with flooded control messages. In contrast, HEAT has constant overhead per node independent of the network size. The overhead for OLSR is also close to constant but still increases slightly because the link state routing protocol requires full knowledge about the whole topology. The hierarchical flooding mechanism used by OLSR mitigates the scalability problem but is not able to eliminate it completely.

Scalability with the Node Density

![Figure 3.4: Scalability by the node density.](image-url)
In a second experiment, we vary the average node degree to see how the protocols scale with the node density. We obtain different node densities by varying the total number of nodes while keeping the simulation area constant. The packet delivery ratio for a static scenario with randomly placed nodes on an area of 5 km by 5 km, 100 active nodes, and a total number of nodes ranging from 200 to 2000 is shown in Fig. 3.4. For node degrees smaller than around 2.5, the network is not always in a connected state (some data sources are partitioned from the group of Internet gateways) and the delivery ratio is thus less than 1 for all protocols. An average node degree of approximately 2.5 suffices to have a connected network and all three HEAT, OLSR as well as AODV manage to deliver almost all packets. However, as the node degree becomes larger than 2.5, the performance of HEAT and OLSR remains mainly unaffected, whereas the performance of AODV significantly drops. The reason in this case is again that AODV produces a large amount of overhead from the flooded route request messages. Note that the performance of HEAT and OLSR also slightly degrades since a higher node degree increases the probability of HEAT beacon, respectively hello, messages to interfere. However, compared to AODV, the degradation is only marginal (a few percent).

**Scalability with the Network Dynamics**

In a next experiment, we investigate how node mobility affects the routing performance. We consider two scenarios: (i) a scenario with mobile nodes moving at pedestrian speeds (i.e., node speeds that are uniformly distributed between 0.5 m/s and 3 m/s), and (ii) a scenario including nodes moving at car speeds in a city (i.e., node speeds that are uniformly between 10 m/s and 20 m/s).

The results for the pedestrian scenario with a simulation area of 5 km by 5 km, 5 gateways placed at strategic positions, and 100
Figure 3.5: Mobile scenario at pedestrian speeds.
3.2 Forward Path: Routing Packets Toward the Gateways

![Graph](image)

(a) Packet delivery ratio.

![Graph](image)

(b) Routing overhead.

**Figure 3.6:** Mobile scenario at car speeds.
active nodes are given in Fig. 3.5. At this node speed, the packet delivery ratio of HEAT is almost as good as in the static scenario (as previously shown in Fig. 3.3) with a slightly higher routing overhead. This additional overhead originates from the protocol enhancements to improve the convergence time as proposed in Section 3.2.1, since the core protocol has an overhead which is independent of the node speed. Looking at the results of OLSR reveals that its packet delivery ratio already decreases slightly for nodes moving at pedestrian speed, particularly in larger networks with longer routes.

Figure 3.6 shows the performance at car speeds using the same settings. At these node speeds, the performance of all three HEAT, OLSR and AODV is worse than at pedestrian speeds. However, the packet delivery ratio of HEAT remains above 70 percent whereas the ratio of OLSR as well as AODV drops below 40 percent for networks of 2000 nodes. At car speed, OLSR is no longer capable to maintain up-to-date routes while HEAT still does. This is caused by the fact that OLSR has to propagate its link state changes through the network while HEAT only requires local information exchange to adjust the temperature field. In addition to this, the early Hello mechanism of HEAT ensures fast convergence of the temperature field. These results show that the routing performance of HEAT scales better by the node speed in the wireless mesh network than OLSR and AODV.

The Effect of the Number of Internet Gateways

Finally, we investigate the scalability with respect to the number of available gateways in the mesh network. In Fig. 3.7, the packet delivery ratio and the routing overhead are presented for the pedestrian scenario with 1000 nodes moving on an area of 5 km by 5 km with randomly placed gateways of a total number ranging from 1 to 30 as well as 100 active nodes generating traffic.
Obviously, the packet delivery ratio increases as the number of gateways increases. This is mainly because the average distance between mesh nodes and gateways decreases as the number of gateways increases in the same area. Therefore, the average paths are smaller and the performance is less prone to link failures due to mobility. Furthermore, when the number of gateways is too small (e.g., only one gateway), the capacity of the radio interface at the gateway(s) becomes a limiting factor. In other words, the available capacity of the gateway(s) is not sufficient to support all the traffic generated by the mesh nodes.

If we consider the number of gateways that are necessary in the mesh network to provide an average packet delivery ratio that is greater than for example 0.99, we conclude that with HEAT, 5 gateways are sufficient. With 5 gateways, OLSR achieves a packet delivery ratio of about 0.9. With an increasing number of gateways, this ratio rises only slightly because the limiting factor of OLSR in mobile scenarios is that routes become invalid quickly. With AODV, a deployment of more than 30 gateways would be necessary to achieve the same performance. We conclude from this experiment that in mobile scenarios, OLSR and AODV require many more gateways than HEAT to achieve a comparable delivery ratio. A large number of gateways can be saved with our protocol, which makes it particularly suitable for mesh network deployments where the cost of the gateways is an important aspect.

3.2.5 Summary

In this section, we investigate the problem of routing between wireless nodes and Internet gateways in wireless mesh networks. We propose a new anycast routing protocol that is based on temperature fields. It takes advantage of the multiple gateways offering Internet access by looking at them as a single anycast group. Our protocol makes use of local beacon exchanges to
Figure 3.7: The effect of the number of gateways (mobile scenario at pedestrian speeds).
establish routing state and is thus particularly scalable for large and dense networks. Furthermore, temperature fields account for the link diversity and redundancy toward the gateways which makes our protocol particularly robust to node mobility.

We evaluate and compare the performance of our protocol with the performance of AODV and OLSR through extensive simulations. Our results show that HEAT and OLSR achieve packet delivery ratios in static dense and large mesh networks which are above 0.95. In the same settings, AODV fails to deliver more than 30 percent of the packets successfully. In mobile scenarios with car mobility, HEAT outperforms AODV as well as OLSR in terms of the packet delivery ratio by more than a factor of two. Finally, we show that HEAT is able to provide a packet delivery ratio that is higher than 0.99 with 5 gateways while the ratio for OLSR is 0.9. AODV would require more than 30 gateways to achieve a comparable performance in the same scenario.

3.3 A Robust Path Metric

Wireless networking is expected to continue to grow in popularity, leading to abounding connectivity in urban areas. Routing in such dense networks involves selecting among multiple paths the one with the most desirable characteristics. Routing protocols determine the path to use based on a routing metric. The most commonly used metric is minimum hop count, which selects an arbitrary path among those with the minimum number of hops to the destination. This routing method is also called shortest path routing and can be represented by a link cost metric that assigns every link a cost of 1; leading to a route cost equal to the number of hops.

Shortest path routing is useful under the assumption that there is no measurable difference in the quality of the links.
In fixed networks, this assumption is reasonable, but in mobile wireless networks, the quality of links varies widely and is subject to many influences, e.g. (i) the geographical length of the link, (ii) physical disturbances the link is exposed to (including interference and obstacles), and (iii) the relative speed of the endpoints of the link. Hence, the shortest path route may involve unreliable wireless links and break quickly, leading to packet loss and a connection interruption until a new route is established.

As proposed earlier, paths can be selected based on more than only their hop count by assigning each link a cost figure that reflects the quality of the link. The straightforward method to rate wireless links is to consider their Received Signal Strength Indicator (RSSI) value, which can easily be obtained from most wireless interfaces. The received signal strength indication measures the energy intensity of a packet received at the antenna, so it reports the strength of the desired signal but may also measure the strength of any interference in the area. In order to avoid links that are very long or cross areas of high interference, links with low RSSI are assigned high cost. However, selecting paths based on this simple link metric leads to routes with a large number of short links. Even though the individual links appear to be very reliable, the probability of a route break increases with the number of hops. Furthermore, every hop increases the end-to-end delay and incurs additional transmissions that add to the interference level.

Thus, selecting paths based on the RSSI values of their links is a trade-off between long paths over many reliable links and short paths over few unreliable links. We propose a link cost metric that differs from earlier proposals as follows. Our metric not simply prefers the strongest links, but rather uses a cost function that essentially rates the received signal strength indicator in relation to a pre-determined Preferable Received Signal Strength Indicator (PRSSI). However, even if every node selects the link with the lowest cost, the determined end-to-end path may not be
the most robust. Therefore, we develop a path metric based on our link metric that assigns the lowest cost to the most robust path. For practical applications, this path metric can for instance be incorporated in a distance vector routing protocol, but it could also be used in other contexts.

In order to demonstrate the effectiveness of our path metric, we compare the routes it selects with the shortest path routes using the network simulation software Glomosim. We use a static and two mobile scenarios with node speeds comparable to pedestrians and cars, respectively. To derive as realistic results as possible, the movement of pedestrians and cars follows the vectorized street map of the city of Zurich. Our results show that the probability of route breaks is decreased considerably compared to shortest path routing. In the car speed scenario, the average route lifetime is increased by a factor of 7 and the packet loss ratio (fraction of packets lost due to route interruptions) is reduced by up to a factor of 15.

This section makes the following main contributions. First, we introduce a link cost metric that allows selecting robust links in mobile wireless networks. Second, we provide a path metric to select robust end-to-end paths based on our link metric. Third, we present an implementation of those metrics in a simple proactive distance vector routing protocol.

The rest of this section is organized as follows. In the following subsection, we introduce our link metrics. Then, we discuss the implementation of our metrics in a routing protocol. In Subsection 3.3.3, we describe the simulation environment we use for our experiments, and in Subsection 3.3.3 we present the results of the evaluation. Finally, we conclude this section.

### 3.3.1 Metric Design

In this subsection, we present the concept of our link cost metric which prefers certain neighbors based on their received signal
strength indicator\(^1\). We first explain how to derive a meaningful average of the received signal strength indicator and then we describe how this average is used to calculate the link cost metric.

**Link Cost Metric**

The aim of preferring certain neighbors over others for routing data is to minimize the probability of link breaks and thus also minimize the route break probability. In mobile wireless networks that route according to the shortest path paradigm, link breaks due to mobility are very common. The reason is that the distance of two neighbors is optimized such that it is as close to the maximal communication range as possible and thus the number of hops are minimized. Thus, links selected by shortest path routing are likely to break as soon as the nodes move away from each other.

Estimating the distance between neighbors without location data is usually done based on the received signal strength indicator since this measurement value provides an indication about the quality of a link and is easily obtainable from most wireless interfaces. However, the RSSI value is susceptible to fluctuations due to fading and similar temporary disturbances. In order to mitigate the effects of random signal fading, we use an averaging technique widely used in telecommunication: Exponential Weighted Moving Average (EWMA).

We calculate the average received signal strength indicator, \(\bar{s}_{i,k}\), for neighbor \(k\) considering the packet with index \(i\) according to Alg. 2. By \(U := [u_1, \ldots, u_{|U|}]\), we denote the set of neighbors and by \(S_k := [s_{k,1}, \ldots, s_{k,|S|}]\), the set of received signal strength indicators measured from the packets received from neighbor \(k\). The smoothing factor \(\alpha\) determines the weight of each measured value \(s_{i,k}\) and reduces the weight of a single measurement, if there

\(^1\)By *received signal strength indicator* of a neighbor we refer to the strength of the signal and interference that is received *from* this neighbor.
are many current measurements available. $n_k$ denotes the number of packets received from neighbor $k$ during the last second.

Algorithm 2 Exponential weighted moving average of the received signal strength indicator.

