Technische Universität München
Fakultät für Informatik
Lehrstuhl für Netzarchitekturen und Netzdienste

Simulation of the "Better Approach to Mobile Adhoc Networking" Protocol

Bachelorarbeit in Informatik

durchgeführt am
Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

von
Fabian Oehlmann

September 2011

Technische Universität München
Fakultät für Informatik
Lehrstuhl für Netzarchitekturen und Netzdienste

Simulation des "Better Approach to Mobile Adhoc Networking" Protokolls

—

Simulation of the "Better Approach to Mobile Adhoc Networking" Protocol

Bachelor's thesis in Computer Science

written at
Chair for Network Architectures and Services
Faculty of Computer Science
Technische Universität München

by
Fabian Oehlmann

Supervisor:	Prof. Dr.-Ing. Georg Carle
Advisor:	Dr. rer. nat. Alexander Klein
Submission Date:	September 22^{nd} 2011

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

I assure the single handed composition of this thesis only supported by declared resources.

Garching, September 22^{nd} 2011

Abstract:

Apart from military applications, multi-hop adhoc networking emerged in several civil applications such as community mesh networks and wireless sensor networks during the last decade. Ever increasing data rates of wireless devices render adhoc networks a feasible alternative to classic infrastructure based networks. Moreover, the growing number of WLAN equipped mobile phones opens up new potentials of networking. A crucial factor for the performance of an adhoc network is the utilized routing protocol. Therefore the "Better Approach to Mobile Adhoc Networking" protocol (B.A.T.M.A.N.) was developed by the open source community. The current version of this protocol introduces its own routing metric, addressing the problem of integrating asymmetric links efficiently in the network topology. In line with this thesis, the two least current versions of the protocol have been reimplemented for simulation in the OPNET network modeler. They have been tested and compared in terms of packet delivery ratio, average end-to-end hop count and generated overhead in several scenarios. Furthermore, an evaluation of their ability to account for asymmetric links has been conducted.

Kurzfassung:

Abseits von militärischen Anwendungen kamen Multihop Adhoc Netzwerke in den letzten zehn Jahren auch in einigen zivilen Anwendungen, wie etwa gemeinschaftlichen Mesh Netzwerken oder Sensor Netzwerken, auf. Immer weiter zunehmende Datenraten drahtloser Geräte machen Adhoc Netzwerke zu einer brauchbaren Alternative gegenüber klassischen infrastrukturbasierten Netzwerken. Die wachsende Anzahl mit WLAN ausgestatteter Mobiltelefone eröffnet außerdem neue Möglichkeiten sich zu vernetzen. Ein kritischer Faktor für die Leistung eines Adhoc Netzwerkes ist das verwendete Routing Protokoll. Deshalb wurde das „Better Approach to Mobile Adhoc Networking" Protokoll von der Open Source Community entwickelt. Die aktuelle Version des Protokolls führt eine eigene Kostenmetrik ein, die sich des Problems annimmt, asymmetrische Verbindungen effizient in die Netzwerktopologie einzubinden. Im Rahmen dieser Arbeit wurden die letzten beiden Protokollversionen für die Simulation im OPNET Network Modeler neuimplementiert. Sie wurden im Hinblick auf ihre Übertragungszuverlässigkeit, die Anzahl durchschnittlicher Ende-zu-Ende Hops und den generierten Overhead untersucht und verglichen. Weiterhin wurde eine Einschätzung der verwendeten Mechanismen zur Erkennung asymmetrischer Verbindungen vorgenommen.

Contents

1 Introduction **2**

2 Related Work **4**

2.1 Version III Performance Studies 4

2.2 Version IV Performance Studies 5

3 Routing in MANETs **6**

3.1 Mobile Adhoc Networks 6

3.2 Competing Routing Protocols 7

3.2.1 OLSRv2 7

3.2.2 Babel 8

3.3 The B.A.T.M.A.N. Routing Protocol 8

3.3.1 Background 8

3.3.2 Protocol Description 9

3.3.2.1 Version III 9

3.3.2.2 Version IV 13

3.3.2.3 Version Comparison 18

3.3.3 Evolution of the Protocol 19

3.4 Asymmetric Link Treatment 20

4 Simulation **22**

4.1 Simulation Environment 22

4.1.1 OPNET Modeler 22

4.1.2 Simulation Framework 23

4.1.3 Transmission Range Model 24

4.2 Performance Metrics 25

5 Evaluation/Results **26**

5.1 Circle Scenario 26

5.1.1 Scenario Description 26

5.1.2 Results 27

5.2 Handover Scenario 29

5.2.1 Scenario Description 29

5.2.2 Results 30

5.3 Grid Scenario 33

5.3.1 Scenario Description 33

5.3.2 Results 33

5.4 Mobility Scenario 38

5.4.1 Scenario Description 38

5.4.2 Results 40

5.5 Evaluation of the Asymmetric Link Mechanism 42

6 Conclusion **44**

Bibliography **46**

List of Figures

3.1 Example MANET 6
3.2 Version III Packet Format 10
3.3 Sliding Window Example 11
3.4 Version III Operation Example Topology 12
3.5 Version IV Packet Format 13
3.6 Receive Quality 14
3.7 Echo Quality 14
3.8 Transmit Quality 14
3.9 Version IV TQ Propagation 15
3.10 Asymmetric Penalty Function 16
3.11 Version IV Operation Example Topology 17

4.1 Framework Node Model 23
4.2 Disc Radio Propagation Model 24
4.3 Asymmetric Link 24

5.1 Circle Example Topology 26
5.2 Circle, 2 Hops, Reliability 28
5.3 Circle, 6 Hops, Reliability 28
5.4 Handover Scenario 29
5.5 Handover, OGM Inter-Arrival-Time 30
5.6 Handover, Window Size/Local Window Size, OGM IAT = 1s 31
5.7 Handover, Window Size/Local Window Size, OGM IAT = 0.5s 31
5.8 Handover, Bidirectional Timeout Range/Global Window Size 32
5.9 Grid Example Topologies 33
5.10 Grid, Topology A, Reliability 34
5.11 Grid, Topology A, Average End-To-End Hop Count 35
5.12 Grid, Topology A, Overhead 35

5.13 Grid, Topology B, Reliability 37
5.14 Grid, Topology B, Average End-To-End Hop Count 37
5.15 Grid, Topology B, Overhead 38
5.16 Mobility Scenario 39
5.17 Random Walk, Reliability 40
5.18 Random Walk, Average End-To-End Hops 41
5.19 Random Walk, Overhead 41

1. Introduction

Adhoc networking is a field under research, since the beginning of computerized wireless networks. Yet, the technological progress has not so far led to an adoption of the underlying principle of relaying information over multiple hops in everyday life. However, during the last ten years adhoc networking displayed itself as an emerging technology in wireless sensor networks or community mesh networks. Ever increasing data rates render adhoc networks possible to overcome their inherent challenges. Furthermore, a largely increasing number of mobile devices, such as smartphones or tablet computers, is nowadays equipped with a wireless network interface, while providing more and more battery capacity. Consequently, there is an increasing number of use cases, that make it desirable to have these linked in an easy way.

There are a lot of challenges, such as mobility, limited power, and unstable connections, adhoc networks have to cope with. Thus, there is a vast amount of routing protocols, that have to date been suggested to solve the problem of efficient packet forwarding under these demanding circumstances. The "Better Approach to Mobile Adhoc Networking" (B.A.T.M.A.N.) protocol is a recent, ongoing development among them. One of its distinctive features is its inherent capability of handling asymmetric wireless links.

Studies have already shown that there is a significant fraction of links in wireless testbeds, that proves not only to be unreliable, but also exhibit asymmetric characteristics [SAZ10, CABM03, WTC03]. This phenomenon occurs particularly in low power setups. In simulation environments, where it is necessary to make simplifying assumptions, this is often neglected [KNE03]. However, the tool of simulation gives the researcher more power to control link characteristics, contrary to a real testbed. Under this circumstance, one is enabled to gain more detailed insight into the mechanisms and algorithms a protocol employs.

The target of this bachelor thesis lies in the implementation and simulation of the B.A.T.M.A.N. routing protocol. Under investigation are two different routing algorithms that were deployed by the developers. Their performance is compared in terms of reliability, end-to-end hop count, and resulting overhead. Additionally a detailed perspective on their functioning is given. Especially an evaluation of the asymmetric link mechanism is conducted.

It is found, that B.A.T.M.A.N.'s first simplistic approach of finding reliable routes is tainted with a significant drawback in asymmetric network topologies. The current approach, though, introduces a routing metric very much capable of detecting paths with

a minimal packet loss. However, the link estimation process comes with some degree of inertia regarding the detection of network topology changes.

In Chapter 2 the thesis begins with references of related studies, which mostly measure the performance of the B.A.T.M.A.N. protocol in small testbeds against that of other solutions. Afterwards Chapter 3 gives a brief introduction to applications and to the topic of routing in adhoc networks. It contains an overview of recent competing routing protocols with similar functionality to the B.A.T.M.A.N. protocol. A detailed description of the implemented protocol versions of B.A.T.M.A.N. is presented in the following. The chapter ends with a short evaluation of the treatment of asymmetric links. In Chapter 4 further insight is given on the simulation framework used in the OPNET modeler simulation software. Moreover, it contains a description of the asymmetric link model used in the according simulation scenarios and highlights the measured performance metrics. The conducted scenarios are explained and evaluated in Chapter 5. At first, the mechanisms and protocol parameters are evaluated in smaller setups. Afterwards, the overall behavior and performance is measured in bigger setups with changing environmental circumstances. A résumé concentrating on B.A.T.M.A.N.'s link cost metrics completes the chapter. Eventually the thesis finishes with a conclusion in Chapter 6.

2. Related Work

This chapter gives an overview of available research papers concerning the evaluation of the B.A.T.M.A.N. protocol. Tests were almost entirely performed in real testbeds with varying size and technology. Most works are comparative studies against other adhoc protocols, where different implementations and versions were used. Moreover, different performance metrics are under evaluation, with throughput being the most popular one. Therefore, the presented papers are not comparable to each other and conduct differing results.

2.1 Version III Performance Studies

Ikeda et al. performed several testbed evaluations of version III of the B.A.T.M.A.N. protocol. In [IMBT08] the effect of the routing protocol on throughput with TCP and UDP traffic is measured in a small setup. Varying characteristics are observed compared to the Optimized Link State Routing protocol (OLSR). In [KIBM10] different moving patterns are examined, while comparing the performance in resulting bit rates, delay, and packet loss with the Adhoc On Demand Distance Vector (AODV) protocol. It is concluded that B.A.T.M.A.N. has a better performance with mobile nodes. In a third work ([BIM+09]) an experimental comparison in a static scenario was done. OLSR with the ETX link quality extension and a differing window size was evaluated against B.A.T.M.A.N. in terms of throughput. The authors obtained the best results with OLSR and a window size of 10.

In [JNA08] the authors compare B.A.T.M.A.N. against OLSR with ETX as link quality estimation in a 7×7 grid testbed. B.A.T.M.A.N. achieves a higher throughput, while maintaining more stable links than OLSR. At the same time it shows a clear advantage over OLSR in terms of scalability, while reducing used memory and CPU load.

The authors of [RCC09] perform a simulation of several routing protocols, with B.A.T.-M.A.N. among them, using a mobile client scenario within a mesh network. "Smart-Window-B.A.T.M.A.N." is introduced as a modification of the original proposal, which uses a weighted average approach on the routing metric. This modification outperforms the compared protocols. In [ACC+09] its performance is then evaluated in a small real testbed.

Another test against OLSR is performed in [AWK+09] using a wireless sensor network testbed. In this study B.A.T.M.A.N. only shows better results than OLSR with a high traffic load in the network.

2.2 Version IV Performance Studies

In [AHW09] the developers' *batmand* routing daemon is tested against those of current implementations of OLSR and the Babel protocol. In an indoor office mesh setup Babel and B.A.T.M.A.N. show better overall results than OLSR. B.A.T.M.A.N. shows a slightly better stability and packet delivery, while Babel offers a quicker reaction to topology changes. The same authors provide an evaluation of the Hybrid Wireless Mesh Protocol (HWMP) in [WHA10]. This protocol is part of the upcoming IEEE 802.11s standard for wireless mesh networks. Its performance levels are below those of OLSR and B.A.T.M.A.N..

In [MDK10] B.A.T.M.A.N. is evaluated using both its developers' layer 3 and layer 2 implementation in an office testbed. The resulting performance is compared to that of OLSR and Babel. In this study Babel considerably outperforms OLSR and B.A.T.M.A.N., while the latter are showing more or less equal performance levels.

In [BC11] the protocol is enhanced with a hysteresis mechanism called "batrytis", which shall prevent it from unnecessary route changes. Tests are conducted both in static and mobile testbeds.

3. Routing in MANETs

In the following chapter a brief overview of current adhoc network applications is presented in Section 3.1. Section 3.2 then consists of a short introduction to the two competing routing solutions OLSRv2 and Babel. The B.A.T.M.A.N. routing protocol is subsequently explained in detail in Section 3.3. At the end of the chapter Section 3.4 evaluates the utilized asymmetric link treatment mechanism.

3.1 Mobile Adhoc Networks

Mobile adhoc networks (MANETs) are infrastructureless multi-hop wireless networks as exemplified in Figure 3.1. By relaying information over the participating devices, the network coverage expands far beyond the transmission range of a single device. Due to the presumed mobility of the nodes forming the network, and the decentralization, self-organization mechanisms have to be employed.

In these characteristics MANETs are closely related to Wireless Sensor Networks [KW03], which are used for environment monitoring. Moreover MESH networks deal with similar difficulties. These are basically multi-hop wireless networks as well, but are equipped with stationary backbone devices, often providing several network interfaces [AW05].

Adhoc networking comes in handy, whenever it is not practical to provide the otherwise necessary infrastructure, e.g. due to the temporary nature of a network, or restrictions caused by the environment. As with many technological developments such circumstances

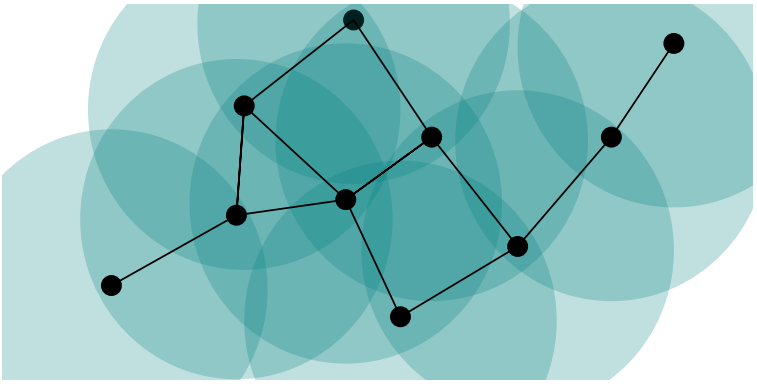


Figure 3.1: Example MANET

emerge especially in the military context [Per01]. Apart from these, the already mentioned wireless sensor networks are probably the most commercially important employment of the technique. The third recognizable current use case are community networks, which have oftentimes been built due to a lack of internet coverage in certain areas. Adhoc networks were found very much useful to provide internet acces over the "last mile". These communities may be found in rural areas as well as in urban areas. Exemplary, the german "Freifunk" community[1] shall be mentioned here, as in its surroundings the B.A.T.M.A.N. protocol was developed, and the austrian pendant "Funkfeuer"[2]. The latter are actively working on the new specification of the next version of the popular Optimized Link State Routing protocol OLSRv2.

