TECHNISCHE UNIVERSITÄT DRESDEN
FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING
Institute of Communication Technology

# DIPLOMA THESIS

*„Optimizing Broadcast Delivery in Dynamic Wireless Mesh Networks“*

by

*Marek Sobe*
Matriculation number: 3972737
Born on 28.10.1994 in Dresden

to achieve the academic degree

## DIPLOMINGENIEUR

(Dipl.-Ing.)

Submitted on: 04.11.2019
Supervisor of Diploma Thesis: Dipl.-Inf. Simon Wunderlich
Supervising professor: Prof. Dr.-Ing. Dr. h.c. Frank H. P. Fitzek

# TECHNISCHE UNIVERSITÄT DRESDEN
# FAKULTÄT ELEKTROTECHNIK UND INFORMATIONSTECHNIK

## Task for Diplom thesis

| | | |
|---|---|---|
| For Mr | Marek Sobe | ET/2013 |
| Topic | Optimizing Broadcast Delivery in Dynamic Wireless Mesh Networks | |

**Task description**

Wireless Mesh Networks (WMNs) enable creating networks over mobile devices with lossy wireless links. Next to route finding for unicast connections, another very important service is broadcast delivery, which is needed for retrofit fieldbus applications, broadcast/multicast services like video streaming or Ethernet Layer 2 services like ARP and DHCP. While "flooding" type mesh networks were the first type of mesh networks to be invented, recent advancements of different modulation rates (e.g. IEEE 802.11 WLANs provides modulation schemes from 1 MBit/s to multiple GBit/s) and neighbor discovery do not seem to be incorporated in modern mesh broadcast protocols.

The challenge is therefore to find a flooding broadcast mechanism which can be used for the mentioned applications. There should be a good tradeoff between different performance aspects: efficiency in terms of airtime can offer high throughput or minimize the effects of other services in the mesh network. High reliability and low latency are important for fieldbus applications.

The student's tasks are to:

1. Research the current state of the art in flooding mesh networks
2. Establish performance metrics for further evaluation
3. Implement the existing candidates in a packet simulator
4. Determine an upper performance reference for further comparisons
5. Compare the candidates and, if possible, propose modifications to optimize for some of the performance aspects
6. Discuss design choices and possible trade offs to optimize for certain performance aspects

| | |
|---|---|
| Supervisor | Dipl.-Inf. Simon Wunderlich |
| Reviewer | Prof. Dr.-Ing. Dr.h.c. Frank Fitzek |
| Second Reviewer | Dr.-Ing. Roland Schingnitz |
| Started at | 27.05.2019 |
| To be submitted by | 04.11.2019 |

The diplom/master thesis is written in English.

Prof. Dr.-Ing. Steffen Bernet
Vorsitzender des Prüfungsausschusses

Prof. Dr.-Ing. Dr. h.c. Frank Fitzek
Verantwortlicher Hochschullehrer

# Statement of authorship

I hereby certify that I have authored this Diploma Thesis entitled

„Optimizing Broadcast Delivery in Dynamic Wireless Mesh Networks“

independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. During the preparation of this thesis I was only supported by following persons:

Dipl.-Inf. Simon Wunderlich
Peter Sossalla
Sandra Zimmermann

Additional persons were not involved in the intellectual preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

Dresden, den 04.11.2019 Marek Sobe

# Abstract

Wireless Mesh Networks (WMNs) started as a niche development for military purposes in the 1990s. With the emergence of 5G and the paradigm shift it entails, WMNs gain in importance. The advantageous nature for a plethora of device to device communication use cases attracted wide research interests over the recent years. However, a topic that has received little research attention so far are broadcast transmissions. Unlike unicast and multicast approaches, current broadcast protocols for WMNs are sparse and reveal a lack of features. Flooding and further fixed rate schemes such as Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.) and Optimized Link State Routing (OLSR) fail to achieve a reliable delivery and at the same time a resource consumption minimization, required by present and future applications. This work therefore proposes a broadcast protocol RAIDER, that incorporates multi-rate capabilities and link quality awareness. To evaluate the performance, it introduces a new performance indicator to combine both the achieved reliability as well as the accumulated transmission time on the wireless medium. In doing so, a performance assessment of Rate Aware Information Dissemination with Extra Reliability (RAIDER) against the state of the art was possible. To complement the comparison, an upper performance bound for the considered broadcast scenario was established. Results show RAIDER accomplishing a clear improvement against existing broadcast protocols in both reliability and airtime combined. RAIDER performed five times as good as flooding. Compared to B.A.T.M.A.N., the airtime could be decreased by up to $95\%$, while maintaining a slight advantage in reliability. The performance of OLSR could be increased by $56\%$, including an increment in reliability of up to $20\%$. Although the performance considerably approached the upper performance bound, it could ultimately not be matched, leaving potential for further broadcast improvement in the future.

# Zusammenfassung

Wireless Mesh Networks (WMNs) begannen in den 90er Jahren als Nischenentwicklung für militärische Zwecke. Mit dem Aufkommen von 5G und dem damit einhergehenden Paradigmenwechsel gewinnen WMNs an Bedeutung. Die vorteilhafte Natur für eine Vielzahl von Anwendungsfällen der Device to Device Kommunikation hat in den letzten Jahren ein breites Forschungsinteresse geweckt. Ein Thema, das bisher jedoch wenig Beachtung gefunden hat, sind Broadcast Übertragungen. Im Gegensatz zu Unicast- und Multicast-Ansätzen sind die aktuellen Broadcastprotokolle für WMNs spärlich und zeigen einen Mangel an Funktionalität. Flooding und weitere Protokolle mit fester Rate wie B.A.T.M.A.N. und OLSR liefern keine zuverlässige Übertragung bei gleichzeitiger Minimierung des Ressourcenverbrauchs, wie sie von aktuellen und zukünftigen Anwendungen gefordert wird. Diese Arbeit schlägt daher das Broadcastprotokoll RAIDER vor, das Fähigkeiten zur Übertragung mit mehreren Raten beinhaltet und Qualität der Verbindungen zwischen den Geräten beachtet. Um die Leistung zu bewerten, wird ein neuer Leistungsindikator eingeführt, der sowohl die erreichte Zuverlässigkeit als auch die kumulierte Sendezeit auf dem drahtlosen Medium kombiniert. Dadurch war eine Leistungsbeurteilung von RAIDER gegenüber dem Stand der Technik möglich. Ergänzend zum Vergleich wurde eine obere Leistungsgrenze für das betrachtete Broadcast-Szenario festgelegt. Die Ergebnisse zeigen, dass RAIDER eine deutliche Verbesserung gegenüber bestehenden Übertragungsprotokollen sowohl in Bezug auf die Zuverlässigkeit als auch auf die Sendezeit erreicht hat. RAIDER hat fünfmal so gut abgeschnitten wie Flooding. Im Vergleich zu B.A.T.M.A.N. konnte die Airtime um bis zu $95\%$ reduziert werden, während ein leichter Vorteil in der Zuverlässigkeit erhalten blieb. Die Leistung von OLSR könnte um $56\%$ erhöht werden, einschließlich einer Erhöhung der Zuverlässigkeit um bis zu $20\%$. Obwohl sich die Performance deutlich an die obere Leistungsgrenze annäherte, konnte sie letztendlich nicht erreicht werden, so dass sich das Potenzial für weitere Verbesserungen in der Zukunft ergibt.

# Contents

# List of Figures

# List of Tables

# Acronyms

**ARP** Address Resolution Protocol

**B.A.T.M.A.N.** Better Approach to Mobile Ad-hoc Networking

**CDS** Connected Dominating Set

**CSMA/CA** Carrier Sense Multiple Access/Collision Avoidance

**DHCP** Dynamic Host Configuration Protocol

**DFS** Depth-First Search

**ETT** Expected Transmission Time

**ETX** Expected Transmission Count

**IEEE** Institute of Electrical and Electronics Engineers

**IETF** Internet Engineering Task Force

**ITU** International Telecommunication Union

**LTE** Long Term Evolution

**MANET** Mobile Ad-hoc Network

**MCS** Modulation and Coding Scheme

**MPR** Multipoint Relay

**NHDP** Neighborhood Discovery Protocol

**OFDM** Orthogonal Frequency-Division Multiplexing

**OGM** Originator Message

**OLSR** Optimized Link State Routing

**PER** Packet Error Rate

**QoS** Quality of Service

**RAIDER** Rate Aware Information Dissemination with Extra Reliability

**RLNC** Random Linear Network Coding

**SNR** Signal-to-Noise Ratio

**UDP** User Datagram Protocol

**WLAN** Wireless Local Area Network

**WMN** Wireless Mesh Network

**WSN** Wireless Sensor Network

# Symbols

| Symbol | Description |
|---|---|
| $G(V, E)$ | Network graph notation consisting a set of edges $E$ and a set of vertices $V$ |
| $R$ | Set of available data rates |
| $r$ | Currently selected data rate |
| $p_{i,j}^{(r)}$ | Link success rate for transmissions with data rate $r$ |
| $\epsilon_{i,j}^{(r)}$ | Link error rate for transmissions with data rate $r$ |
| $p_{tr}$ | Link reliability threshold for RAIDER |
| $l$ | Side length of the square-shaped network area |
| $n$ | Number of nodes in the network |

# 1 Introduction

The introduction of the next generation of mobile networks in 5G is a reaction to the increased future traffic demand [15]. Since existing network structures struggle to keep up with these demands, this development comes with a shift in focus, as device to device communication significantly gains in importance [67].
The current hierarchical cellular network design relies on the performance of a single, central access point or base station. WMNs are an opposing concept to offer flexibility and the support of highly dynamic environments in combination with device to device integration. Mesh networks have already proven their value in numerous applications, such as community networks [22], industrial [73] or automotive applications [77]. The versatility allows for the deployment on top of a variety of wireless technologies in for instance the IEEE 802.1x family [4], Bluetooth [78] or Powerline Communications (PLC) [68].
However, the renunciation of a central network entity raises coordination challenges and therefore demands the development of a distributed solution. One of those challenges is traffic routing, which experiences a significant increase in complexity, due to the error-prone wireless links. So far, routing in wireless networks was trivial, as all traffic was handled and forwarded by the coordinating entity. As WMNs implement direct device to device communication, they can exploit of the wireless multicast advantage [72]. It offers the possibility to constructively overhear transmissions and include this into a more diverse routing. This has been successfully incorporated into unicast routing protocols, such as the opportunistic protocols ROMER [79] and MORE [12], as well as multicast schemes [34][71].
In contrast to that, broadcast has not received as much research attention [9]. Existing approaches include OLSR [17] and B.A.T.M.A.N. [49], which have also been successfully instated in multiple applications [54]. However, they can not take advantage of multi-rate capabilities of for instance IEEE 802.11 Wireless Local Area Network (WLAN) environments, that they are deployed in.

## 1.1 Motivation

The lack of multi-rate broadcast schemes forces the usage of very reliable, yet slow data rates in transmission. A plethora of protocols relies on classical flooding to deliver broadcast transmissions throughout the network [9]. Although the surrounding environmental conditions would allow for the exploitation of available advancements in increased transmission rates, this is so far mostly considered for unicast only. The

latest multi-rate advancements [66][70][14] failed to properly adapt to the properties of WMNs and to deliver a general performance improvement. Complicated prerequisites and dependencies, such as central coordination or an up-to-date overall network knowledge, render approaches hardly applicable to actual deployments, which leads to the usage of flooding as preferred fallback.
Broadcast in particular is required by numerous lower-layer services, Address Resolution Protocol (ARP) and Dynamic Host Configuration Protocol (DHCP) for instance, to ensure general network operability. Furthermore, broadcast is used in industrial applications, such as industrial ethernet for factory and process automation, as well as for media applications, such as live streaming. Therefore it must receive attention for the development of a general broadcast scheme, that is available in all situations and includes transmission rate awareness to increase throughput and latency, while also decreasing broadcast overhead.

## 1.2 Contribution

This work contributes to the solution of the broadcast routing problem in WMNs, in that it firstly outlines the current requirements. These are compiled through an analysis of the deployed state of the art protocols, namely B.A.T.M.A.N., flooding and OLSR, paired with a discussion of additional related work approaches to multi-rate routing in mesh networks. It points out the innovations as well as the weaknesses in design choices, to develop and propose a novel scheme on that basis. Rate Aware Information Dissemination with Extra Reliability (RAIDER) combines and extends previous work for an advancement in general broadcast. With RAIDER,the first general broadcast solution in WMN, incorporating link quality awareness and the dynamic selection amongst multiple transmission rates, is presented.
Its performance is subsequently assessed in a custom WMN broadcast simulation environment for comparison against existing protocols, as well as an additionally established upper performance bound. For a suitable comparison taking the reliability as well as the airtime into account, an overall performance determination is established.
The results show an average improvement over B.A.T.M.A.N. of $80\%$ in consumed airtime, while at the same time also slightly raising the reliability by one percent. By investing between $20\%$ more airtime compared to OLSR for good surrounding network conditions and $105\%$ for bad conditions, RAIDER can improve the reliability up to $21\%$. Both airtime and reliability metric were also combined, to assess an overall broadcast performance improvement of up to $885\%$ compared to B.A.T.M.A.N. and $56\%$ for OLSR.

## 1.3 Outline

The second chapter includes an introduction to the theoretical background of WMNs in general, as well as the fundamentals on broadcast routing approaches. As such, the currently most important schemes are presented and discussed, supplemented by further related work approaches. The last sections will provide an overview of routing metrics and introduce the upper performance bound calculations.
The third chapter provides a description of the simulator used as evaluation environment for the assessment of the broadcast performance of each protocol. The network model and the simulation procedure are presented in detail. In the last sections, the parameters to create the different simulated scenarios and the optimal performance bound are explained.
In chapter 4 the candidate protocols presented and discussed in chapter 2 are implemented in the simulation environment and subsequently evaluated. The performance results are presented and the protocols are analyzed for any benefits and disadvantages, as well as possible conclusions for the ensuing protocol design.
With the previously gained knowledge, this thesis presents a new approach on broadcasting in WMNs in chapter 5. Therefore the criteria for the protocol design, as well as the requirements deduced in the preceding chapters are explained. Afterwards the key challenges are formulated and a custom solution considering all criteria is developed.
The performance analysis of the proposed protocol in comparison to the existing state of the art in broadcasting in WMNs is presented in chapter 6. The protocol is tested against a number of circumstances, which are described and analyzed.
The last chapter contains a summary of the results with an evaluation and review considering the defined goals in chapter 3. Finally, it provides an outlook on open research directions to improve the work of this thesis in the future.

# 2 Background And Related Work

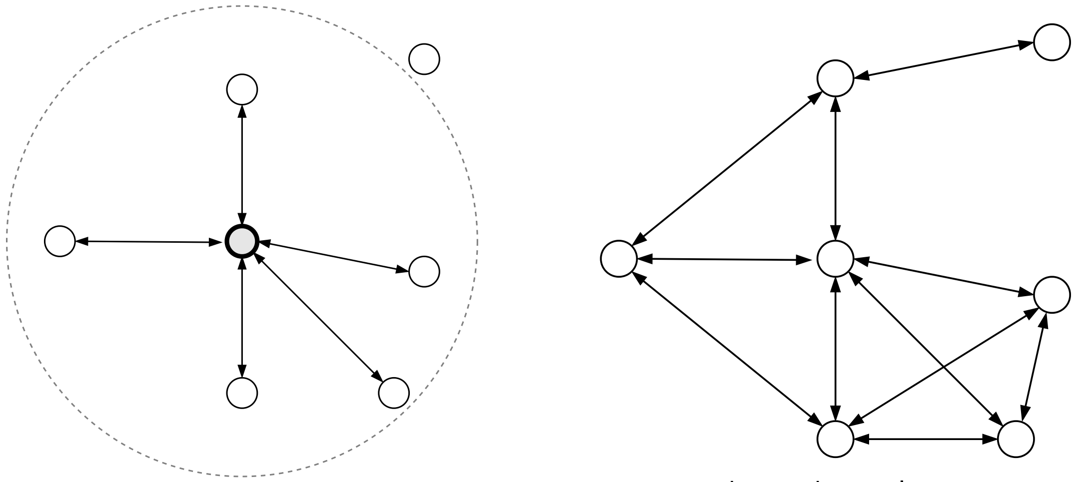
## 2.1 Network Model

To describe the network mathematically, it is modelled as a graph $G = (V, E)$, where $V$ is a set of *vertices*, also referred to as *nodes*, representing network devices, such as routers, phones or computers. The nodes are connected through a set of *edges* $E$, whereas each edge $e$ is a tuple defined by its two end-point nodes $i, j \in V$ and constitutes a directional connection between these nodes. Furthermore, each link carries a weight $\epsilon_{i,j} \in [0, 1]$, expressing an error or failure probability for a single transmission attempt via this link. For the continuous transmission of a complete packet, $\epsilon_{i,j}^{(r)}$ corresponds to the Packet Error Rate (PER) for a data rate $r \in R$. The probability of a successful transmission of a packet $p_{i,j}$ between two nodes $i$ and $j$ via a link $e = (i, j)$ can thereby be described as $p_{i,j} = 1 - \epsilon_{i,j}$. All error probabilities are defined separately for each direction. Provided that both directional values are equal, meaning that $\epsilon_{i,j} = \epsilon_{j,i}$, the link is considered symmetrical.
For this work, the utilized network model includes exclusively bi-directional and symmetric links.

## 2.2 Wireless Mesh Networks & Mobile Ad Hoc Networks

Wireless Mesh Networks (WMNs) are one possible concept used to satisfy the demand of device to device communication in the upcoming 5th generation of mobile networks. Before their introduction, wireless networks such as WLANs according to the IEEE 802.11 standard family or Long Term Evolution (LTE) mobile networks were mostly organized in the same structure, namely in a star topology [29].
As a consequence, devices in a wireless network had different roles. Each device was involved only with its own transmissions and would disregard all other overheard traffic. The only exception was the central node, with was specialized to organize and coordinate all traffic and hence would be receiving, transmitting and relaying the traffic in the network. Additionally it would act as a gateway to provide the local wireless network with a connection to further networks. With the growing traffic volume and the at the same time increasing traffic demands, the star topology shows a major disadvantage. The overall network performance is limited to the central node, as it has to manage the traffic for each of the participating devices in the network. Thus the network size was also limited by the reach of a single device and could not be

extended due to transmit power regulations. Nodes outside of this coverage could not connect to the network.