1: for $u_k \in U$ do
2: \[ \alpha = \frac{n_k}{n_k+1} \]
3: for $s_{i,k} \in S_k$ do
4: \[ \bar{s}_{i,k} = (1 - \alpha) \bar{s}_{i-1,k} + \alpha s_{i,k} \]
5: end for
6: end for

Using the average received signal strength indicator $\bar{s}_{i,k}$, we calculate the link cost according to the cost function defined in Alg. 3. Note, that we have also run simulations with exponential cost functions. But since we have not found a parameterization that outperforms the simple linear function, we use the linear function for all simulations reported in this thesis. The linear function consistently delivers high performance, is easy to understand, and its parameters can be chosen intuitively as shown at the end of this subsection.

Our linear cost function has three parameters: $c_{prssi}$, $c_{out}$, and $c_{in}$. These parameters determine, which link cost is assigned to a link with a given received signal strength indicator. By defining $c_{prssi}$ as lower than $c_{out}$ and $c_{in}$, links whose received signal strength indicator is close to the Preferred Received Signal Strength Indicator (PRSSI) will have minimal link costs corresponding to highest preference, as depicted in Fig.3.8. Links with much lower or much higher RSSI will be assigned higher cost, indicating lower preference.

The preferred received signal strength indicator depends on the scenario and in particular on the mobility of the nodes. At high node speeds, a value close to the maximum received signal strength indicator (rxMax) should be chosen because links that
Figure 3.8: Dependence of the link cost metric on the received signal strength indicator. Note: for simplicity, we depict the communication range (rxThresh) of a node and the power level of the preferred received signal strength indicator as circles, but our algorithm by no means assumes that the range is indeed circular.
**Algorithm 3** Calculation of link cost $c_x$.

1. if $s_{i,k} > PRSSI$ then
2. 
   $c_k = \frac{PRSSI - \bar{s}_{i,k}}{PRSSI - rxThresh} \cdot (c_{out} - c_{prssi}) + c_{prssi}$
3. else
4. 
   $c_k = \frac{s_{i,k} - PRSSI}{rxMax - PRSSI} \cdot (c_{in} - c_{prssi}) + c_{prssi}$
5. end if

.. include:: code_snippet

... do not have high received signal strength indicators are likely to break quickly. In such an environment, paths consisting of several short links provide better performance.

In contrast, if nodes move slower, the preferred received signal strength indicator should be closer to the minimal received signal strength indicator that still allows successful reception ($rxThresh$ value). The reasoning is that at low speed, the probability that two nodes move outside the communication range is much lower, even neighbors with low received signal strength indicators may remain within range for a considerable period of time. Hence, the routing performance is better if paths with lower hop count are used.

**Robust Path Metric**

After describing our link cost metric, we show how it can be used as a path metric. An optimal path with respect to the link cost metric comprises exactly those nodes whose received signal strength indicators is very close to the preferred received signal strength indicator. An example is depicted in Fig. 3.9. Node $C$ is exactly at the PRSSI of node $D$ and node $B$ is exactly at the PRSSI of node $C$. However in reality, it is unlikely that every node will always have a neighbor that is located exactly at the preferred received signal strength indicator range. And if there exists such a path, it might not lead to the destination on a straight line but might involve detours across a large number of paths.
links. In most cases, the robustness of such an overly long path would be no higher than the robustness of the shortest path.

Therefore, we propose a path metric that provides a trade-off between the preferred received signal strength indicator and the hop count to the destination. Our definition of the optimal path \( p_{opt} \) from the set of possible paths \( L \), is the path that minimizes the sum of its link costs:

\[
p_{opt} = p_l : \min_{p_l \in L} \sum_{k \in p_l} c_k
\]  

(3.3)

This definition is commonly used in networking. For example, the minimum hop-count metric uses the same definition except that the link costs \( c_k \) are set to 1 for all links.

To illustrate the impact of this new path metric on the link selection, we plot in Fig. 3.10 the path chosen by this metric and compare it with the path that would be chosen by the minimum hop-count metric. The minimum hop-count path is one hop shorter, since our path metric trades in low hop count for more reliable links in order to increase the robustness of the path.

**Cost Parameter Selection**

The robustness of a route depends mainly on how good the trade-off between hop count and hop reliability fits the scenario. The link cost function proposed in Alg. 3 can be parameterized to optimize the path selection for many scenarios.

As a case study, we describe how to optimize the parameters \( c_{prssi}, c_{out}, \) and \( c_{in} \) for an arbitrary chosen scenario with uniform distribution of link breaks with the following probabilities. For nodes at the border of the communication range, the link break probability be \( b_{out} := 40\% \); for nodes that are next to each other, the link break probability be \( b_{in} = 1\% \); and for nodes at the preferred received signal strength indicator, the link break probability be \( b_{prssi} := 10\% \).
Figure 3.9: Path costs for a route from A to D using our link cost metric.

In order to minimize the route break probability we have to set $c_{prssi}$, $c_{out}$, and $c_{in}$ according to the following consideration. The probability of a route break should be the same for a path over several links with preferred received signal strength indicator as for a path with much fewer links where the nodes are at communication range distance. Assuming uniform distribution of link breaks, the probability of a route break can be approximated as follows. Let $b$ be the link break probability and $h$ be the number of hops. Then the probability of a route break is $r = 1 - (1 - b)^h$.

Since the proposed cost function is linear and the costs over a path are accumulated, the cost parameters $c_{out}, c_{in}, c_{prssi}$ can be
interpreted as the number of hops. Inserting the parameters and
the route break probabilities into the above equation leads to the
following system of equations:

\[
(1 - b_{prssi})^{c_{prssi}} = (1 - b_{out})^{c_{out}} \quad (3.4)
\]
\[
(1 - b_{prssi})^{c_{in}} = (1 - b_{in})^{c_{prssi}}. \quad (3.5)
\]

By arbitrarily defining \( c_{prssi} = 1 \), we find

\[
c_{out} = \log(1-b_{out})(1-b_{prssi}) \approx 5 \quad (3.6) \]
\[
c_{in} = \log(1-b_{prssi})(1-b_{in}) \approx 10. \quad (3.7)
\]
Thus, in our example, the probability of a route break is the same for a path with five links with preferred received signal strength indicator as for a path with one long link where the nodes are at the maximal communication range distance: \(5c_{\text{prssi}} = c_{\text{out}}\).

### 3.3.2 Implementation

In order to compare the effect of link and route metrics with shortest path routing, we have implemented a proactive distance vector routing protocol. We have deliberately chosen a very basic implementation to minimize the influence of protocol specific features. We will first briefly describe the distance vector routing algorithm and then discuss the link and path metric implementation in more detail.

**Exchange of Distance Vector**

In our basic proactive distance vector routing protocol, every node maintains the distance for every known destination. Note that in the context of distance vector routing, the term distance corresponds to the cost of a link or a path. The distances are computed based on periodically exchanged messages called distance vectors. A distance vector is a table that provides destinations and their distances. To avoid the typical problems of distance vector routing (loops and the count-to-infinity problem), we use a widely established method called split horizon with poison reverse [Mal98]).

In our implementation, all nodes build and maintain a data structure called *neighbor table* that contains the distance vectors received by neighbors and a timestamp of the last update.

Based on the neighbor table, nodes maintain their own distance vector. Whenever an entry is added, removed, or changed in the neighbor table, the node re-computes its distance
vector. There are essentially three cases that necessitate an update of the distance vector:

1. New neighbor: If a distance vector from an unknown neighbor is received, a corresponding entry is added to the neighbor table. In addition, the distance vector of the node is re-computed.

2. Maintain neighbor: If the distance of a known neighbor to a destination changes, the distance of this node to this destination is recomputed.

3. Missing neighbor: If no distance vector is received from a neighbor for a certain period, its entry is removed and the distance vector of the node is re-computed.

Based on the distance vector, routing packets is straightforward and implemented on a hop-by-hop basis: A packet is always forwarded to the neighbor with the shortest distance to this destination.

**Implementation of Robust Path Metric**

The implementation of our path metric is split into two modules. The *link cost module* maintains the link cost to all neighbors based on the link cost metric, as described in Subsection 3.3.1.

The *path cost module* maintains the distances to the destinations and provides the corresponding next hop. The distance $c_d$ to a destination $d$ is calculated in two steps. First, for every neighbor $n_k$, the sum of the distance to this neighbor, $c_k$, and the distance to the destination reported by this neighbor, $c_{d,k}$, is calculated. Second, the neighbor $c_{k*}$ with the lowest distance is chosen as the next hop and stored in the distance vector along with the associated cost $c_d$. In the rest of the thesis, we will refer to this path selection method as *route preference* scheme.
3.3 A Robust Path Metric

To compare our solution with shortest path routing, we also implement a path cost module using hop count as the path metric. The only difference to the algorithm presented above is that a constant link cost of 1 is used instead of \( c_k \).

Since some implementations of distance vector routing protocols have a rather low limit for the distance, as for example 255 with AODV [PBRD03], they cannot really profit from our path metric. Because in the worst case the number of hops is very limited, as for example with a \( c_{in} \) of 10 and a resolution of 0.1 the maximum number of hops is 2. However, the effect of preferring links at the preferred received signal strength indicator can still be shown, as follows. The distance \( c_d \) to a destination \( d \) is calculated as with the route preference scheme. However, the distance that is stored in the distance vector along with the chosen next hop is simply the sum \( c_{d,k*} + 1 \). In the following, we refer to this route selection scheme as the link preference scheme. Including this link preference scheme in the evaluation shows clearly the advantage of our route preference scheme over link preference schemes with a local view proposed in the literature discussed in Subsection 2.5.

Parameter Selection

The preferred received signal strength indicator value should be set according to the maximal expected movement speed of the nodes and the update interval of the deployed routing protocol. To determine the preferred received signal strength indicator, we consider two nodes being at this received signal strength indicator and moving away from each other with the maximal expected speed. The preferred received signal strength indicator should now be chosen such that these two nodes do not lose their connection (reach the maximal communication range) during the next two update intervals of the routing protocol. This allows the routing protocol to (i) detect the leaving node and (ii) fix the
affected paths. The preferred received signal strength indicator is calculated individually per node based on the receiver sensitivity (rxThresh) of the deployed hardware, thus ensuring hardware independent operation.

We assume according to the established traffic speed regulation for cities a maximal node speed of $60 \text{ km/h}$. The PRSSI is calculated for two routing update intervals of 1 second. The detailed parameters for the calculations of preferred neighbor routes are listed in Tab. 3.1. Note, that since the distance values are implemented as integers, we scaled the link costs $c_{\text{prssi}}$, $c_{\text{out}}$, and $c_{\text{in}}$ by a factor of 10000 to increase the precision and to avoid rounding effects. As the distance vector update interval, we use 1 second.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{\text{prssi}}$</td>
<td>10000</td>
</tr>
<tr>
<td>$c_{\text{out}}$</td>
<td>50000</td>
</tr>
<tr>
<td>$c_{\text{in}}$</td>
<td>100000</td>
</tr>
<tr>
<td>PRSSI</td>
<td>rxThresh + 10dB</td>
</tr>
</tbody>
</table>

**Table 3.1:** Parameters of the robust path metric for the evaluation.

### 3.3.3 Simulation Study

To evaluate and compare the performance of our approach, we performed simulations with Glomosim [ZBG98], a network simulator for wireless networks. As a reference for the performance of our path metric, we use the minimum hop count metric. The settings and assumptions of our experiments are described in this subsection; the results are given in Subsection 3.3.3.
Set-up

Our simulations are based on a WiFi network. All nodes are equipped with a 802.11b radio with a bandwidth of 11 Mbps and a nominal range of 250 meters. As MAC layer protocol, we use the 802.11 DFWMAC-DCF w/RTS/CTS [IEE07] and the two-ray ground propagation model [Rap96]. Due to the large network sizes we use, we were unable to model the effect of intermediate buildings in our city scenarios. However, we expect that the trends of our results also hold when such obstacles are present.