3.2 Competing Routing Protocols

There exists a vast number of different routing protocols for multi-hop adhoc networks. Among those, B.A.T.M.A.N. belongs to the table-driven or proactive protocols, which maintain a routing table of the topology regardless of the necessity of a connection. Opposing to the concept of reactive protocols, which only establish routes on demand, routing information is exchanged constantly leading to a higher overhead. On the other hand there is no delay from the view of the application, because routes are already at hand whenever needed.

This section gives a short introduction to competing routing protocols that show similar characteristics to those of the B.A.T.M.A.N. protocol, as they are both proactive protocols, while representing the different approaches of link-state and distance vector routing.

3.2.1 OLSRv2

The Optimized Link State Routing (OLSR) protocol was first specified in [CJ03]. It is very popular in MESH networks and is the common standard in the above mentioned community networks. Since these communities ran into problems concerning scalability and performance, the actually deployed software differs in its behavior from the RFC specification. The protocol has since received several improvements and an extensible modular architecture, which is newly specified as OLSR version 2 in [CDJ11]. The main functionality and basic algorithms remain the same. A comprehensive examination is given in [Her11].

As the name already states, OLSRv2 is a link-state protocol. As such it has three distinct functions: Neighborhood discovery, link state advertising, and shortest path calculation.

For neighborhood discovery OLSRv2 puts in charge the *Neighborhood Discovery Protocol* (NHDP). This executes the task of discovering the two-hop neighborhood of a node by the dissemination of *HELLO* messages in regular intervals. The content of these messages is basically a nodes' address and a list of its direct neighbors. Based on these the protocol performs a bidirectional link check, to rule out the usage of unidirectional links. Additionally, the NHDP may perform a link quality evaluation, which is not a link metric, but a mechanism to not propagate links, that are considered not good enough for being below a certain threshold. The nature of this link quality evaluation is not defined.

Link state advertising is done by flooding the network with so called *topology control* (TC)) messages. These are relayed throughout the network in order to have every node informed about the resulting topology. Based on this knowledge, every node performs a shortest path calculation and can afterwards make the next-hop decision when forwarding data packets.

[1] http://start.freifunk.net/
[2] http://funkfeuer.at/

In order to save bandwidth, OLSRv2 chooses a *multipoint relay* (MPR) approach for flooding TC messages. Instead of having every node rebroadcast every yet unknown message, nodes select among their direct neighbors a subset of nodes, via which every two-hop node can be addressed. TC messages are only rebroadcasted by a node, if it is aware of it being selected as an MPR node by the sender of the message.

Due to the modular and extensible character of OLSRv2's core specification, it does not force a distinct routing metric to be used. Nonetheless the restriction is made, that the link metric must be additive, so that path costs result from summing up the costs of the links it consists of. Furthermore the representation of the link metric is defined in an exponential form. This consists of 12 bits, with 4 bits for the exponent and 8 bits for the mantissa. By utilizing this technique the amount of numbers near the lower boundary of representable numbers is higher than that near the upper boundary. This yields a higher accuracy on calculations involving low costs. The link metrics are explicitly forming a directed graph, on which the shortest path calculation is applied. Therefore, OLSRv2 offers in principle the possibility to make use of asymmetric metrics.

3.2.2 Babel

Opposing to OLSRv2, Babel follows the paradigm of distance-vector routing protocols and as such relies on the principle of the Bellman-Ford algorithm. A lot of emphasis is put on avoiding the occurrence of routing loops. Therefore, certain techniques from preceding protocols, such as the Destination Sequenced Distance Vector (DSDV), Adhoc On Demand Distance Vector AODV, and Enhanced Interior Gateway Routing Protocol (EIGRP) are adapted. An experimental RFC has been published describing the inner works of the protocol [Chr11].

Babel detects neighboring nodes by the periodical broadcast of *Hello* messages, to announce the presence of a node. The reception rate of these *Hello* messages allows nodes to evaluate the receiving link quality. In order to perform a bidirectionality check, *I Heard U* messages are sent containing the perceived receive quality. Hence, by the exchange of these two message types, nodes gain a cost evaluation of local links in both directions. Similar to OLSRv2, it is not further specified on how to implement the cost calculation of a routing metric. However, as a default the expected transmission count (ETX) is recommended [CABM03]. A 16 bit integer representation is chosen to evaluate link costs.

Both in regular intervals and on triggered events, every node propagates its routing table in compliance to the distributed Bellman-Ford calculation. Though, routes will only be adapted, if certain feasibility conditions are fulfilled, that guarantee loop-freeness. One problem arising is that through e.g. varying cost estimations, a node may run out of feasible routes towards a destination. To overcome this challenge, nodes suffering from this "starvation" send out requests for new feasible routes. These are flooded throughout the network, to cause the destination to repropagate them with a new sequence number. The protocol is therefore not strictly proactive.

3.3 The B.A.T.M.A.N. Routing Protocol

In the following the B.A.T.M.A.N. routing protocol is explained in detail. Version III and IV of its algorithm are presented as well as its development history.

3.3.1 Background

The B.A.T.M.A.N. routing protocol has been developed by the German "Freifunk" community [Aic07, NAL07] in Berlin. Based on their experiences with their own implementation of the OLSR protocol, it was concluded that there was a need for a new routing algorithm

to be used within their network. The main reason for this decision was the size of the network with over 400 participating nodes. Since the OLSR protocol, as it is a link state routing protocol, disseminates topology information and requires every node to perform a shortest path calculation on the whole network topology, it became a computationally intensive task for the employed embedded hardware to maintain the routing table. The time consuming shortest path calculations may lead to inconsistencies in the topology and therefore produce routing loops, which of course are very much desired not to occur.

For this reason, the designers chose an approach that would do without having every node to carry information on the whole network topology, while proactively maintaining link information. The latter design choice stems from the experience that on the one hand, it is inevitable to have some sort of proactive mechanism involved in the process of making a routing decision based on the measurement of link qualities. On the other hand it is more practical to have routes readily available for small amounts of traffic, as it is the case with common internet traffic. Browsing and e-mail, apart from VoIP, or streaming applications, do not make use of the route for a long period of time.

The core principle of the routing algorithm is this: After a given time interval every node broadcasts an originator message (OGM), that basically states the existence of the node. In order to make other nodes, that are not within reach, aware of the originator, these messages are rebroadcasted by its recipients according to certain rules. A node that received such a message memorizes the direct neighbor, via which it has been received. By maintaining statistics on how successful messages spread over the network, nodes conclude, which one of their direct neighbors is the best one to forward a packet towards its destination. As a consequence routes and the topology of the network are unknown to the participants, apart from the nodes, which are within direct reach. The routing information is distributed over the whole network. This is sufficient as the participating nodes only have influence on the next hop decision.

The development of the B.A.T.M.A.N. protocol is highly oriented on the practical needs and experiences its developers gain from employing the protocol in real scenarios. Thus, the protocol is an ongoing work-in-progress, that according to its developers runs through evolution stages depicted with roman numerals. These describe the different versions of the algorithm, separate from the implementation versions of the Linux kernel module *batman-adv* and the user space routing daemon *batmand*.

At the time of the writing of this work, version IV is applied, while version V is under development. This thesis concentrates on version III and IV of the routing algorithm, as these are fully developed and provide deviating routing metrics, that are to be investigated. As version V is still in its design phase, a further look upon its mechanisms may be a task for future examinations.

3.3.2 Protocol Description

This section describes the protocol versions III and IV, as they have been implemented for simulation in the OPNET Modeler.

3.3.2.1 Version III

The implementation of version III of the protocol has been done according to the behavioral description in the RFC draft accessible via the developers' website [WLT09] and in [Aic07]. It adapts to the simulation framework, that is later on described in section 4.1.

Originator Message Format

Every node periodically broadcasts in an interval depicted by the *interarrival time* (IAT) an *originator message* (OGM), that is relayed over the network to indicate the nodes'

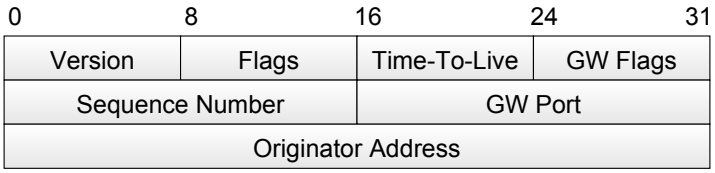


Figure 3.2: Version III Packet Format

existence. This node is called the originator of the message. It consists of 12 bytes as described in Figure 3.2. It is encapsulated within a UDP packet, that is sent via IP broadcast. Together with the typical IP overhead the effective size of an OGM results in 52 bytes.

An OGM holds the following information essential for the routing process:

- A *unidirectional flag* indicating, that the link towards the originator of the message is not (yet) considered bidirectional.
- An *is-direct-link flag* indicating, if the OGM has been rebroadcasted from a node, who is a direct neighbor towards the originator.
- A *time-to-live* (TTL), that defines an upper limit of rebroadcasts of the OGM.
- A *sequence number*, that is being incremented for every OGM a node emits.
- The *originator address* of the node promoting its existence.
- The *address* of the intermediate sender rebroadcasting the OGM. This information is obtained from the IP header of the transmitted packet.

The gateway flags and the gateway port are not essential for the routing mechanism, as their purpose is to announce the originator's possibility to function as a gateway to the internet.

Neighbor Ranking

Upon reception of an OGM the receiving node knows of the existence of its originator. Furthermore, a route to this originator leads via the intermediate sender of the message. In case there exist several routes towards the originator, it might happen that OGMs are received via more than one direct neighbor. Therefore the receiving node for every known originator maintains a list of every direct neighbor, over which the originator is reachable. In order to make the routing decision, a node performs a ranking of its direct neighbors by comparing the amount of OGMs, that have been received via every one of the direct neighbors. Since OGMs can be identified by their sequence numbers, the ranking of a direct neighbor only improves when being the first one to transmit the OGM to the receiving node. Thus, the best ranking neighbor is the one providing the route, over which packets are received faster (i.e. with less hops) and more frequently. Sequence numbers are recorded within a sliding window. The size of this sliding window represents the maximum value for the neighbor ranking. It is shifted, when a yet unknown sequence number, i.e. a non-duplicate, is received, which leads to an update of the neighbor ranking.

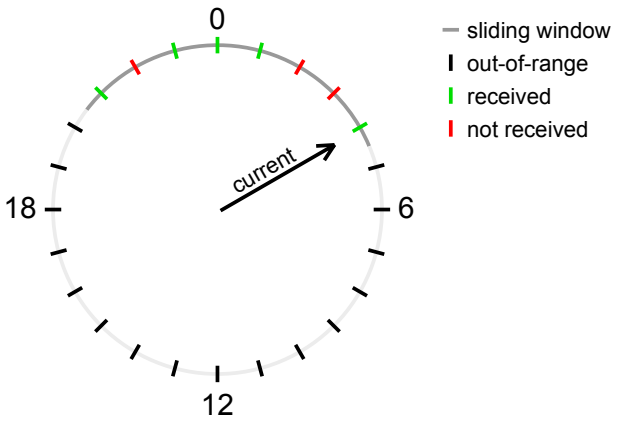


Figure 3.3: Sliding Window Example

Sliding Window

The set of possible sequence numbers is determined by the length of the sequence number field. With 16 bits this leads to a maximum sequence number of $2^{16} - 1 = 65535$. Sequence numbers are parted into *in-window sequence numbers* and *out-of-range sequence numbers*. The protocol keeps track of the last received sequence number as the *current sequence number* received from each originator. The in-window-sequence numbers are the $window size - 1$ numbers below the current sequence number calculated modulo $2^{16} - 1$. The reception of these has to be memorized as well, since the number of received in-window sequence numbers over a distinct neighbor is the used metric for determining the link quality towards the concerning originator.

Whenever an out-of-range sequence number is received, it is set as the new current sequence number and the sliding window is moved accordingly. Recorded sequence numbers that fall out of the sliding window during that process, are not memorized anymore.

Figure 3.3 shows a small example sliding window with a set of possible numbers of 0 to 23. With a sliding window size of 8 and the current sequence number set to 4, the in-window sequence numbers range from 21 to 4. Accordingly the out-of-range sequence numbers range from 5 to 20. The memorized received sequence numbers are 21, 23, 0, 1, and 4. Hence, the packet count of such a sliding window would be 5.

The reception of an out-of-range sequence number, higher than the current sequence number plus the sliding window size, empties the sliding windows of all of the entries in the neighbor list of an originator, except for the one, via which the regarding OGM was received, which then has a packet count of 1.

Bidirectional Link Check

The algorithm performs a bidirectional link check between neighboring nodes. This is essential, since it has to be ensured that the propagated routes are working in both directions, in case MAC layer acknowledgments are used. Hence, OGMs from multi-hop originators are only rebroadcasted, when the bidirectional link check towards the intermediate sender was successful.

When receiving an OGM from a direct neighbor, to which the link is not yet considered bidirectional, it is rebroadcasted with the unidirectional link flag set. OGMs with the unidirectional link flag set are only processed by the originator itself. For the originator of the message to be sure that the answering neighbor has received the message directly, the is-direct-link flag has to be set as well. The message having travelled in both directions proves

the link to be bidirectional. This procedure has of course to be done in both directions of the link, so that both nodes consider the link bidirectional. A link is not considered bidirectional anymore, when the last OGM, that has been successfully rebroadcasted by a neighbor, contained a sequence number, that is not within a defined range. That means a node has not *re*received a certain amount of sent OGMs from the neighbor in question.

Route Deletion

The last valid time, when the last OGM has been received from a node is saved. When this time exceeds a certain timeout interval, called *purge timeout*, the respective node will be removed from the routing table, as it is considered not reachable anymore. This node might have been disabled or moved out of reach.

Similarly for every direct neighbor, that is registered to serve as a possible next hop towards this node, a last valid time is kept track of. This memorizes the last valid OGM, which has been received via this neighbor. When it exceeds the purge timeout interval, the node will be removed from the neighbor list of the according node.

The authors of the protocol suggest to make use of a rather long period of time for the purge timeout interval. In case there still exists a routing table entry for an unreachable node, the network might be used in vain, when traffic is forwarded to an unreachable destination. However, a purge interval too short might lead to valid routes being deleted, which would have a negative impact on the routing functionality.

Example

Consider the topology given in Figure 3.4 of a network with 6 nodes:

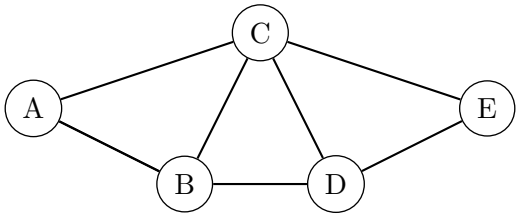


Figure 3.4: Version III Operation Example Topology

When node A sends its first OGM, it is received by nodes B and C. These will rebroadcast the message with the unidirectional link flag and the is-direct-link flag set, for the OGM was received from the originator. The rebroadcasted messages will be ignored by every receiving node, except from A, which then has successfully performed the bidirectional link check towards B and C. Assuming that B and C have successfully performed the bidirectional link check towards A, they will then rebroadcast A's next OGM without the unidirectional link flag set, so that nodes D and E get to know of the existence of A. By this means OGMs get flooded throughout the net. When furthermore E receives an OGM from A via D, that has the same sequence number as a formerly received OGM from A via C, it discards the message, as it is already known.