(a) Star topology

(b) Mesh topology

Figure 2.1: Comparison of network topologies

These problems can be solved with the concept of connecting the network nodes in a meshed topology. Every node is allowed to establish a direct connection towards all reachable nodes in its vicinity. The resulting structure, resembling the looks of a mesh, offers significantly more transmission paths compared to the same network in a star topology configuration.
Figure 2.1 shows the two topologies in direct comparison, applied to an identical set of nodes.
It also allows the extension of the network coverage with every additional participating node. In WMNs there is no designated central node for coordination. Every node can have different properties and abilities, thus the WMN can include *mesh clients* as well as *mesh routers* or *mesh gateways* [5]. Moreover, this enables the WMN to be easily integrated into existing networks, such as Wireless Sensor Networks (WSNs) or WiMAX [4]. A WMN in general must not include every type of node, as a *hybrid WMN* does, but can also consist purely of clients to form a *client WMN* or solely of routers or gateways to form a *backbone/infrastructure WMN* [4]. This distinguishes WMNs from Mobile Ad-hoc Networks (MANETs), as they do typically only contain clients, that are connected ad-hoc. Regarding intra-network traffic, there is however no difference between both types [46].

The open nature of WMNs elevates the topology compared to cellular networks. Mesh networks have the ability of self-organization, self-discovering, self-healing as well as self-configuration [46]. For that reason, they are low-cost, easily maintainable, while offering an increased robustness and a reliable coverage [32]. Another advantage of a WMN is the ability to support moving nodes and therefore enabling

the deployment in a dynamic environment. On the contrary, the complete network coordination has to be distributed, leading to numerous open research issues in that field.
These properties nevertheless enable WMNs to be a suitable candidate for a wide range of applications. These include home networking, enterprise networking, for instance at Microsoft or Google [52], transportation systems, metropolitan area or community networks [22] such as *Roofnet* [13] or *Freifunk* [54] and medical and security systems [46][11].
WMNs were also covered by Institute of Electrical and Electronics Engineers (IEEE) standardization work. Specialized task groups were established to create new or adapt existing WLAN technologies to the requirements of mesh networks [11]. The early work, starting at 1999, focused on IEEE 802.16a to create a "first-mile/last-mile" extension for flexible point to point connections at the edge of fixed wireless networks. Later work on IEEE 802.15.5 and 802.20 attempted to define the physical layer and medium access details to connect small groups of fixed or portable consumer devices, such as laptops or phones, in a short range vicinity. The latest approach of IEEE 802.11s created a standard to extend the existing WLAN standard towards a high-speed capable adaptation specifically for WMNs. It concentrated on the shift in requirements from the more dynamic nature of the topology and improved multicast and broadcast support. In the end, the IEEE advancements could not reliably establish themselves and and are therefore not relevant for this work.

### 2.2.1 Broadcast in Wireless Mesh Networks

As it is the case with each type of network, the general operability in WMNs depends on lower-layer network protocols, i.e. protocols at the physical, data link and network layer of the OSI network management model. Representatives, such as ARP [55] and DHCP [23] require to reach every device in the network, to be able to provide all network services to them. To fulfill this requirement, both protocols make use of a broadcast transmission throughout the whole network. If this broadcast fails repeatedly to reach all nodes, unserved nodes will be effectively excluded from the network, rendering them unusable. This should be avoided under as much as possible. Hence the maximum available reliability is required by these type of network protocols. The ARP and DHCP traffic occurs infrequently, as usually only new connections to the network or timed updates trigger protocol activity. In the case of ARP, the frame size of $28$ bytes is relatively small, compared to the several hundred bytes for a DHCP message.
A further application of broadcast specifically in WMNs is the replacement of *industrial ethernet* with a wireless alternative. Current technologies, such as *Modbus* [47], *Profinet* [57] or the open-source *Powerlink* [25] aim at delivering real-time communication for factory automation, process and motion control, in the scope of the latest 5G use cases. They therefore rely on a high reliability, while also demanding a very low delivery time [28]. Dependent on the actual use case, the required

| Broadcast application | Packet size | Traffic volume | Traffic requirements |
|---|---|---|---|
| Address Resolution Protocol | $28$ bytes | low, infrequent | high reliability |
| Dynamic Host Configuration Protocol | several $100$ bytes | low, infrequent | high reliability |
| Industrial ethernet technologies | $40...250$ bytes | medium, frequent | high reliability, low latency |
| Video streaming | $> 1000$ bytes | high, continuous | large amount of data support |

Table 2.1: Overview of different broadcast applications

latency ranges between several hundred to several tenth of a millisecond [30], while payload sizes can vary between $40$ bytes for the smallest and up to $250$ bytes for the largest frames [28]. In current industrial ethernet deployments however, the resulting minimum frame size is limited to $64$ bytes. Industrial traffic is furthermore transmitted frequently, however not in direct succession [30]. Therefore it will not generate a data stream, but instead a series of recurrent transmissions.
Finally, broadcast will be included in future developments regarding end-user oriented traffic, for instance live video streaming. With a broad range of available content, media delivery services will increase their traffic volume [15]. The resulting traffic will mainly include large amounts of data, that can tolerate packet losses and delays, since this type of traffic is continuous and therefore applicable for coding schemes, that can repair damages in separate transmissions.
Table 2.1 summarizes the different broadcast applications with their traffic characteristics.

In the scope of this work, the focus is placed on network protocols and industrial traffic applications, as they share common or similar requirements and constitute a basic traffic, in that it is comparably sparse and small.

## 2.3 State of the Art Routing

The flexible and open structure of WMNs compared to cellular or wired networks does not allow for a central entity for orchestration. Consequently, there is a plethora of problems, that need a custom solution adapted to the WMN environment.
An important problem is the routing inside of the network without central coordination. Routing is determining a transmission path from the source to the destination and thereby selecting which nodes in particular have to forward the received data.

For this purpose, topology information about the surrounding network has to be gathered and processed. For the unicast case, i.e. finding a path from one source to one receiver, research has come up with numerous propositions. This includes traditional routing approaches [40] as well as the modern opportunistic approach, which exploits the broadcast nature of transmissions even in unicast routing [79][12]. Extending unicast to multiple destinations, multicast has also a broad range of developed protocols [34], covering the opportunistic approach as well [71].
Although the majority of routing protocols depend on a working broadcast mechanism for network operability or route discovery, broadcast transmissions have experienced a significantly reduced interest in research in comparison to unicast and multicast [9]. This generally constitutes a contrast between the usage, described in section 2.2.1 and the offer of broadcast routing solutions. The majority of routing relies for this task on basic flooding [9], with exceptions such as OLSR, implementing an individual scheme. Additional dedicated broadcast routing approaches are presented in detail in further subsections.

### 2.3.1 Routing Protocol Classification

To classify routing schemes, they can be divided into three main strategies regarding their operation principles [39] [1]. *Proactive* or "table-driven" routing protocols actively select a route in advance without the direct and imminent necessity caused by incoming traffic. As soon as a change in the network topology is detected by a node, the routing algorithm calculates a forwarding decision for this node, to reach any other node in the network. On the one hand, this enables a fast reaction to incoming traffic, as a routing decision has already been made. On the other hand it increases the computational cost, as every single change in the network topology, i.e. an added or removed node or a change in link quality, triggers a partial or complete recalculation. For highly dynamic networks this might constitute a problem, not only for an intricate or inefficient algorithm.
Conceptually opposite to the proactive protocols are the *reactive* or "on-demand" protocols. Instead of always precalculating the routing decisions, they are generated on a source-initiated demand. As soon as a packet from a new source has to be relayed, the protocol gathers the necessary information to decide, how the packet should be routed next. Therefore an inquiry is disseminated to locate the destination of the transmission [27]. This eventually benefits the usage in dynamic environments, as expensive calculations happen only at a minimum on demand. Compared to proactive routing, source nodes might suffer from a longer delay for the route setup, whereas they exhibit a superior scalability, since not every change in the network topology triggers a recalculation [37].
The third group of protocols combines elements of pro- and reactive strategies. *Hybrid* routing operates in two different scopes. For the direct neighborhood, the routing decisions can be proactively calculated, while the longer paths towards

nodes further away require an on demand and up-to-date forwarding decision. This accounts for the different probabilities of forwarding traffic from the two network regions.

Apart from calculating the routing decision itself, the protocols also need to maintain an information base, on which each decision is based. Each node in the network broadcasts routing information, regarding its detected neighborhood, periodically, to let other nodes update their information base. The two fundamental strategies in managing this information base are realized in *distance-vector* and *link-state* algorithms. The approach taken by link-state protocols is to construct a complete representation of the whole network, based on periodical broadcasts through the network. These contain information about the direct link quality of each node toward its neighbors [2]. The goal is to have sufficient information at any node, so that paths can be calculated locally from source to destination as a whole. This requires expensive maintaining in terms of up-to-date information at every node in the network, which is under realistic circumstances not achievable, since it requires nearly a constant information exchange, with no delays. Nevertheless, it leads to a significant increase in control overhead and thus a performance decrease.
To avoid this dependency, distance-vector routing reduces the network knowledge, that needs to be gathered. All nodes share only a table with their neighbors, which contains their distance to each respective node in the network. Combining this information from the neighborhood, the cheapest next forwarding hop can be locally determined, without knowledge of the complete path to the desired destination [43]. Therefore distance-vector protocols exhibit a better scalability. The local information exchange delivers the information fast and requires only a minimum in transmissions. But on the other hand it is the main disadvantage compared to the link-state approach, as the lack of knowledge can lead to undetected routing loops.

### 2.3.2 Flooding

The most intuitive and simplest approach to disseminate information to all nodes in the network is to use a classical flooding mechanism. It is the minimum effort strategy in complexity and requirements and therefore suitable for all circumstances, as it requires only a duplicate detection mechanism. When a node receives a message, that has not been seen before, i.e. it is innovative, it rebroadcasts the information to all direct neighbors, as depicted in figure 2.2. For this purpose it uses the fixed broadcast rate, which is commonly the lowest available data rate. This guarantees robustness against lossy wireless links through redundant transmissions as well as the maximum probability of reaching every node in the direct neighborhood. However, since there is no further control mechanism in place, it potentially also leads to contention and eventually collisions, as described in the *broadcast storm problem*

[50]. The subsequent cost in time to recover from the problem, makes it mandatory to set in place a strategy to prevent a broadcast storm, since otherwise the whole network operation can be compromised. These strategies include probabilistic schemes [60][61][80], counter-based schemes to assess and act according to the expected additional coverage [48], less used distance-based schemes to decide upon the distance to the previous sender [33], location-based schemes and cluster-based schemes using a hierarchical approach or pruning to restrict a subset of nodes from retransmitting [42].

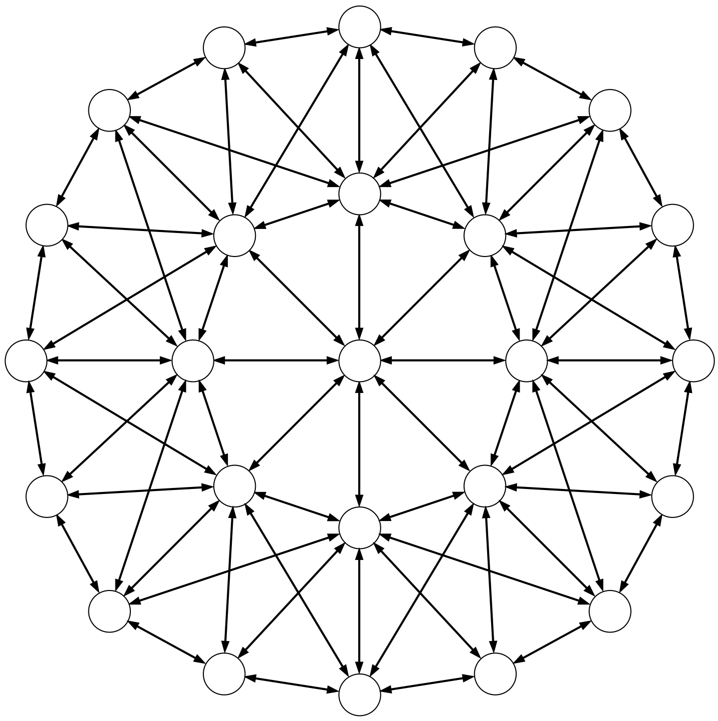


Figure 2.2: Individual transmissions for a flooding broadcast initiated by the central node

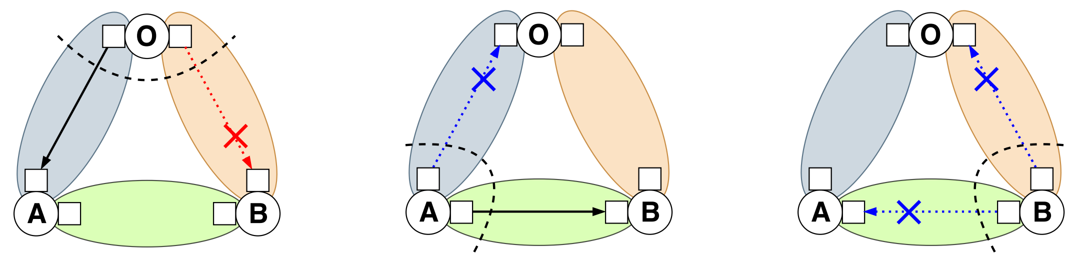
### 2.3.3 B.A.T.M.A.N.

B.A.T.M.A.N. is a proactive routing protocol implementing a simple route finding algorithm for mobile ad-hoc networks. The B.A.T.M.A.N. protocol was first specified in an IETF Internet-Draft [49] in 2008 and evolved into the commonly used version of B.A.T.M.A.N. advanced [63]. It is a distance-vector protocol maintaining a small information base to proactively calculate routing decisions. It uses a custom type of packets to determine single-hop neighbors and the quality of these link-local connections, as well as to generate a list of participating nodes in the network. Each node periodically announces itself via Originator Messages (OGMs) to all nodes connected to the transmitting interface, so that they are detected by the nodes in the direct neighborhood. The nodes themselves store this neighborhood information in an originator list. Since each neighbor shares its own OGMs, all nodes can establish a channel quality estimation from the average of received messages. To account for both directions separately, B.A.T.M.A.N. calculates its link metric, the directional transmission quality towards the neighbor node $TQ$ from the measured receive quality $RQ$ and the measured echo quality $EQ$, counting the overheard rebroadcasts of own OGMs. The receive quality is calculated from the expected number of OGM transmissions by a neighboring node in relation to the actually received number. Upon reception of an external OGM, a node will rebroadcast it, only if it was received through the neighbor with the best metric for the original source of the OGM. As a consequence, the information travels along the best available path, which is then registered and stored by all receiving nodes. In the case of all direct neighbors, their connection and therefore the metric is already the best, hence all neighbors will rebroadcast the OGMs. This enables the originally transmitting source to overhear all rebroadcasts of its own OGM, to obtain an echo quality $EQ$. After dividing $EQ$ by $RQ$, B.A.T.M.A.N. applies penalties for the number of hops and asymmetric links to retrieve the final link metric [7], resulting in a value between $0$ and $255$. To obtain the transmission quality, B.A.T.M.A.N. uses the previous $64$ OGMs, to offer a sufficient link quality resolution.

Besides a unique channel quality metric, B.A.T.M.A.N. also employs a dedicated broadcast mechanism. It is conceptually very close to the classical flooding presented in subsection 2.3.2. To compensate for missing retransmissions, as known from unicast connections, every packet is broadcasted three times in succession. For interference avoidance, a delay of $5$ ms is applied between the separate transmissions. The broadcast is executed at the fixed broadcast transmission rate, determined by the lower-layer specifications. It is not controlled by the protocol itself. To avoid unnecessary transmissions, B.A.T.M.A.N. also implements a redundancy avoidance strategy, depicted in figure 2.3. Each interface will cancel the transmission if it is only connected to the originator of the message, as demonstrated in figure 2.3b or if it is only connected to the previous sender as shown in figure 2.3c. These constitutes the only cases, where a node without any further prior knowledge can exclude, that the interface is connected to any neighbor, that still might require the information. Therefore the transmission would be certainly redundant and can be omitted.

With its broadcast scheme, B.A.T.M.A.N. mimics the flooding approach with an extension to add further redundancy. This generally increases the number of transmissions in the network, but offers a more reliable delivery. Actual performance measurements have shown, that B.A.T.M.A.N. effectively achieves a high level of stability and packet delivery when deployed in a WMN [1].

For this work, B.A.T.M.A.N. is selected as a candidate protocol, since it emphasizes a specific broadcast strategy, which has successfully been deployed and at the same time shown promising results.




(a) Erroneous transmission from O to B

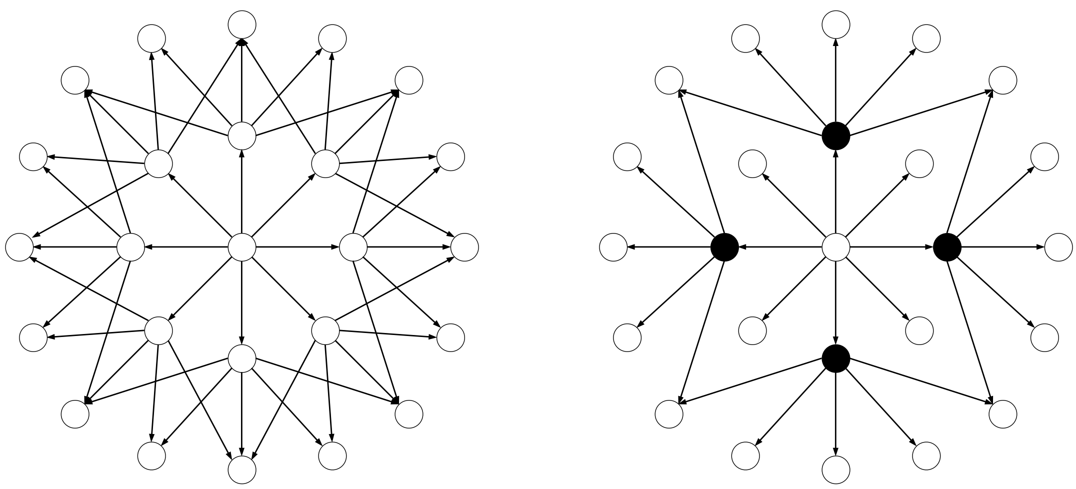
(b) Originator avoidance between A and O

(c) Previous sender avoidance between B and O

Figure 2.3: Demonstration of redundant broadcast avoidance in B.A.T.M.A.N.

### 2.3.4 OLSR

The Optimized Link State Routing (OLSR) protocol is an optimization of the classical link state routing protocol for MANETs, first described in [18]. Link state routing is a network routing approach, that aims at distributing a complete routing database throughout the network [2]. Each node collects information regarding all its connections to any other neighbor with the help of *HELLO* messages. These link quality reports are periodically shared via a local broadcast among all nodes in the direct neighborhood, to subsequently enable the creation of a local information base of the current 2-hop neighborhood at every single node. Thus, the topology information required for the link state information base is collected in a distributed manner. To gather and merge this local data into the complete data base, it is shared throughout the network using *TC* messages. In this way, every node can regularly update the required link state routing information.