The actual node movement is modeled according to the city mobility model presented in Subsection 5.2. In each scenario, we model 1000 nodes on an area of 5km by 5km in the city of Zurich. As data traffic model, we use the same Internet like mix presented in 3.2.4. The simulations have a duration of at least 10000 seconds and the results are always an average over at least 20 runs with different random seeds.

Results

In this subsection, we show the results of the evaluation of our route preference scheme. We compare this scheme with the link preference scheme and with shortest path routing. We expect that the route preference scheme outperforms the shortest path approach by a significant margin, especially in highly mobile scenarios. In the subsequent subsections, we will provide detailed results, based on the following metrics.

- **Packet loss ratio** – Ratio between the number of lost packets and the number of packets sent by the source.

- **Path length** – Average path length of a route (number of hops).

- **Route break probability** – Average probability that a route breaks within a second. This corresponds to the probability
that a packet is forwarded to a neighbor that is no longer in the communication range of a node.

In highly dynamic networks, such as mobile wireless networks, the routing performance is mainly measured by the packet loss ratio and the path length. We deliberately do not consider the route delay because this metric heavily depends on the traffic load and the packet loss ratio. Since packets on long routes with high delay are typically lost by protocols with high loss rate, those protocols would have a shorter average delay and protocols with low packet loss ratio would be penalized by a delay metric.

For the optimization goal of minimal route break probability, there is a trade-off between path length and link break probability. However, since there is no direct correlation between link break probability and end-to-end packet loss ratio, we do not discuss the characteristics of individual links further. Instead, we focus on the route break probability, i.e. the probability that a route breaks at any link.

**Packet loss ratio**

In Fig. 3.11, we plot the packet loss ratio against the speed of the nodes. In the static scenario, there are no link breaks; consequently, there is no significant difference among the three schemes. However, already at pedestrian walking speed, noticeable differences emerge. As expected, the differences are most pronounced at car speed. Here, the link preference scheme has a packet loss ratio of 35%, which is around one half of the ratio of shortest path routing. The route preference scheme leads to an even lower packet loss ratio of 8% – only one quarter of the packet loss ratio of link preference and less than one tenth of the shortest path method’s loss ratio. These results indicate two things. Firstly, the distance between the nodes of a route is critical for communication in highly mobile networks. Secondly, optimizing
these distances individually helps, but balancing these distances along the whole path is even more effective to minimize the overall packet loss ratio, as demonstrated by our route preference scheme.

![Graph showing packet loss ratio](image)

**Figure 3.11:** Packet loss ratio.

The ability to select neighbors is only useful if there are multiple neighbors to choose from. In order to understand the influence of the node density, we analyze the scenario with nodes moving at car speed in more detail. The packet loss ratio is plotted against the node density in Fig. 3.12. Obviously, if the node density is 6 or more nodes per communication range, the packet loss ratio does not decrease anymore. Again, the packet loss ratio of shortest path routing is the highest and the ratios of the link and route preference schemes are again much lower – by as much as a factor of 2 and 10, respectively.

At very low node densities of 3 or less nodes per communication range, the packet loss ratio increases almost exponentially not only with shortest path, but also with the link and route preference schemes. This is not surprising given that the freedom
to select among multiple neighbors evades as the node density decreases, leaving not much room for optimization.

![Graph showing packet loss ratio vs. node density](image)

**Figure 3.12:** Packet loss ratio vs. node density with nodes moving at car speed.

**Path length**

The path length correlates with the end-to-end delay and also has an influence on the route break probability. In Fig. 3.13, we plot the path lengths for the scenarios used previously in Fig. 3.11. The shortest path routing method optimizes for this metric and is obviously the optimal case here. Any scheme that uses nodes that are not on one of the shortest paths is bound to extend the path length. This plot shows that the route preference scheme incurs much longer routes. However, the links used by those routes seem to be much more reliable, such that the route preference scheme still leads to substantially more robust routes and lower packet loss ratio as shown previously in Fig. 3.11.
3.3 A Robust Path Metric

The robustness of routes manifests itself in the packet loss ratio. In Fig. 3.14, we plot another metric that corresponds to route robustness: the probability that a route breaks within the next second. As expected, in the static case, no difference is visible. However, as the node speed increases, the effect of preferring certain neighbors becomes apparent. The route preference scheme, which aims at optimizing the route break probability, has by far the lowest route break probability. At car speed, the probability of a route break is almost a factor of 7 lower than with shortest path routing.

Consequently, the average route lifetime of the route preference scheme is the highest – around 8 times higher than with shortest path (see Tab. 3.2).
Figure 3.14: Route break probability.

<table>
<thead>
<tr>
<th>Routing scheme</th>
<th>Average route lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shortest path routing</td>
<td>6.8s</td>
</tr>
<tr>
<td>Link preference scheme</td>
<td>10.5s</td>
</tr>
<tr>
<td>Route preference scheme</td>
<td>50.5s</td>
</tr>
</tbody>
</table>

Table 3.2: Average route lifetime with nodes moving at car speed.

3.3.4 Summary

In mobile wireless networks, the problem with shortest path routing is that the routes often involve links at the edge of the communication range that are likely to break due to interference or node mobility. In contrast, in this section, we propose to select paths that provide a good trade-off between their hop count and the reliability of their links. We introduce a new link cost metric for mobile wireless networks. Furthermore, we propose a path metric that allows selecting robust paths based on our
link metric. Compared to shortest paths, the paths selected by our metric exhibit a lower packet loss ratio and a lower break probability because they consist of more reliable links.

Our link cost metric gives preference to links that offer a good compromise between length and reliability using a cost function. Every node assigns its neighbors a link cost that depends on how close to its preferred received signal strength indicator the neighbor’s RSSI value is. Based on this link cost, our path metric then selects the most robust path.

We demonstrate the effectiveness of our approach in a simulation-based study with realistic mobility traces where nodes move according to the vectorized street map of the city of Zurich. With nodes moving at car speeds (up to $60 \text{ km/h}$), paths selected by our path metric are a factor of 7 less likely to break than paths chosen by minimum hop count. In terms of end-to-end performance, the packet loss ratio caused by route interruptions during the transmission of data is reduced by a factor of up to 15. We conclude that selecting paths based on the RSSI value of their constituting links allows increasing route robustness considerably.

3.4 Reverse Path: Routing Packets into Wireless Mesh Networks

Routing packets into a wireless mesh network is different from routing packets out, because for routing packets out of a wireless mesh network only a single destination has to be maintained, the default route to the gateways. In contrast for routing packets into a wireless mesh network, a proper route has to be maintained for every single node in the mesh as depicted in Fig. 3.15.

As with forward path routing, optimizing the performance of the backward path is a trade-off between accuracy of the routing state information and control traffic overhead. We compare three backward path routing protocols, each of which represents one of
the three most widely used families of routing protocols: reactive hop-by-hop routing (AODV-CGA), proactive hop-by-hop routing (FBR), and proactive source routing (GSR). One major difference between GSR and the other two protocols is that with GSR, all routing state information is kept on the mesh gateways, while with the other two protocols; the state information is distributed on the mesh nodes. Furthermore, GSR re-uses forward paths collected by data packets while the other protocols depend on dedicated routing control packets to determine routes.
AODV-CGA is a version of AODV extended for mesh networks as proposed by Braun et al. [MB05]; the other two protocols, FBR and GSR, we have developed ourselves.

1. **Reactive hop-by-hop routing (AODV-CGA):** We use an extended version of AODV specialized for wireless mesh networks. This extension extends the AODV routing domain to include a network border router that is connected to all gateways. Upon receiving a packet from the Internet addressed at a mesh node, the border router floods a route request to all gateways from where a route is then determined in the mesh network.

2. **Proactive field-based routing (FBR):** We propose a proactive field-based routing protocol similar to HEAT. With this protocol, every mesh node maintains a scalar field that is propagated by beacon messages through the mesh network. Routing toward a specific mesh node is achieved by forwarding along the steepest gradient of the field of the destination node.

3. **Gateway source routing (GSR):** We present a source routing protocol that re-uses the forward paths that are recorded by data packets and stored on the gateways. These paths are then used for source routing on the backward path.

Based on the experiences we made during the evaluation, we introduce modifications to the FBR and the GSR protocols that mitigate some of their limitations and improve their performance.

With the Glomosim [ZBG98] network simulator, we compare the performance as well as the communication overhead of these three routing protocols. In order to account for worst-case situations, we assume that all nodes except the gateways are moving constantly. We simulate an application similar to radio streaming, where traffic is sent by a few Internet servers to mesh
nodes. In order to account for the frequently required “keep-alive” feedback packets, the receiving mesh nodes periodically send a data packet to the Internet server. Note, that there is no traffic on the forward path except these feedback packets to ensure that our evaluation only considers the performance of the backward path protocols.

In our simulation experiments, FBR outperforms the other protocols with respect to the packet delivery ratio. However, FBR does not scale well with the network size, especially if routes are updated frequently, because for every mesh node a proper field has to be maintained. GSR scales much better to the network size because it just reuses routing information from the forward path. For instance, with an update rate of one update per second, the packet delivery ratio of GSR is almost as good as with FBR and the communication overhead is much lower. We show that, besides offering very good performance, GSR offers the best scalability properties with respect to the network size. For instance with a route update frequency of five seconds, GSR has a somewhat lower delivery ratio than FBR but still outperforms AODV-CGA in both delivery ratio and communication overhead.

The rest of this section is organized as follows. First, we present the three routing protocols we compare in this section. Then we present our evaluation study and discuss the results.

3.4.1 AODV-CGA (Reactive Hop-by-hop Routing)

As the representative of the class of reactive routing protocols, we use an extended version of AODV that we call AODV-CGA. This extended AODV protocol allows the use of multiple gateways to the Internet using a Common Gateway Architecture (CGA) proposed by Braun et al. [MB05]. AODV-CGA shares most mechanisms with the well-known AODV protocol and thus may be seen as a benchmark in our comparison. The basic principle of operation of AODV is as follows. AODV constructs
3.4 Reverse Path: Routing Packets into Wireless Mesh Networks

routes *on demand* by flooding route requests using an expanding ring search mechanism. If a request reaches a destination, the destination node answers with a route reply message that is forwarded using temporarily stored information from the request. The reply concurrently establishes the route by registering it at every intermediate node. In AODV, route error messages and sequence numbers are used to deal with broken routes.

As proposed by Braun, all gateways are connected to a dedicated router that acts as a proxy to the Internet. This router has two tasks: (i) on the forward path, it sends route replies on behalf of hosts in the Internet; (ii) on the backward path, it initiates route requests for nodes in the wireless mesh network. To improve scalability with respect to the network size, expanding ring search [PBRD03] is used for the route requests, which slightly increases the route set-up time.

### 3.4.2 FBR (Proactive Field-based Routing)

As the representative of the proactive routing protocol family, we designed a field-based routing protocol similar to HEAT. In HEAT, wireless mesh nodes periodically exchange beacons. These beacons contain a list of all known destinations with their respective field value. When a new destination appears, it announces its presence with beacons to its neighbors in order to establish a field. With this mechanism, a field on the network is constructed for every destination. This field assigns a value to every node in the network; the destination bears the maximum value. Packets are then routed along the steepest gradient toward the destination.