Node A may receive OGMs from E via several paths. The shortest being E-C-A, while others would be e.g. E-D-B-A or E-C-B-A. The task of choosing the right gateway towards E depends on whether node C or node B delivered most of E's OGMs first.

With respect to the examplary neighbor ranking in Table 3.1, maintained by node A, the gateway to choose towards E would be B. Thus, a longer route, such as E-C-B-A, is preferred to the short route of E-C-A, since it is considered to be more reliable.

Node	Neighbor	Packet Count
B	B	43
	C	3
C	C	30
	B	10
D	B	26
	C	12
E	B	18
	C	16

Table 3.1: Version III Neighbor Ranking Example

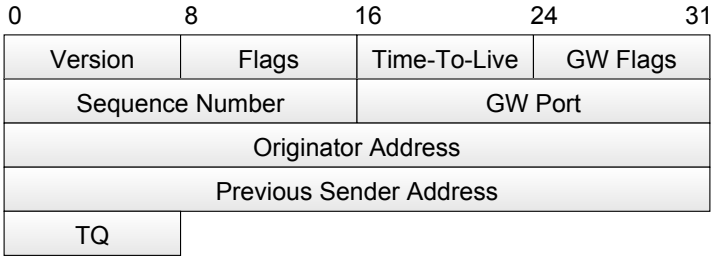


Figure 3.5: Version IV Packet Format

3.3.2.2 Version IV

Version IV has been implemented according to the description in [WLT09]. Due to the loose character of the document, the source code of the implementation in *batmand 0.3.2* was consulted, whenever matters were unclear and demanded a more detailed investigation.

Originator Message Format

The originator message format is similar to the one used in version III of the protocol. The differences important for the routing mechanism lie in an additional TQ field, which denotes the path quality towards the originator. This value is described in more detail later in this chapter. Furthermore, a previous sender address field is introduced, which contains the address of the node, that has previously forwarded the message. The unidirectional flag is not used anymore. The addition of 5 bytes results in an effective transmission size of 57 bytes per packet. The datagram of the packet format can be seen in Figure 3.5.

Local Link Quality Estimation

The protocol takes measurements of the link qualities between direct neighbors. It obtains both, a value representing the probability of the successful transmission of a packet towards another node - this is called *transmit quality* (TQ) - as well as the probability of a successful reception of a packet from a neighbor, called *receive quality* (RQ) respectively. The estimation once again takes place by maintaining sliding windows, which count the amount of the last received originator messages. For the used routing metric the TQ is of most interest. Without any form of indication, a node cannot directly measure the amount of successfully transmitted packets towards another node. This is why the protocol performs a simple calculation targeting to approximate the local TQ value.

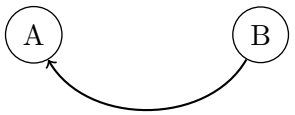


Figure 3.6: Receive Quality

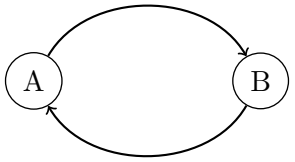


Figure 3.7: Echo Quality

Firstly, node A obtains the RQ (see Figure 3.6) by dividing the "charging level" of the sliding window, that records the successful reception of the last packets, that originated from node B, by the size of the local sliding window.

Secondly, node A calculates an *echo quality* (EQ), which denotes the probability of a successful reception of an OGM originating from node A itself, that has been rebroadcasted by node B (see Figure 3.7). This, similarly to the measurement of the RQ, happens by maintaining a sliding window for each direct neighbor, that records the successful echo receptions. The EQ therefore approximates the probability of the twice successful transmission of the packet. This is expressed as a formula in Equation 3.1.

$$EQ = TQ \cdot RQ \tag{3.1}$$

Figure 3.8: Transmit Quality

The TQ (see Figure 3.8) is finally back calculated by simply solving the above equation for TQ resulting in Equation 3.2.

$$TQ = \frac{EQ}{RQ} \tag{3.2}$$

Global Link Quality Propagation

When sending an OGM the TQ field is initialized with its maximum value of 255. Upon reception of an OGM, the receiving node multiplies the TQ of the incoming OGM with its measured TQ towards the last hop of the OGM (see Equation 3.3).

$$TQ_{new} = TQ_{OGM} \cdot TQ_{local} \tag{3.3}$$

So far, the TQ field of an OGM represents the probability of a successful transmission of a packet towards the originator of the message, along the path the OGM has travelled. This shall be illustrated by Figure 3.9.

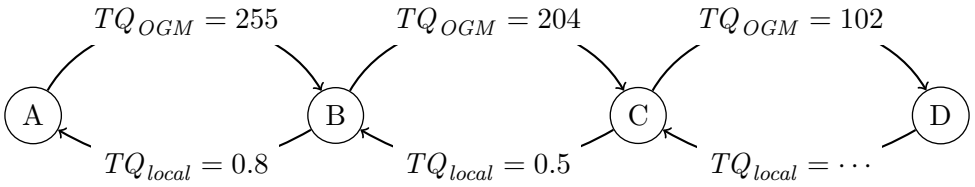


Figure 3.9: Version IV TQ Propagation

Since the protocol is naturally supposed to forward data packets along the path, that is being ascertained as the one providing the highest probability of transmission success, the rebroadcasted TQ value is exchanged by the one, that applies for the best ranking neighbor, instead of the one received. Thus, a node always advertises its best ability to successfully forward packets towards the originator.

Asymmetric Penalty To account for the necessity of links to be used in both directions, the protocol imposes a penalty on links, that show a bad receiving link quality. This stems from the usage of acknowledgement packets on the medium access layer for unicast packets. While the transmission of a packet may always be successful, a link may turn out useless, if acknowledgement packets are not received in the reverse direction. This may induce the retransmission of the same packet until the sender runs into a timeout.

The penalization is performed by multiplication of the TQ value with the weighting function in Equation 3.4 plotted in Figure 3.10.

$$TQ_{new} = TQ_{old} \cdot (1 - (1 - RQ)^3) \tag{3.4}$$

The influence of the RQ on the resulting TQ is nonlinear in order to decrease penalization of the link by higher RQ values. Yet a fully asymmetric link leads to a TQ value of zero, causing the protocol to completely avoid usage of that link.

Hop Penalty Analog to the asymmetric penalty, the protocol additionally applies a *hop penalty* (HP), which is a fixed value between the maximum TQ value and zero. It is applied before every retransmission of a packet according to Equation 3.5.

$$TQ_{new} = TQ_{old} \cdot (TQ_{max} - HP) \tag{3.5}$$

Hereby, the protocol punishes paths with a longer hop count, that might provide a similar transmission quality. This is expected to save bandwidth, reduce latency, and avoid the decrease of throughput caused by self-interference.

OGM Rebroadcasting

The regular dissemination of OGMs throughout the network assures the constant evaluation of local link qualities, advertises the existence of the participating nodes, and renders possible the neighbor ranking by informing receiving nodes of the measured path quality towards the originator. To make this mechanism work the participating nodes rebroadcast received OGMs, after having altered the fields in question under certain conditions.

The changes applied are an update of the status of the is-direct-link flag, so that receiving nodes know, whether the originator is a direct neighbor of the sending node. By this

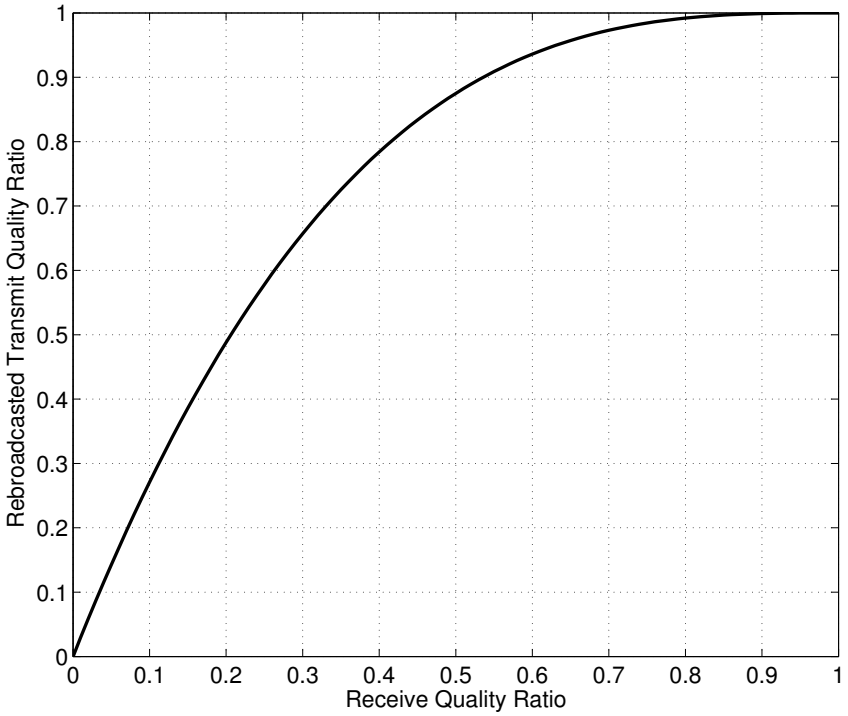


Figure 3.10: Asymmetric Penalty Function

flag the originator of the message determines, whether the OGM is a valid echo or has travelled over several hops back to the originator. Most importantly, the forwarding node enters the current global TQ value in the correspondent field, that resulted from the preceding procession of the OGM, so that the information of the signaled path quality is up-to-date. Finally, the TTL is decremented by one and the previous sender field is filled with the address of the former last hop of the message. OGMs, that reach a TTL of zero, are dropped before being rebroadcasted. The previous sender field allows to make a distinction between messages, that are echoes of an OGM, that has already been processed, and is just a *re*rebroadcast by a neighboring node. These messages do not hold new information relevant to the path quality estimation, and have therefore to be distinguished from the ones that have travelled over another path. An OGM, whose previous sender field contains the address of the receiving node, is consequently identified as an echo and dropped immediately.

OGMs are always rebroadcasted, when received from the originator itself, i.e. when the last hop address equals the originator address field. OGMs, received over more than one hop, are only rebroadcasted, when they are not a duplicate, meaning the contained sequence number is not known and the path, via which it was received, is considered bidirectional. This serves the purpose of not advertising a connection towards a node that presumably cannot be reached.

Bidirectional Link Check

The bidirectional link check is performed considering the TQ value of the currently processed OGM. Bidirectionality towards the sending node, and towards the originator via this link, is tested by calculating the new TQ value for the path. This is done by the already illustrated global link quality propagation mechanism, except for the hop penalty, which is applied just before the OGM is rebroadcasted. If the local TQ towards the last hop proves to be too low, the resulting global TQ value will turn out below a predefined

value, that serves as a limit for the link to be considered bidirectional. By default, this limit is set to 1. A higher limit is not likely to change routing decisions, but reduces the range of coverage of an OGM in a network afflicted with lossy links.

Neighbor Ranking

Version IV ranks its neighbors with regard to the received global TQ values, having taken into account the local TQ and the asymmetric penalty of the last hop. The global TQ thereby serves as a metric describing the path quality towards an originator. The neighbor with the highest global TQ value is considered as the one providing the path with the highest likelihood of a successful transmission towards the packet's destination privileging shorter and non-asymmetric paths. Global TQ values are stored within a sliding window for every entry in the neighbor list of a node. The desired value for the neighbor ranking is gained by averaging over the sliding window entries. OGMs with a TQ of zero are not taken into account, as they are considered non-received packets.

Packet Aggregation

In order to make an efficient use of the transmission channel, the protocol utilizes a packet aggregation mechanism. Instead of rebroadcasting OGMs at once, a node waits for a maximum aggregation time, in which new packets are appended to already queued ones waiting for transmission. The aggregated packet is sent, either when the maximum aggregation time is up, or the aggregated packet has reached a maximum size.

Especially in areas with a high node density this shall reduce the probability of collisions. Additionally, the relative overhead to the amount of OGMs transmitted is reduced, since several OGMs get encapsuled in just one UDP broadcast packet. On the other hand, the loss of an aggregated packet, due to e.g. interference, leads to the loss of all the contained OGMs, while otherwise just one might have been lost.

Example

Consider the example topology in Figure 3.11, which is similar to the example in the previous section. Opposing to the previous, precise probabilities of a successful transmission of each direction are given.

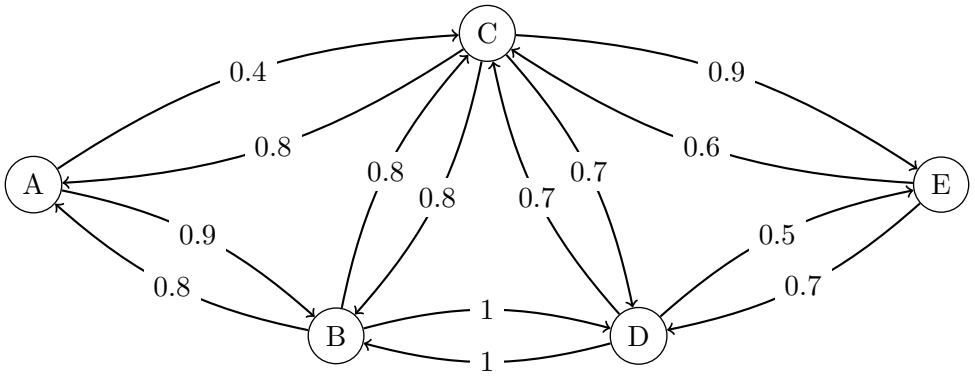


Figure 3.11: Version IV Operation Example Topology

Nodes B and C receive OGMs, that have been emitted by node A with a TQ value of 255. As the nodes are direct neighbors of A, they will rebroadcast them, which enables node A to increase its echo count towards nodes B and C. First the rebroadcasted messages will contain a TQ value of zero, which is why nodes D and E will simply drop these.

Meanwhile, node A receives OGMs from node B and C and rebroadcasts them as well. After receiving the first echo from A, nodes B and C will then rebroadcast A's OGMs with a first TQ value higher than zero and thereby make A known to D and E. The TQ values become most accurate after the $OGM\,interval \times local\,sliding\,window\,size$ seconds, i.e. when the number of recorded sequence numbers divided by the local sliding window size equals approximately the given link quality.

Node	Receive Count	Echo Count	Local TQ
B	49	42	218
C	57	23	102
D	0	0	0
E	0	0	0

Table 3.2: Version IV Originator List

Table 3.2 gives an example originator list of node A, with a local sliding window size of 64. Entries for nodes D and E are all zero, because they are not direct neighbors of A, which is why A neither received a one-hop OGM originating from them, nor an echo of an own OGM, resulting in a local TQ of zero. With the given numbers, node A would estimate the local TQ towards node B as $42/49 \cdot 255 \approx 0.857 \cdot 255 \approx 218$, which is slightly differing from the value of 229 that would be the expected outcome. The measured link quality towards node C given with $23/57 \approx 0.404$ is in contrast quite accurate compared to the real one of 0.4.

Table 3.3 is a neighbor ranking table that is likely to occur with the given topology at node A.

Node	Neighbor	Global TQ
B	B	222
	C	75
C	B	192
	C	93
D	B	187
	C	79
E	B	133
	C	65

Table 3.3: Version IV Neighbor Ranking table

As can be seen, the bad link towards node C is successfully recognized for its bad characteristic, which is why node B is preferred as next hop for every destination. Even the direct connection towards node C is avoided, as well as the shorter path via C towards E.