(a) Information flow using flooding

(b) Information flow using MPRs in OLSR

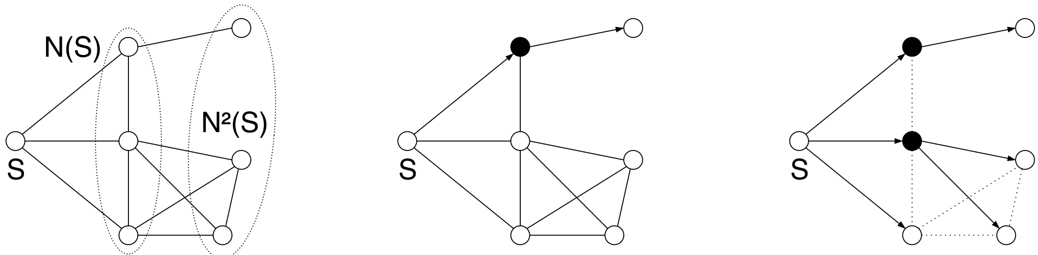
Figure 2.4: Comparison of classical flooding and MPR flooding in OLSR as in [51]

To improve the performance by relieving the network from the frequent broadcast load induced by the *TC* messages, OLSR introduced *multipoint relays*. This concept in the style of a dominating set scheme assigns only a subset of nodes the task to forward flooded messages, in order to reduce the number of transmissions compared to the traditional flooding mechanism. However it does not optimize for the amount of forwarding nodes. Figure 2.4 shows the dissemination of information through the network, comparing flooding and OLSR with MPRs colored in black. Note that arrows do not mark transmissions but only indicate the direction of the information flow. Each node $x$ calculates its own set of relaying nodes $MPR(x)$ from the set of symmetric one-hop neighbors $N(x)$, according to the following procedure, describing the MPR selection in detail [59][18]. The strict 2-hop neighborhood $N^2(x)$ includes direct neighbors of all members of $N(x)$, except members of $N(x)$ themselves, as well as $x$. In the following algorithm, $N^2(x)$ is initialized as described and from there on represents uncovered 2-hop neighbors of $x$. A 2-hop neighbor node $z$ is considered uncovered, if it has no neighbor $y \in N(x)$, that is an element of $MPR(x)$. $d(y)$ denotes the degree of a node $y \in N(x)$, defined by the number of symmetric neighbors $z$ of that node, that are also strict 2-hop neighbors of $x$.

1. Start with an empty set $MPR(x)$.
2. Calculate $d(y)$ for all $y \in N(x)$.
3. Select the nodes in $N^2(x)$, that are only connected to a single neighbor node $y$ in $N(x)$ and add these one-hop neighbors $y$ of $x$ to $MPR(x)$.
4. While there still exist nodes $z$ in $N^2(x)$, that are not connected to a node $y$ from $MPR(x)$, proceed as follows:
   a) Calculate the reachability for each node $y$ in $N(x)$, i.e. the number of uncovered 2-hop neighbors in $N^2(x)$, that are connected to $y$.
   b) Select the node $y$ for which the reachability is maximum and add this node to $MPR(x)$. Break a tie between nodes with the node with the maximum $d(y)$.
   c) Remove from $N^2(x)$ all by the previously selected MPR $y$ covered nodes in $N^2(x)$.

The algorithm applied to a demo network is shown in figure 2.5, with $S$ being the respective source node. The second step illustrates the evident emphasis on isolated nodes in the 2-hop neighborhood, that are connected only through one direct neighbor of $S$. This marks an important feature for guaranteeing a reliable coverage of all 2-hop neighbor nodes.

The specified OLSR additionally introduces a metric *WILLINGNESS* to each node, that represents the nodes self decided willingness to participate in transmitting as a relay. This reflects individual constraints on power, processing capabilities and expected lifetime in the network and allows a node to refuse the service as an MPR, if it is




(a) Step 1:
Start with an empty MPR set

(b) Step 2:
Cover isolated 2-hop neighbors first by selecting the first MPR

(c) Step 3:
Select additional MPRs to cover all 2-hop neighbors

Figure 2.5: The original MPR selection algorithm in OLSR applied to a demonstration network

presumably not capable or if it can not fulfill this role reliably. The *WILLINGNESS* parameter is omitted for this work, since it can not be qualitatively selected without the underlying physical demands. However, it would allow a final algorithm step of removing expendable members of $MPR(x)$, sorted by their increasing *WILLINGNESS*. If all neighbors of a node $y$ are also covered by another node from the set of MPRs, it can be removed from $MPR(x)$.

The OLSR Version 2 protocol [17] is the most recent evolution of OLSR. It includes the Neighborhood Discovery Protocol (NHDP) [16] for managing the *HELLO* messages. It also adds the support for an additive link metric, other than the hop count still used by default. However, it does not specify or propose, which metric should be used. A further addition to the second version of OLSR is the option to include a second set of MPRs for unicast routing. As this is a separate functionality, it has no effect on the previous usage of MPRs for the dissemination of *TC* messages. This MPR mechanism is identical to the first OLSR version. Hence, in the scope of this work, both versions OLSR and OLSRv2 are not considered different, since both employ the same broadcast scheme.

The original OLSR specification provides only an algorithm proposal to populate the set of MPRs. At the same time it allows for other approaches, that pursue different optimizations, than the specification. These schemes can be grouped into three categories [10][41]. Pure MPR schemes are oriented on the original heuristic, whereas MPR-based Connected Dominating Set (CDS) schemes aim at the maximum reduction of nodes in the relay set or optimizing the individual power usage efficiency. Lastly, Quality of Service (QoS) based schemes make use of QoS metrics to meet requirements such as a certain bandwidth or a minimum delay, to guarantee a fixed service level for audio or video applications.

For this work, OLSR is selected as a candidate protocol, since it implements a novel strategy specifically for the broadcast, which is currently deployed in numerous applications. Furthermore the included and originally specified MPR selection algorithm will be considered, as it is currently implemented. It will be complemented by a second algorithm, with offers a double coverage, i.e. if possible two separate MPRs covering each 2-hop neighbor.
OLSR offers a good performance in high density networks with highly sporadic traffic and is therefore suitable for the broadcast applications in WMNs, presented in section 2.2.1. In contrast to that, in more dynamic environments, the OLSR struggled to provide a stable performance [45][36], which demands for improvements.

### 2.3.5 Further Related Work

In [53], Peng *et al.* pursue a similar approach to OLSR. Their protocol is designed for broadcast transmissions and utilizes fixed-rate transmissions on a shared common wireless channel. It operates from a base of only a duplicate table, as well as a 2-hop neighbor table. To distribute the information in the network, they make use of dedicated broadcast relay gateways (BRGs), that in contrast to OLSR, constitute a CDS in the network graph. Therefore it optimizes directly for the least amount of transmitting nodes possible.
All presented broadcast solutions for WMNs so far concentrated solely on determining, which nodes participate in the broadcast. However, they do not make use of the wireless environment that they are deployed in, regarding the capability to adapt the data rate for the transmission according to the measured channel and link qualities. This significantly reduces the efficiency in terms of the time, that the wireless medium is occupied, as well as the overall latency of the broadcast. Thus, the remainder of this section presents several approaches for broadcasting in WMNs, considering the multi-rate capabilities of the wireless network.

In [66] Subramanian *et al.* propose *UFlood*, a flooding protocol for wireless mesh networks, targeting a high throughput and low delays. This is mainly achieved by a careful selection of the transmitting nodes, as well as a dynamic usage of different data rates. The distributed approach reacts to the immediate environment and the current state of the direct neighbors, including the link properties in available bit rates and error probability. To furthermore improve the efficiency of the distribution of the target data, randomized network coding is used. This ensures that no data is transmitted, where none is needed. To achieve this, UFlood implements a feedback, reporting and acknowledging the previously received data. Routing decisions are eventually made from the data, all neighbors currently possess and the topology overview including the neighborhood, available links and respective rates. For this

purpose, UFlood uses the Expected Transmission Time (ETT) metric, which is further discussed in section 2.4.
To summarize, UFlood can increase the performance for applications such as software updates where all nodes need to receive the same large data files. It can achieve this by optimizing forwarding decisions with the knowledge of the complete network topology, as well as valuable information gained by dedicated feedback transmissions. The scheme incorporates the individual link loss rates and utilizes multiple data rates to improve the broadcast throughput and delay.

Another latency focused optimization was introduced in [14] by Chou *et al.*. Based on the ability to dynamically adjust the transmission rate in recent IEEE 802.11 standards, Chou *et al.* developed a heuristic algorithm to solve the minimum latency broadcast problem in multi-rate WMNs. This algorithm is divided into three parts. At first, a weighted connected dominating set is constructed to obtain a spanning tree, that covers all nodes in the network, while at the same time taking the multi-rate ability into consideration. In the following step, the concrete number of transmissions on any data rate for each node needs to be determined. A special focus is placed on guaranteeing a prioritized delivery to "critical" nodes, that forward to a larger set of nodes. The last step schedules all transmissions determined in the previous step, so that as much transmissions as possible can happen simultaneously, without creating interference. For this purpose, they assume a central unit with complete network knowledge.
Chou *et al.* accomplish a significantly reduced broadcast delivery latency for the whole network. The achieved gains are enabled by the usage of efficient transmission rates, while accounting for the relevance of each node in the overall forwarding process. Eventually, this is made possible by a centralized scheduling with a complete topology overview as well as the knowledge of all transmissions to be executed.

The Distributed Rate First (DRF) algorithm [70] presented by Wang *et al.* applied the minimal connected dominating set strategy to the multi-rate wireless mesh to reduce the broadcast latency. At first mathematically modelling and describing the minimal latency broadcast problem for multiple available transmission rates, they offer a solution strategy based on their network model. It includes variable transmission rates, that successfully transmit up to a certain distance from the source node, modelling the error rates. The assumed network is connected, meaning that each node will be reached by at least one other node at any rate. To minimize the latency, every node is allowed at maximum one transmission, while maintaining a connected dominating set, when combining all transmitting nodes. To solve this problem, the last constraint is dropped to create sub-trees, that match the remaining constraints. If these optimal sub-trees are not connected, the algorithm uses bridges to establish a connection using the available link with the least cost, regarding the transmission time. In summary, the Distributed Rate First algorithm achieves a latency optimal broadcast for sub-trees by the usage of optimal data rates. Since they do not necessarily cover the complete network, they have to be connected with central coordination and complete knowledge of all available bridge links in between the sub-trees.

### 2.3.6 Summary

To conclude the state of the art examination in broadcasting in WMNs, it has been firstly shown, that broadcast is a mandatory requirement for the operability of the network itself, as well as for instance routing as a further necessity for successful data transportation. Most of current routing protocols still rely on the most basic form of the classic flooding scheme to handle the task. This includes both unicast and multicast routing, as well as broadcast routing. Therefore it can be stated, that flooding undeniably still makes up for a large share of all deployed broadcast schemes.
Nevertheless there have also been developments, such as B.A.T.M.A.N. or OLSR with the MPR scheme, that introduced individual approaches, to improve the broadcast.
However the majority of currently implemented broadcast schemes use a fixed rate for the broadcast transmissions, ignoring the multi-rate capabilities of modern wireless environments. This significantly hurts the performance in terms of latency or airtime. On the other hand, the usage of low fixed broadcast rates secures a reliable delivery in all possible circumstances. Approaches implementing a multi-rate solution show promising gains, but rely on advanced services and mechanics, such as a complete and accurate network knowledge at all times or a central entity, that coordinates network activities.

## 2.4 Routing Metrics

To evaluate the transmission quality of a link and to compare different links to make the best possible routing decision, every routing protocol has to make use of a metric to convert the link sensing results into a single performance number. For this purpose, most of the currently used routing protocols in wireless networks use the *hop count* as dominant metric for the route determination, without taking the link quality into account [11]. The reason for the wide-spread usage of this metric is its simplicity in processing and retrieval. However, it is assuming uniform links and this does not reflect the actual channel conditions in terms of the loss rate or the bandwidth, potentially leading to an unexpected performance decrease. In the scope of this work, the hop count is for instance in OLSR MPR flooding of *TC* messages the only metric determining, which nodes have to be served and which will forward.

To account for the link condition, there are numerous metrics, that focus on properties, measured through link sensing. The Expected Transmission Count (ETX) metric [20], in case of a symmetric link and no expected feedback transmission, is directly derived from the success probability $p_{i,j}$ measured on the link, according to equation 2.1. It estimates the number of transmissions, that on average compensate for the errors on the link to achieve a reliable successful transmission. In other words it increases the number of initially intended transmissions, so that the redundant transmissions statistically cover for the lost transmissions. The ETX can be also applied to a path $p$, consisting of a chain of several links, described in equations 2.2 and 2.3. This also reflects the similarities between ETX and the hop count, as they are identical in the case of $p_{i,j} = 1$. Although this metric is derived from an actual link property and therefore offers a simple but important improvement over the hop count, it does not account for the link load or different data rates.

$$ETX_{i,j} = \frac{1}{p_{i,j}} = \frac{1}{1 - \epsilon_{i,j}} \tag{2.1}$$

$$ETX(p) = \sum_{e \in p} ETX(e) \tag{2.2}$$

$$= \sum_{e \in p} \frac{1}{p_e} \tag{2.3}$$

An extension of the ETX, including the ability to transmit in different data rates is the ETT [21]. It is considered a bandwidth-adjusted ETX, in that it is calculated directly from the ETX by dividing it by the data rate of the transmission $r$. To retrieve the time for the whole transmission to be completed, this subsequently has to be multiplied with the size of the data in the transmission $N$, as equation 2.4 describes. Since the ETT combines the most important link properties, namely the error rate and the available data rate, it is perfectly suitable for consideration in broadcast protocols.

$$ETT_{i,j} = ETX_{i,j} * \frac{N}{r} \tag{2.4}$$

## 2.5 Optimal Performance Bound

The presented state of the art of broadcast routing schemes yields with classical flooding a suitable lower performance bound for optimizing the broadcast performance in WMN, since it enables a comparison to the currently most important deployed mechanic. However, to determine an upper or optimal performance bound is not as trivial. On account of the *wireless multicast advantage* [72], the usage of omnidirectional antennas in a wireless environment enables nodes other than the intended receiver to also receive the transmission successfully, depending on their distance to the sender. This renders purposeful data transfer even more complicated, in terms

of optimal efficiency. For the single-rate broadcast, finding a CDS of minimal size has already been proven to be NP-hard [44], leaving no simple solution for a true minimum cost broadcast. Similarly, optimizing the expected transmission time [81] or the latency [58] in a multi-rate broadcast was found to be NP-hard as well.
Nonetheless, to evaluate the broadcast performance gain, an upper performance bound for the minimum airtime broadcast must be established. Therefore, the following optimization problem, described in formula 2.5 has to be solved. The goal is to minimize the overall transmission time, that sending nodes spend transmitting on a rate $r$ for all possible rates and for all nodes in the network. However this must be minimized subject to three constraints. First, each node is allowed to transmit at maximum one time. $s_i^{(r)}$ describes the binary decision, if node $s$ will transmit at rate $r$. Second, each node in the network has to be connected to a transmitting node, while the error probability using the transmission rate ensures a successful delivery. The last constraint makes sure, that all transmitting nodes are connected for the respective transmission rates, so that the transmissions will span not only a sub-graph of connected nodes.

$$\min \left( \sum_{r \in R} \sum_{i \in V} \frac{1}{r} * s_i^{(r)} \right) \tag{2.5}$$

$$\begin{aligned}
s.t.\ (1) \quad & \forall i \in V : \sum_{r \in R} s_i^{(r)} \leq 1 \\
(2) \quad & \forall j \in V : \sum_{r \in R} \sum_{i \in V} \left\lceil p_{i,j}^{(r)} \right\rceil * s_i^{(r)} \geq 1 \\
(3) \quad & \left\{ i \in V \middle| \sum_{r \in R} s_i^{(r)} = 1 \right\} \text{ are connected}
\end{aligned}$$

### 2.5.1 Simulated Annealing

To approximate an optimal solution, simulated annealing, first proposed by Kirkpatrick *et al.* [38], can be used. It is a probabilistic method to approximate the global minimum of an optimization problem, that may additionally exhibit multiple local minima. The basic elements of simulated annealing are a set of states $s \in S$ and a cost function $c(s)$ defined on $S$. The general procedure of simulated annealing starts with an initial state $x(t = 0)$. For each iteration in time, a random neighbor state $s'$ of $s = x(t)$ is selected and the cost of both states are compared. If the cost decreases, i.e. $c(s') \leq c(s)$, the new state will be selected for the next iteration, $x(t + 1) = s'$. Otherwise, the following state will be determined as follows:

$$x(t+1) = \begin{cases} s', & \text{with probability } \exp\left(\frac{-(c(s')-c(s))}{T(t)}\right) \\ s, & \text{otherwise} \end{cases} \tag{2.6}$$

where $T(t)$ is the temperature in time step $t$. It determines the likelihood of selecting a cost-wise worse state to explore neighboring states, attempting to prevent the convergence on a local minimum. For higher temperatures, the probability for the system to select a less preferable state is higher and the exploration of states is greater. Therefore, if the temperature is sufficiently high, i.e. the annealing simulation has just started, the state can leave the range of an local minimum, by selecting a state with a worse cost. To finally converge on a minimum, the temperature has to decrease over time according to a *cooling schedule* [8]. This prevents the state to escape the range of the current minimum and enables a step-wise convergence towards the local minimum itself, approximating an optimal solution. The result of simulated annealing, is therefore not guaranteed to be the global optimum. The success of the process greatly depends on the cooling schedule. On the one hand, it must allow for a initial period of time to change to states with higher costs, to explore and increase the chances to approach the global minimum. On the other hand, it must ensure towards the end of the simulation, that a low temperature optimizes the final result to reach a minimum. In addition to that cooling schedule, the definition of the state space and more importantly the definition of neighboring states has a significant influence on the result. If the neighbor states are defined in a way, that it is closely related to the original state, the expected cost will be similar for both states. If however the neighbor is loosely defined and includes extremely different states, the cost of the neighbor does not have to be similar to the original state. This hinders the convergence to a minimum.

### 2.5.2 Monte Carlo Simulation

An alternative approach to obtain an approximation for the optimal solution is a *monte carlo simulation* [62]. This rather simple technique pursues a probabilistic approach as well. Similarly to annealing, for each simulation run it determines an effectively achieved cost. For the selection of the simulated states however, it relies only on a random selection. It therefore iterates through a preferably large set of configurations, to cover a representation of as much scenarios as possible. For smaller sets of iterated states, the probability to select the best performing state decreases. The monte carlo simulation eventually gives an estimation of the optimum through the best achieved cost. As this includes only an overview of possible solutions, it is not suitable for the determination of the true optimum of the formulated broadcast problem. However it can provide a good overview over the complete range of possible results, granted that it covered enough states with sufficient iterations.

To approximate a solution to the broadcast optimization problem, this work primarily uses simulated annealing. This enables that a reliable upper performance bound can be found. Additionally, it will be complemented by a second monte carlo simulation to provide a further overview, as later explained in section 3.3.1.