Note, that FBR—being a proactive routing protocol—incurs a small communication overhead since it proactively maintains all routes, regardless of whether there is data traffic or not.
3.4.3 GSR (Gateway Source Routing)

With gateway source routing (GSR), we propose to reuse the forward path information from the packets that arrive at the gateways. In the routing header of every packet, the intermediate hops from the mesh node to the gateway are recorded. These paths are then stored in the gateways. To route packets to a mesh node, the mesh gateway inverts the recorded forward path and copies it to the packet header. The gateway then sends the packet to the first node of the backward path. Each node updates the path in the header by removing its entry and forwards the packet to the given next hop until the packet reaches the destination.

By design, this approach is scalable to the number of mesh nodes as it imposes no overhead that depends on this number. Only the gateways have to maintain up-to-date routes to individual mesh nodes. Also, this approach does not increase the number of control packets exchanged between the mesh nodes, and thus reduces the chance of collisions.

Obviously, GSR requires that a packet toward a host in the Internet is first sent by a mesh node in order to establish the backward path. However, since the majority of communication is initiated by mesh nodes, we consider this to be only a small limitation.

Should a mesh node act as a server, a dedicated addressing mechanism (e.g., HIP [MNH04,Per96]) would probably be used. HIP and most other addressing mechanisms require periodic registration messages from the mesh node toward a gateway. Those periodic registration messages serve also to initiate and maintain the path at the gateway. Should traffic be unidirectional from an Internet node toward a mesh node, this poses a problem, as the backward path cannot be maintained. This problem can only be solved by requiring the mesh node to periodically send some sort of ping packets to a gateway. Note, however, that most applications have a feedback channel
(i.e., TCP acknowledgments or RTCP messages for streaming applications) and hence generate bidirectional traffic. In our evaluation study in Subsection 3.4.4, we evaluate a broad range of feedback intervals to shed some light on the trade-off between communication overhead and the quality of the backward path.

3.4.4 Enhancements

Based on simulation experiments, we design enhancements to our protocols. With FBR-GW we aim to improve the scalability to the network size of FBR and with GSR-PN we strive to improve the packet delivery ratio of GSR.

FBR-GW (Enhancement for Scalability)

The FBR protocol is designed for maximum packet delivery ratio, but it scales poorly with the network size. Routing from the mesh nodes to the Internet gateways requires only a single field. In contrast, routing from the gateways back to the mesh nodes requires a separate field for every mesh node. Thus, a field of every node is propagated through the entire network.

Assuming that there are no data connections among the mesh nodes, the scalability to the network size can be improved as follows: We propose to let the field information of mesh nodes only be propagated toward the gateways. Implementing this enhancement is straightforward. Instead of establishing the “per-node” field with all neighbors, only the neighbors with an increasing “gateway field” value are used to establish a mesh node field. As a result, state information about individual mesh nodes is only established in nodes that might be used for packet forwarding.
GSR-PN (Enhancement for Performance)

The GSR protocol is designed for scalability to the network size. However, its packet delivery ratio drops rapidly if the routes are not frequently updated by feedback packets. Most packets are lost due to paths that contain links that are broken because the nodes moved away from each other. Such link breaks happen mostly between nodes that are almost at the maximal transmission range to each other. In order to reduce this effect, we use the robust path metric presented in Section 3.3 where the routing mechanism prefers nodes at an optimal distance.

3.4.5 Simulation Study

Set-up

We performed our simulations with Glomosim [ZBG98], a network simulator for wireless networks. With Glomosim, we evaluated the performance, overhead, and set-up time of the three presented approaches. All simulations run for a duration of at least 10000 seconds. Note also that the results presented in the next section are always an average over at least 20 runs with different random seeds.

Our simulations are based on a WiFi network. All nodes are equipped with a 802.11b radio with a bandwidth of 11 Mbps and a nominal range of 250 meters. As MAC layer protocol, we use the 802.11 DFWMAC-DCF w/RTS/CTS [IEE07] and the two-ray ground propagation model [Rap96]. Due to the large network sizes we use, we were unable to model the effect of intermediate buildings in our city scenarios. However, we expect that the trends of our results also hold when such obstacles are present.

The actual node movement is modeled according to the city mobility model presented in Subsection 5.2.

Internet like traffic consists of a mix of streaming and request-response traffic [FGHW99, Fie03]. In this section, we focus
on routing packets from the Internet into the wireless mesh network. For obtaining a diversified picture with respect to the packet transmission ratio, we focus our simulations on highly asymmetric streaming traffic as web radio.

According to Anderson and Stokas [And00, Sto02], we use high quality audio streaming with a constant bit rate of 96 kBits. 200 nodes act as streaming clients. The durations of streams are exponentially distributed with an average of 480 seconds and an exponentially distributed pause time of 120 seconds.

All streaming traffic in our simulations is initiated from hosts in the Internet and forwarded to nodes within the wireless mesh network. The wireless nodes regularly send back control packets of 100 Bytes. To examine route stability in our simulation study, we varied the interval of feedback control packets from 1 to 40 seconds.

Results

We use the following metrics to compare the performance and overhead of the three routing protocols:

- **Packet delivery ratio**: The ratio between the number of packets that are received and the number of packets sent. This metric only considers backward path traffic, i.e., the data packets from the gateways to the mesh nodes.

- **Routing overhead**: We use the following metrics to capture several aspects of routing overhead:
  - The average number and size of routing control messages sent per second by a mesh node.
  - The average additional header space required for routing information in a data packet.
• **Route set-up time:** The time elapsed until a demanded route is available. Unsuccessful route establishments are ignored.

**Packet delivery ratio**

When we compare the packet delivery ratio in the static scenario, all routing protocols deliver over 99.6% of the data packets. However, already if nodes move at pedestrian speed, the results differ. In Fig. 3.16, we plot the delivery ratio of the different routing protocols vs. the feedback packet interval. Increasing the feedback interval decreases the freshness of the routing information. Due to its proactive route maintenance mechanism, the FBR protocol is almost independent of the feedback interval. The enhanced FBR-GW protocol shows similar behavior at a lower ratio of roughly 90%. In the simulation log file, we found that the packet losses occur when a node that is directly connected to a gateway moves away. During the period where the node re-establishes its field, packets are lost since there is no alternative path to such a node.

The AODV-CGA protocol performs very well if the feedback interval is long and achieves a packet delivery ratio of up to 95%. However, with shorter feedback intervals, the routing packet broadcasts interfere with the data traffic and the packet delivery ratio decreases to 84% at a feedback interval of 1 second. Note, this feedback interval corresponds to less than 1kbit/s. If the traffic were symmetric, the delivery ratio of AODV-CGA would presumably drop further. Apparently, AODV-CGA is very sensitive to the network load. A detailed study of the routing overhead is presented following.

The GSR protocol performs almost as good as the FBR protocol when the feedback interval is short. But when the paths stored at the gateways are updated less frequently, the packet delivery ratio drops quickly. The enhanced protocol, GSR-PN,
reduces this problem. With a feedback interval of 5 or even 10 seconds, GSR-PN still achieves a higher packet delivery ratio than all other protocols of our study except FBR.

In Fig. 3.17, we plot the packet delivery ratio for nodes moving at car speed. The results are similar to those of the pedestrian model. The FBR protocol still outperforms all other protocols, but, due to the high mobility, the delivery ratio is only around 90%. Note that the GSR-PN protocol performs almost as good as the AODV-CGA protocol at a feedback interval of 5 seconds.

A profile over all scenarios with a feedback interval of 5 seconds is provided in Fig. 3.18. Obviously, all routing protocols are affected by the higher node speed, but the FBR-GW protocol is particularly susceptible to this parameter.
Routing overhead

Besides packet delivery, routing overhead is the most important property that we evaluate. In our simulation setting, we compare the following routing overhead metrics:

1. Number of routing packets sent per second by a mesh node
2. Average size of routing packets
3. Average additional header space required per data packet

We evaluate these metrics with a feedback packet interval of 10 seconds.

(i) Figure 3.19 shows that for AODV-CGA, the number of routing packets per data packet raises steeply with increasing mobility while it remains almost constant for the proactive
protocols. This is not surprising, since AODV-CGA is a reactive routing protocol. Its advantage lies in the low communication overhead if there is little or no traffic.

(ii) Figure 3.20 shows the size of routing packets in a logarithmic scale. While both GSR and GSR-PN do not require additional routing packets, the other protocols are based on dedicated routing control packets. The routing packets of AODV-CGA are small and have a constant size. The FBR protocol produces larger routing packets. As the plot shows, the limitation of field information propagation used in FBR-GW helps to reduce the packet size and thus increases the scalability to the network size. However, with increasing speed, the packet size increases considerably.
(iii) GSR and GSR-PN require additional space for routing information in the header of each data packet. This overhead slightly increases for scenarios with higher mobility because the average routes tend to become longer (see Fig. 3.21). Of note, with GSR-PN, the space required is higher than with GSR because the Preferred Neighbor enhancement leads to paths with higher hop count that provide shorter and more reliable links.

**Route set-up time**

We compare the route set-up time for the investigated routing protocols. By design, the proactive protocols, i.e., FBR, FBR-GW, GSR, and GSR-PN do not require any route set-up time if initialized or converged. With the AODV-CGA protocol, when a route is required that has not recently been used, the protocol...
initiates the route discovery process. Table 3.3 shows the route set-up time for different mobility schemes. At a first glance, the set-up time seems to decrease with increasing node speed. However, by analyzing the simulation logs in more detail, we found that this is due to the fact that the discovery of longer routers fails more frequently. In order to eliminate this effect, the average set-up time only comprises route discovery processes that succeed.

<table>
<thead>
<tr>
<th>scenario</th>
<th>static</th>
<th>pedestrian</th>
<th>car</th>
</tr>
</thead>
<tbody>
<tr>
<td>routing set-up time</td>
<td>0.50s</td>
<td>0.34s</td>
<td>0.28s</td>
</tr>
</tbody>
</table>

**Table 3.3:** Route set-up time of the AODV-CGA protocol.

---

*Figure 3.20:* Average size of routing packets.
3.4.6 Discussion

In Table 3.4, we summarize the strengths and weaknesses of the evaluated routing protocols. Reactive routing protocols are widely used in the ad hoc network area. However, in our mesh network scenarios, the AODV-CGA protocol exhibits poor performance.

If the goal is to maximize the packet delivery ratio, the proactive field-based routing protocol, FBR, should be considered. FBR inherently does not scale to the network size, but for smaller networks it seems to be a good fit.

Considering scalability with respect to the network size, the gateway source routing protocol, GSR, outperforms the other protocols. GSR is scalable to the network size and still achieves a very high packet delivery ratio given frequent feedback packets from the mesh nodes. With the proposed enhancement that
3.4 Reverse Path: Routing Packets into Wireless Mesh Networks

<table>
<thead>
<tr>
<th></th>
<th>AODV-CGA</th>
<th>FBR</th>
<th>FBR-GW</th>
<th>GSR</th>
<th>GSR-PN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet delivery ratio</td>
<td>-</td>
<td>++</td>
<td>-</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Routing packets per node and second</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Average size of routing packets</td>
<td>0</td>
<td>- -</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Average additional space for routing in each data packet</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Routing set-up time</td>
<td>- -</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Scale with the network size</td>
<td>0</td>
<td>- -</td>
<td>0</td>
<td>++</td>
<td>++</td>
</tr>
</tbody>
</table>

Table 3.4: Qualitative comparison of evaluated routing schemes.

prefers neighbors at a certain distance, GSR-PN, the protocol performs better even if feedback packets are sent less frequently.

3.4.7 Summary

In this section, we compare three protocols for routing from the Internet to mesh nodes in static and mobile scenarios. These protocols all represent different classes of routing strategies. Based on our findings, we then design and evaluate enhancements of two protocols.

In the simulation experiments, AODV-CGA, a reactive hop-by-hop routing protocol based on AODV, exhibits limited scalability to the network size. Furthermore, due to the high route setup time of AODV-CGA, the packet delivery ratio is rather low.