3.3.2.3 Version Comparison

The mechanisms of both protocol versions are divided into three main tasks. Assertion of the usability of local links, flooding the network with the originator messages, and providing a metric estimating the quality of the path, that is used, when routing a packet through the network. Both versions hereby manage to work without the need of more than one packet format, which has a fixed size, providing a minimum amount of information. Similar to distance-vector routing, in both versions of the protocol participating nodes have no knowledge of the topology of the network. Its representation is distributed over all nodes.

The assertion of the local link usability in both versions depends on the bidirectional link check. Version III does this basically by assuring that OGMs are still travelling in both directions. Version IV's approach is the same, but in an indirect manner, and creates statistics on local link quality. The number of lost echoes, causing a link to be considered unidirectional, is in version III explicitly given as a parameter. In version IV this number is defined by the local window size.

The flooding of the network depends on the rules, when to rebroadcast an OGM. While both versions rebroadcast messages received by direct neighbors at all times, to maintain the local link considerations, they behave differently with OGMs of distant nodes. The basic principle of just forwarding non-duplicates, is extended in version III, with the strict policy of just repeating messages, that have been received via the currently best ranking neighbor. The path quality metric otherwise might become unreliable and may even cause routing loops. On version IV, however, this problem is not inflicted, since the path quality metric is changed to represent the best possible value. Version IV therefore floods the network more reliably, but also causes more overhead in doing so.

The path quality metric in version III is simply the amount of messages that reach a node via its direct neighbors. In a certain sense the OGMs therefore do not hold the routing information but are in itself the routing information. In contrast, version IV's OGMs propagate path qualities for every direct neighbor determining the ranking among them. The major difference of both approaches lies in different assumptions regarding the conditions of the paths. Version III assumes that the path, by which most packets reach a node, ought to be the best one to forward packets to. Version IV evaluates path qualities by merging information of every segment.

3.3.3 Evolution of the Protocol

The evolution of version I to III is described in [NAL07], while version IV is explained in [WLT09]. Version V design considerations are stated on the developers' website [ALL+11].

- Generation I served mainly as an experimental implementation for testing purposes. Link qualities are measured in dependence of the amount of OGMs, which have been received over one neighbor. It does not yet check for bidirectional links in the topology.
- Generation II introduces a classification of links in bidirectional and unidirectional as well as a mechanism to prevent nodes from propagating links, that are not classified as bidirectional.
- Generation III introduces a strict policy of rebroadcasting OGMs, i.e. they are only rebroadcasted, when having been received via the best ranking neighbor. This prevents the occurrence of routing loops. Furthermore, the recognition of bidirectional links has been revised, so that it is time-independent. In addition, the algorithm supports multiple interfaces per nodes, as well as a configurable TTL value for OGMs.
- Generation IV fundamentally changes the neighbor ranking mechanism. While former generations base their routing decision on how many OGMs were received, generation IV introduces the TQ metric, that measures the probability of a successful transmission of a packet on a certain link. Secondly packet aggregation may be applied to make better use of the medium.
- Generation V is planned to have several major modifications. First of all, the task of measuring link qualities between direct neighbors will no longer be accomplished by the use of OGMs, but by another packet format coexisting with OGMs. Secondly, a

reactive mechanism is planned to detect link failures in the network. This is intended to increase the responsiveness to changes in the topology.

3.4 Asymmetric Link Treatment

Several studies have shown that links in a wireless network have a lossy character and are oftentimes asymmetric in their transmission quality depending on the direction [KNE03, ZK07]. In regard to the underlying technology, there are many reasons for a link to be asymmetric. The surrounding area has a great impact, since radio wave propagation is affected by obstacles and different height levels of the devices. In addition to that, devices might suffer from different noise levels in areas of different node density. Technological reasons might be varying power levels of the transmission of some devices, due to different hardware or simply different battery levels. Moreover, different antenna directions or patterns play an important role of radio wave propagation.

The correct evaluation of this kind of links greatly affects the performance of the used routing techniques as shown in [WTC03]. It is easy to see that the simple avoidance of fully symmetric links is a waste of network resources. The involvement of fully asymmetric, i.e. unidirectional links, on the other hand demands for more complex techniques to find routes in both directions. These are mandatory to account for transport layer end-to-end acknowledgements. Moreover, networks with medium access layer hop-to-hop acknowledgements rely on link bidirectionality. Thus, the general approach of most protocols, as the ones in this work, is to avoid these completely.

OLSRv2, Babel and B.A.T.M.A.N.'s evaluations of paths follow a similar principle. Link quality is measured in some fashion and then propagated over the network. The routing decisions are based on using the path providing the lowest cost. While in OLSRv2 every node performs the shortest path calculation on its own, Babel and B.A.T.M.A.N. IV follow the distance vector paradigm, in which every node only knows about costs towards his direct neighbors. Contrary to this, B.A.T.M.A.N. III does not perform any local link measurement at all. Hence, in the rest of this section it will just be referred to version IV.

The utilized algorithms of the protocols are all capable of finding the shortest path in a directed graph. Thus, the usage of asymmetric links strongly depends on the cost metric, which determines the weights of the edges in the graph. Equal costs for every link lead to the utilization of the path with the shortest hop count. This has the tendency to use links with a longer distance, which has oftentimes a negative impact on the reliability. Therefore OLSRv2 and Babel use the *expected transmission count* (ETX) [CABM03] in their default implementations. This metric is the inverse of the probability of a successful transmission in both directions of a link. Therefore it calculates the amount of transmissions, necessary for a packet to be transmitted and acknowledged successfully, hence the name. This metric takes asymmetry into consideration, although having symmetric characteristics, because it generates the same value for both directions. It merely punishes links with an asymmetric characteristic instead of using them.

The *expected number of transmissions over forward links* (ETF) as proposed in [SAZ10] similarly counts the expected number of transmissions but just in the desired direction. Therefore, the metric does not punish asymmetric links, but rather exploits those, which provide a good quality in at least one direction. The authors concluded that this metric outperforms ETX not only in networks without hop-to-hop acknowledgements, but also in those making use of them. This relies on the fact that acknowledgements are small in size and sent right after reception, when the channel is still likely to be unoccupied.

B.A.T.M.A.N.'s TQ metric is somehow a mixture of hop count, ETX and ETF. Just like ETF it basically measures the probability of a successful transmission towards the next

hop. Yet, it tries to avoid asymmetric links, for the asymmetric penalty punishes links with a bad quality in the opposing direction akin to ETX. Additionally a high hop penalty obfuscates the resulting TQ values, increasing the tendency to use the route with the shortest hop count.

4. Simulation

This chapter describes the simulation environment, in which the scenarios were evaluated. Section 4.1 concentrates on the software used, and which presumed assumptions are made. Section 4.2 gives a concise explanation of the performance metrics in question.

4.1 Simulation Environment

For this evaluation of the B.A.T.M.A.N. protocol, simulation was chosen instead of employment in a real testbed. Firstly, this accounts to the general advantages of simulation. These would be practical reasons, as less time and effort is spent on building different scenarios and setups, and computing time is the only limitation to their size. Secondly, opposing to a real testbed evaluation, simulation permits exact control of the procedures. Indeed, the modeling of real physics entails making simplified assumptions. Yet, it renders the possibility, to form the environment just as needed for consideration of the protocol. In this case, this particularly refers to the modeling of asymmetric links in the network. Controlling link qualities in a real wireless setup would be a task connected with massive difficulties.

In the following the OPNET modeler simulation software, the simulation framework, and the physical modeling with the characteristics of asymmetric links are explained in further detail.

4.1.1 OPNET Modeler

The simulations were performed using the OPNET Modeler [OPN08], version 14.5 by OPNET Technologies Inc. It is a commercial discrete event system simulator, used in the industry and in academic environments to simulate and measure the performance of computer and communication networks.

As described in [Kle10], the simulator displays a hierarchical representation of the network. The top level is the network level, which displays the environment, in which the communicating nodes are placed. These nodes are described by a node model, which is further explained in section 4.1.2. A node model consists of process models interacting with each other, typically representing the layers of the ISO/OSI communications model. Process models are defined as a finite state machine, holding state variables and attributes. Upon reception of an interrupt by the simulation kernel, the process changes its state according to the given transition rules. The behavior of the system is written in C code, which is

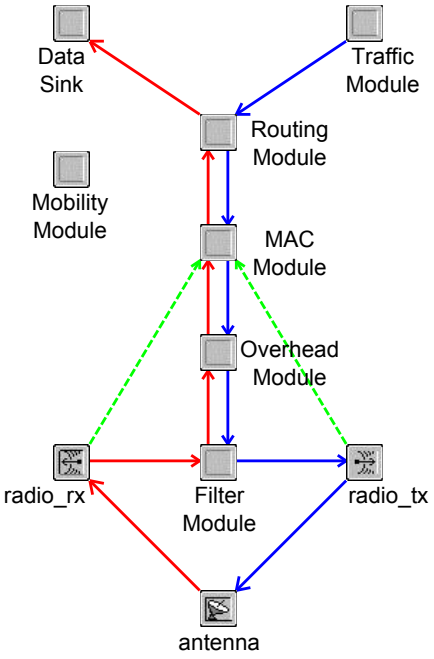


Figure 4.1: Framework Node Model

executed on the occurrence of an event, thereby changing the state variables and attributes of the system.

Events scheduled along the simulation time line are processed consecutively. Consequently, the condition of the system remains the same in-between and the simulation time proceeds discretely towards the next scheduled event.

4.1.2 Simulation Framework

The simulations executed in the context of this work have not been conducted using the default MANET node models provided by OPNET. Instead the framework provided by the chair for network architectures and services of the TUM was used. The basic framework is described in [Kle10]. It consists of a library of processes, which are assembled to simulate the characteristics of a wireless node.

The node model has been slightly altered from the basic model to meet the needs of the simulations performed. Figure 4.1 shows its components, which are mainly reflecting the ISO/OSI model.

Traffic Module/Data Sink These Modules abstract the layers 4 to 7. Generic data packets are generated by the traffic module to simulate application traffic. The data sink module receives packets and forwards statistics to a central instance.

Routing Module The two recent versions of the "B.A.T.M.A.N." routing protocol were implemented as processes for use in the routing module abstracting the corresponding layer.

MAC Module This module simulates the medium access control layer. For the purpose of this study simple carrier sense multiple access (CSMA) with a uniform back off distribution was used. It is connected to the radio transmitter and receiver modules with statistical wires to gain knowledge of their states.

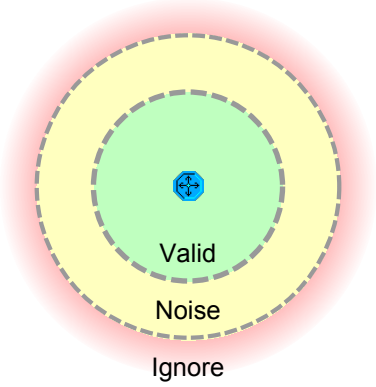


Figure 4.2: Disc Radio Propagation Model

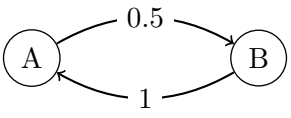


Figure 4.3: Asymmetric Link

Overhead Module The overhead module counts the amount of outgoing routing traffic. This information is forwarded to the central statistics collection mechanism.

Filter Module In order to perform the simulation scenarios described in Sections 5.1 and 5.3 a simple filter module was added to the framework to artificially simulate the presence of asymmetric links. A central instance generates a filter matrix, which describes the link condition between every node pair in both directions. Upon reception of a packet, the filter randomly eliminates packets, with respect to the successful transmission probabilities given in the filter matrix. Together with the transmitter, receiver, and antenna this abstracts the physical layer. The OPNET simulator makes use of a free space propagation model in its calculations for the physical layer. Since the filter module is simply put on top of that, the resulting probability of a successful transmission is below the one specified in the filter matrix. Furthermore, every filtered packet is still perceived by the MAC module as one that occupies the transmission medium. The data rate of the physical layer was set to 1 Mbit/s.

Mobility Module This module coordinates the movement of the node as in the scenario in Section 5.4. Several common random mobility models are available.

4.1.3 Transmission Range Model

The nodes transmission range depends on a simple disc model, that limits the free space propagation, and has the purpose of speeding up simulation time [Kle10]. As shown in Figure 4.2, two radii define three areas of reception. In an early computational stage of transmission, packets are flagged according to the distance they are transmitted over. Only those within the inner circle, flagged valid, can be successfully transmitted. Noise flagged packets will be perceived as interference at the receiver. Transmissions of nodes in the outer circle are ignored.

Figure 4.3 depicts the modeling of an asymmetric link between two nodes within communication range. A link is considered asymmetric, whenever the probability of a successful transmission differs significantly depending on the direction. A link, that only permits

transmission in one direction is considered unidirectional. Since the B.A.T.M.A.N. protocol does not make use of unidirectional links, these are not modeled in the according simulation scenarios. The link asymmetry is evoked by the filter module, realizing the transmission probabilities given in the filter matrix. An asymmetric link, between nodes A and B, is randomly chosen to be affected by packet loss in only one direction. This is exemplified in Figure 4.3 with 0.5. Reception in the other direction remains only an issue of the free space propagation model.

4.2 Performance Metrics

Reliability, average end-to-end hop count, and generated overhead are the performance metrics under view in this study. The focus lies on the evaluation of the mechanisms of the protocol, and particularly its applied routing metrics. Thus, other metrics, such as delay and throughput, are not taken into account. Of course, these are of great importance in the performance consideration of a network, but are also highly dependent of the higher layer traffic, the MAC protocol, and the hardware platform used. The latter is especially of interest, when researching operating expenses, such as energy, CPU and memory consumption. A comparative performance metric for an evaluation against other protocols would be the ratio of overhead to successfully transmitted data packets. However, this work concentrates on the aforementioned metrics, as they give most insight to the routing results, and survey the involved expenditure.

Reliability

Reliability denotes the end-to-end packet delivery ratio of data packets routed within the network. It is an assessment for the protocol's ability to either avoid lossy links within the topology or adapt the routing table to changes in the given topology.

Hop Count

The average end-to-end hop count gives information about the length of the chosen routes. It is the average number of hops, data packets have travelled, until they arrive at the respective destination. Since the B.A.T.M.A.N. protocol does not perform transmission delay techniques, the path length is the only source of transmission delay in the simulation. The more a packet needs to be sent over a link, the more delay accumulates.

Overhead

Overhead is the amount of routing information exchanged. This is measured by each transmission or retransmission of a B.A.T.M.A.N. packet. Supposedly, an increasing overhead of a protocol should lead to a better information base the routing decisions are made upon. On the contrary, a high overhead decreases the performance of a protocol especially in terms of throughput, since bandwidth is mostly limited. Since throughput is not under investigation in this study, the measured overhead shall simply monitor the "talkativeness" of the protocol. With an unpartitioned network and perfect links, the maximum overhead caused by broadcasting one OGM in both protocols results in $(number\,of\,nodes - 1) \times packet\,size$.