# 3 Implementation

In order to assess the performance of broadcast routing schemes for WMNs, I created a dedicated simulation environment. It provided the implementation of a basic network model, which is required to deploy the different routing protocols, while maintaining the control over any additional structural and functional overhead, that might influence the results. This ensures a lightweight solution, tailored to the exact needs of the protocol evaluation.
To evaluate and compare the broadcast routing performance of each protocol, the minimum requirements include an implementation of a WMN, i.e. a set of nodes, that are connected among each other to form the network, as well as means to transmit data between multiple nodes. Data transmission is realized inside a IEEE 802.11 [75] WLAN environment, by simulating the correspondent wireless channel characteristics. The evaluation of a broadcast scheme does not require a realistic, but complicated channel implementation, but only a basic mean of determining a successful data transport depending on the node distance. Since the evaluation is focused on the protocol performance, the simulated network environment does not account for any irregularities, that may impose an influence on the wireless connections. This includes asymmetrical links as well as signal spread, obstruction or reflections. The nodes are placed in an empty space and connected via a wireless broadcast channel. This results in noise-free, perfectly bi-directional channels, as there are no external influences.
The custom environment on top of that channel model allows to omit the regular protocol stack for packet transmission in favor of a basic broadcast transmission functionality. Nodes can directly exchange the data, if the channel model allowed for a successful transmission. Therefore the broadcasted packets constitute the sole type of messages in the network. No additional transmissions for general operability, e.g. establishing a connection or coordinating the data exchange, are necessary. Furthermore the transmitted packets do not carry external protocol header information, assuring a reduction to the bare minimum. As additional network protocols are not implemented, the packets do not require to include for instance a TCP or IP header. This narrows the focus on the sole evaluation of the broadcast performance, as all those possible additions are not required for this purpose. The goal of the final implementation was therefore to only fulfill these general preconditions, to ensure a lightweight reduction to the minimum requirement.
As such, a modular network simulator for the testing of interchangeable protocols is described in this chapter. Section 3.1 describes the details of the implementations, while section 3.2 is focused on the parameter space for each simulation.

## 3.1 Simulator

The custom developed simulator serves three main tasks, namely the network simulation, protocol deployment and performance evaluation. It is implemented in python, to take advantage of its lightweight and intuitive character. The general operation is organized in a series of discrete events to simulate an advancement in time. The simulation steps are grouped and organized by recurrent transmission slots. This enables the broadcast transmission itself to be divided into and modelled as a series of singular local transmission steps. However, the broadcast procedure is preceded by a static setup sequence, including the network generation and protocol setup.

### 3.1.1 Setup Procedure

The simulator setup starts with the random placement of the nodes in a specified area. Node objects are identified by an alphabetical label and furthermore divided into source nodes and destination nodes, triggering different setup procedures later in the simulation process. For the broadcast, only a single node at a time is assumed to be the source, while the rest of the network nodes are considered as destinations. To create the WMN from the set of nodes, they have to be connected through a wireless transmission channel. Since the simulator aims to create a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) environment according to IEEE 802.11 [39] in a single shared channel for the whole network, no interference between the data transmissions is assumed. Therefore the channel between two nodes is characterized only by the respective Signal-to-Noise Ratio. To obtain the Signal-to-Noise Ratio (SNR) pair-wise from the node distance, the simulator implements the *TGn model D* channel model by the IEEE [65][24], which is similar to the ITU *Indoor Path Loss Model* [64].

$$L(d,f) = L_{FS}(d,f) \qquad d \leq d_{BP} \tag{3.1}$$

$$L(d,f) = L_{FS}(d,f) + 35 * log_{10}\left(\frac{d}{d_{BP}}\right) \qquad d > d_{BP} \tag{3.2}$$

It is derived from a basic free space loss model, but accounts for signal spread as well as obstacles in the environment, that interfere with the signal after a breakpoint distance $d_{BP}$ of $10$ m, see equations 3.1 and 3.2, as well as figure 3.1. Therefore the node deployment in an empty space can be justified, since the propagation model considers the deployment environment. The selected channel model represents transmission conditions in an open commercial environment and is used for indoor as well as outdoor scenarios. It is therefore suitable for the industrial use case, presented in section 2.2.1.
The free space loss model in the channel model is described as follows:

$$L_{FS}(d_{[m]}, f_{[MHz]}) = 20 * log_{10}(d) + 20 * log_{10}(f) - 27.55[dB]. \tag{3.3}$$

The signal loss depends on the distance $d, [d] = m$ and the frequency of the signal $f, [f] = MHz$. Within the selected WLAN environment, $f = 2400$ MHz is assumed.

$$SNR_{dB}(d) = P_{tx,dB} - L(d, f) - P_{noise,dB} \tag{3.4}$$

To retrieve the final SNR from the distance of the node pair, the obtained path loss as well as the background noise level has to be deducted from the transmission power. This simple model can be assumed for the simulator, although it may not model the environment realistically. Further interference or different signal disturbances impair the ability of the receiver, to successfully decode the transmission, since he might not be able to distinguish these errors from the received signal.
The resulting formula 3.4 converts the distance of a node pair to the SNR value of the transmission. For the simulator, an unidirectional antenna with a transmit power of $15$ dBm is assumed, concurring with the german federal regulations [6] and leaving headroom for the reduced possible transmit power on higher data rates of several WLAN industrial and consumer devices [26]. The noise floor is at $-95$ dBm, which is commonly assumed, e.g. in the Linux kernel and therefore affecting a variety of hardware, as well as supported by the majority of WLAN devices [26][69].

Dynamic routing protocols usually rely on a link discovery, that does not detect the link quality in terms of SNR, but instead gains a success probability from actual channel probing. As this would require a lot of coordination overhead and furthermore is not necessary, as the channel conditions do not change over time, the success probabilities of the connections are derived from the SNR values by a look-up conversion. Instead, to imitate the characteristics of a real link probing, any link discovery was only successful, if the success probability matches or exceeds $0.1$. This was necessary to eliminate the discovery of nearly unusable links, as many of them would later result in a PER greater than $0.95$, due to the utilized SNR to PER look-up data. Although realistic, since e.g. B.A.T.M.A.N. uses 64 probing packets and therefore would most likely discover such a link, this creates a significant drawback for error rate insensitive schemes.

Since this work aims to utilize different transmission rates, an IEEE 802.11 environment was chosen. More specifically, the 802.11ax standard [76], which is one of the latest versions in commercial use, but not completely finalized yet. It supports the usage of CSMA/CA [3] and different MCSs, that offer a range of data rates by selecting an unique combination of a modulation scheme and a coding rate for the transmission. In the scope of this work, a reference of a specific MCS refers to the connected data rate. With a deteriorating received signal quality, the modulation and coding must be more robust to counter the occurring transmission errors. For this work, the MCSs 0

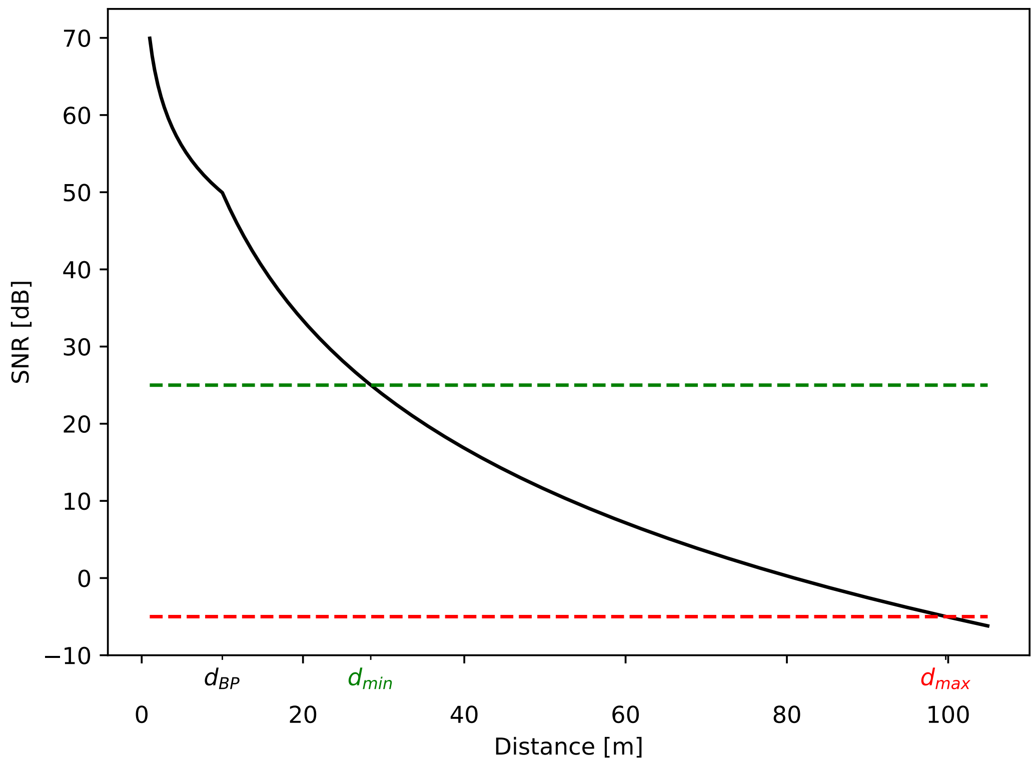



Figure 3.1: SNR over distance according to the TGn channel model D

to 9 are used, as they are included in the PER look-up data. Furthermore, the physical layer transmission parameters were assumed as follows: the lowest available *channel bandwidth* of $20$ MHz is used with *OFDM* with a *guard interval* of $1.6$ µs, while operating with a single *spatial stream*. These settings are a general baseline, that do not require advanced compatibilities.
The selection of MCSs is shown in table 3.1.

For each of these MCSs, the look-up yields a different PER, since more advanced MCSs demand an increased channel quality. [56] provides a look-up a detailed measurement collection, linking the SNR to a PER for MCS0 through MCS9, as figure 3.2 depicts. The results were gathered averaging over various independent measurements by multiple companies and provide precise PER values for a broad range of SNRs between $-5$ and $25$ dB. Outside this SNR range, the PERs will be always $0$ for higher and $1$ for lower SNR. This results in a minimum and maximum node distance, $d_{min}$ and $d_{max}$ respectively, in which errors randomly occur. That can be seen in figure 3.1, showing the SNR over the distance according to the channel model.
When all links in the network have been discovered, every node has a set of neighboring nodes, containing additional information on the available MCSs for each link, as well as the associated PERs. This reproduced the result of for instance the *HELLO* message implementation in OLSR or the OGMs for B.A.T.M.A.N., which distribute link

| MCS Index | Modulation | Coding Rate | Data Rate [Mbit/s] |
|---|---|---|---|
| 0 | BPSK | 1/2 | 8 |
| 1 | QPSK | 1/2 | 16 |
| 2 | QPSK | 3/4 | 24 |
| 3 | 16-QAM | 1/2 | 33 |
| 4 | 16-QAM | 3/4 | 49 |
| 5 | 64-QAM | 2/3 | 65 |
| 6 | 64-QAM | 3/4 | 73 |
| 7 | 64-QAM | 5/6 | 81 |
| 8 | 256-QAM | 3/4 | 98 |
| 9 | 256-QAM | 5/6 | 108 |

Table 3.1: Modulation and Coding Schemes (MCSs) in IEEE 802.11ax

quality reports in the neighborhood. Therefore the direct neighborhood of every neighboring node can be seen as well, resulting in a 2-hop neighborhood knowledge for every node in the network.

When the nodes are connected and the network is created, it has to be checked for complete connectivity. All nodes must be connected via at least one link to the rest of the network. There can be no separate single nodes or connected node groups in the network, since the broadcast could not be successful at all and the results would be falsified. To assure the connectivity throughout the network, a Depth-First Search (DFS) algorithm [19] is used, starting at the designated source node. It creates a set of connected nodes by looking at the neighbors of each node and marking checked nodes as already visited. This resulting set of nodes must be identical to the set of nodes in the network to guarantee the connectivity. If the check fails, the simulator setup has to be restarted at the network creation.

Figure 3.3 shows a representation of a generated demonstration network. Each node is placed according to its position in meters, as it is required by the SNR calculations. The color for each link represents the highest available MCS for each specific node pair. The apparent correlation between the link quality and the distance is clearly visible. Node pairs not connected by a line exceed the maximum distance $d_{max}$ and are not connected on any MCS.

### 3.1.2 Simulation Procedure

After the successful network creation, the simulator is ready to conduct the broadcast simulation. At first, the designated broadcast protocol is specified and initialized, if necessary. To start the simulation, the source needs to have at least one packet queued for transmission. The simulation itself is structured in a transmission cycle,

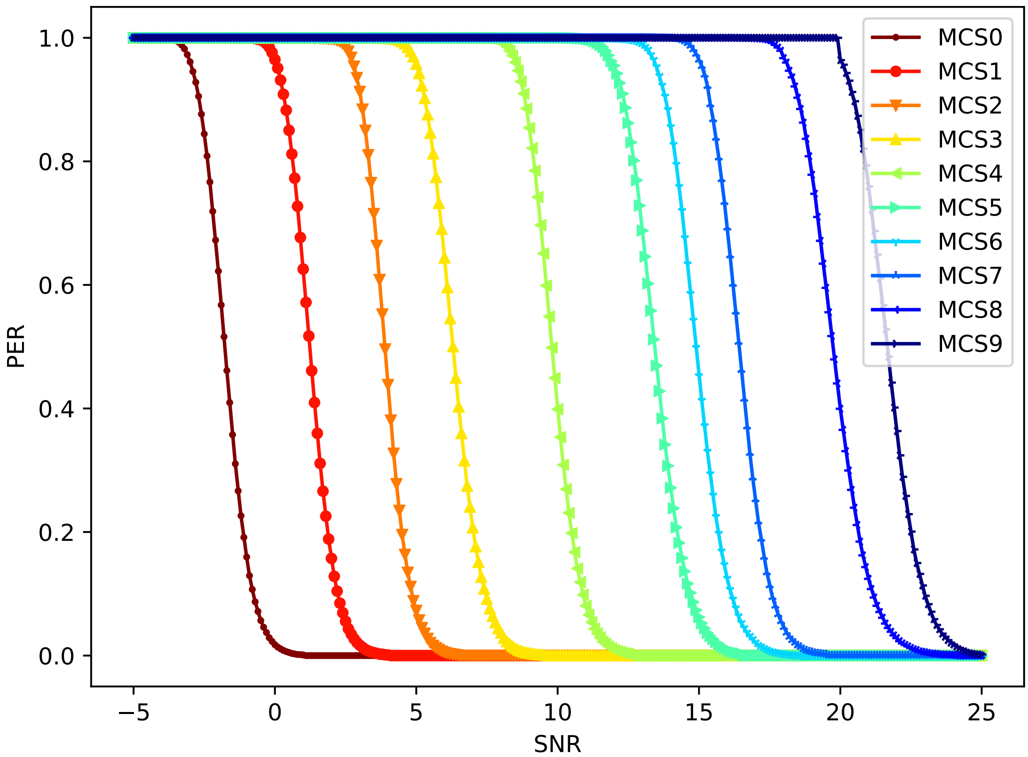



Figure 3.2: SNR to PER mapping for MCS0 through MCS9

executing one transmission attempt at a time. This ensures, that there is no interference in the network, as the transmission channel is shared among all nodes and only one transmitting node is allowed. This transmitting node is randomly selected out of all nodes, that have a transmission scheduled, i.e. in the context of this simulator, a transmission waiting in the queue. This represents the CSMA/CA scheme, as all other nodes, that could not send in the current time slot have to back-off their next transmission for an unknown, random time, until they are selected for transmission. From the now selected transmission sender, the first scheduled packet is retrieved. In the next step, the appropriate MCS is selected by the protocol, based on the information from the node itself. In the case, that this transmission is a relaying transmission for a different broadcast, the previous packet header information is additionally taken into account. The protocol section of the current packet's header is updated according to the protocol details and the packet is sent through the network. For every neighboring node it is determined, if it was able to successfully receive the packet, which happens probabilistically based on the pre-determined PER between the sender and the potential receiver for the previously selected MCS. If a transmission was successful, the corresponding receiving node inspects the packet and discards it, in case that it is not innovative, i.e. it does not contain new information, that has been seen before. Afterwards the current protocol determines, if a new rebroadcast transmission is to be scheduled. At the end of every transmission

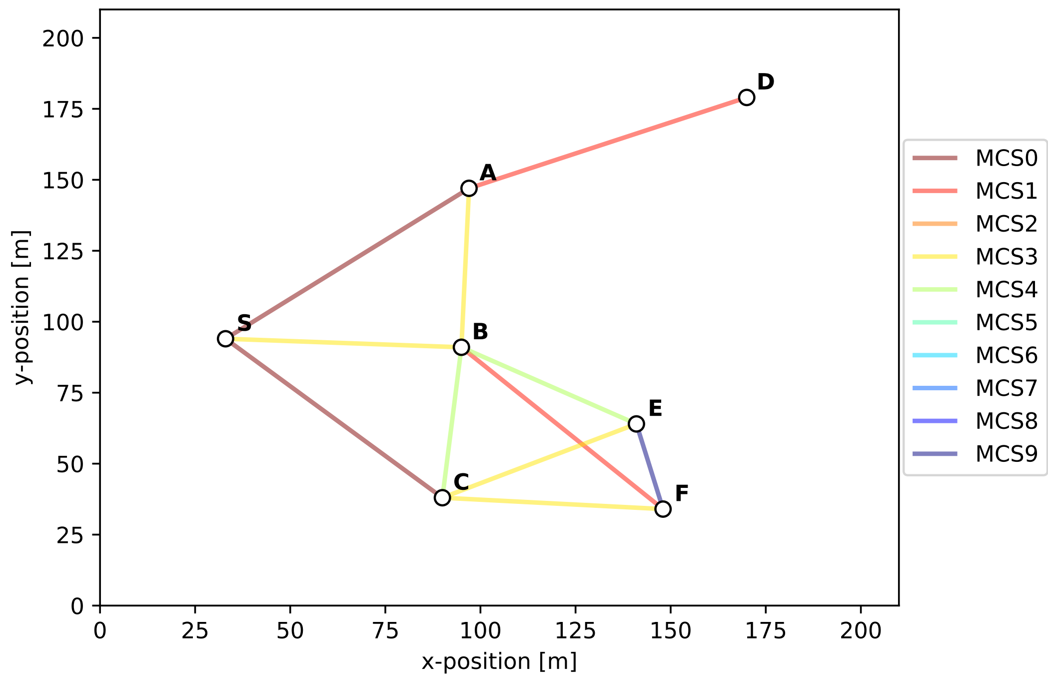



Figure 3.3: Representation of a demonstration network generated by the simulator

phase, the simulator checks if the simulation is complete, i.e. that no more transmissions are scheduled by any node and every node has the intended information. A simulation can also end, when it reaches a specified number of steps and times out.

Upon completion of the network-wide transmission of the source packets, the statistics are gathered and analyzed. The whole transmission process is then repeated, to guarantee statistically representative results. As soon as all those *batches* of the transmissions are finished, the broadcast simulation for this specific network-source pair is completed. To increase diversity in the simulations, the same network is simulated with all other nodes being the source as well.