FBR, our proactive field-based routing protocol, outperforms all others with respect to the packet delivery ratio. However, FBR is not scalable to the network size. Furthermore, compared to the other protocols, FBR incurs the highest communication
overhead. With FBR-GW, an enhanced version of FBR, the communication overhead decreases slightly.

GSR, our gateway source routing protocol, delivers promising results. Its source routes gained from recorded routes of packets destined to the Internet prove to be quite reliable. GSR is scalable to the network size and has no route set-up delay (assuming that the receiver has sent at least one packet). We find that if the receiver sends a feedback packet toward the Internet host every five seconds, the packet delivery ratio remains high, even in scenarios where nodes move at car speed. With an enhanced version called GSR-PN, we achieve a higher packet delivery ratio in highly mobile scenarios.

We conclude that gateway source routing is a promising routing approach, since in our study, it delivers the best trade-off between packet delivery ratio, routing overhead, and scalability to the network size.

3.5 Simulation Study

In the previous sections, we present our routing solution for sending packets toward the gateways and back to the mesh nodes. In addition, we show how to make this solution robust to node mobility using our robust path metric. In this section, we combine these three approaches by using the robust path metric with HEAT. Using Glomosim, we compare the performance of HEAT combined with the robust path metric against OLSR and the extended AODV implementation presented in 3.2.4. To obtain highly meaningful results, we use a realistic city mobility model that we introduce in Section 5.2.
3.5 Simulation Study

3.5.1 Set-up

Our simulations are based on a WiFi network. All nodes are equipped with a 802.11b radio with a bandwidth of 11 Mbps and a nominal range of 250 meters. As MAC layer protocol, we use the 802.11 DFWMAC-DCF w/RTS/CTS [IEE07] and the two-ray ground propagation model [Rap96]. Due to the large network sizes we use, we were unable to model the effect of intermediate buildings in our city scenarios. However, we expect that the trends of our results also hold when such obstacles are present.

The actual node movement is modeled according to the city mobility model presented in Subsection 5.2. In each scenario, we model 1000 nodes and 5 gateways on an area of 5km by 5km in the city of Zurich. As data traffic model, we use the same Internet like mix presented in Subsection 3.2.4. The simulations have a duration of at least 10000 seconds and the results are always an average over at least 20 runs with different random seeds.

3.5.2 Results

We use the following metrics to compare the performance of HEAT using the robust path metric with AODV and OLSR.

- **Packet delivery ratio** - Ratio between the number of packets that are successfully received and the total number of packets sent. This metric includes all data packets sent from the mesh nodes to the gateways as well as packets from the gateways sent back to the mesh nodes.

In scenarios with nodes moving at pedestrian speed, the packet delivery ratio of our routing solution is around 99%. Looking at the results of OLSR reveals that its packet delivery ratio already decreases slightly for nodes moving at pedestrian speed, particularly in larger networks with longer routes. For
(a) Mobile scenario at pedestrian speeds.

(b) Mobile scenario at car speeds.

Figure 3.22: Packet delivery ratio.
AODV, the packet delivery ratio drops almost linearly with the number of nodes as depicted in Fig. 3.22(a).

Figure 3.22(b) shows the performance at car speed. Clearly, the performance of all three protocols is worse than at pedestrian speed. However, the packet delivery ratio of HEAT using the robust path metric remains around 80 percent whereas the ratio of OLSR drops to almost 40% and of AODV to 25% AODV for networks with 2000 nodes. These results show that HEAT combined with the robust path metric scales better to the network size and is at the same time robust to node mobility.

3.6 Demonstrator

In this section, we validate our scalable and robust routing protocol by a proof of concept implementation. We have implemented HEAT on Linux 2.6.20 with support for IPv4 as well as IPv6. The prototype implementation design of our demonstrator consists of a user space daemon and a kernel module. For gateways, an additional kernel module is required. We have evaluated our demonstrator in a small testbed with 10 nodes and up to 3 gateways.

3.6.1 User Space Daemon

The user space daemon consists of two main modules, a timing engine and a routing engine (see Fig. 3.23). The timer controls the periodic emission of the hello messages, and the aging mechanism for the neighbor table as well as the recorded routes in the gateway. The routing engine processes hellos, maintains the neighbor table and calculates the field. After executing the field calculation, the neighbor with the highest temperature value is stored as default next hop in the kernel routing table using netlink sockets.
3.6.2 Kernel Module for Mesh Nodes

The kernel module for mesh nodes has three major purposes (see Fig. 3.24). First, it forwards hello broadcasts from and to the user space daemon. Second, it forwards packets toward the gateways using the default route. At each hop, while forwarding, the module adds the forwarding node to a header that records the route. Third, the module forwards packets back to other nodes by using the next hop from the gateway source routing header.

3.6.3 Kernel Module for Gateways

The kernel module for gateways is more complex than the one for common mesh nodes (see Fig. 3.24). The module also forwards hello broadcasts from and to the user space daemon. But when a packet to the Internet is received, the header that records the

Figure 3.23: Demonstrator: User space daemon.
route in the wireless mesh network is removed and the route is stored in the route database. When a packet has to be forwarded to the wireless mesh network, the backward route is looked up in the backward route database. Then this route is added to a gateway source routing header and the packet is forwarded to the first hop of that route. An aging mechanism, driven by the user space timer daemon, maintains the backward route database. In addition, Network Address Translation (NAT) can be performed on all data packets.

---

**Figure 3.24:** Demonstrator: Kernel space module for nodes.

**Figure 3.25:** Demonstrator: Kernel space module for gateways.
3.6.4 Validation

To evaluate our demonstrator, we set up a small testbed consisting of 10 nodes. If connected by wire to the Internet, all of these nodes can be used as gateways with network address translation. For assessing the correctness of the routing functionality we set up several static and dynamic scenarios with failing nodes and gateways. Two exemplary scenarios are shown in Fig. 3.26 and 3.27. For the validation, we check the connection of a mesh node $A$ by permanently pinging a host in the Internet.

![Figure 3.26: Validation of demonstrator in a static scenario with node failure.](image)

![Figure 3.27: Validation of demonstrator in a dynamic scenario.](image)

3.7 Summary

In this chapter, we present a complete routing solution for scalable and robust routing in a wireless mesh network. We
present HEAT, a scalable and robust field-based anycasting routing protocol designed for routing traffic toward the gateways. Our analysis of different protocols for routing packets back from gateways to mesh nodes has shown that gateway source routing combined with our robust path metric is a promising choice. We perform separate evaluations (i) by theoretical considerations and (ii) by simulations for all contributions of this chapter. Furthermore, we evaluate a complete routing solution for wireless mesh networks comprising HEAT, gateway source routing and our robust path metric. In addition, we present a proof-of-concept implementation of this routing solution on Linux.
Chapter 4

Mobility Management

4.1 Introduction

For building city-wide wireless mesh networks, many gateways are required to provide enough bandwidth to the Internet. In large-scale wireless mesh networks, multiple gateways are attached to a multitude of different access networks (see Fig. 4.1). When a node in the wireless mesh network communicates with a node in the Internet, the IP packets are relayed through the mesh to any of the available gateways. As a result, when a node moves, routes in the mesh network might change and its IP traffic is forwarded to another gateway. If these two gateways belong to the same access network, we refer to this kind of mobility as \textit{micro mobility}, whereas if they belong to different access networks, we refer to it as \textit{macro mobility}. There are situations, where a node is attached to multiple gateways at a time. When these gateways belong to multiple access networks we refer to the node as \textit{multihomed} node [Hus05].

For enabling macro mobility and multihoming, several IP mobility management protocols and extensions have been proposed [Hen06,FT03,EN06]. They all require that a node is aware
of the access networks it is attached to. However, in a wireless mesh network the selection of the access network depends on the implemented routing and forwarding strategy. There are two possible mechanisms: service discovery or anycast routing. In the first case, a node uses a gateway discovery protocol to find neighboring gateways (see [RRGS05, WMP+06, JAL+00]). Based on this information a node decides which gateway to use for relaying packets to the Internet. Then, packets are sent to the chosen gateway by means of unicast. With anycast routing, a node leaves the choice of gateway to the routing protocol. A node only indicates that a packet should be sent to any gateway without specifying it (see [VP99, BHLMO07b]). The routing protocol then routes the packets in an anycast manner to one of the gateways. In the first case, a node knows which

Figure 4.1: A wireless mesh network connected to the Internet through different access networks.
gateway it relays its packets to and thus is aware of its macro mobility. However in the second case, the node is not aware of the selected access network and is hence not able to adapt to changes caused by its macro mobility.

But, since anycast is a very efficient means to implement gateway selection in wireless mesh networks, we aim at complementing this approach by adding a notification protocol. Specifically, we propose a notification protocol that is driven by the gateways and that is independent of the used routing service. In our approach, the gateway detects the macro mobility of nodes by monitoring the source addresses of packets sent to the gateway. If the addresses do not match the access network of the gateway and the node is not multihomed, the gateway sends a notification message to the sending node with the configuration information for its new access network. This node then adjusts its configuration accordingly. If necessary, the node also informs its communication peers about its new address. In cases where the nodes are multihomed, the gateway periodically informs the mesh node that some of its packets are relayed through its access network and performs a network address translation of the network prefix to fit the packet address to the routing topology. If necessary, the node also informs its communication peers about its additional locator address.

4.2 Micro Mobility

Micro mobility in city-wide wireless mesh networks is no specific issue. Unlike for handling macro mobility, handling micro mobility does not require any IP mobility management protocols. But the routing services of the access networks have to be capable to route packets to the designed gateways.
4.3 Macro Mobility and Multihoming

In this section, we describe our notification protocol for IPv6 which allows to handle macro mobility and which also supports multihoming independently of the used routing service. The proposed protocol is independent from existing IP mobility management protocol; hence the protocol enables nodes in a mobile mesh network to use MobileIPv6, HIP or similar IP mobility management protocols and its multihoming extensions.

First, we give an overview of the proposed protocol and specify the mobility notification message. Then, we explain how gateways detect macro mobility and how multihoming is supported. Following, we specify the handling of mobility notification messages at the mobile nodes as well as the procedure for node joins.

4.3.1 Protocol Overview

When a node sends packets to the Internet, the gateways detect macro mobility and multihoming of a node by means of the source address of the sent packets and a list of known nodes (see Fig. 4.2). If the address of a multihomed node is not known or the address of a non-multihomed node does not match the access network, the gateway sends a Mobility Notification Message (MNM) with the configuration information for its access network to the mobile node. A non-multihomed node then adjusts its configuration according to the mobility notification message. A multihomed node includes the new access network in its list of locators. The gateway periodically informs the nodes that some of the sent packets are relayed through the access network the gateway belongs to. Moreover, the gateway translates the network prefix of the source address of the packets going to the Internet to topologically fit the packets to the access network.
4.3.2 Mobility Notification Messages (MNM)

Mobility Notification Messages (MNM) are sent from gateways to mobile nodes to inform them about the access networks which are relaying their packets. Mobility notification messages are implemented based on ICMP [Pos81] router advertisement messages according to [NNS98] (see Figure 4.3). A mobility notification message contains two important information: (i) the notification interval for multihoming; and (ii) the prefix of the access network the sending gateway belongs to. The optimal choice of the notification interval depends on the mobility of the nodes as well on the amount of traffic sent. For moderate
mobile networks up to pedestrian speed, we propose to set the notification interval to a default value of 60 seconds.