5. Evaluation/Results

This chapter evaluates the performance of the protocol versions in the conducted simulation scenarios with regard to end-to-end packet delivery ratio, average hop count, and generated overhead. Simulations were run ten times with changing seeds, the errorbar plots in this chapter show the 99 percent confidence interval to the average of the resulting value. Sections 5.1 and 5.2 evaluate the protocols' general behavior in small setups concerning link asymmetry and topology changes. Sections 5.3 and 5.4 then investigate on the functioning in bigger and less controlled setups. Finally Section 5.5 concludes the chapter with an evaluation of the mechanism that employs the asymmetric link metric.

5.1 Circle Scenario

In this scenario the protocols' handling of asymmetric paths in a circle scenario is looked upon.

5.1.1 Scenario Description

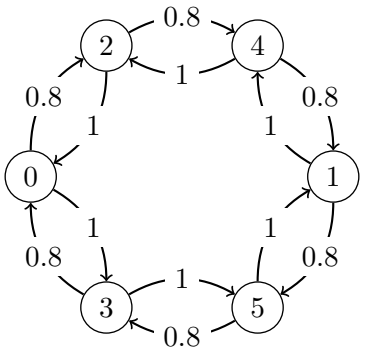


Figure 5.1: Circle Example Topology

The circle topology consists wholly of asymmetric links as depicted in Figure 5.1. In order to create asymmetric links, the filter module filters packets according to the setup given in the figure. The links behave in a way that packets routed clockwise will suffer from packet loss, while the ones routed counter-clockwise are not to be filtered at all. Data packets are sent between node 0 and node 1 in both directions with a constant interval of 1 second.

Parameter	Version III	Version IV
OGM IAT	1s	1s
OGM TTL	50	50
OGM Jitter	0.2s	0.2s
Purge Timeout	200s	200s
Window Size	64	-
Bidirectional Timeout Range	10	-
Local Window Size	-	64
Global Window Size	-	10
Hop Penalty	-	5
Maximum Aggregation Time	-	0.1s

Table 5.1: Protocol Configuration

The purpose of the simulation is to measure the protocol's ability to react to asymmetric paths in the topology. For a more thoroughly examination of the mechanism the simulation is carried out with circles of a different size. Topology A consists of only 4 nodes, while topology B increases the circle size to 12 nodes. While in topology A nodes 0 and 1 have 2 hops in-between them, in topology B there are 6 hops that must be overcome.

The duration of the simulation is 1200 seconds. Whereas the routing mechanism is started after 10 seconds, the data traffic is started after 200 seconds. This means the transient phase of the protocol has passed, as the link and path estimation has already reached its maximum accuracy.

5.1.2 Results

The protocols are configured as shown in Table 5.1. The clockwise link quality changes from 0.7 to 1 in steps of 0.05.

Topology A

Figure 5.2 shows very contrary results. Version III has a linear curve, increasing proportional to the increasing link quality. This points to the fact that it is always picking the lossy path. Therefore the packet delivery ratio matches quite good with the squared link quality on the lossy links. The result is not surprising as the path quality metric used by version III correlates directly with the receive quality, which is in this case better over the lossy path.

Protocol IV's estimation of the transmit quality leads to the correct path choice, leading to an almost constant reliability of 1. Just in the cases of low asymmetry of the links it differs a little bit from 1. The link estimation cannot properly determine the lossy path, because of its limited accuracy.

Topology B

As can be seen in Figure 5.3 the increased size of the circle does not have an impact on the behavior of version III. Its packet delivery ratio still corresponds directly to the lossy path quality to the power of 6.

Version IV shows a different behavior than in the foregoing experiment. When dealing with a low packet loss ratio along the lossy path the reliability remains near 1 meaning that the lossy path is always correctly avoided. This is not the case anymore with a link quality below 0.8. The reason is the limited size of the global sliding window. As originator messages over the lossy path are reliably received, the current sequence number

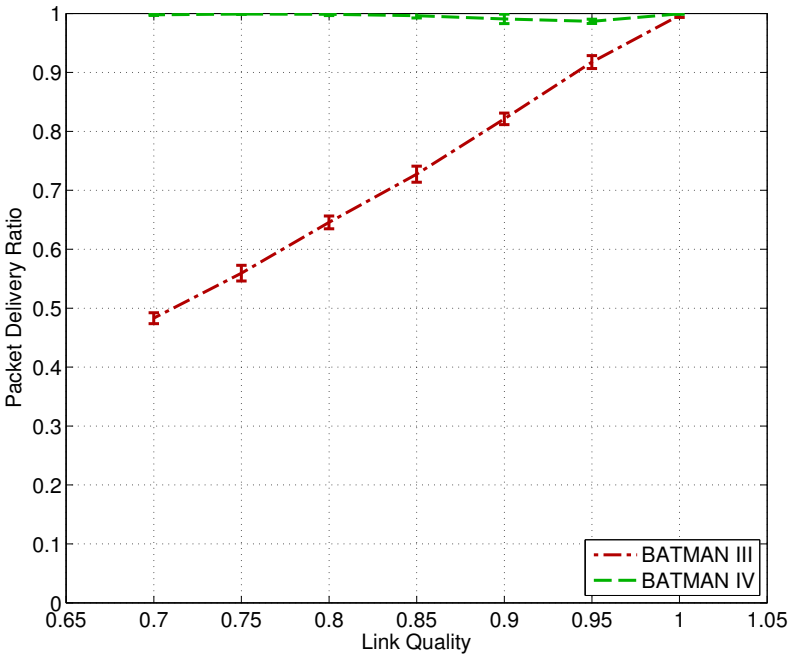


Figure 5.2: Circle, 2 Hops, Reliability

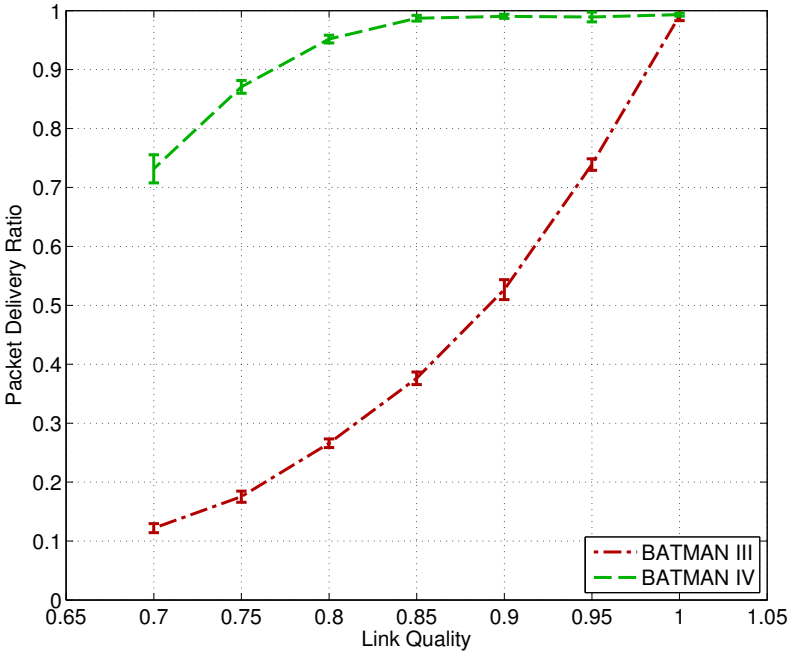


Figure 5.3: Circle, 6 Hops, Reliability

of the distant originator is always correctly recorded. Therefore, the sliding window for the path estimation over every neighbor is put forward on a regular basis. The global TQ value is the average of the sliding window entries unequal zero. Hence, if a node does not receive an OGM over the good path for a certain time, the global sliding window for the corresponding neighbor is emptied leading to a global TQ value of 0. This leads to a change of the routing decision towards the bad path until a new OGM is received over the good path.

5.2 Handover Scenario

This section evaluates the protocols' ability to detect topology changes, caused by a single moving node. Moreover, the impact of differing configuration of the protocols' parameters are monitored.

5.2.1 Scenario Description

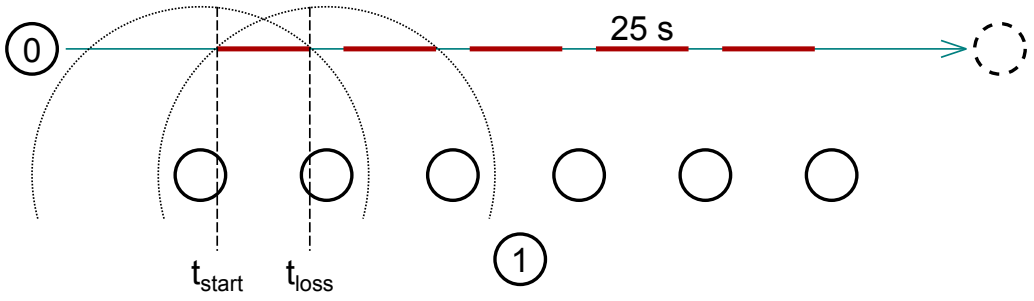


Figure 5.4: Handover Scenario

This scenario holds seven static nodes and one mobile node moving along the other ones. The setup is shown in Figure 5.4. Node 0 moves over a distance of 575 meters. The rest of the nodes are placed 75 meters apart with a distance of 75 meters forming a chain topology, except for node 1, which is placed nearby the center of the other static nodes. The movement of node 0 takes 250 seconds, with a speed of 2.3 m/s. With a transmission range of 100 meters of each node this leads to a range of ≈ 57 meters, in which the mobile node is within reach of 2 nodes on its path. Therefore the time between t_{start} and t_{loss}, when a node enters and leaves the according areas of reception, is about 25 seconds.

The target of this scenario is to investigate the protocols' abilities of detecting changes in the topology, that occur due to the movement of a node within the net. For a better insight, the simulation is run with changing values of different parameters of the protocols. These are the inter-arrival time of originator messages, the size of the used sliding windows, and the bidirectional timeout range.

Apart from the varying parameters of the simulation runs, the protocols are configured as in the circle scenario (Section 5.1) shown in Table 5.1. The duration of the simulation is 600 seconds. The routing mechanism is started after 10 seconds. After 350 seconds nodes 0 and 1 send data packets to each other, with a constant interval of 1 second. The movement phase of node 0 and the traffic phase overlap perfectly, while the routing phase is initiated earlier, in order to have the nodes fully adjusted to a topology without node 0. During the movement/traffic phase there is a small period of time at the beginning and the end, when node 0 is not within reach of the rest of the network.

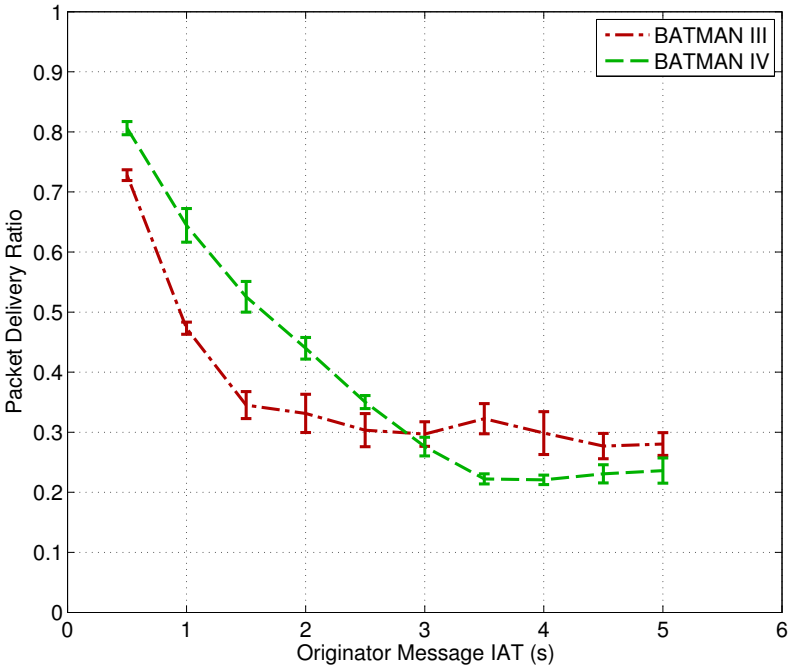


Figure 5.5: Handover, OGM Inter-Arrival-Time

5.2.2 Results

OGM Inter-Arrival Time

Figure 5.5 shows the results of the simulation run with the inter-arrival time of the OGMs changed from 0.5 to 5 in steps of 0.5. The dissemination of OGMs is a sampling of the current topology. The changes are at least recognized after *sliding window size* × *OGM inter-arrival time* seconds. In the given setup this time ranges from 32 to 320 seconds. However the topology change is likely to be recognized earlier, since the memorized sequence numbers partly reflect the new topology, after the first OGM was received over the new route. This accounts for the local link estimation in both protocol versions. Version IV though will switch the route earlier, due to the size of the global sliding window, which will be emptied after a time ranging from 5 to 500 seconds, after leaving the transmission range of the first node. In practice it may take longer, since OGMs may be delayed by the packet aggregation and by the loss of OGMs, which hinders the global sliding window from moving forward.

The simulation shows that the longer inter-arrival time has a severe impact on the reliability, as both curves are decreasing very fast. With a high frequency version IV profits from the earlier route changes mentioned above. On the other hand it then falls below the curve of version III with a constant reliability just above 0.2. This matches about the number of packets transmitted during the time, when node 0 is within reach of the first node. This means, once that route had been established, it was not changed anymore during the rest of the simulation. Version III in these cases features a shorter time to recognize the topology changes and has therefore a slightly better reliability.

Sliding Window Size

The effects of a varying sliding window size and local sliding window size respectively, are plotted in Figures 5.6 and 5.7. Simulations were run with a size of 5 to 100 incremented in steps of 5. The OGM interval in the first figure has been set to 1 second, in the second figure it has been set to 0.5 seconds.