## 3.2 Parameter

To create different simulation scenarios, there are a number of parameters, that have to be specified separately for each simulation. They determine the network setup as well as the procedure of the simulation itself.

### 3.2.1 Network Parameter

The creation of the implemented network model depends on a set of parameters. Naturally, the size of the network is directly determined by the *number of nodes* in it. The more nodes are connected, the more intricate the network structure and therefore also the complexity of the routing problem becomes. By varying the node count, the scalability can be evaluated. With an increasing amount of participants in the network, the number of links increases. Depending on the density of the network, the addition of the $i$th node can generate $i - 1$ new links. Given the specific number of nodes to be placed in the network, the network density can be influenced by manipulating the dimensions of the area, that the nodes are randomly placed in. Since this area is squared, this leaves a single parameter, the *size multiplier* to create the desired side length $l$. The most notable impact of a change in the network area is the influence on the number and quality of the connections between the nodes. When the number of nodes stays constant, the network area can on the one hand spread the nodes, resulting on average in an increased node distance, which translates directly into a worse SNR. On the other hand a decrease of the network area moves all nodes closer together. This leads to an increase in the SNR, enabling the usage of higher data rates for the transmissions.
Both parameters in conjunction influence the complexity of the routing problem. If a higher number of nodes is placed in a smaller area, the increase of useful connections as well as the additional available MCSs for each connection constitute an increment in possible routing choices for each node. The direct neighborhood for each node will also include more nodes. Therefore the protocols must regard more nodes in their calculations, potentially challenging the routing algorithm. This is stressing and eventually testing the scalability, as the computational complexity and resource demand increases, as well as the amount of protocol decisions to be shared.
In a realistic deployment of any protocol, this might also influence the performance. Depending on any information, that needs to be shared among the nodes as a basis for the routing decision, the network traffic might increase. As the protocol adds either to the packet header or creates dedicated transmissions to disseminate this information, it adds to the total of transmitted data. For a large number of nodes, this could congest the network.

### 3.2.2 Traffic Parameter

To execute a complete simulation, the network must be tested by creating and transmitting actual traffic. This traffic is subject to multiple parameters, such as the *packet size*, the *number of transmitted packets* and the *interval* between the source packets. For this thesis, the interval will not be regarded, as all traffic will be singular and not periodically, i.e. there is only one set of source packets, that are fixed from the beginning of the simulation. This complies with the primarily considered use cases of

lower-layer network protocols or industrial applications. Furthermore, the character of the simulator can not guarantee any correct timing in between specific events, as the simulated time span for each transmission slot depends on the selected transmission rate. For a plethora of industrial use cases, the frequency exceeds the actual time, that the complete broadcast in the network consumes. This would eliminate any 'inter-broadcast' influence, that newly created traffic might cause.
The packet size for the simulations is set to $32$ bytes. This provides the highest possible accuracy in terms of the usage of the PER look-up data, since the reference measurements were conducted with that exact packet size [56]. Furthermore this presents a compromise between the traffic characteristics of the potential applications in table 2.1.
These characteristics also include a relatively spare usage for the case of lower-layer protocols, such as DHCP or ARP. Therefore the broadcast is simulated with a single packet to be transmitted at a time. This matches, as previously described, also industrial traffic characteristics as well, as they transmit periodically, but with enough time to complete the previous broadcast, before the next packet arrives. To create this scenario, each transmission is executed a number of times in succession, specified by the number of *batches*.

## 3.3 Performance Evaluation

Firstly, to evaluate and compare the final performance of each broadcast scheme, a common evaluation metric has to be established. Typical WMN broadcast use cases, as presented in section 2.2.1, have different requirements to their broadcast delivery. Neither of the in this work represented use cases demands a high throughput as media delivery services, although most of the future user-oriented use cases will [15]. Instead all traffic is small and requires to be delivered fast, while consuming as little shared and individual resources as possible.

However, all of these applications have a clear similarity, in that they all rely on a assured delivery. Therefore the primary optimization objective of any broadcast scheme should be to ensure a reliable delivery. The reliability stands out from all other metrics, as it is the only value, that needs to be maximized, to perform best. Additionally it is critical for the overall performance, that the reliability is the highest priority. Since any further optimization would reduce the extend and number of transmissions, this would otherwise eventually lead to a failure to reach every single node in the network. In the context of broadcast transmissions in this work, reliability is defined as the proportion of nodes, that successfully received all of the initial source data at the end of the transmission. This can not be achieved in every case, as long as there are errors occurring in a single transmission, thus multiple transmissions need to be taken into account. If this is not possible, the result must be adjusted accordingly.

The next priority when evaluating the performance is the minimization of airtime. Airtime refers to the time, the shared transmission medium is occupied by a transmitting node, holding the other nodes from sending. In real use cases this would expand to the time, the next sender is negotiated and also to any potential feedback, but all of this is omitted in the simulator. Therefore it is beneficial to select a preferably high MCS for each transmission, as a transmission with a faster data rate consumes less time for the same amount of data, compared to a slower, more reliable data rate. For the broadcast case, the overall airtime is determined by the sum of airtime of all transmissions originating in the network-wide broadcast of the source packet. It hence includes all transmissions, that were scheduled or executed, although all nodes already had the necessary information. This can occur, since the absence of feedback takes away the means to check, if any neighboring nodes still requires additional information. Since the transmitted data in the simulator consists only of $32$ bytes of actual payload, complemented by only virtual packet and header data, the airtime for a single transmission is calculated by dividing the size of the payload by the data rate of the selected MCS.

Lastly, the achieved latency could be examined. Although the latency is a separate metric, it is already partially covered by the airtime, in the case of transmitting only a single packet. In the broadcast case, the latency is defined by the time from the start of the transmission by the sender until the last node receives the whole information. For the case, that every node sends only a single time, the delay differs only in the last transmissions from the airtime. At the end of the broadcast, the nodes at the edge of the network have further transmissions scheduled, although every node in the vicinity may already have the information. In this case, the remaining transmissions would increase the airtime compared to the latency. As this occurs predominantly at the edge of the network, it makes only for a small proportion of the overall airtime in a large network. Furthermore it can even be prevented in advance by additional coordination. All in all, latency can best be minimized, by avoiding failures in reaching single nodes, i.e. a high reliability, as well as utilizing the best available data rate to increase the transmission speed, i.e. a small airtime. Therefore latency will not be specifically examined in the evaluation.

### 3.3.1 Optimal Performance Bound Determination

To determine the optimal performance bound, the simulator includes an implementation of a monte carlo simulation, as well as an implementation of simulated annealing. Applied to the broadcast problem, both implementations operate on a state space $S$ as the set of routing strategies, where each node is assigned a binary value

representing if the node is transmitting. Furthermore each strategy includes the selection of the transmission rate for each transmitting node. To also compensate for errors in transmissions with the lowest MCS, the strategy is expanded for a number of up to ten retransmissions, if a node in the strategy has been selected to transmit on the lowest data rate. This is not included in the candidate state of the art protocols in section 2.3, but complies with the protocol design, presented later in chapter 5. As described in section 2.5, simulated annealing additionally requires a definition of neighbor states. Therefore two states are considered to be neighbors, if they differ in either a single transmitting node, a single data rate or the number of retransmissions for one node transmitting on the lowest transmission rate. This carries the risk, that simulated annealing experiences problems when converging to a local or global optimum. As the removal of a single node might completely or partially break the broadcast, the resulting evaluated performance can be immensely different for two neighboring states. However, this can not be avoided in advance.
To improve the performance bound simulations, the set of allowed strategies for both simulations is restricted. Nodes, that connect otherwise isolated nodes, are firstly forced to transmit and secondly also forced to select the lowest data rate or an alternative, that enables an error-free transmission towards the isolated node. This prevents foreseeable impossible strategies and accelerates the simulations.
For the final evaluation of each annealing and monte carlo simulation run, a cost function is required. However, instead of minimizing a cost function, this work implements the maximization of a performance indication, to account for the reliability in combination with the airtime. The previous section established priorities for this eventual performance evaluation. The first priority of the reliability consequently dominates the performance. In general, a reliability less than $1$ can void the performance, as all nodes have to be served, if possible. However, the cost function must support continuous reliability values to also compare two not completely reliable states to step-wise approach to optimum. Otherwise two neighboring states, in which one strategy covers only ten percent of all nodes successfully and the second strategy all but one node, would be evaluated equally. This could result in the selection of each state, although one of both is clearly closer to an optimum.
The airtime has to contribute inversely, as a lower airtime is valued higher. Balancing both values constitutes the difficulty in finding a suitable performance function, as an increased reliability could justify an increase in airtime, but only to some measure. This balance has to be adjusted according to the respective application, as the results could otherwise diverge a lot from the desired optimum. To adjust the balance, the performance function in equation 3.5 includes a variable parameter $\alpha$. The performance is obtained from the reliability of the simulation to the power of $\alpha$, divided by the achieved overall airtime.

$$performance(reliability, airtime) = reliability^{\alpha} * \frac{1}{airtime} \tag{3.5}$$

For the performance evaluation, the reliability is emphasized by selecting a larger value for $\alpha$. It is mandatory to consider the network conditions for this selection as well. If they include links, that are essential for the broadcast transmission and offer only the lowest overall MCS, any error probability on this link could prevent a successful transmission, independent of the number of retransmissions. There is always a chance, that all transmission attempts fail, which consequently would exclude at least one node from the rest of the broadcast transmission. Subsequently, complete reliability can not be achieved and the performance measure has to account for this circumstance. Therefore, later simulations with $l = 100$ m and $200$ m are allowed to select the maximum value of $\alpha = \inf$ to optimize for complete reliability and minimum airtime. This also includes all iterations for a second simulation over the number of nodes $n$ for $l = 200$ m. However, the simulated annealing for a larger network with $l = 300$ m or $400$ m, accounts for possible unavoidable errors by selection $\alpha = 5$. In contrast to simulated annealing, that needs to apply the performance function in between each step, a monte carlo simulation can evaluate the results after the simulation has finished. Therefore, the selection of $\alpha$ includes a wider range of $\alpha \in \{2, 3, 4, 5, \inf\}$. Furthermore, each monte carlo simulation is repeated $100000$ times.

$$T(t) = 1 - \frac{t}{k}. \tag{3.6}$$

The cooling schedule selected for the simulated annealing is presented in equation 3.6. The simulation concludes after a fixed amount of $k = 10000$.

To differentiate between both simulations, they each pursue different goals. The simulated annealing implementation determines an upper performance bound, allowing only such rates, that offer an error free delivery. This ensures, that the result represents a strategy, which delivers a reliable performance, if applied successively. In contrast to that an application might tolerate unreliable transmissions to some extend, that are compensated with a network coding scheme. Random Linear Network Coding (RLNC) [31] could for instance compensate lost transmissions. RLNC is a coding scheme, that decouples the dedication of specific packets to specific information and transmits a random combination of the original information instead. This allows the sender to create redundancy with supplementary transmissions, whereas compared to regular end-to-end coding schemes, every single additional packet can recover one failed transmission. Alternatively, the application could also value the airtime a little bit more, in exchange for a small reliability decrease. In such an application, transmissions could be executed on higher rates with success probabilities smaller than $1.0$, gambling on a successful transmission. Then the result would be a lower airtime, while at the same time a probabilistically decreased reliability. These use cases are not included in the simulated annealing result. To obtain an estimation of the maximum possible performance, regarding only a single broadcast attempt, the monte carlo simulation is included in this work. It allows the selection of any rate with a success probability larger than $0.0$. Furthermore, it will simulate each strategy only once, while the simulated annealing simulates at least a closely related strategy repeatedly. Therefore simulated annealing is not able to value a strategy, that was successful by a marginal chance. The monte carlo simulation, however, will include these strategies and hence generate a different performance bound. Its result will consequently also represent a completely different approach, attempting transmissions more aggressively towards an airtime improvement.

# 4 Performance Evaluation of State of the Art Routing

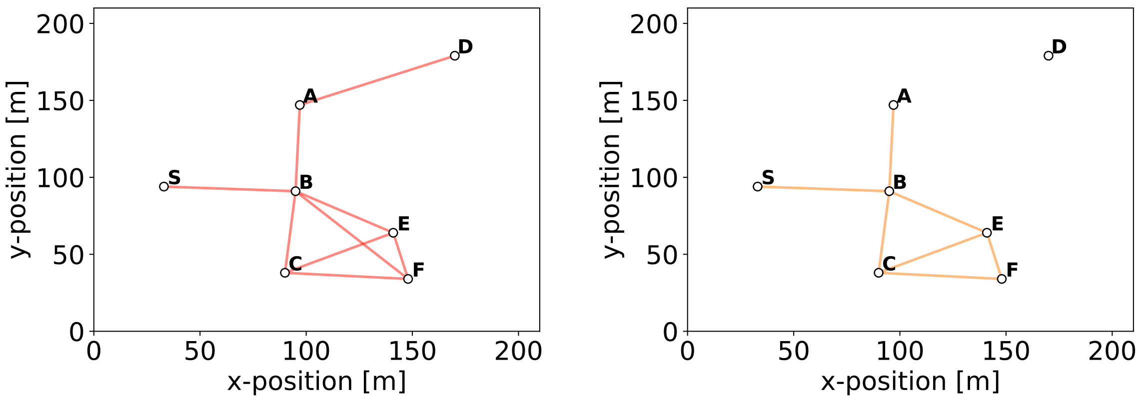
In this chapter, the reliability and airtime performance of the selected state of the art broadcast schemes is evaluated, to work out potential strengths and weaknesses for the protocol design in the following chapter. As discussed in section 2.3, these schemes include:

- Classical flooding, as it is still widely used and relied on, being virtually the standard in broadcasting,
- B.A.T.M.A.N., as a different approach to flooding with already actual deployments in real applications,
- OLSR, offering a unique scheme to overcome the shortcomings of flooding.

Further approaches presented in section 2.3.5 will not be considered here, as they violate basic structural or conceptual principles. The often assumed complete network knowledge can not be achieved for every node without a reliable broadcast in the first place to distribute the topology information. Likewise, the transmission scheduler for the latency optimization defies the idea of a mesh network, in that mesh networks do not have a central entity for signalling and controlling.

All considered state of the art solutions have an important feature in common, since all three operate on a fixed transmission rate. Depending on the wireless environment, which is used for the transmissions, this data rate is usually set to the available minimum, to maximize reliability and counter the lack of a feedback to restore possible losses. With prior general network knowledge, this rate can be increased under circumstances, that already guarantee an error-free delivery. However, this does not happen in the scope of the protocol, rather than the network operator or independent lower-layer functionalities [35][74].
This provides the possibility of an elementary data rate adaption, although it can not be directly compared to the regular ability to dynamically adapt transmission rates. The fundamental difference lies in the prerequisite, that the majority of links in the network have to support the increased rate. Otherwise the network could potentially be not fully connected, making a successful broadcast impossible. Furthermore, there is no verification in place, that the amount of links, supporting a higher rate, is actually sufficient to build a fully connected network. This is why only the clear majority of links and not just a smaller proportion is assumed sufficient. As a result, already a small set of links, that do not support increased data rates, can easily cause a single or a set of nodes to be isolated. Figures 4.1a and 4.1b show the connectivity in the

demonstration network for MCS1 and the next increment MCS2. It can be seen, that only two of eight links, $(A, D)$ and $(B, F)$ are effected, but the network has already lost the ability to connect node $D$ to any other node. This must exclude MCS2 from consideration in the case of the demonstration network for all fixed rate protocols.




(a) MCS1 (b) MCS2

Figure 4.1: Representation of the demonstration network allowing only a minimum fixed MCS

Following this principle, figure 4.2 displays the distribution of the highest available MCS for 250 random graphs each. Varying the number of nodes and the network sizes, different network conditions and node densities are generated. The general tendency shows, that if the node density increases, i.e. more nodes in a constant area, the highest MCS providing connectivity to all nodes increases as well. However, contrary to this intuitive approach, large networks show a similar behaviour for a very small number of nodes. This can be explained, since the possibility for five nodes to be placed in a close vicinity is relatively high, compared to more than five nodes. The more nodes, the higher is the probability, that one of these nodes is placed outside of the reach of higher MCS. In that cast, only the lowest MCSs can provide connectivity. This holds true, until the network space is filled and the best rate increases again, with additionally added nodes. Nevertheless, the overall available rates for larger network sizes are reduced to lower MCSs. Smaller networks will, however, also offer a higher MCS connecting the network, as the average distance can hardly be as high, that only a low data rate is allowed. This leaves only the base rates for networks spread over a bigger area or with fewer participants.

The second major disadvantage in the usage of a preset transmission rate, is also showcased by the critical link $(A, D)$ in the demonstration network. Given that MCS1 had previously been selected as highest available rate, to ensure connectivity, the distance between nodes $A$ and $D$ is still comparatively large. The link has only a low success probability of $0.12$ on the selected rate, which however is not regarded by the routing. All candidate protocols only detect the available link with the previous

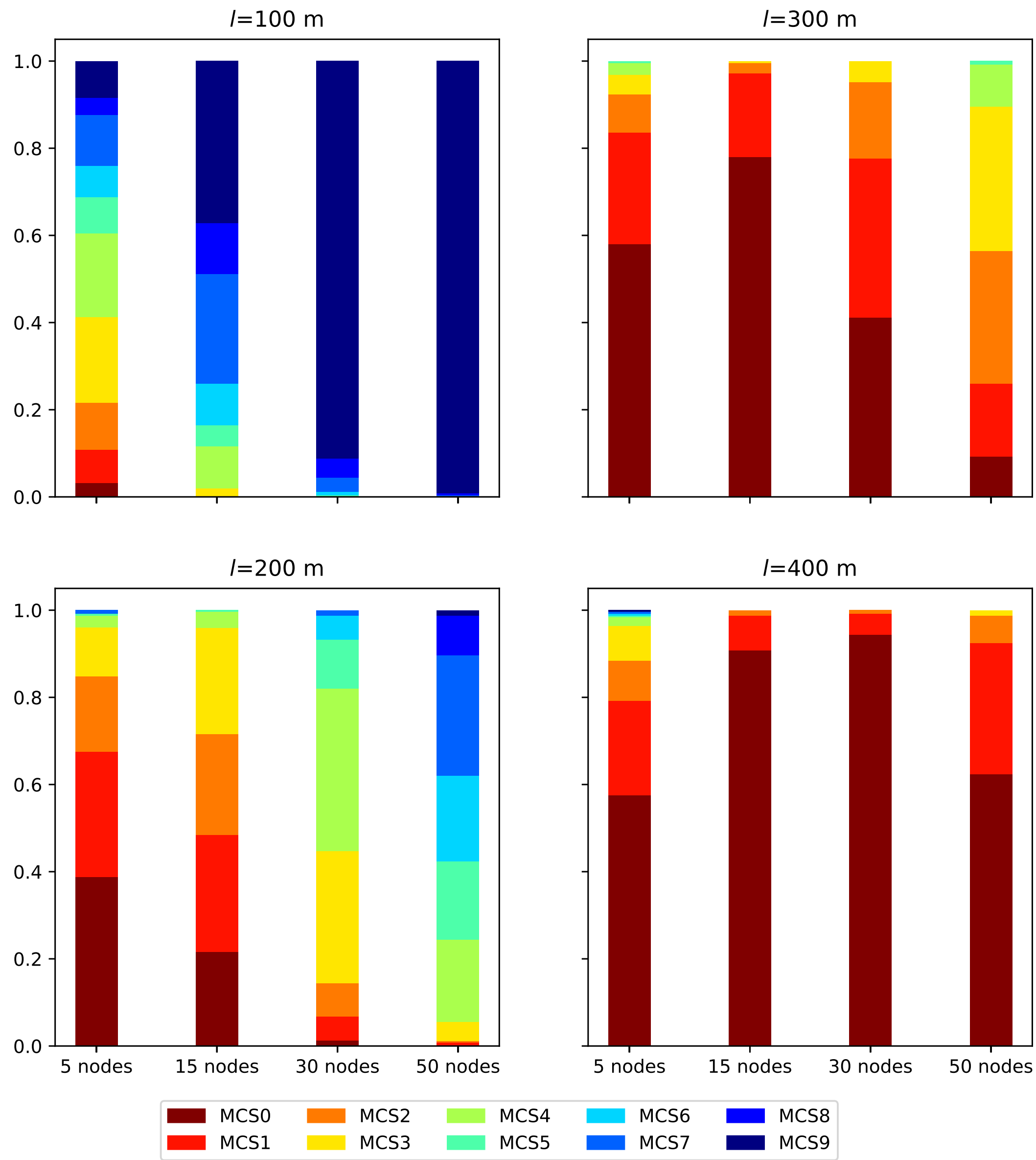



Figure 4.2: Proportion of each maximum MCS connecting the complete network for several number of nodes and network sizes

link sensing. By the time it comes to make the decision on whether to use the link for routing or not, which is obviously mandatory, the link quality is not taken into account. The link is blindly assumed functional, once it has been discovered. In the case of B.A.T.M.A.N. for instance, the possibility of the discovery is with one successful probe over the last 64 attempts relatively high.