For integrating security in the mobility notification message, we propose to use the authentication header described in [Ken05a].

\begin{figure}
\centering
\includegraphics[width=\textwidth]{mobility_notification_message.png}
\caption{Mobility notification message format.}
\end{figure}

### 4.3.3 Macro Mobility Detection and Multihoming Support at the Gateways

Gateways distinguish between mobile nodes supporting multihoming or not by looking at the network prefix of the nodes addresses. Nodes supporting multihoming always use a link-local address with the prefix \textit{FE80::/64} according to [TN98] while nodes not supporting multihoming use global addresses.

In the latter case, when the relaying access network of a node changes, the alteration is detected by the gateways of the new access network since the gateways constantly examine all packets they are relaying toward the Internet. If the source address of a packet that is topologically incorrect (i.e., the routing prefix
does not match the access network), the gateway sends a *Mobility Notification Message* to the sending node (see Fig. 4.4).

Processing of packets from multihomed nodes is more complex and requires the gateway to perform two tasks. First, the gateway has to verify if a node has recently been informed that its packets are relayed through this access network. If this is not the case, the gateway sends a mobility notification message to the mobile node to inform it about the actual access network. For reducing the amount of mobility notification messages, the gateway records the node address combined with a timestamp in a lookup table. After a *notification interval*, the gateway deletes the entry and if it is still relaying packets for this node, notifies the mobile node again.

Second, the gateway substitutes the link-local address prefix of the IP source address of the packet with the prefix of the access network it belongs to and forwards the packet to the Internet.

The forwarding algorithm for packets destined to the Internet at the relaying gateways is depicted in Fig. 4.5.

### 4.3.4 Handling Mobility Notification Messages at the Mobile Nodes

Handling of mobility notification messages is different at nodes supporting multihoming and those that do not support multihoming. When a multihoming node receives a mobility notification message, it adjusts its address prefix to topologically fit the new access network. Subsequently, it informs about its address change using its IP mobility management protocols. In the case where packets of a node are continuously forwarded over different access networks, multihoming support is an advantage to prevent continuous address changes.

When a multihoming node receives a mobility notification message, it checks if it already is aware of that access network. If this is not the case, it informs its communication peers about its
new locator using the multihoming extension of the IP mobility management protocol. Again, we distinguish two methods to detect if an access network does no longer relay packets for a mobile node. First, a communication peer informs a mobile node that it is no longer reachable over a certain access network. Second, a mobile node keeps a list of its relaying access networks with the time stamp of the last mobility notification message received from this access network. From time to time, the mobile node checks its list for outdated access networks. The appropriate choice for the \textit{MNM time out} highly depends on the mobility message notification interval of the gateways, the amount of traffic sent and on the mobility of a node. For moderate mobile

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{A gateway detects a packet with a topologically incorrect routing prefix (1). It sends a mobility notification message to the sending node (2). This node then updates its address (3).}
\end{figure}
4.3 Macro Mobility and Multihoming

Figure 4.5: Algorithm for processing packets destined to the Internet at the relaying gateway.

networks, we set the MNM time out to a default value of 3 times the notification interval.

The algorithm for handling mobility notification messages at mobile nodes is depicted in Fig. 4.6.

4.3.5 Joining of a Mobile Node

When a node joins a mobile mesh network, it automatically configures its address according to [TN98] as a link-local address if it supports multihoming, otherwise as a site-local address. These addresses use a specific prefix and an interface identifier as suffix which is derived from the Ethernet address (the prefix \textit{FE80::/64} for link-local addresses and \textit{FEC0::/64} site-local address). Using the automatically configured address, the node
Figure 4.6: Algorithm for handling mobility notification messages at a mobile node.

immediately participates in the mobile mesh network and no further initialization is required.

4.4 Discussion

In this section, we want to briefly discuss two issues coming up when deploying the proposed protocol: supporting of secured connections and making macro mobility transparent to routing protocols.
To support secured connections, only multihomed nodes have to be considered. For those nodes, a problem occurs when IPsec authentication headers [Ken05a] are used, since the gateways have to change the (outer) IP header of a packet. Note that IPsec encapsulating security payload [Ken05b] is supported since the encryption and authentication is not applied to the (outer) IP header.

Macro mobility is not transparent to routing protocols for wireless mesh networks, because they use the entire IP address as a unique identifier for routing. They do not have any support for nodes which change their address as required for macro mobile nodes. Thus, an address change is treated as a node leave and join which creates unnecessary overhead. A possible solution is that routing protocols for wireless mesh networks only use the interface identifier as identifier for routing. In addition, such a mechanism also reduces routing overhead and storage requirement.

4.5 Summary

There are scenarios where nodes in a wireless mesh network are unaware of the access network that relays their packets. For these scenarios, we propose a detection mechanism and a notification protocol supporting multihoming which informs the nodes about their macro mobility and thus about the access networks they are using.
Chapter 5

Realistic Mobility Models

5.1 Introduction

Today, the evaluation of protocols and algorithms for wireless mesh and vehicular ad hoc networks is based mostly on theory or simulation experiments as large-scale testbeds would be too expensive to be practical. However, high predictability can be expected from simulation studies only if the mobility model resembles reality closely.

In this chapter, we present two novel mobility models. The first mobility model uses vectorized street information, including speed limit data, from the Swiss Geographic Information System (GIS) [Fed]. The second model is based on traces from a leading microscopic, multi-agent traffic simulator. Both models have advantages and limitations. The trace-based model imitates reality very closely, but requires a huge amount of computing power for the generation of traces. In contrast, the GIS-based model requires only little computing power but it does not
produce scenarios that are as close to reality as those of the trace-based model.

Later in this chapter, we investigate the influence of the choice of the mobility model on the performance of routing. We consider two popular routing protocols AODV [PBRD02] and GPSR [KK00]. These routing protocols represent two major classes of ad hoc routing protocols: AODV represents reactive, non-geographic routing, whereas GPSR represents geographic routing with greedy forwarding. Both AODV and GPSR are well documented, have been tested in many research studies and are known as good performers in their classes of routing protocols.

Our results indicate a significant drop of the packet delivery ratio with our presumably more realistic models compared to the delivery ratio achieved with the random waypoint model.

### 5.2 Mobility Model using GIS-data

With the mobility model based on GIS-data, we aim to mimic the movement of pedestrians and vehicles. To this end, we constrain the movement of the nodes to streets. We extract street maps of real Swiss cities from the Swiss Geographic Information System (GIS) [Fed]. This database includes vectorized building and street maps including speed limit information and many other data. As an example, the vectorized maps of the city centers of four large Swiss cities are displayed in Fig. 5.1. A major advantage of this data source is its high level of detail. The precision of the street geometry is 1 meter, thus even small turns and the number of lanes are mapped. Another advantage is the availability of an elevation profile as well as additional information about the maximum velocities.

In our GIS-based mobility model, the actual node movement is generated according to the steady-state random trip mobility model [BV05] on the vectorized street maps as follows. A node
chooses a random destination in the city and moves to this position using the fastest available path, i.e., the shortest path in terms of time. In order to determine this path, we calculate a weighted shortest path \[\text{AMO93}\] whose edges are weighted according to the time required to pass through a street segment at the maximal allowed speed. The model captures the behavior of pedestrians as well as cars since both move along the streets of a city. We generate three scenarios: (i) a static scenario for model verification and benchmarking, (ii) a scenario with mobile nodes moving at pedestrian speeds (i.e., node speeds that are uniformly distributed between 1 m/s and 4 m/s), and (iii) a scenario with nodes moving at car speeds according to the maximal allowed speed on that street segment. In order to account for cars that do not move at the maximal speed, the speeds are uniformly distributed between 75% and 100% of the speed limit, corresponding to speeds of approximately 10 m/s–20 m/s.

5.3 Mobility Model using Multi-agent Microscopic Traffic Simulator

Since real vehicular traces are not available, a traffic simulator can be used to generate the movement of vehicles. However, driver behavior on a road is very complex since it depends on the environment. Drivers react to changing road conditions such as for instance congestion, which in turn depend on the drivers’ plans and behavior. Furthermore, weather and other environmental factors also influence the behavior of the individuals. Thus, the choice of the traffic simulator in the end determines the relevance and viability of the obtained results. Vehicular traffic simulators can in general be classified into microscopic and macroscopic simulators. A macroscopic simulator considers such system parameters as traffic density
Figure 5.1: Vectorized street maps of four Swiss cities.

(number of vehicles per km per lane) or traffic flow (number of vehicles per hour crossing some point, usually an intersection) to compute road capacity and the distribution of the traffic in the road network. From the macroscopic perspective, vehicular traffic is viewed as a fluid compressible medium, and therefore is modeled based on an equation deduced from the Navier-Stokes equations [LL87]. In contrast, microscopic simulators determine the movement of each vehicle that participates in the road traffic. Thus, a microscopic traffic simulator is potentially a better choice for our research.

The Multi-agent Microscopic Traffic Simulator (MMTS) [RCV+03, RVC+02] developed at ETH Zurich is capable of simulating public
and private traffic over real regional road maps of Switzerland with a high level of realism. MMTS models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment. For instance, one individual might board their car and go to work in the early morning, another one that gets up later goes shopping using public transportation, etc. Travel plans are made based on road congestion; congestion in turn depends on the travel plans. To resolve this situation, a standard relaxation method is used [Mat92, Pre92].

The street network that is used in MMTS was originally developed for the Swiss regional planning authority (Federal Office of Spatial Development). The major attributes of each road segment are type, length, speed, and capacity. The street network is simulated on a Beowulf [BM02] Pentium cluster of over 30 CPUs. With the help of MMTS, the consequences of construction sites, road modifications, new roads, etc. can be simulated and potential economical influence (e.g., travel time and price changes for public and private transport) can be estimated.

For generating vehicular mobility traces, we use a 24-hour, detailed car traffic trace file generated by MMTS. The file contains detailed simulation data of a certain geographic region in the canton of Zurich. This region includes the part where the main country highways connect to the city of Zurich, the largest city in Switzerland. Around 260'000 vehicles are involved in the simulation with more than 25'000'000 recorded direction/speed changes in an area of around 250 km × 260 km (see Figure 5.2).
The car traffic simulator file describes in XML format the step by step movement of each vehicle within the 24-hour time period. To use that data we convert the XML input from the car traffic simulator into simulator movement pattern. However, the file resulting from car traffic simulator movement pattern contains too many nodes (vehicles) to be processed by any available network simulator. Thus to allow simulation on a network simulator, we select smaller subregions that are suitable for simulations as shown in Figure 5.3. The selected region contains the 24 hour movement pattern of all the vehicles that travel inside or through a region. As a next step we capture different levels of activity of vehicles in the region (the number of events per time slot) and select three time periods that correspond to high density rush hour (more than 50 vehicles per km of road), medium density (30-40 vehicles/km), and low vehicle density (less than
15 vehicles/km). Each period has a length of 0.5 hours. Finally, from each of these periods, we create 10 scenario files of 300 seconds, each shifted by 150 seconds from the previous one.

![Visualization examples of different inner city regions of Zurich.](image)

**Figure 5.3:** Visualization examples of different inner city regions of Zurich.

In Figure 5.4 a screen-shot of mobile nodes (vehicles) moving about the center of Zurich-city is superimposed above a map of the region.

### 5.4 Simulation Study

We now discuss the influence of the chosen mobility model on the performance of two popular routing protocols: AODV and GPSR. For this comparison, we use the two proposed mobility models as well as the random waypoint mobility model. Before presenting the results, we describe the simulation and mobility model set-up.
5.4.1 Simulation Set-up

The literature shows that the results of performance studies of ad hoc network depend heavily on the chosen mobility model [YLN03, BRS03, CBD02]. To allow our study to be compared with prior work, we investigate AODV and GPSR\(^1\) based on the random waypoint mobility model [BMJ+98] and compare the results with our mobility models using release 2.31 of the ns-2 [ns202] network simulator. For the simulations, we use the default settings for the GPSR and AODV routing protocols.