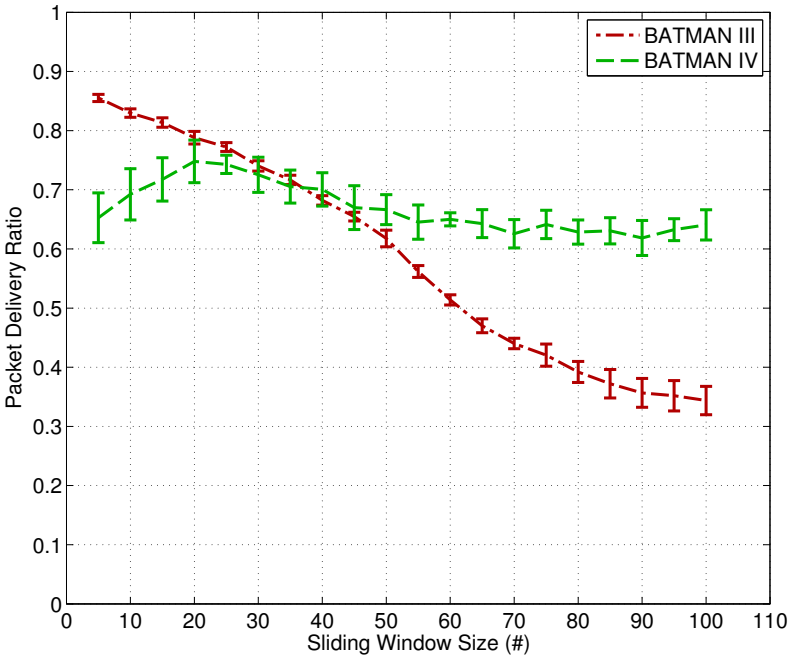


Figure 5.6: Handover, Window Size/Local Window Size, OGM IAT = 1s

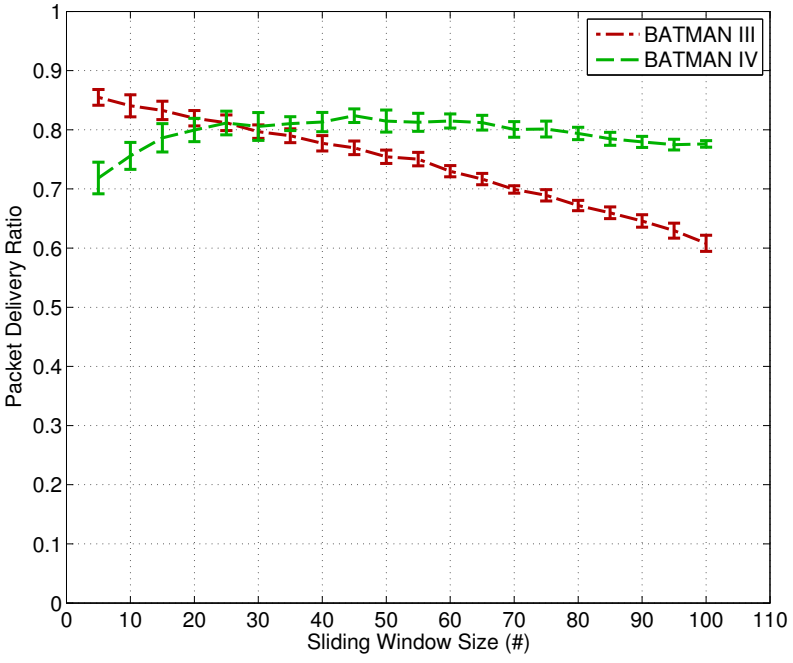


Figure 5.7: Handover, Window Size/Local Window Size, OGM IAT = 0.5s

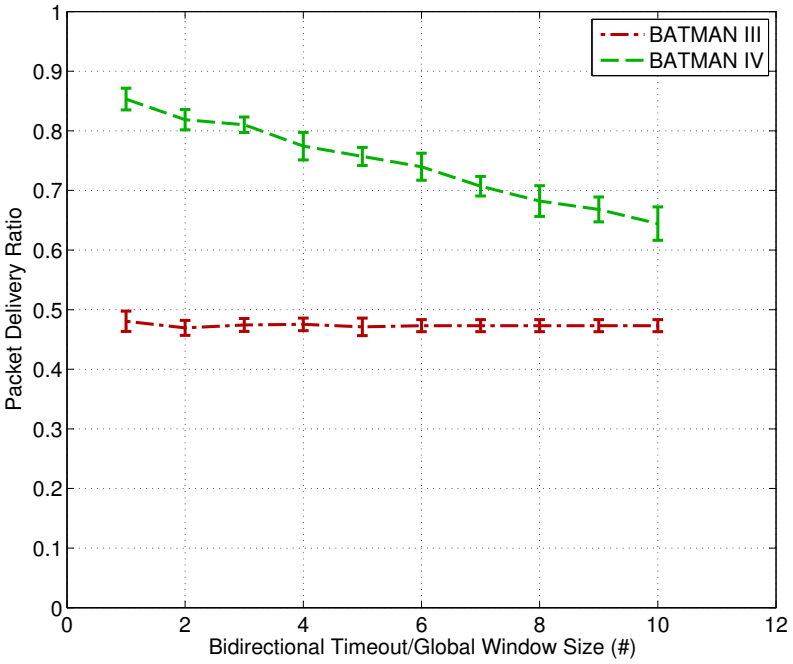


Figure 5.8: Handover, Bidirectional Timeout Range/Global Window Size

In version III, this parameter carries much weight of the resulting performance in this scenario, due to the fact that it directly influences the protocol's reaction time on topology changes. This ranges from 5 to 100 seconds in the present setup. That explains the fast decline of the curve.

In version IV, the routing decision is changed at last, when the global sliding window does not contain any global TQ values for the outdated path anymore. In this setup this is most likely the case, about 10 seconds after t_{loss} for the first node. By then the incoming OGMs over the new path will have set the entries for the outdated path to zero. The handover may be performed earlier, when the new path provides a higher average of its window entries than the outdated one at an earlier point in time. It is remarkable that for sizes smaller than 20, the behavior worsens although the local link estimation reflects the new topology sooner. But then the small window size negatively affects the accuracy of the resulting TQ values, causing a later path change. Bigger local window sizes on the other hand less accurately reflect the changing topology, but the negative impact keeps within a limit due to the smaller global window size. Figures 5.6 and 5.7 show that there is an extension of the peak when the highest reliability occurs with an increasing number of link probing packets (OGMs). As the sliding window represents a shorter time interval, in which the probing took place, even a higher window size better recognizes the topology change. The higher OGM frequency does not have an impact on the inaccuracy of the local link estimation occurring with small sliding windows. In general the local TQs merely state the existence of links than an actual link quality in this scenario, while the behavior is remaining constant, due to the constant global window size.

Bidirectional Timeout/Global Window Size

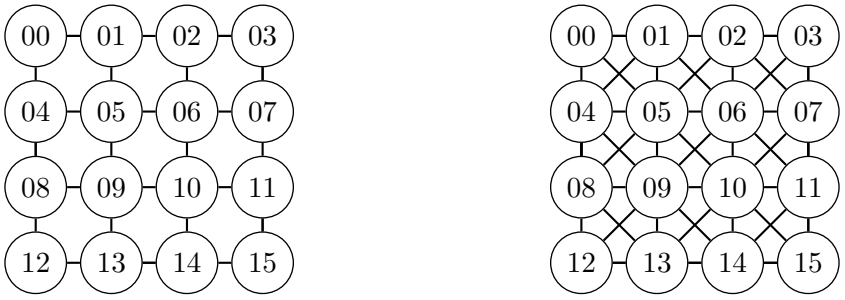
Figure 5.8 for version III of the protocol shows the influence of the bidirectional link timeout value on the performance in the scenario. The reliability is not affected by a changing value, as it has no effect on the resulting metric values. The mechanism prevents the protocol from using unidirectional links, which are not present in the given scenario.

For version IV Figure 5.8 shows the impact of the global sliding window size. The reliability worsens linearly with a bigger sliding window size. The reason for this effect is, as stated in the section before, that the global sliding window size directly determines the time when the path change occurs lastly. This is its size multiplied with the OGM inter-arrival time.

5.3 Grid Scenario

In this section a random lossy asymmetric link setup is given to test the protocols' ability of finding reliable paths within.

5.3.1 Scenario Description



(a) Topology A (b) Topology B

Figure 5.9: Grid Example Topologies

In this scenario $7 \times 7 = 49$ nodes are set up as a static grid with a differing node distance, in order to create a differing network topology. In case A the node distance is chosen to be 75 meters, whereas in case B the distance is chosen to be 60 meters. With a transmission range of 100 meters this leads to two topologies. As depicted in Figure 5.9 exemplary for 16 participating nodes the varying node density causes the middle nodes in case A to have 4 neighbors. In case B the middle nodes have 8 neighbors each. Therefore these topologies offer 84 and 228 links.

The purpose of this setup is to measure the performance of the protocol with regard to an increasing number of asymmetric links within the topology. The chosen scenario does not have any network partitioning, while having a pairwise average hop count between nodes of 4.66 and 3.29 respectively, in order to render the routing decision rather challenging.

The simulation is carried out with a duration of 1100 seconds. The routing mechanism is started after 30 seconds. Traffic is sent after 100 seconds. Therefore, the sliding windows of the nodes will have filled already so that the routing decision is made upon fully informed states of the links. All nodes generate data packets with a constant packet size of 1024 bits, which are sent in constant time intervals of 1 second. Each node at the beginning of the simulation picks a random destination to send packets to.

To simulate a level of asymmetry within the topology, originator messages and data packets are being filtered. Hence, at the beginning of the simulation, links are randomly chosen to be asymmetric with a link quality between 0.2 and 0.8 in one direction and 1 in the other. The asymmetric ratio determines the percentage of all present links on which the filter is inflicted.

5.3.2 Results

Figures 5.10 and 5.13 show the resulting reliability of the simulation with an increasing asymmetric link ratio from 0 to 1 in steps of 0.2. Apart from the performance of the two

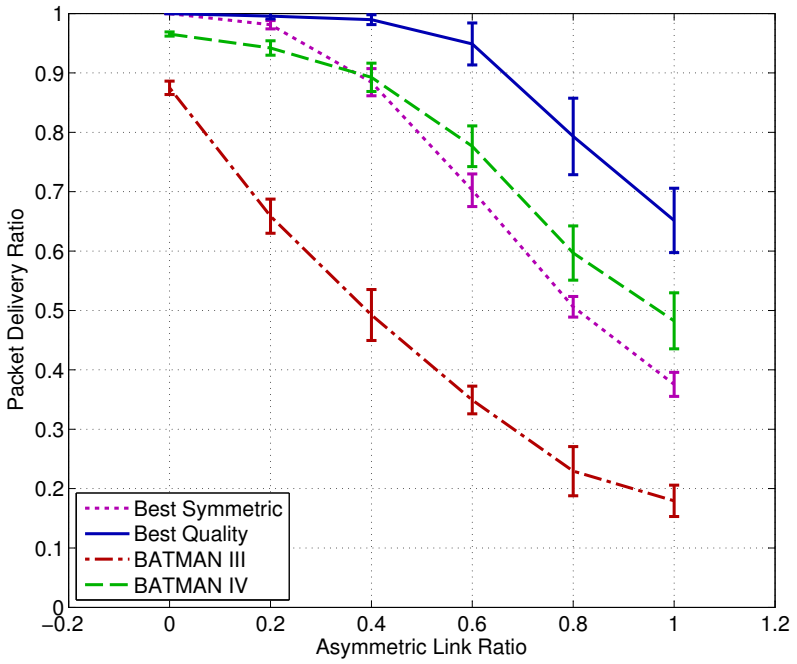


Figure 5.10: Grid, Topology A, Reliability

versions of the B.A.T.M.A.N. protocol, another curve shows the best reliability, that is theoretically possible to achieve. This curve is obtained by calculating the shortest path from every node to its traffic destination and averaging this value for every node and run. As link costs the inverse path quality was used. Path costs result in a multiplicative combination of link costs. While the amount of asymmetric links remains the same in the compared simulation runs, the best possible average path costs differ. This is due to the random link qualities in every simulation run.

Additionally another curve is drawn, called best symmetric. This curve describes the expected reliability the filter module would allow, when paths are chosen in the following way: The underlying graph, the shortest path calculation is performed upon, is turned into an undirected graph. As link costs then the ETX metric value is taken for both directions. The purpose of this curve is to have a curve corresponding to a symmetric routing metric, that nonetheless punishes the use of asymmetric links.

Topology A

The curve of the best expected reliability shows a clear decreasing tendency with an increasing number of asymmetric links. With a low asymmetry ratio the best possible paths remain at a value near 1, as there are almost always alternative routes completely avoiding lossy links. In the given topology this is not the case anymore with a high asymmetric link ratio, but the resulting values differ more from one another, since the level of randomness is higher with a higher number of links affected by a random loss in one direction.

The curve of version III decreases almost linearly with the increasing number of asymmetric links, which is symptomatical for its tendency to route in the wrong direction with an asymmetric link topology, as was already the case in the circle scenario (Scenario 5.1). Conforming with this outcome is the picture drawn in Figure 5.11, which shows that the average end-to-end hop count of the received packets decreases as well. This points to the fact that particularly packets routed over long paths are lost. As longer paths accumulate

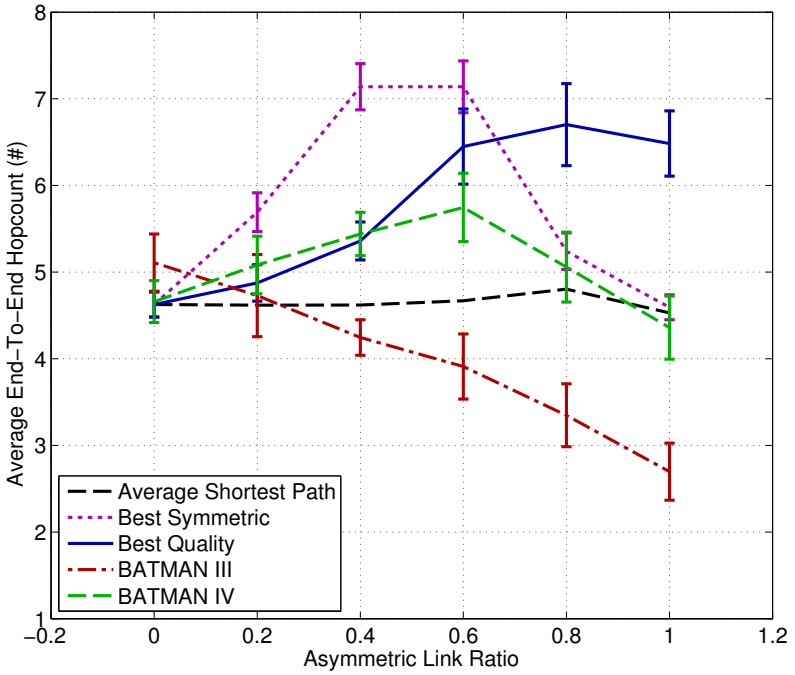


Figure 5.11: Grid, Topology A, Average End-To-End Hop Count

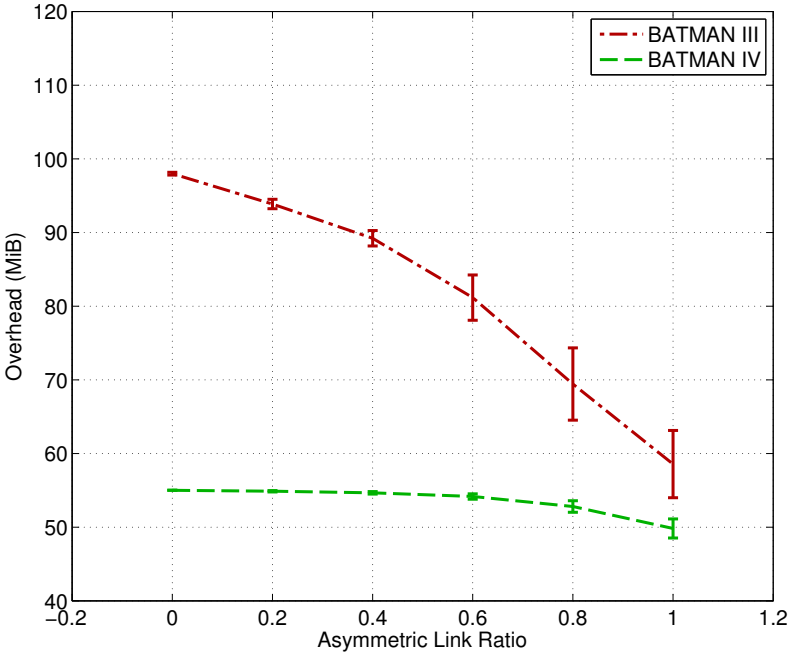


Figure 5.12: Grid, Topology A, Overhead

lower reliabilities, the ones chosen apparently prove to be the lossier ones. Figure 5.11 also shows that version III does not detect the shortest paths available in the topology, without link loss present. This accounts to the circumstance, that OGMs do not spread as successfully in the center of the setup, where the node density is higher than at the border. Therefore it has a slight tendency to prefer longer paths along the border over the shortest path.