To preliminarily evaluate the actual performance, two simulations were conducted. The first shows the number of transmissions of all protocols applied to the demonstration network in figure 4.3. For the evaluation of the performance results,

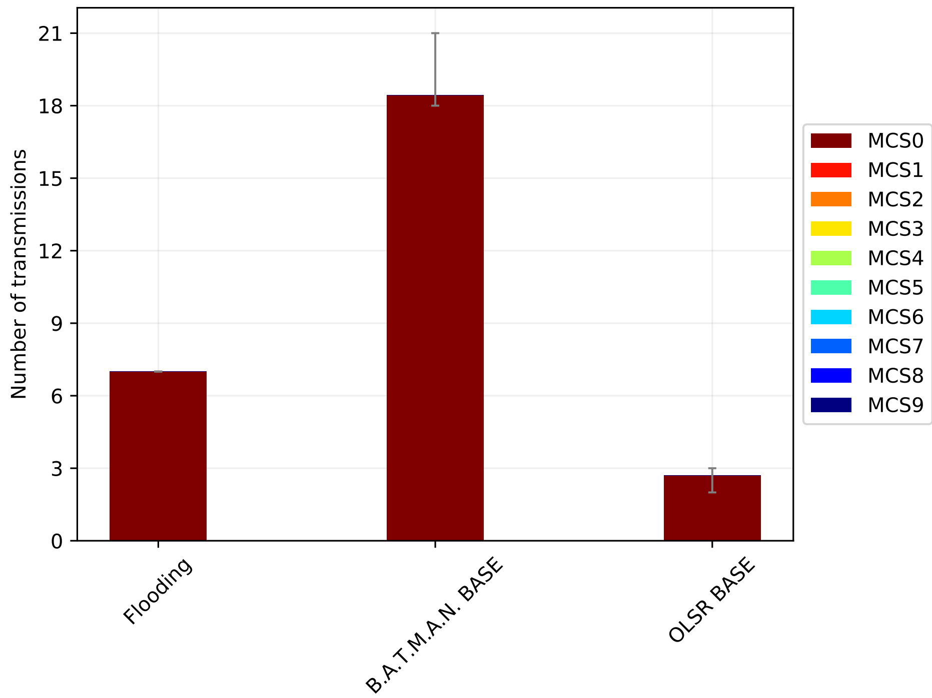



Figure 4.3: Number of transmissions for the demonstration network for state of the art protocols

B.A.T.M.A.N. and OLSR refer to the general protocols respectively, i.e. the group of all implemented versions. The labels *B.A.T.M.A.N. BASE* and *OLSR BASE*, referring to the base line versions, indicate the usage of the basic transmission rate, realized by using the lowest MCS0. It shows, that flooding uses a constant number of transmissions per network-wide broadcast, to reach all seven nodes. Each of the seven nodes locally rebroadcasts the received transmission upon the first reception. *B.A.T.M.A.N. BASE* clearly exhibits the similarity to flooding, in that it requires nearly three times the number of transmissions. At the same time, the broadcast avoidance strategy can be observed, since *B.A.T.M.A.N. BASE* mostly skips the transmission of the seventh node $D$, resulting in $18$ transmissions overall for the six remaining nodes. The only case, that all seven nodes will transmit is when $D$ itself is the source node of the broadcast. In contrast to that, OLSR successfully manages to serve all nodes with $A$ and $B$, clearly demonstrating the reduction of transmitting nodes. Only for the case, that node $D$ is the broadcast source, three nodes have to transmit.

The second simulation covers a wider range of $250$ random networks, with the default amount of $15$ nodes in a variable network area with $l = [100, 200, 300, 400]$ m, repeatedly conducted for $10$ batches. The change in the network size with a constant number of nodes results in a larger node distance on average and therefore overall worse link conditions.
Figure 4.5 shows the overview for the results. It can be clearly seen, that all pro-

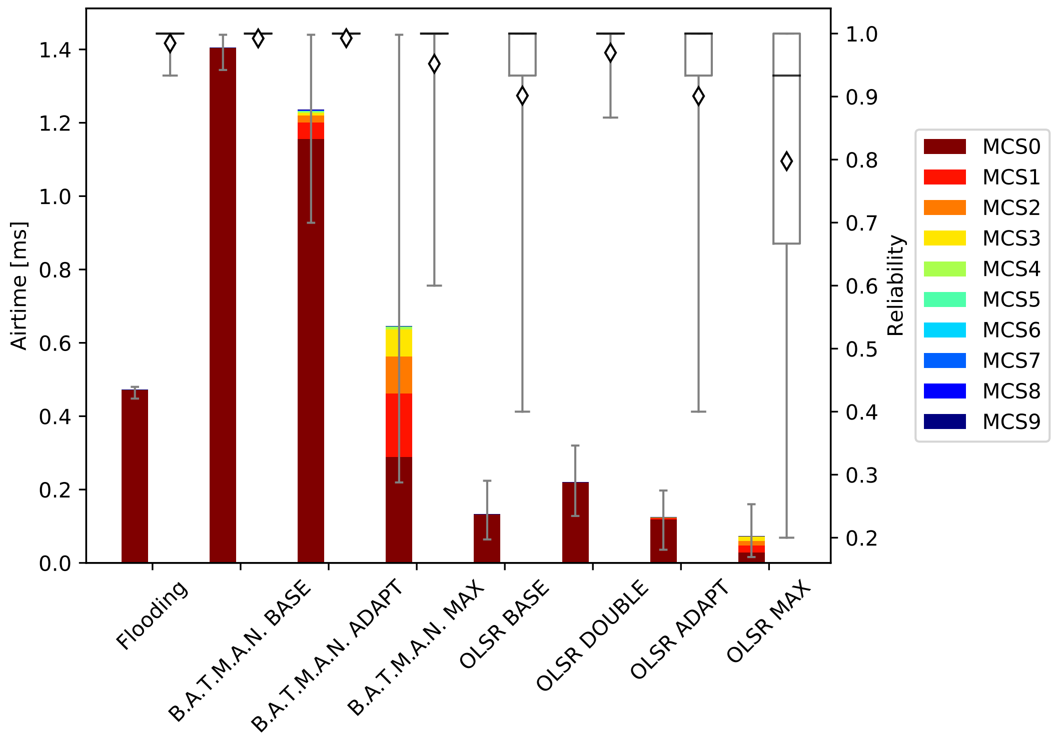



Figure 4.4: Detailed performance of state of the art protocols and enhancements for $l = 200$ m

tocols share the same general tendency, that the performance deteriorates with an increasing $l$. In the smallest network, nearly all protocols exhibit a very high reliability around $95\%$, however with completely different airtime consumptions between $0.05$ and more than $1.4$ ms. B.A.T.M.A.N. consumes in general the most airtime, while OLSR requires the least. In the case of OLSR it trades the airtime savings for the worst reliability results of all protocols.

It is noticeable, that OLSR offers a different behavior for larger $l$, than B.A.T.M.A.N. and flooding. OLSR is able to cover a larger set of nodes in a smaller network, and therefore requires less transmissions. If the network size grows, it has to attempt more transmissions, as the coverage of a single MPR does not include as many nodes as before. However, for flooding and B.A.T.M.A.N., the number of transmissions is already fixed and can only decrease. This happens, since both protocols fail to reach all nodes in the network, which therefore will not transmit, decreasing the overall airtime.

Both airtime and reliability results are displayed in detail for $l = 200$ m in figure 4.4. For each protocol, the bar plot displays the cumulative airtime required by the transmission. It is divided according to the airtime spent on each individual MCS. However the proportions do not represent the actual transmission amounts on the respective rate, as they are multiplied by the according data rate. Additionally, the $5$th and $95$th percentile of the total airtime are indicated. The reliability is added in form of a box

plot to the right of each airtime bar. The box plot demonstrates the distribution of the overall achieved reliabilities by each simulation run. For each protocol, the median is marked with a black line, while the mean reliability is marked with the diamond. Similar to the airtime, the whiskers represent the $5$th and $95$th percentile.
The simulation is extended by the addition of further implementations of both B.A.T.M.A.N. and OLSR to explore possible performance enhancements. Additionally to the previous basic version, a version utilizing the highest available rate connecting the whole network was added to each protocol, named *B.A.T.M.A.N. MAX* and *OLSR MAX* respectively, as well as two rate adaptive versions *B.A.T.M.A.N. ADAPT* and *OLSR ADAPT*. The rate adaptive version includes the regular protocol mechanics, with the same neighborhood information as decision base. The difference to the basic version is, that for the same set of neighbors as with MCS0, the transmission rate is adaptive. Only if all of these neighbors support an error-free transmission on a higher rate, this rate is selected. Lastly, for OLSR, another version *OLSR DOUBLE* was added, which employs a different MPR selection algorithm, providing double MPR coverage for each 2-hop neighbor, if possible.

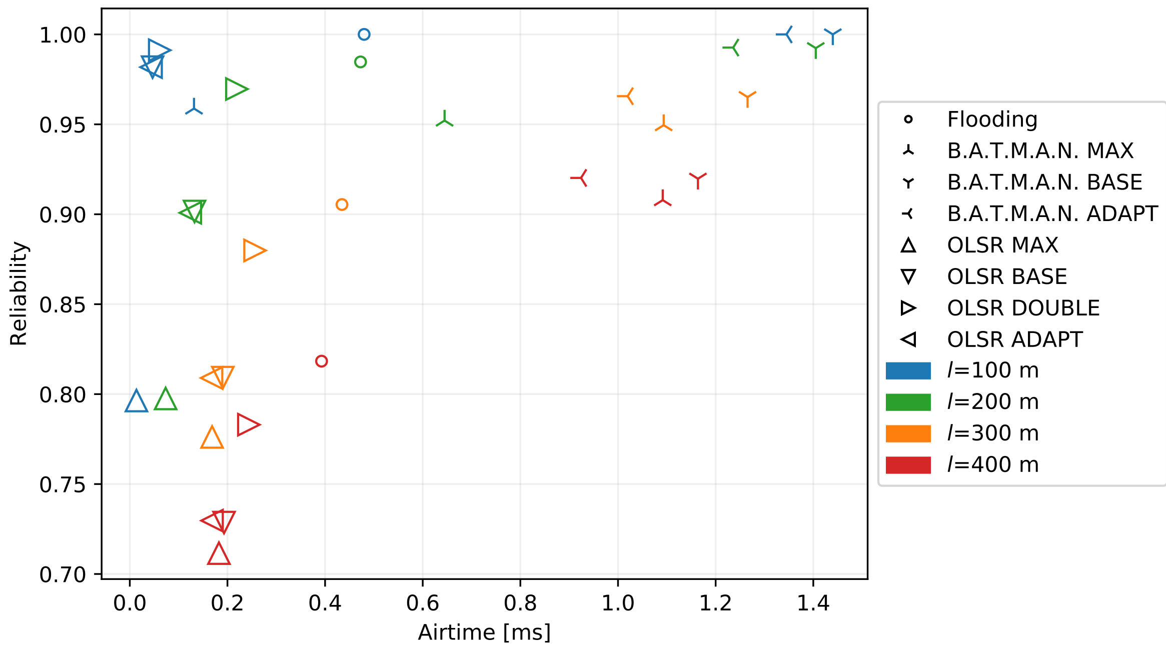



Figure 4.5: Performance overview for state of the art protocols for different $l$

## 4.1 Flooding

Classical flooding offers a good balance between reliability and the number of transmissions for the fixed rate operation on the lowest available data rate. For a case as in 4.4, with $l = 200$ m, flooding achieves a solid reliability. Given, that the resources

for all these transmissions are available, the simplicity makes it a suitable solution. However, when optimizing for airtime, the blind forwarding approach obviously creates a lot of redundancy, which unfavorably weighs in more, when using a low MCS. Each transmission itself consumes a lot of airtime, compared to higher MCS. In that case, the usage of low rates is only efficient in less dense networks, where the possible transmission rates for each link do not exceed MCS0 and MCS1. The higher MCS would save on airtime, but it comes with lower success probabilities for link local transmissions. The lower MCS0 however, would offer reliable transmissions, with a slightly higher airtime and thus, perform better overall. Nevertheless, even then the balance is very fragile and can easily become inefficient. When the network is even more sparse, the error probabilities for the base rate increase and reliability can only be maintained by increasing the number of transmissions, which flooding does not support. This can be observed in figure 4.4, where the reliability already slightly decreases below $1.0$, as already errors on the lowest data rate occur on few links for $l = 200$ m. This would be identical for an overall worse average link quality, as shown in figure 4.5, with the reliability quickly declining for deteriorating link qualities. If the network is denser, transmitting on higher data rates would be as reliable, but significantly more efficient.
Overall, flooding achieves in general a good reliability for smaller networks, that support connectivity through higher MCS, but since it lacks flexibility, the airtime spent on the transmission is often either too high or too low to be considered efficient. For networks with worse link conditions however, flooding looses quickly in reliability. Nevertheless flooding nowadays is still considered the base line regarding the broadcast performance.

## 4.2 B.A.T.M.A.N.

In comparison to flooding, it clearly can be stated, that B.A.T.M.A.N. only offers a marginal improvement in efficient performance, if any. It can increase the reliability, but at the cost of nearly three times as much airtime, as shown in 4.5. B.A.T.M.A.N. consumes by far the most airtime overall. For the case, that the network supports and allows for higher transmission rates, the same disadvantages apply as for flooding. The fixed rate approach becomes quickly inefficient, since it does not provide means to make use of the improved conditions. Not only the rate is fixed, but also the number of transmissions per node, further increasing the airtime total. In contrast to flooding, B.A.T.M.A.N. implements a strategy to avoid unnecessary transmissions, if it is possible. This awareness is crucial towards reaching the optimal efficiency. However, B.A.T.M.A.N.'s conservative approach generally leads to an abundance in redundancy for most cases, with the worst possible network conditions being the exception. B.A.T.M.A.N. is able to compensate much lower link success probabilities, than most other protocols, by transmitting three times on the lowest rate. As depicted in figure 4.5, B.A.T.M.A.N. is able to maintain a reliability of more than $0.9$, while all other protocols clearly slide below $85\%$ for $l = 400$ m. However, with three transmissions, this holds true only to success rates greater or equal than $0.33$. In addition to the retransmission avoidance strategy, this is the second important feature, since already low loss rates on the lowest rate statistically require a second transmission to deliver reliably. All other approaches are not able, to guarantee a successful delivery on average.

As shown in figure 4.4, the airtime performance could be potentially increased, with the usage of enhanced data rates. The most gain can be achieved, if the protocol has prior knowledge, which maximum rate still provides full connectivity. On average, the airtime could be cut in half, comparing *B.A.T.M.A.N. BASE* and *B.A.T.M.A.N. MAX*. However this can be expensive considering the constant probing for a dynamic network and it will also significantly impair the reliability. With no link quality awareness, the usage of higher MCSs increases the probability of errors on the link, which subsequently could separate a part of the network. For maintaining a high reliability, the rate adaptive approach of *B.A.T.M.A.N. ADAPT* can be used to at least slightly reduce the required airtime. As the rate is only increased, if errors are not possible, the protocol manages to preserve its original reliability, while decreasing the airtime. Although it gains an advantage in airtime, this strategy does not significantly improve the performance. Since it considers the exact same neighborhood, the reliability of all of these nodes supporting the error-free transmission on a higher rate is relatively small.
In conclusion it can be assessed, that B.A.T.M.A.N. has an advantage in deteriorating network conditions, through the introduced redundancy and the usage of low data rates. The additionally invested airtime generates a clear gain in reliability, compared to all other protocols, as demonstrated in figure 4.5. It is however conceptually not suited for the usage of significantly increased transmission rates. Higher MCS will reduce the airtime, but without a link quality awareness, this comes at the cost of significantly lower reliability. This is counteracted by the three total transmissions, but as previously explained, only to a certain extend. A slight increase of one or two data rates is nonetheless possible and could be deployed, in a corresponding scenario. To attain a general performance estimation for comparison of all scenarios, both versions *B.A.T.M.A.N. BASE* and *B.A.T.M.A.N. MAX*, will be carried into the final evaluation. Since the broadcast rate can vary somewhere in between both utilized rates, the actual performance is somewhere in between.

In this work, B.A.T.M.A.N. is implemented without the specified $5$ ms delay in between the retransmissions. The subsequent transmission can be initiated in the transmission slot directly after the first transmission. This has no impact on the reliability, nor the required airtime.