To gain more realistic signal propagation than with the deterministic free space or two-ray ground reflection model, we use the shadowing model for radio propagation [Rap96]. According to real world experiments with inter-vehicle communication [EDS05, SBSC02], we choose \(\beta = 3.5\) and \(\sigma_{dB} = 6\). All nodes are

---

\(^1\)The code for GPSR is taken from [KK00].
equipped with a 802.11b radio with a bandwidth of 11Mbps and a nominal range of 250 meters. As MAC layer protocol, we use the 802.11 DFWMAC-DCF w/RTS/CTS [IEE07].

Traffic sources transmit data at a fixed data rate. We are not interested in the maximum achievable throughput in VANET; we rather want to investigate the influence of the different mobility models on the performance, i.e. on the packet delivery ratio. Thus we consider randomly 25 source-destination pairs with a sending rate of 100 kbit/s. The sources start generating data packets within the first 50 s of the simulation time and stop generating data packets 50 s before the simulation ends to avoid transient effects on our results.

5.4.2 Set-up of Mobility Models

For the random waypoint model we simulate an area of $5 \text{ km} \times 5 \text{ km}$ populated by 1100 nodes that move about the area. The node speeds vary between 15 km/h and 55 km/h. The pause time equals 0 s (constant motion), 200 s and 300 s. For the mobility model using GIS-data, we select the same area size, the same number of nodes using the scenario with nodes moving at car speed and the same pausing times as for random waypoint. Also for the vehicular traces we select the same area size representing the city region. We identify the 300 s time interval when the selected area is populated by at least 1000 nodes and at most by 1200 nodes. Pause time is modeled like follows: for 300 s pause time (static picture) we cancel all node mobility; for 200 s pause time we keep nodes static for 200 s and then allow movement; for 0 s pause time no restrictions are applied.

5.4.3 Results

For measuring the performance of routing protocols, many different evaluation metrics could be used. Following, we focus...
on a very popular, the packet delivery ratio. Figures 5.5 and 5.6 show the packet delivery ratio of AODV and GPSR with the random waypoint model and with our more realistic models. The performance of both protocols shows noticeable dependence on the chosen mobility model. With the random waypoint model, GPSR outperforms AODV; with the vehicular traces the situation changes to the opposite (see tables 5.1 and 5.2). Using the proposed mobility models, the packet delivery ratio of both protocols turns out to be much lower than in the corresponding random waypoint model scenarios. Following, we explain how these performance problems occur.

The performance problems of AODV can be ascribed to uncontrolled or blind flooding. This uncontrolled flooding generates many redundant transmissions, which may cause the so called broadcast storm problem [YCSYYSJP02]. The performance problems of GPSR can be ascribed to inconsistencies in the neighbor tables. As described in [KLH03], these inconsistencies lead to significant problems and low throughput. We further address these performance problems of AODV and GPSR in [NBG06].

Comparing the two proposed mobility models shows that the performance using the GIS-data based model is close to the performance of the model based on the traffic simulator traces. Since the GIS-data based model requires much fewer computing resources for generating mobility patterns than the trace-based model and still provides quite similar results, it provides a viable alternative.

Our analysis of the degraded packet delivery ratio compared to random waypoint has revealed the following main causes.

1. We found that with both realistic models, the average path length (in number of hops) is around 1.3 to 2 times longer than with the random waypoint model. This increased path length is due to the restriction of the node movements
to the roads, which leads to routes with detours. These detours increase the average path length, which incurs a higher packet loss ratio.

2. With the random waypoint mobility model, nodes move along a straight line from a starting point to a destination at a constant speed. In contrast, the two novel models force nodes to change their direction and speed more frequently. This leads to an increased link breakage ratio, mainly for nodes close to the maximal communication range, and also increases the packet loss ratio.

<table>
<thead>
<tr>
<th>Pause time</th>
<th>Random waypoint</th>
<th>City model GIS-data</th>
<th>City model vehicular trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 s</td>
<td>AODV 99</td>
<td>AODV 42</td>
<td>AODV 34</td>
</tr>
<tr>
<td>200 s</td>
<td>AODV 90</td>
<td>AODV 40</td>
<td>AODV 29</td>
</tr>
<tr>
<td>0 s</td>
<td>AODV 83</td>
<td>AODV 37</td>
<td>AODV 27</td>
</tr>
</tbody>
</table>

**Table 5.1:** Packet delivery ratio of AODV.

<table>
<thead>
<tr>
<th>Pause time</th>
<th>Random waypoint</th>
<th>City model GIS-data</th>
<th>City model vehicular trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 s</td>
<td>GPSR 99</td>
<td>GPSR 25</td>
<td>GPSR 13</td>
</tr>
<tr>
<td>200 s</td>
<td>GPSR 93</td>
<td>GPSR 24</td>
<td>GPSR 12</td>
</tr>
<tr>
<td>0 s</td>
<td>GPSR 88</td>
<td>GPSR 22</td>
<td>GPSR 12</td>
</tr>
</tbody>
</table>

**Table 5.2:** Packet delivery ratio of GPSR.
Figure 5.5: Packet delivery ratio of AODV.

Figure 5.6: Packet delivery ratio of GPSR.
5.5 Summary

In this chapter, we have proposed two novel mobility models that model reality more closely than simple models such as random waypoint. Our first model is based on street maps with speed limit information from the Swiss geographic information system. The second model is based on realistic traces from a microscopic multi-agent traffic simulator. The latter model provides potentially more realistic mobility patterns. However, it requires much more computing resources than the first model.

Our evaluation has validated the finding that performance results of wireless mesh networks as well as vehicular ad hoc networks depend highly on the underlying mobility model. The simple random waypoint mobility model tends to considerably overestimate the performance of the routing protocols that we used for our evaluation (AODV and GPSR).
Chapter 6

Conclusions and Future Work

In this chapter, we summarize our main results and contributions and conclude our work on wireless mesh networks. Following, we describe three research projects inspired by our research. Finally, we discuss some open research issues for further investigation.

6.1 Summary and Conclusions

In this thesis, we propose a sophisticated routing solution to build wireless mesh networks. We show that the proposed solution meets the requirements with regard to scalability to the number of nodes as well as gateways. The solution also fulfills the requirements regarding robustness to node mobility. In addition, we propose a mobility detection and notification protocol that allows supporting macro mobility and multihoming using common IP mobility management protocols independent of the routing service used. For the evaluation of our proposals by simulations, we develop realistic mobility models based on geographical data and traces from vehicular simulations. In
addition, we validate our scalable and robust routing protocol by a proof of concept implementation in a demonstrator for Linux.

For routing between wireless nodes and Internet gateways in wireless mesh networks, we propose HEAT, an anycast routing protocol that is based on temperature fields. Our protocol makes use of local beacon exchanges to establish routing state and is thus particularly scalable to the network size and suitable for large and dense networks. Furthermore, the temperature field accounts for the link diversity and redundancy toward the gateways, which makes our protocol particularly robust. We show the scalability and robustness of HEAT by a simulation study as well as theoretical considerations. We present a comparison of HEAT with AODV and OLSR based on simulations. In large networks with nodes moving at car speed, HEAT outperforms AODV as well as OLSR in terms of the packet delivery ratio by a factor of two.

We present an investigation of three different protocols for routing packets from the Internet to a wireless mesh node. We show by simulation that gateway source routing using recorded routes from packets destined to the Internet is a promising solution. It is scalable with respect to the network size and has no route set-up delay. We found that, if routes are no older then 5 seconds, the packet delivery ratio remains high, even in mobile scenarios where nodes move at car speed.

We introduce a new link cost metric for mobile wireless networks. Furthermore, we propose a path metric that allows selecting robust paths based on our link metric. Compared to shortest paths, the paths selected by our metric exhibit a lower packet loss ratio and a lower break probability because they consist of more reliable links. Our link cost metric gives preference to links that offer a good compromise between length and reliability using a cost function. Every node assigns its neighbors a link cost that depends on how close to its preferred
signal strength indicator the neighbor’s RSSI value is. Based on this link cost, our path metric then selects the most robust path.

For investigating our proposals by simulation, we have developed two realistic mobility models using vectorized street maps as well as vehicular simulation traces. In addition, we have built a prototype implementation to show that the proposed routing solution is practical.

Finally, we propose a mobility detection and notification protocol for mesh nodes that are unaware of the access network that relays their packets. This protocol enables multihoming as well as macro mobility and thus allows using multiple access networks independent of the used routing service.

6.2 Weaknesses and Shortcomings

In this section we address two weaknesses and shortcomings of this thesis.

The scalable and robust routing protocol has been evaluated in detail by simulation but not in a real city-wide wireless mesh network. The demonstrator has been used for validating the protocol while the performance analysis has been done by simulation. However, based on the results gained from simulations using realistic mobility models, we expect that our protocol performs well in real city-wide wireless mesh networks.

The detection and notification protocol for managing macro mobility and multihoming in wireless mesh networks is designed for IPv6 only. For adapting it to work with IPv4 two additional mechanisms would have to be specified: (i) an auto address configuration mechanism similar to [TN98] and (ii) a mechanism to distinguish nodes supporting multihoming from nodes which do not. The major problem with such an auto address configuration mechanism for IPv4 is that assigned addresses must be unique
and that the nodes in an access network have to be tracked because of the limited number of available addresses.

6.3 Future Work and Outlook

6.3.1 Link-Diversity Routing

In this thesis, we present HEAT, a scalable anycast routing protocol for wireless mesh networks. HEAT is inspired by the properties of temperature fields and aims at constructing robust routes. This idea has led to a novel routing paradigm. In [BL07], we apply this routing paradigm and the properties of thermal physics to unicast routing in wireless mobile ad hoc networks. We design and implement link-diversity routing, a robust unicast routing paradigm that increases the packet delivery probability in networks with frequent link failures. Link-diversity routing incorporates the amount of alternative paths along the delivery path in the forwarding decision and thus manages to route around broken links more often than traditional minimum hop-count routing. We propose a routing model based on heat theory and an according routing algorithm to implement link-diversity routing in wireless ad hoc networks. Through simulations, we show that in the studied networks, link-diversity routing manages to increase the probability of successful packet delivery by a factor of up to 2 compared to minimum-hop count routing. This is achieved without considerably increasing the path length and the control message overhead of shortest path routing.

6.3.2 Toward Realistic Simulations

In Chapter 5, we propose a city mobility model based on the Swiss geographic information system. We aim at further extending the use of this geographic data toward highly realistic simulations. For this purpose the calculation of the signal propagation as
well as the mobility model itself should be extended. The signal propagation should be extended to consider the scenery, mainly buildings but also forests and other obstacles. The mobility model should be extended with an agent-based behavior model. In addition to pedestrian and car mobility, it should also consider public transport systems such as trains, buses and trams as well as fixed nodes in buildings. Finally, we plan to incorporate these realistic models with simple models such as random waypoint in a public web-based trace generator. The aim of this generator is to make simulations better comparable by enabling multiple research groups to use exact the same mobility models.

### 6.3.3 Transport Protocols for Wireless Mesh Networks

Recent research has shown that reliable end-to-end transport protocols such as TCP perform poorly in wireless mesh networks, primarily due to their inability to cope with the dynamics incurred by node mobility. We reconsider the design decisions that lead to the end-to-end design of the transport layer. To this end, we present and evaluate a framework for reliable hop-by-hop transport protocols in [HBMP07, HBMP06]. Furthermore, we plan to investigate the transport layer task analytically. We aim at deriving upper bounds for both end-to-end and hop-by-hop transport in mobile wireless networks.