Version IV's curve on the other hand orients itself much on the best expected curve. This shows the protocol indeed tends to choose the path showing the highest quality. Yet the resulting values are lower for the hop penalty applied to the metric may lead to a path with a shorter hop count to be favored over one with a better quality. In addition, due to the asymmetric penalty, links with a high difference of link qualities in each direction may be avoided, even though they're part of the best path. Figure 5.11 shows that the protocol chooses the shortest possible path without asymmetry in the topology. Then the average hops increase, since the asymmetric links can at first be avoided by evading the asymmetric links with the usage of a longer alternative. At the end of the curve the value decreases again, since the growing number of asymmetric links entails a decreasing number of better alternative paths. In addition, the qualities of the most reliable path and the shortest path are less likely to differ much from one another.

Noticeably, version IV's routing metric outperforms the best symmetric reliability in Figure 5.10 already with an asymmetric link ratio around 0.4. Since the best symmetric curve is an idealized calculation, not taking into account the packet loss caused by the free space propagation model of the simulation, a routing protocol performing a metric alike this would achieve worse results. This indicates that there can be achieved a high benefit in terms of reliability by choosing an asymmetric metric in general.

The generated overhead may be seen in Figure 5.12. For version III it decreases dramatically with a decreasing link quality, while version IV maintains an almost constant level. This owes to version III's strict packet drop policy. In an ideal unpartitioned network without lossy links, both protocols rebroadcast every non-duplicate once. With lossy links, version III just rebroadcasts OGMs incoming over the best ranking neighbor or if the non-duplicate was received over a path with an equal length. This policy is necessary to prevent routing loops, similar to the feasibility conditions in the Babel protocol. Version IV on the other hand basically rebroadcasts any non-duplicate. There remains a high probability that any OGM reaches every node at least once, even with bad link conditions. Therefore, the amount of overhead remains at a constant level. While one version IV OGM has a bigger size, the packet aggregation mechanism significantly reduces the relative size of one packet. Version IV's overhead is thus much lower, although supposedly the same amount of messages is exchanged.

Topology B

The higher node density in the second topology example leads to a much higher number of links offered by the setup. Both protocols clearly profit from this circumstance, in that they have more routing options available.

As can be seen in Figure 5.13, version III still has a linear decrease with a growing number of asymmetric links. The received packet count metric results in paths, with a high probability of a successful reception from a certain node. In the other direction, these paths have a linear increasing probability of being afflicted with a link loss. This mirrors in the resulting curve.

Version IV maintains an almost constant reliability up to an asymmetric link ratio of 0.6. In comparison to the foregoing simulation it manages to find high reliable paths more

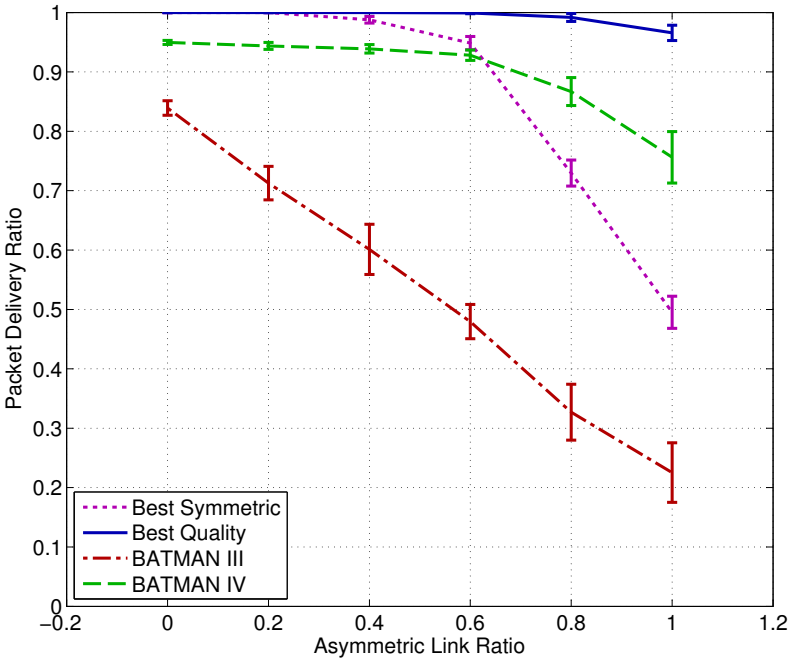


Figure 5.13: Grid, Topology B, Reliability

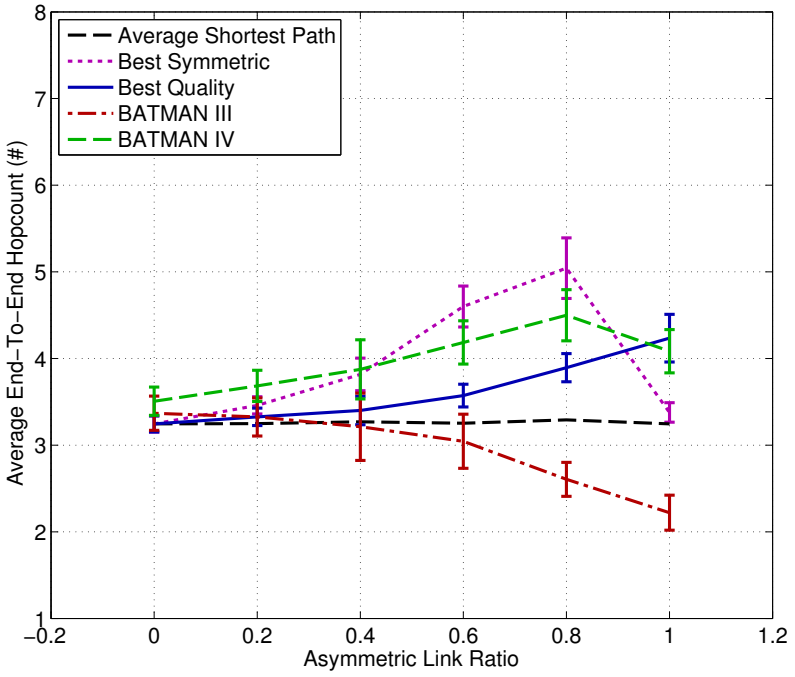


Figure 5.14: Grid, Topology B, Average End-To-End Hop Count

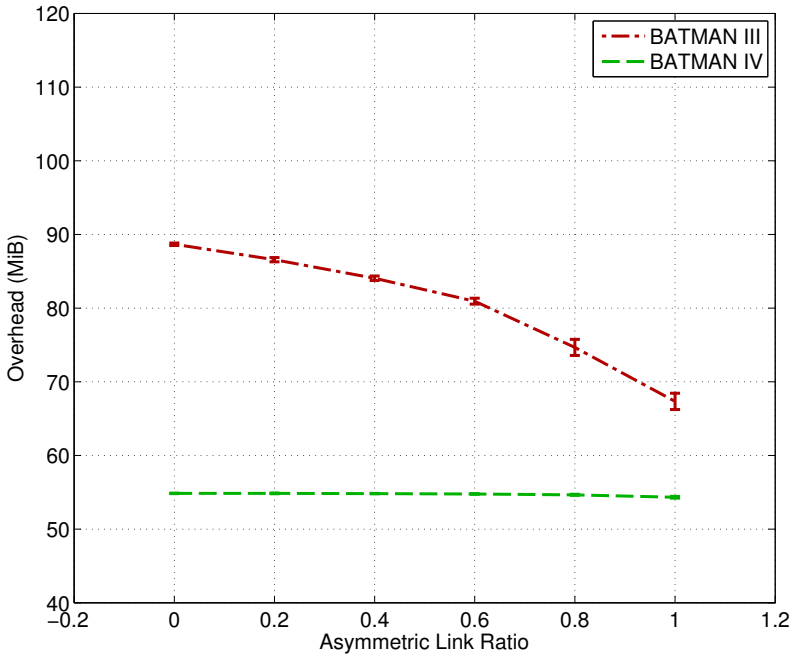


Figure 5.15: Grid, Topology B, Overhead

easily. While the best expected average reliability remains almost 1, even with every link being asymmetric, version IV's reliability becomes lower. Since every link is then penalized with the asymmetric penalty, the better routing decision is harder to make. Particularly on longer paths, this has a severe impact on the resulting global TQ value, and makes the best path indistinguishable from one with less asymmetry.

Compared to the values of the best symmetric curve, version IV's benefit of its link metric exhibits itself only with a high ratio of asymmetric links, then being equally significant as in topology A. Thus, this approach is particularly promising with a lower number of alternative paths.

Figure 5.14 draws the same picture as Figure 5.11 in section 5.3.2. Except for both curves in the beginning staying closer to the average shortest path hop count. This stems from the higher probability of finding a high reliable short path. Noticeable is that, while there are many paths present with a reliability of 1, version IV does not always decide for the shortest one among them. Due to the low chosen hop penalty, equally reliable paths with a low hop difference are not reliably distinguishable.

The overhead generated in the topology B scenario (see Figure 5.15) is very similar to that generated in topology A (Figure 5.12). The higher number of available links does not affect the behavior significantly. Version III's overhead shows a lower overall level owing to the fact that the average shortest path between a node pair has less hops in this topology. The rebroadcasting mechanism does not make use of longer alternative paths.

5.4 Mobility Scenario

This section comprises of a scenario, which exposes the protocols to a random walk mobility model.

5.4.1 Scenario Description

In this scenario 50 nodes are placed randomly on a field of 1000×1000 meters. The nodes are moving randomly to the rules of a mobility model with an increasing maximum speed.

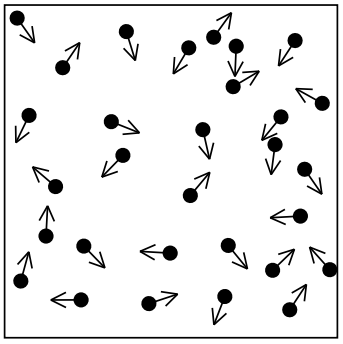


Figure 5.16: Mobility Scenario

Parameter	Version III	Version IV
OGM IAT	0.5s	1s
OGM TTL	16	16
OGM Jitter	0.1s	0.1s
Purge Timeout	2000	2000
Window Size	16	-
Bidirectional Timeout Range	3	-
Local Window Size	-	24
Global Window Size	-	3
Hop Penalty	-	24
Maximum Aggregation Time	-	0s

Table 5.2: Mobility Protocol Configuration

The transmission and interference range of the nodes is limited to 200 meters. Figure 5.16 shows the principle of the setup.

The goal of the scenario is to simulate a mobile network with a randomly changing topology over time. The increasing maximum node speed causes more changes of the topology. Each node emits a constant data traffic to a random destination per packet. Thus, the reliability measures the protocols' overall ability to maintain the connection of the network.

The duration of the simulation was 1500 seconds with a transient phase of 500 seconds, before the traffic model was started. The routing protocol was activated after 400 seconds. Every data packet was sent to a random destination, thus randomly sampling the existence of an active route between unfixed node pairs. As in the foregoing scenarios, the data packet size is set to 1024 bits and the inter-transmission time is set to 1 second constantly.

In order to increase the performance of the protocols, their parameters were adjusted to better react on mobility. The exact configuration is listed in Table 5.2. In order to improve version III's topology change recognition, essentially the OGM inter-arrival time was set to a half second and the window size was reduced to 16. For version IV a small global window size was chosen to narrow the emphasis of the path estimation on the most recent received TQ values. The local sliding window size was reduced analogously. A higher hop penalty was chosen, taking into concern the fact that a longer route is more likely to break in a mobile scenario, since more moving nodes are involved. The maximum aggregation time is set to zero seconds, thus effectively turned off, since the mechanism obfuscates the topology representation through slowing down the dissemination of recent information. The purge time-out is set to a value longer than the simulation duration to induce a most opportunistic routing behavior. Thus, packets that otherwise might be dropped for the

Parameter	Value
Start Time	10s
Stop Time	∞
Minimum Speed	1m/s
Maximum Speed	2 - 16m/s
Movement Time	10s
Pause Time	0s

Table 5.3: Random Walk, Parameter Configuration

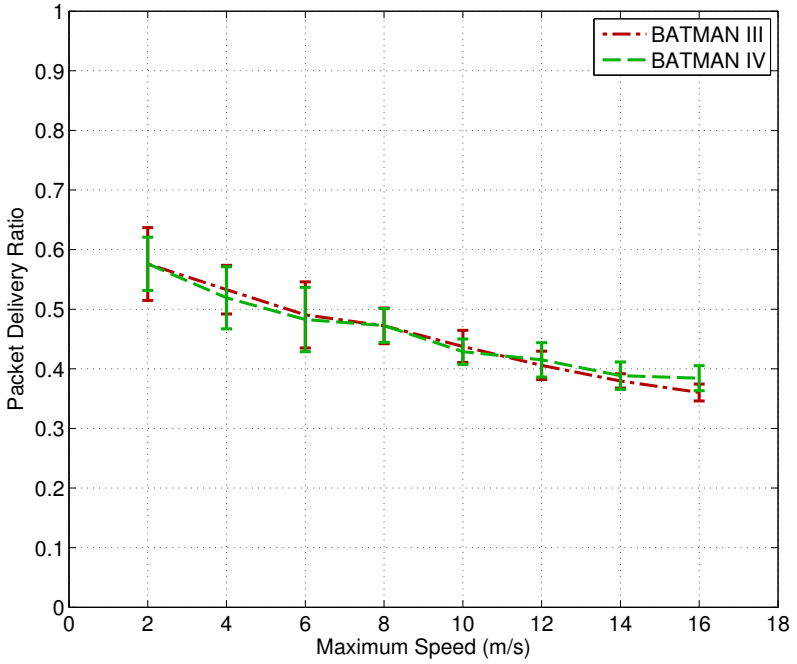


Figure 5.17: Random Walk, Reliability

destination node's removal from the routing table, are still being relayed. By this behavior, the probability of a packet being dropped, because of a falsely deleted route, is eliminated.

The nodes move according to a time-based random walk mobility model. The configuration of its parameters is shown in Table 5.3.

5.4.2 Results

Figure 5.17 displays the resulting reliability of the simulation, which turns out to be quite similar to each other. Since the scenario does not hold asymmetric links, version IV cannot profit in any way from its transmit quality considerations. Instead, the performance of both versions is rather determined by their mean time of detecting changes in the topology, which is more or less even.

The general approach of information dissemination by rebroadcasting single OGMs over the network, remains similar in both versions. As a result, the impact of a higher maximum node speed is about even. Thus, the curves are both decreasing in a similar manner.

In Figure 5.18 can be seen that with the increase of the node speed longer routes are failed to be maintained. Apart from that version IV has a slight tendency to choose longer routes. This accounts for version III's strict OGM drop policy, which leads to a more concise selection of the shortest path than version IV's hop penalty allows.