## 4.3 OLSR

OLSR clearly offers a significant improvement in airtime. Compared to flooding, as well as compared to B.A.T.M.A.N., it manages to reduce the total number of transmissions and therefore the overall consumed airtime. Figure 4.5 shows, that it is nearly the only candidate, that manages an airtime below $0.3$ ms. Furthermore, this holds true for all protocol variants and network sizes. However, it operates on a fixed rate as well, without any link quality awareness. This inflexibility causes a much more relevant problem in OLSR, than in flooding and B.A.T.M.A.N.. As OLSR disseminates information exclusively via a set of predetermined MPR nodes, the eventually utilized network becomes smaller, considering the utilized links. Thus, it is more dependant on a successful transmission on the links between the MPRs. A single link failure between two relay nodes could impact a much larger set of nodes, connected to that specific MPR. Without the link quality awareness, OLSR relies on the previously discovered links to be reliable, without an alternative in case of failure. Therefore the increase in saved airtime comes at the price of a decreased reliability. Figure 4.4 shows the disadvantage in reliability for the mean values, as well as especially for the deviation. This affects sparser networks more than dense networks, as the sparsity increases the average node distance and therefore also the link quality including the lowest transmission rates. However, in dense networks a decrease in the efficiency is caused by the usage of the lowest data rate, as the short links reliably support higher MCS.

The implemented multi-rate OLSR enhancements in figure 4.4 show similar results to the previously discussed B.A.T.M.A.N. enhancements. The ability of *OLSR MAX* and *OLSR ADAPT* to transmit with a higher MCS can reduce the airtime consumption, but in the case of *OLSR MAX* drastically decreases the reliability in return. It is noticeable, that the dependency on the links between the different MPRs is causing a distinct decrease in reliability, paired with an immense uncertainty. The range of achieved reliabilities can vary heavily, as the deciding error can occur comparably early in the MPR transmission chain, cutting off the whole remainder. This makes the OLSR core concept of Multipoint Relaying, as originally specified and without link quality awareness, undesirable in error-prone environments. However, this can be partly countered with a scheme, as implemented in *OLSR DOUBLE*. It provides a second opportunity to reach any node, to increase chances of a successful reception. For the usage of the lowest rate, it achieves the by far the best reliability among all OLSR implementations. On the other hand and also as expected, the double MPR coverage increases the airtime by nearly $100\%$. Eventually *OLSR DOUBLE* can not recover from losses on MCS0 and it still achieves less than $0.8$ reliability and is therefore worse than B.A.T.M.A.N. and flooding. Even with the second connection option it relies too much on a successful transmission to each MPR. Therefore it can not overcome the inflexibility of OLSR to significantly increase the reliability.

In this work, OLSR is omitting the *WILLINGNESS*, as discussed in section 2.3.4. However this comes with a positive effect on the performance. In that, if a shorter or more efficient forwarding path would become available through a node with a low *WILLINGNESS*, than the airtime would improve for the usage in the simulator. The actual implementation would be in contrast to that not allowed to utilize this path and would be forced to select an alternative, that requires more airtime. For the case, that the alternative offers only the lowest data rate, combined with an increased error rate, the reliability would additionally deteriorate.

# 5 Rate Aware Information Dissemination with Extra Reliability (RAIDER)

## 5.1 Design Criteria

> "A perfect routing protocol should be able to provide a route between any two nodes in the network (reachability). Packets that travel along these routes should arrive without being dropped (packet loss), timely enough for interactive applications (delay) and fast enough for high-bandwidth transfers (throughput)." [63]

The main objective of any routing protocol is to connect any desired nodes, by determining a transmission path, that enables reliable data transport. Projected on the case for this work, broadcast delivery in Wireless Mesh Networks, the goal is to provide complete reachability throughout the network and to achieve a low airtime consumption. This ensures, that the data is transported as efficient as possible to the entirety of nodes in the network, while also reducing the utilized bandwidth and the induced delay. To find the solution to this routing problem, the available concepts were described in section 2.3. On account of the usage in WMNs, the protocol should be **reactive**, in the scope of the topology knowledge of each node. The constant precalculation for any change in link quality or node quantity requires an enormous effort in processing and would be no match for the dynamic nature of a WMN. The signalling overhead is typically increased in mesh networks, since there is no central coordination. Hence, it should not be further increased, if possible. Furthermore, a reactive nature would result in always up-to-date routing information, to make the most effective decision possible. Therefore a link state solution would be impractical. As the topology can change very quickly and routes potentially span a large portion of the network, nodes should consider only their extended neighborhood, instead of the complete network. The trade-off here lies in the fact, that routing with more prior information can be arranged more efficient. However at the same time the overhead caused by the required network information and precision of that information deteriorate. A good compromise is offered by using only the **2-hop neighborhood** of every node for each routing decision. This requires every node to locally broadcast its neighborhood information periodically, which is lightweight and simple to realize. The reduced overhead additionally ensures scalability, which is a definite requirement for routing in mesh networks. A 2-hop neighborhood as information base complies with the findings of section 2.3.5. The current related work on multi-rate approaches relies on assumptions in the network knowledge, that are impractical and can hardly be realized as required. This averts a deployment in any arbitrary scenario, since the dependencies are not generally fulfilled. Therefore the goal is to develop a **distributed** universal **multi-rate** scheme with limited knowledge, with as little external dependencies as possible. The 2-hop neighborhood is also compatible with lower-layer network protocols, such as ARP or DHCP, since link local transmissions do not require higher layer services. Furthermore, both protocols require primarily a **reliable delivery** and secondly a **low airtime**, since their service must not unnecessarily interfere with higher layer and network operation. Therefore these are the desired priorities for an improved broadcast scheme as well. In contrast to the disadvantages in the usage of fixed transmission rate approaches such as B.A.T.M.A.N. and OLSR, as analyzed in chapter 4, the algorithm should include a **link quality awareness**, to react and adjust the transmission rate according to the link qualities. The last design choice, that needs to be considered and discussed is the use of feedback in any form. Dedicated feedback transmissions are clearly impractical for their transmission overhead. A viable option of feedback is overhearing the next senders transmissions as a feedback, as implemented in [66]. However in a multi-rate environment, the next sender could select an increased data rate for the forwarding, resulting in the inability to overhear the feedback reliably. Therefore feedback is not included.

## 5.2 Protocol Design

To fulfill all requirements and design criteria, a newly created broadcast scheme must offer a solution to three main challenges. These three challenges divide the complex routing problem in different steps, that need to be revised according to the prerequisites.

1. **Targeted Nodes**
   Which nodes should be considered as recipients for a node's local broadcast transmission?

2. **Rate Selection**
   Which data rate is best to reach all desired recipients under consideration of the optimization objective of the protocol?

3. **Forwarder Selection**
   Which recipients of a node's local broadcast should participate in forwarding the data?

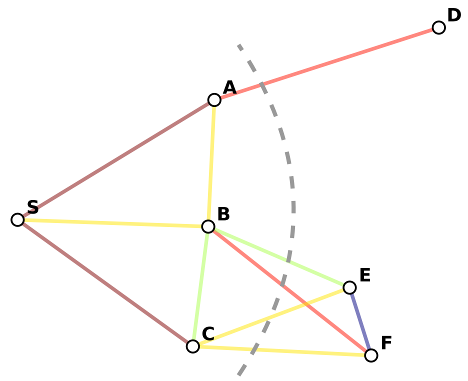
The first challenge is the foundation to match the predefined requirements and performance goals, as both remaining objectives depend on the selection of step one. The protocol has to be aware of the direct surroundings of each transmitting node. If it can determine, which surrounding nodes it should serve, than it can also decide, whether a transmission is necessary at all. As pointed out in section

4.2, B.A.T.M.A.N. offers a broadcast avoidance mechanism. This demonstrates the fact, that in certain circumstances a blind relaying transmission can be prevented, with little prior knowledge. The broadcast protocol must be able to recognize, if a transmission is necessary and if so, to what extend. To do so, it depends on the network knowledge, that is provided to each node. The more knowledge, the better can a node decide, if it should transmit and which nodes should be targeted with this transmission.
If a set of target nodes has been identified, the protocol can assess, which transmission rate is the most suitable to maximize the performance in terms of airtime, while at the same time maintaining sufficient reliability. Each use-case might require a different realization of this balance. The selection of the rate is straight forward, for the case that the PERs for each link and each data rate are known to the sending node. It again depends on the network knowledge, but can also change, if a node is allowed to transmit multiple times. This may be the case, if the performance is not optimized for the overall consumed airtime, but for instance for the latency performance of prioritized nodes.
The last challenge directly arises from the available network knowledge. It ensures, that the broadcast will cover every single node in the network. If no network knowledge is taken into consideration, each node has to assume, that it has to serve all neighbor nodes itself and consequently will transmit. However, the forwarding decision can also be calculated in advance for nodes that have a knowledge exceeding the own direct neighborhood. A previous node can determine, if its direct neighbors cover the same set of nodes and subsequently assign only one of them with the forwarding. This prevents the second node from transmitting redundantly. The forwarding problem also becomes important, if the protocols use a different metric, than the hop count, since it might make use of rerouting to choose the path with the best metric.

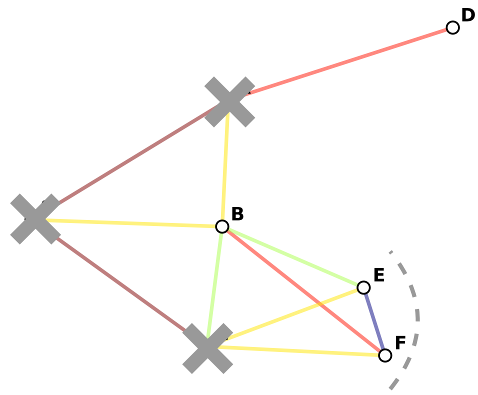
The following subsections describe in detail, how Rate Aware Information Dissemination with Extra Reliability (RAIDER), the approach presented in this work, solves these challenges.

### 5.2.1 Targeted Nodes

Generally stated, the target node selection in RAIDER is based on the assumption, that every node should serve its complete direct neighborhood. As already deployed in flooding, B.A.T.M.A.N. and OLSR, this ensures complete reachability. However, to improve the efficiency, this is extended by a redundancy avoidance strategy, similar to B.A.T.M.A.N.. Since RAIDER also takes the topology knowledge, i.e. the 2-hop neighborhood, into consideration, the avoidance can even be elevated. In B.A.T.M.A.N., only the transmission source and the previous sending node can be excluded from the set of targets, since one can be sure, that if they transmitted the data, they also




(a) Example sender $S$




(b) Example sender $B$

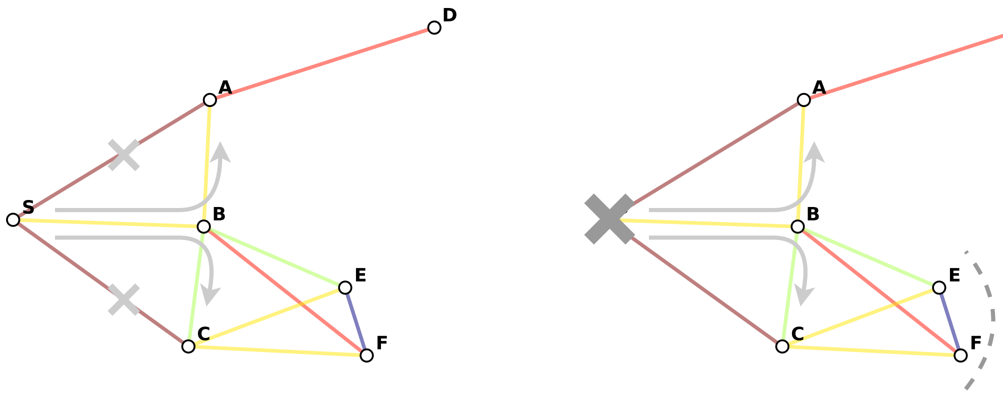
Figure 5.1: Target node selection step for RAIDER applied to the demonstration network

must already possess it. Applied to the presumption, that every node already covers its own direct neighbors, this results in a whole set of nodes, that can be ruled out as potential targets. Therefore RAIDER considers all of its 1-hop neighbors as a target, that at the same time are strict 2-hop neighbors from the previous sender. This excludes all 1-hop neighbors from the previous sending node, as well as the sending node itself.
The only exception to this are so-called *forwarding assignments*, that can be determined by previous nodes. A further explanation how they are defined is given in the next section. These assignments have to be strictly obeyed, as they are a fundamental part of the protocol mechanics. It may contain the instruction, that a direct neighbor has to be served, although it is also in the 1-hop neighborhood of the previous sender.

The target selection applied to the demonstration network in figure 3.3 is shown in figure 5.1. Example sender $S$ would begin a network-wide broadcast and select all direct neighbor nodes $A$, $B$ and $C$ as targets for its transmission. An example ensuing transmission by node $B$, would initially see nodes $E$ and $F$ as target nodes, since $S$, $A$ and $C$ are considered already covered.

### 5.2.2 Rate Selection

If the set of target nodes was determined, the next step is to set the best suitable data rate for the transmission. From the presumed 2-hop neighborhood discovery, the PER for each MCS is known for all targets. The challenge is to pick the most efficient rate from this limited set. The sender should only lower the rate, when there is no faster path to a recipient via nearby nodes. In the scope of the requirements,




(a) Forwarding assignments by $S$

(b) Effect on the target nodes for $B$

Figure 5.2: Forwarding assignment procedure for RAIDER applied to the demonstration network

these nearby nodes regularly include the 2-hop neighborhood. However, this knowledge base only includes the links in between the direct neighbors and not the links in between 2-hop neighbors. Between two direct neighbors, the optimal but undetected route might include two 2-hop neighbors and the connection between them. Therefore the protocol does assume, that it can not reliably make the most efficient routing decision in the whole 2-hop scope without this information. Instead, the routing decision is based on the direct neighborhood, including links in between the neighbor nodes. Following this principle RAIDER calculates the best path to each target node, utilizing a weighted dijkstra algorithm. To optimize for airtime, the dijkstra algorithm uses the Expected Transmission Time metric according to equation 5.1, to consider all possible rates $r \in R$ for each hop. The result is the path with the lowest sum of ETTs for all edges in the path. If the algorithm proposes a path diverging from the direct link to the target node, a forwarding assignment is created. It contains the calculated path and is transferred to the next hop via the packet header. Internally, the protocol temporarily stores the forwarding assignments and the determined optimal rates for each step of the path.

$$ETT_{i,j} = \min_{r \in R} \left( \frac{1}{1 - \epsilon_{i,j}^{(r)}} * \frac{N}{r} \right) \tag{5.1}$$

So far, the dijkstra path finding solely optimizes for airtime and could potentially favor significantly more error-prone links with higher rates. To suppress this behavior, RAIDER introduces a link local *reliability threshold* $p_{tr}$, which has to be matched for the dijkstra algorithm to consider the link at a specific rate in the first place. If $p_{i,j}^{(r)}$ is less than the threshold, a reduced data rate has to be used on that specific link. Setting this link quality threshold to $1.0$ would result in the selection of only those

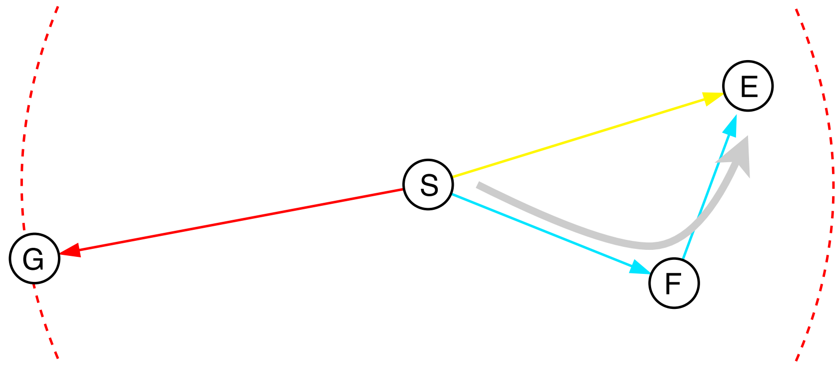



Figure 5.3: Example scenario in RAIDER for an obsolete forwarding assignment

links, that have a guaranteed successful delivery when sending on the selected data rate. Selecting $p_{tr} = 0.0$ on the other hand will not have any influence, since every link matches this criterion. The only exception to the threshold is the lowest MCS0, which has to be selected, since the threshold can not be matched by further decreasing the transmission rate. This introduces the option to additionally optimize considering the selected level of reliability.

This path finding step could optionally also include a penalty for any additional hop, that comes with the selection of a diverging path. For dynamic environments, this increases the risk of an unforeseen network change, while the transmission occurs along the path. As a consequence, this could interrupt the path and prevent a successful transmission. Therefore a penalty would constitute an incentive to route over shorter paths.

To take the forwarding assignments into account for the rate selection, the protocol must consider a different set of nodes, than the original targets. For each target, that had a forwarding assignment created, the target itself has not to be reached, but instead the first hop in the assignment. Therefore the adjusted targets contain the original targets, which are served directly, and the first hop in the assignment for each original target, that is served through a forwarding assignment. For this set of adjusted targets, RAIDER selects the minimum stored data rate, determined earlier as optimal for that link by the dijkstra path finding.

If the final transmission rate was determined, all previously created forwarding assignments are compared to this rate. All assignments contain only the optimal path and therefore the optimal data rates to each original target node, regardless of the other target nodes. As a consequence, another target node could force the selection of a significantly lower rate, potentially rendering assignments obsolete.

Let $E$ be an example target node, that is reliably connected to the source on MCS3, as depicted in figure 5.3. However, a more efficient forwarding path is determined and $E$ will be served via $F$, with MCS6 as the determined best rate towards $F$. Let $G$ be another target node, which is considered isolated in the scope of RAIDER, so that it must be served directly. If $G$ can only be served on MCS1, the selection of this maximum rate is forced, since $G$ must be served. Eventually, although a faster path

was available for $E$ via $F$, the in the end selected transmission rate does already cover $E$. This makes the forwarding assignment obsolete and it must be removed, since it would force an unnecessary transmission from $F$ to $E$.
If the decision has been made and an MCS was selected, it is checked if the protocol should also initiate duplicate transmissions for the current packet. As demonstrated by B.A.T.M.A.N., repetitive transmissions can be beneficial, if the links support only MCS0 with a success probability smaller than $1.0$. The duplicates then provide redundancy to overcome the error probabilities. Since the lowest, but still detected success rate is $0.1$, the protocol allows for maximum $10$ retransmissions. Furthermore, they are only enabled, if the node is sending on the lowest rate, MCS0, as otherwise the rate could simply be lowered. Therefore only retransmissions on MCS0 are allowed. The protocol determines the number of retransmissions from the maximum of all error probabilities towards all adjusted target nodes $j$ for transmitting on the lowest data rate, according to $\max(10, \lceil \max_j(1/\epsilon_{i,j}^{(r_{min})}) \rceil)$.