### 6.3.4 Other Issues

In addition to the projects mentioned above, we think that the following open research issues should be addressed

- How to build large-scale testbeds for wireless mesh and vehicular ad hoc networks. Even with very realistic simulations, there is always a gap to reality. Thus, real testbeds are required for final evaluations.
• Evaluation of the robust path metric presented in Section 3.3 in a real-world study.

• Implementation of the mobility detection and notification protocol presented in Chapter 4.
This appendix provides several extensions to Chapter 3. First of all, we present the terminology of field-based routing, we developed during our work on this thesis. Then we provide information about the support of different routing metrics for HEAT. Following, we briefly present a visualization tool for HEAT and other field-based routing protocols. And finally, we present our evaluation of parameters of HEAT.

A.1 Terminology of Field-based Routing

In this thesis, we often argue with field-based routing. Hence, we would like to define a terminology. Therefore, we establish a relation between the well-known concept of field theory in physics and routing in IP networks. Key to understanding this relation to field theory is to understand that we need to distinguish between the field dynamics itself and the effect of a field on a probe. We argue that field dynamics corresponds to the field construction and the effect of the field on a probe to packet routing (see Fig. A.1).

Additionally in [Bau06], we show how to use this terminology for a unifying classification of the numerous IP routing protocols.
A.1.1 Field Construction

The purpose of field construction is to prepare all necessary information for packet routing. These are namely the exchange of all necessary information using the field propagation mechanism and the calculation of the routing field.

Field Propagation

The purpose of field propagation is to exchange the information about the network state, measured using routing metrics. These metrics can be as straightforward as hop count or composed from (i) several routing metrics like ETX, or (ii) a complete network topology. The information exchange can be done in a proactive, reactive or also in a hybrid way and the propagation range can vary from neighboring (one hop) up to global (unlimited number of hops).
A.2 Support for different Routing Metrics for HEAT

Field Calculation

The field calculation function uses the known current network state to calculate a routing field at a node.

A.1.2 Packet Routing

The packet routing is responsible for the handling of the packets. When it receives a packet it determines the next hop using the route selection. Following the packet is handed over to the packet forwarding for forwarding to the determined next hop.

Route Selection

The purpose of the route selection is to determine the next hop of a packet using the route selection function and the routing field.

A.2 Support for different Routing Metrics for HEAT

In this section, we explain how to apply almost every routing metric with field-based routing or more specifically with HEAT.

Routing metrics are used to determine routing paths according to some optimization goal as for example the shortest path, the minimal delay, or the maximal bandwidth. The challenge is to properly encode these routing metrics into a routing field. Important is, that the resulting field rises strictly monotonically.

For this purpose we categorize routing metrics into the following three groups: additive, multiplicative, and concave metrics. For the discussion of these groups, we focus on a node i and its neighbor n. \( \phi(x) \) is the field intensity at node x and \( d_{x,y} \) is the value of the routing metric for the link between node x and y.
1. **Additive metrics**: e.g. delay, hop count, ETX; \( \phi(i) = \phi(n) - d_{i,n} \). In order to guarantee strict monotony of the potential field, all routing metric values must comply with \( d_{i,j} > 0 \).

2. **Multiplicative metrics**: e.g. packet loss ratio, packet reordering ratio; \( \phi(i) = d_{i,n} \cdot \phi(n) \). In order to guarantee strict monotony of the potential field, all values of the routing metric must be normalized in order to comply with \( 1 > d_{i,j} > 0 \).

3. **Concave metrics**: e.g. bandwidth, connectivity, minimal remaining power; \( \phi(i) = \min(\phi(n), d_{i,n}) - \varepsilon \). The constant \( \varepsilon \) is set to the smallest possible value. It preserves the strict monotone rise of the potential function.

In addition to these classification of routing metrics, we present in [BHSW06] an extensive survey on routing metrics.

### A.3 Visualization

The concept of field-based routing can be visualized very well. But when working with real routing fields, the imagination is highly limited. Dumps of routing fields and trajectories of data packets fill uncountable many lines with heaps of numbers. Trials to get a picture of what is going on are from the outset in condemns to failure. Thus, we have developed a visualization tool for analyzing routing field and data packet dynamics in detail. A screen shot of this tool is printed in Fig. A.2.
A.4 Selection of Parameters for HEAT

In this Appendix, we present an initial set of experiments to find appropriate values for the timing parameters of HEAT. These parameters are:

- HEAT beacon interval,
- HEAT beacon timeout,
- delay of early HEAT beacon,
- and the conductivity value $\kappa$.

Figure A.2: Screen shot of the visualization tool for field-based routing protocols.
For these simulations, we use a static scenario of 1000 mesh nodes and 5 gateways. All nodes (including the gateways) are randomly distributed over an area of $5000 \times 5000 \text{ m}^2$.

Determining the HEAT beacon interval is a classical trade-off between communication overhead and convergence time. In order to measure the convergence time, we wait until the routing has converged and then eliminate 10% of all nodes (mesh nodes and gateways). The convergence time is then determined as the time it takes until the routing topology has converged again. Note that we consider the routing as converged as soon as there are no more route changes. Figs. A.3 and A.4 plot the convergence time and the routing overhead vs. the HEAT beacon interval. As expected, the convergence time increases linearly with the HEAT beacon interval. Also, the overhead decreases with an increasing HEAT beacon interval. As a result, we set the HEAT beacon interval to 1 second and the HEAT beacon timeout equal to $3 \times$ HEAT beacon interval. This setting of the HEAT beacon interval allows up to two HEAT beacon messages to be lost before a link is considered to be down. Further, we set the delay for early HEAT beacon messages to two times the backoff interval of the 802.11 MAC layer, which results in an early HEAT beacon delay of 20 ms. Longer delays would have a negative impact on the convergence time (see Fig. A.5). Shorter delays would lead to a greater number of regular HEAT beacons (see Fig. A.6). We set the conductivity value $\kappa$ for the temperature field calculation function to $1/4$. This value has shown to provide the best tradeoff between the convergence time, the communication overhead, and the robustness from the path and link redundancy.
Figure A.3: Parameter selection: Convergence time versus hello interval.

Figure A.4: Parameter selection: Per node convergence time versus hello interval.
Figure A.5: Parameter selection: Convergence time versus delay of early hellos in backoff intervals.

Figure A.6: Parameter selection: Hellos sent per second versus delay of early hellos in backoff intervals.
Bibliography


[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. Network Flows: Theory,


[BHSW06] Rainer Baumann, Simon Heimlicher, Mario Strasser, and Andreas Weibel. A Survey on


In *Proc. IEEE Workshop on Wireless Mesh Networks (WiMesh)*, Santa Clara, USA, sep 2005.


152

BIBLIOGRAPHY


[RVC+02] Bryan Raney, Aandreas Voellmy, Nurhan Cetin, Milenko Vrtic, and Kai Nagel. Towards a Microscopic Traffic Simulation of All of


[Sto02] Panos Stokas. OGG vs. MP3 vs. WMA vs. RA, 2002.


BIBLIOGRAPHY


[ZBG98] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel
Simulation of Large-scale Wireless Networks. In Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS '98), Banff, Alberta, Canada, May 1998.

Acknowledgements

The success of a dissertation depends on the support of many people, whom I would like to thank at this point.

First, I would like to thank my advisor Prreector Prof. Dr. Bernhard Plattner for his continuous support, the stimulating working atmosphere, the huge amount of freedom and the challenging time in his laboratory. I am also very grateful to my co-advisor Prof. Dr. Ralf Steinmetz from the Technical University of Darmstadt.

Special thanks go to Dr. Vincent Lenders for his critical questions, his tireless help and for giving me the opportunity to base my work on his. I am also very grateful to Dr. Martin May for his ongoing support during my dissertation. Then, I also would like to say thank you to Dr. Ulrich Fiedler, my advisor during the first year in the laboratory and to Dr. Merkourios Karaliopoulos.

Very special thanks go to Simon Heimlicher, my first master student and later on my major collaborator in the laboratory. I will never forget the hours we spent together writing papers and discussing ideas. Another thank you goes to my second room mate Mario Strasser for the good collaboration and interesting time.

I also want to thank my colleagues from the laboratory Marcel Baur, Andreas Bleuler, Matthias Bossardt, Daniela Brauckhoff,
Acknowledgements

Martin Burkhart, Thomas Dübendorfer, Matthias Dyer, Karoly Farkas, Stefan Frei, Theus Hossmann, Konstantinos Katrinis, Ariane Keller, Michael Kuhn, Andreas Meier, Georgios Parissidis, Dominik Schatzmann, Stefan Schmid, Bernhard Tellenbach, Arno Wagner, and Nathalie Weiler for the great time we had together. Another thanks goes to all the professors in the laboratory Prof. Dr. Lothar Thiele, Prof. Dr. Roger Wattenhofer, Prof. Dr. Eckart Zitzler and to the service group Julia Breddermann, Alain Breedt, Hans-Jörg Brundiers, Monica Fricker, Beat Futterknecht, Catrina Sposato, and Thomas Stein-gruber.

I would also like to acknowledge the contributions from students doing a semester or master thesis under my supervision: Sebastian Altorfer, Giacomo Amoroso, Olga Bondareva, Gian-matteo Costanza, Severin Hafner, Thomas Hug, Ariane Keller, Niels Kistler, Till Kleisli, Pascal Mathis, Martin Pfister, Tobias Reint, Philipp Sager, Rafael Schönenberger, Ting Jia Shi, Philipp Sommer, Jelena Stamenkovic, Michael Steiger, Sacha Trifunovic, and Andreas Weibel.

Then I would like to add a thanks to my master mentor Valeri Naoumov. Additional thanks goes to Kai Nagel, Bryan Raney, and Hinnerk Spindler for providing us with the car traffic simulator output and for their support of our work.

Finally, I would like to thank my family for their unconditional and untiring support during all these years as well as my girl friend for the relaxing time we spent together.
Rainer Baumann was born in Zurich on the 30th of March 1979. He is a citizen of Bubikon ZH and Langnau am Albis, Switzerland. He grew up in Langnau am Albis, where he attended primary school. From 1994 on, he attended the Kantonsschule Rämibühl in Zurich from where he obtained his Matura degree in mathematics and natural sciences in 1999. During this time, he worked for Kyburz Zug as well as Ingeno and started his own small IT business. After receiving the Matura, he served for 4 months in the Swiss army as an information technology pioneer in the electronics war and communications division. Afterwards, he joined UBS Warburg for half a year as a computer specialist.

In autumn 1999, Rainer Baumann started his studies in computer science at the Swiss Federal Institute of Technology, ETH Zurich. During his studies, he established his own business as an IT consultant and worked for half a year at the world’s largest particle physics laboratory CERN in Geneva. In autumn 2004, he obtained his Masters of Science degree in Computer Science with a major in software engineering and a minor in physics. In parallel with his master degree, he finished his Master of Advanced Studies in Secondary and Higher Education. After his graduation, he was employed for two months at the Kantonsschule Bülach as a teacher in mathematics.

In autumn 2004, Rainer Baumann joined as a PhD student the Computer Engineering and Networks Laboratory at ETH
Zurich. There, he worked on two industry technology transfer projects in the area of telecommunication and exerted his position in teaching. He focused his research on mobile, wireless, mesh, and ad hoc networking. During his doctoral studies, he passed the exams for the Cisco Certified Network Professional (CCNP) and the Certified Information Systems Security Professional (CISSP). In autumn 2005, he started as a lecturer at the Applied University of Rapperswil, HSR. In autumn 2007, he received his Doctor of science degree from the Swiss Federal Institute of Technology, ETH Zurich.