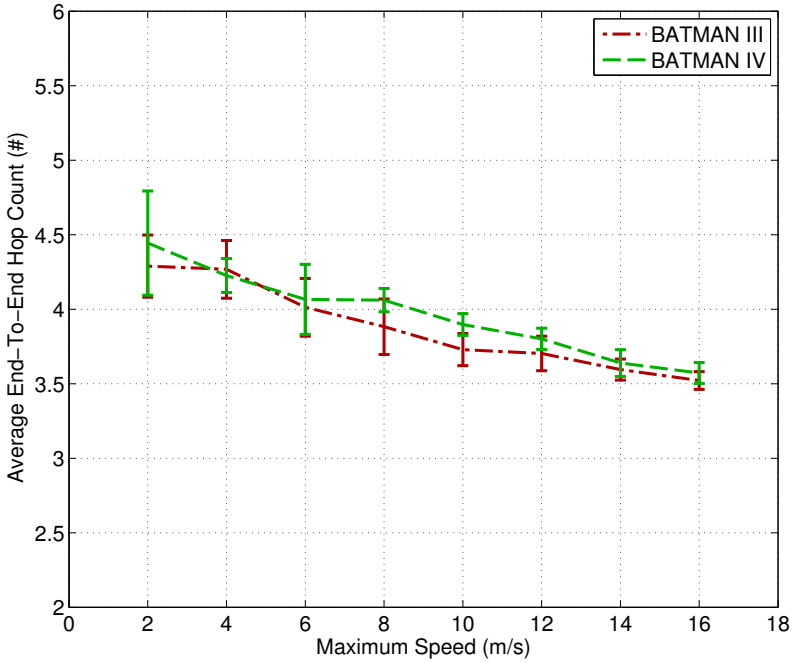


Figure 5.18: Random Walk, Average End-To-End Hops

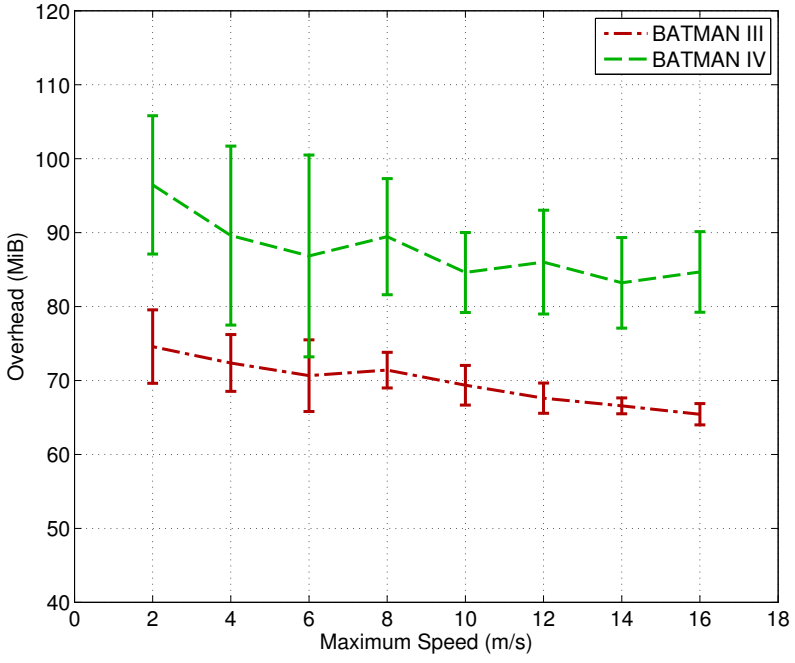


Figure 5.19: Random Walk, Overhead

The produced overhead depicted in Figure 5.19 behaves, as the other values measured, declining with an increasing maximum speed. As the packet aggregation mechanism of version IV was turned off, the generated overhead becomes more comparable. The maximum possible overhead is in both protocols produced, when every OGM is rebroadcasted once by every other node. This results in about 133 MiB for version III and 146 MiB for version IV. This value is not reached due to temporary network partition and packet loss. Apart from that, version III does not rebroadcast OGMs as easily as version IV does, which too leads to a lower overhead. The randomness of the mobility model causes especially in version IV of the protocol much varying numbers. In general, the amount of overhead is determined by the connectivity of the network.

5.5 Evaluation of the Asymmetric Link Mechanism

This section provides a concise evaluation of the impact of asymmetry in the network topology concerning the two protocol versions' mechanisms.

Version III

Version III's approach of measuring path quality has a certain appeal, because of its simplicity. Yet the underlying assumption, that a path considered well suited in one direction may be used in the other as well, is not correctly reflecting the network topology. As expected, simulations have shown, that with asymmetric paths between nodes, the protocol will not choose the right one. Due to the simplicity of the protocol, there is also no possibility to adjust for an avoidance of the usage of asymmetric links. However, the protocol is able to perform quite well if the overall link quality is high. It soon adjusts to changes in the topology when a small sliding window is set. Additionally the relative small size of an originator message does limit the produced overhead. Though, the protocol configuration has to make a compromise between how fast and accurate the topology is recognized, and how much overhead is produced while doing so.

Version IV

Version IV's mechanism gains with little extra effort more information of the network. Simulation has shown that it is quite effective in measuring the TQ metric. Thus the metric is very well suited to finding high reliable paths in the topology. Its closeness to the ETF metric, mainly incorporating the directional transmission success probability, is more obfuscated by the asymmetric link penalty than the hop penalty in its default configuration. Therefore, it shows a slight tendency towards choosing longer routes.

The mechanism putting the metric into effect suffers from similar difficulties, as it is with version III. There exists the same tradeoff between generated overhead and topology change reaction time. Compared to other routing protocols mentioned in this work it is disadvantageous to have the task of measuring local link qualities and propagating path qualities bound to the same packet format. This circumstance forces to do so in the same time interval. Especially in static networks the secondary task might be performed less often to save overhead.

In addition, in extreme cases of asymmetric paths the mechanism is confronted with the problem of recognizing a very reliable path, which is seldom propagated. As shown in the circle scenario, the limited global window size brings about a behavior of switching the route decision towards a path that is propagated more often. Contrariwise, the global sliding window immediately affects the mechanism's ability to adapt to a new topology, which is why it cannot be set too big.

Furthermore, the TQ representation is integer based, linearly ascending from a minimum value, depicting a link as not present, to a maximum value, considering it perfect. However

the link metric is combined in a multiplicative manner. It is more desirable to have a better distinction between two lossy paths than between two, where either alternative is a reliable choice. Thus, the protocol might profit from another number representation, such as a logarithmic scale, that offers higher accuracy on lossy path estimations.

6. Conclusion

Routing in adhoc networks is a challenging task. There is a wide variety of use cases, causing differing demands to the protocols in use. For example in a sensor network environment, where energy consumption should be kept down to a minimum level, a proactive measurement of link qualities is rather unreasonable. On the other hand, in a highly mobile environment it is inevitable to have a constant evaluation of the network topology to avoid outdated routing decisions.

The scenario, the B.A.T.M.A.N. protocol initially was created for, is a mesh environment, where participating nodes are normally attached to a wall and connected to a power supply system. While its first approach of measuring path quality instead of link quality shows some flaws, the link metric put in charge in the current version is an adequate estimation to find reliable paths, if available. By this modification of the former approach the functioning aligns more with the principles of concurrent solutions, but remains fairly simplistic.

The protocol offers parameters that allow a better configuration for certain scenarios. It is obviously more aware of topology changes when originator messages are sent with a higher frequency. The core principle of both protocol versions lies in the usage of the sliding window approach for link detection. In version III, a small sliding window yields a fast detection of available paths, which makes it very well suited in highly mobile environments. Under such circumstances detection of links is more important than making assumptions on their characteristics. Due to the fact, that the metric is incapable of exploiting asymmetric links, but rather employ them to the network's disadvantage, it does not qualify for utilization in a static environment.

Version IV on the other hand proves the concept to be applicable for local link estimation. Depending on the window size, it can give a more or less accurate average evaluation. Unfortunately, more accuracy leads to a higher inertia of detecting topology changes. Smaller sliding window sizes are not suited here to reduce this inertia of the mechanism. Reduction of the local sliding window decreases the informative value of the link metric. With a small global sliding window the protocol will not put the measured metric in charge anymore, but simply detect shortest hop paths. In immobile scenarios, though, the asymmetric link weighting makes a beneficial use of the resources provided by the network. In general, version IV's approach is similar to a distance vector mechanism and as such profits from a low computational complexity.

The conducted simulations have also shown the effect of making simplified assumptions on a complex system. Indeed these are inevitable, just to keep calculation time and effort

within boundaries, yet a simplified model may lead to wrong conclusions. This can be seen in the handover and mobility scenario, where a strictly bidirectional radio wave propagation model is applied. Under these circumstances both protocol versions perform equally well, when configured appropriately. Version III's significant drawbacks can only be witnessed, when lossy link reliabilities are introduced. Simulation results should therefore always be viewed critically.

In the future it might be evaluated how improvements on the mechanisms pay off, such as the planned reactive link break detection mechanism of the upcoming generation of B.A.T.M.A.N.. Furthermore, the commonality of the OLSRv2 and Babel protocol to not determine the metric in use brings up the question, how their performance would be affected, if B.A.T.M.A.N.'s metric was used. In general, a comparison between B.A.T.M.A.N. and other routing protocols in appropriate simulation scenarios could be of interest. Of course, a thorough consideration of the protocol and routing metrics in a real testbed is worthwhile as well. Eventually, an analysis of cross-layer effects is conceivable, since the utilized MAC protocol may strongly influence the interaction with the used routing metric.

Bibliography

[ACC+09] S. Annese, C. Casetti, C. F. Chiasserini, P. Cipollone, A. Ghittino, and M. Reineri: *Assessing mobility support in mesh networks.* In *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization*, WINTECH '09, pages 19–26, New York, 2009. ACM.

[AHW09] M. Abolhasan, B. Hagelstein, and J.C. P. Wang: *Real-world performance of current proactive multi-hop mesh protocols.* In *Proceedings of the 15th Asia-Pacific conference on Communications*, APCC '09, pages 42–45, Piscataway, 2009. IEEE Press.

[Aic07] C. Aichele: *Mesh: Drahtlose Ad-hoc-Netze.* Open Source Press, München, 2007.

[ALL+11] C. Aichele, L. Lüssing, M. Lindner, S. Wunderlich, and S. Eckelmann: *Open mesh website*, August 2011. http://www.open-mesh.org/, visited on 13.8.2011.

[AW05] I. F. Akyildiz and X. Wang: *A survey on wireless mesh networks.* IEEE Communications Magazine, 43(9):23–30, September 2005.

[AWK+09] A. Al Basset Almamou, R. Wrede, P. Kumar, H. Labiod, and J. Schiller: *Performance evaluation of routing protocols in a real-world WSN.* In *Proceedings of the Second international conference on Global Information Infrastructure Symposium*, GIIS '09, pages 9–13, Piscataway, 2009. IEEE Press.

[BC11] M. Britton and A. Coyle: *Performance analysis of the B.A.T.M.A.N. wireless ad-hoc network routing protocol with mobility and directional antennas.* http://downloads.open-mesh.org/batman/papers/SANLABrytis.pdf, 2011.

[BIM+09] L. Barolli, M. Ikeda, G. Marco, A. Durresi, and F. Xhafa: *Performance analysis of OLSR and BATMAN protocols considering link quality parameter.* In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications*, AINA '09, pages 307–314, Washington, 2009. IEEE Computer Society.

[CABM03] D. Couto, D. Aguayo, J. Bicket, and R. Morris: *A high-throughput path metric for multi-hop wireless routing.* In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 134–146, New York, 2003. ACM.

[CDJ11] T. Clausen, C. Dearlove, and P. Jacquet: *The Optimized Link State Routing protocol version 2 internet draft.* http://tools.ietf.org/html/draft-ietf-manet-olsrv2-12, July 2011.

[Chr11] J. Chroboczeck: *The Babel routing protocol.* Technical Report RFC 6126, Network Working Group, April 2011.

[CJ03] T. Clausen and P. Jacquet: *Optimized Link State Routing protocol (OLSR).* Technical Report RFC 3626, Network Working Group, October 2003.

[Her11] U. Herberg: *Performance, Scalability, Automatic Management and Internet Integration of Ad Hoc Networks.* PhD thesis, École Polytechnique ParisTech, May 2011.

[IMBT08] M. Ikeda, G. Marco, L. Barolli, and M. Takizawa: *A BAT in the lab: Experimental results of new link state routing protocol.* In *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications*, AINA '08, pages 295–302, Washington, 2008. IEEE Computer Society.

[JNA08] D. Johnson, N. Ntlatlapa, and C. Aichele: *Simple pragmatic approach to mesh routing using BATMAN.* In *Proceedings of the 2nd IFIP International symposium on Wireless Communications and Information Technology in developing countries*, WCITD '08, Pretoria, October 2008. CSIR.

[KIBM10] E. Kulla, M. Ikeda, L. Barolli, and R. Miho: *Impact of source and destination movement on MANET performance considering BATMAN and AODV protocols.* In *Proceedings of the 2010 International Conference on Broadband, Wireless Computing, Communication and Applications*, BWCCA '10, pages 94–101, Washington, 2010. IEEE Computer Society.

[Kle10] A. Klein: *Performance Issues of MAC and Routing Protocols in Wireless Sensor Networks.* PhD thesis, Universität Würzburg, May 2010.

[KNE03] D. Kotz, C. Newport, and C. Elliott: *The mistaken axioms of wireless-network research.* Technical Report TR2003-467, Department of Computer Science, Dartmouth College, July 2003.

[KW03] H. Karl and A. Willig: *A short survey of wireless sensor networks.* Technical Report TKN-03-018, TU Berlin, October 2003.

[MDK10] D. Murray, M. Dixon, and T. Koziniec: *An experimental comparison of routing protocols in multi hop ad hoc networks.* In *Proceedings of the Australasian Telecommunication Networks and Applications Conference*, ATNAC '10, pages 159–164, Piscataway, 2010. IEEE Press.

[NAL07] A. Neumann, C. Aichele, and M. Lindner: *B.A.T.M.A.N. status report.* http://downloads.open-mesh.org/batman/papers/batman-status.pdf, June 2007.

[OPN08] OPNET Technologies, Inc: *Opnet Modeler 14.5*, 2008. http://www.opnet.com/solutions/network_rd/modeler.html.

[Per01] C. E. Perkins: *Ad hoc networking.* Addison-Wesley, Boston, 2001.

[RCC09] M. Reineri, C. Casetti, and C. F. Chiasserini: *Routing protocols for mesh networks with mobility support.* In *Proceedings of the 6th international conference on Symposium on Wireless Communication Systems*, ISWCS '09, pages 71–75, Piscataway, 2009. IEEE Press, ISBN 978-1-4244-3583-8.

[SAZ10] L. Sang, A. Arora, and H. Zhang: *On link asymmetry and one-way estimation in wireless sensor networks.* ACM Transactions on Sensor Networks, 6(2):1–25, March 2010.

[WHA10] J.C. P. Wang, B. Hagelstein, and M. Abolhasan: *Experimental evaluation of IEEE 802.11s path selection protocols in a mesh testbed.* In *Proceedings of the 4th International Conference on Signal Processing and Communication Systems*, ICSPCS '10, pages 1–3, Piscataway, 2010. IEEE Press.

[WLT09] S. Wunderlich, M. Lindner, and W. Tsai: *B.A.T.M.A.N.-advanced documentation draft*, May 2009. http://git.open-mesh.org/?p=batman-adv-doc.git, visited on 30.8.2011.

[WTC03] A. Woo, T. Tong, and D. Culler: *Taming the underlying challenges of reliable multihop routing in sensor networks.* In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 14–27, New York, 2003. ACM.

[ZK07] M. Z. Zamalloa and B. Krishnamachari: *An analysis of unreliability and asymmetry in low-power wireless links.* ACM Transactions on Sensor Networks, 3(2), June 2007.