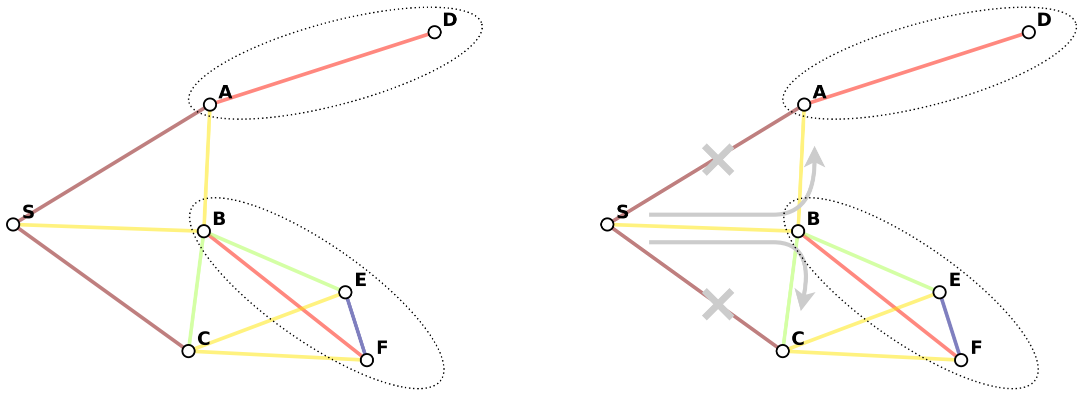
For the first transmission in the demonstration network, node $S$ checks the paths to its target nodes $A$, $B$ and $C$. The direct links to $A$ and $C$ support only MCS0 with a data rate of $8$ Mbit/s, while the connection to $B$ supports MCS3 with a significantly increased data rate of $33$ Mbit/s.

$$ETT_{S,B} + ETT_{B,A} = \frac{N}{33\text{Mbit/s}} + \frac{N}{33\text{Mbit/s}} < ETT_{S,A} = \frac{N}{8\text{Mbit/s}} \tag{5.2}$$

$$ETT_{S,B} + ETT_{B,C} = \frac{N}{33\text{Mbit/s}} + \frac{N}{49\text{Mbit/s}} < ETT_{S,C} = \frac{N}{8\text{Mbit/s}} \tag{5.3}$$

According to the dijkstra optimization, RAIDER selects to reach $A$ through a connection via $B$. For node $C$, the path through $B$ is selected as well, since its second hop supports the even higher MCS4. Equations 5.2 and 5.3 demonstrate this optimization, for an example case with no errors for all rates, $\forall i, j, r : \epsilon_{i,j}^{(r)} = 0$, so that the maximum available rate is always the best choice. For the actual case with different error probabilities, the ETT might be minimal for a lower selected rate. Eventually, two forwarding assignments, $B \rightarrow A$ and $B \rightarrow C$, are required, as depicted in figure 5.2a. Therefore node $S$ appends the information to the header of the packet it will send to $B$, which thereupon knows to include nodes $A$ and $C$ in its target set for the relaying transmission, demonstrated in figure 5.2b. Until then, both nodes were already considered covered by the previous transmission and therefore excluded from the target set of $B$. As soon as all forwarding assignments are handled, a MCS for the transmission is set from the lowest rate of all adjusted targets. For the demonstration network, serving nodes $A$ and $C$ was delegated to node $B$, reducing the original target set including $A$, $B$ and $C$ to the now adjusted target set containing only $B$. Hence, $S$ picks the maximum efficient rate to reach $B$, which corresponds to MCS3.

### 5.2.3 Forwarding Nodes




(a) Forwarder selection for $S$

(b) Overview of complete protocol decisions by node $S$

Figure 5.4: Final step for RAIDER applied to the demonstration network

The last step in the RAIDER algorithm is the choice of the forwarding nodes. With the 2-hop neighborhood knowledge, each node already has a preliminary overview of the required next transmission step. However, it can not foresee unexpected influences from outside of its 2-hop neighborhood scope. Nevertheless, it can estimate from the current knowledge, which of the targeted nodes are from the sender's perspective required for forwarding. This saves nodes from unnecessarily transmitting, although covering no additional nodes. On the contrary this also ensures, that 2-hop neighbors, that are connected to only a single 1-hop neighbor, will be covered by the transmissions as well. This emphasis on these rather isolated notes could also be found in the first step of the OLSR MPR selection. Similar to that, RAIDER assigns such nodes as forwarders first, that definitely have to transmit. This includes mentioned connecting 1-hop neighbors as well as the first hop nodes in all forwarding assignments. At this point, the set of forwarders is extended by the node with most uncovered 2-hop neighbors, until there are no uncovered 2-hop neighbors left.

Applying this scheme to the demonstration network, node $S$ would firstly select node $A$ as a forwarder. Although $A$ is not expected to receive the transmission, since the direct link is much slower than the proposed forwarding path via $B$, node $A$ still has to be included in the set of forwarders. In the case, that a particular direct link is not chosen, only because it has an increased error rate for the best MCS, that respective MCS could independently still be selected for the transmission. Therefore, this would enable node $A$ to overhear the transmission successfully by an unlikely chance.

Eventually, after selecting node $A$ as a forwarder, $S$ also selects node $B$ as the first hop in a forwarding assignment, which already covers the complete 2-hop neighborhood and finishes the forwarder selection. The resulting forwarder selection is shown in figure 5.4a.

Finally, the set of forwarding nodes is added to the packet header. It thus includes the set of forwarders, as well as the forwarding assignments. This constitutes the complete protocol decision for source node $S$, as depicted in figure 5.4b, that needs to be communicated to the respective nodes. In a realistic implementation, this would translate into a list of node identifiers, e.g. a MAC or an IP address and a nested list of identifiers. The size of the packet header would scale with the number of nodes in the 2-hop neighborhood scope.

# 6 Simulation Results

In the following chapter, the performance for the RAIDER protocol is examined. Therefore it will be compared to the determined upper performance bound, as well as the state of the art protocols discussed in chapter 4. Table 6.1 gives an overview over the respective protocols, how they are referred to in this chapter and the implementation details for the protocol mechanics.

| Protocol | Implementation details |
|---|---|
| Flooding | Unmodified classic flooding |
| B.A.T.M.A.N. BASE | Basic B.A.T.M.A.N. implementation, usage of MCS0 |
| B.A.T.M.A.N. MAX | Modified B.A.T.M.A.N. implementation, usage of the highest connecting MCS |
| OLSR BASE | Basic OLSR implementation, usage of MCS0 |
| OLSR MAX | Modified OLSR implementation, usage of the highest connecting MCS |
| RAIDER STD | Basic RAIDER implementation |
| RAIDER $l = 0.75$ | RAIDER implementation, $p_{tr} = 0.75$ |
| RAIDER $l = 0.9$ | RAIDER implementation, $p_{tr} = 0.9$ |
| RAIDER $l = 1.0$ | RAIDER implementation, $p_{tr} = 1.0$ |
| RAIDER RETX | RAIDER implementation, $p_{tr} = 1.0$, retransmissions on MCS0 enabled |

Table 6.1: Overview of compared protocols with implementation details

## 6.1 RAIDER Performance

The first evaluation attempts to assess the performance of the different possible RAIDER configurations. With the link reliability threshold $p_{tr}$ and the option to allow for additional retransmissions, the protocol allows for customizations to adapt to different requirements and environments. In this regard, $p_{tr}$ is adjusted to match the overall network properties, concerning poor link error rates. For the case, that the application expects a guaranteed delivery under all circumstances, the protocol offers an option to not rely on the probabilistic chance of successful transmissions. If $p_{tr}$ is increased, that consequently raises the reliability of the link local transmissions, but will on the other hand increase airtime by selecting lower rates. Inversely, in the case, that the application can tolerate a small amount of losses to optimize the airtime, the threshold can be lowered. A deployed network coding scheme could, depending on its configuration, compensate a certain extend of losses and allow a narrow focus on the airtime optimization. For instance the usage of RLNC would be a suitable choice. This however applies only to use cases, that do use continuous transmissions, that can be combined for coding, before the information expires.

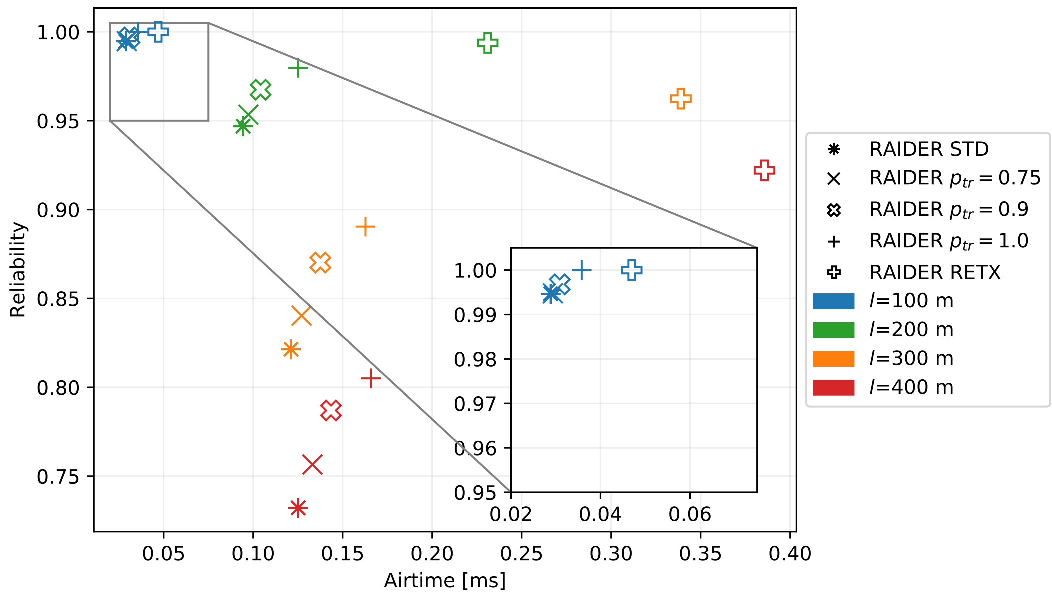



Figure 6.1: Performance overview for RAIDER implementations for different $l$

Eventually, the four different threshold configurations $p_{tr} = [0, 0.75, 0.9, 1.0]$ are compared. A threshold value of $0$ corresponds to no regulation at all, while a value of $1$ requires links to have a PER of $0$.
The second tested customization is the operation with enabled retransmissions per node. These multiple attempts on transmitting a single packet to the neighbor nodes constitute a conservative approach, that distinctly values reliability over consumed airtime. Thus, this configuration has the link reliability threshold set at $1.0$. Retransmissions are only allowed in conjunction with the usage of the lowest transmission rate. Complying with the ETT optimization towards all target nodes, RAIDER foregoes retransmissions on a higher MCS and selects a lower transmission rate instead. This ensures, that each neighbor node, that is targeted by the transmission is expectedly served as fast as possible.

The simulation scenario to assess the different RAIDER configurations includes four network configurations of the different area side lengths $l = [100, 200, 300, 400]$ m. This corresponds, with a static number of nodes in the network, to an increasingly sparse network with on average larger distances between the nodes. Therefore this scenario simulates with increasing $l$ overall deteriorating link conditions, i.e. lower

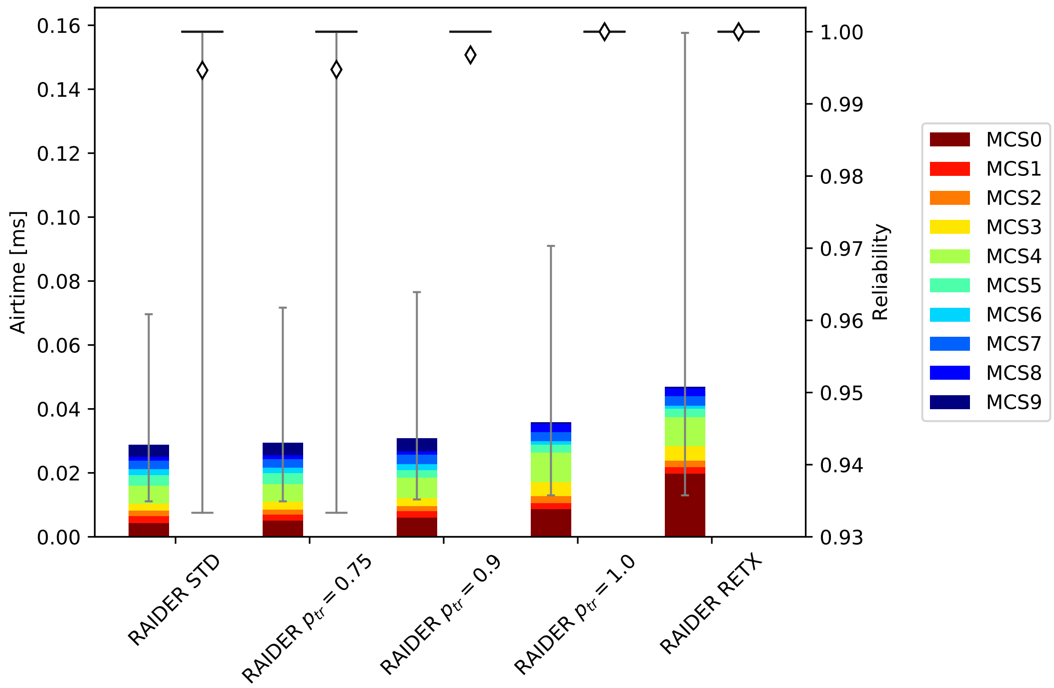



Figure 6.2: Detailed performance for RAIDER variations for $l = 100$ m

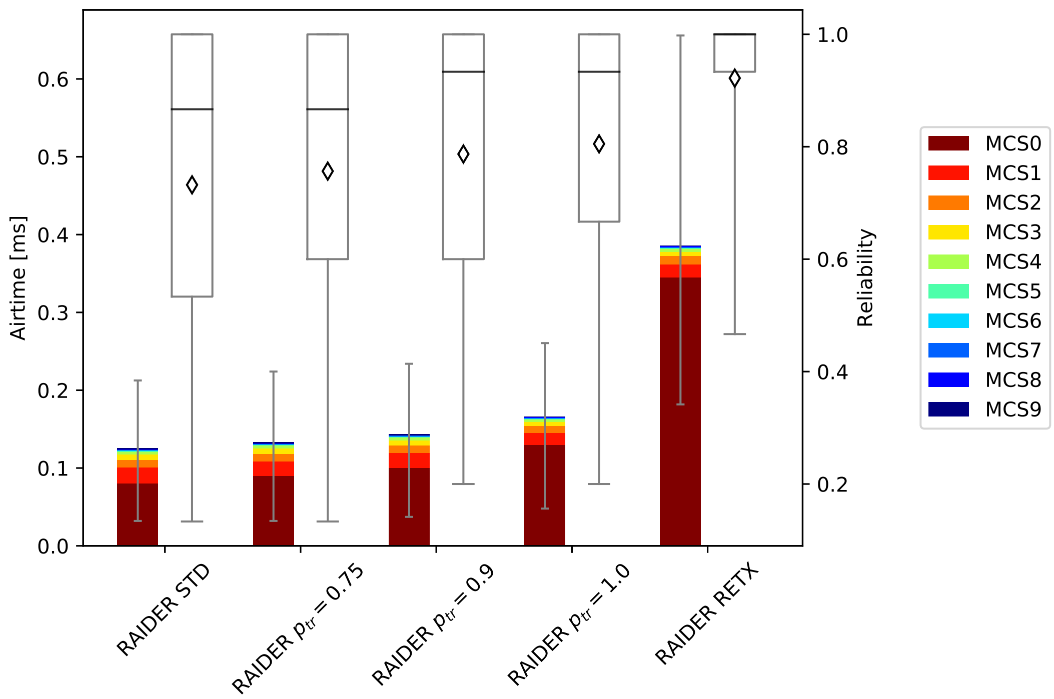



Figure 6.3: Detailed performance for RAIDER variations for $l = 400$ m

MCSs and higher PERs. Figure 6.1 gives an overview over the general RAIDER performance in reliability and airtime. It strikingly shows the decreasing performance in both reliability and airtime, if the network conditions deteriorate. Generally, the different link error threshold variations perform relatively similar, with little deviations compared to the configuration with added retransmissions. The difference between both groups increases with the decreasing average link quality. If the protocol is forced to transmit on MCS0, it can not compensate for errors any longer without retransmissions. Additionally the compensation must theoretically include at least $\lceil \max_j(1/\epsilon_{i,j}) \rceil$ transmissions, considering all targeted nodes $j$. This number of retransmissions quickly increases the airtime considerably, as seen in the performance overview. For a small $l$ of $100$ m, the difference is marginal, compared to the $0.2$ ms for $l = 400$ m. This equals an increase of more than $100\%$ compared to RAIDER implementations without retransmissions. It is also justifying the limitation to maximum ten retransmissions. With the minimum detected link success rate of $0.1$, the expected amount of successfully received packets is $1$ packet. Any otherwise potentially invested surplus in airtime would not be in proportion to the small increase in the tolerated error percentage. In return, the extended airtime effort prevents the reliability from dropping, as figure 6.3 shows a clear difference in the average reliability. RAIDER versions without the ability to retransmit, exhibit a gap in reliability of at least $0.1$. Eventually, the maximum 10 retransmission can not assure an unconditional reliability on average, as it drops well below $95\%$ for $l = 400$ m. *RAIDER RETX* can not completely prevent errors on individual links, which might affect or disconnect a portion of the network nodes, accounting for the large span of fliers. However, this holds true for all other variants as well. To differentiate between the link error threshold configurations in detail, the reliability and airtime are plotted in figures 6.2 and 6.3 for a small and a large network area respectively. The accumulated airtime shows an increase in the usage of the lowest data rate for worse link qualities, as RAIDER manages to adapt dynamically. Naturally it also exhibits a wider distribution of the achieved reliabilities, compared to better network conditions, i.e. smaller $l$. For $l = 100$ m, all implementations achieve a median reliability of $1$, whereas only the lowest two $p_{tr}$ configurations exhibit significant deviations. As a consequence, the average reliability exceeds $99\%$ for $l = 100$ m. All network sizes show the trade-off between an increased airtime demand to achieve a better reliability performance for different threshold values. This furthermore includes a preference for the usage of lower rates for increased $p_{tr}$, to adjust to the link error rates. The detail plots confirm this rate usage, as well as the increased reliability for both $l = 100$ m and $400$ m. In contrast to that, the deviation regarding the span of fliers does not show the same improvement, as these errors mostly are due to transmission failures on lower rates. That can only be corrected with the enabled retransmissions and not by a different selection of $p_{tr}$.

To summarize, the suitable configuration regarding the link reliability threshold $p_{tr}$ and the ability to retransmit depends on the network conditions and the targeted

application. Networks with predominantly low quality links rely on retransmissions to achieve reliability. If this is not required by the application, the threshold $p_{tr}$ can be lowered to allow for a slightly lower airtime.
Further comparisons will include the configuration with enabled retransmissions and the two limits for the link threshold, to estimate the range of achievable reliabilites and airtimes.

## 6.2 Upper Performance Bound

The complete comparison of the RAIDER performance to the state of the art requires an additional estimation regarding the overall maximum achievable performance. Otherwise the achieved improvement could not be assessed in proportion to the overall possible performance. Therefore the two approaches of a monte carlo simulation, as well as simulated annealing, as explained in section 2.5, were used. Figure 6.4 displays the results of both simulations, compared for the targeted scenario of different average link qualities, i.e. as previously $l = [100, 200, 300, 400]$ m.

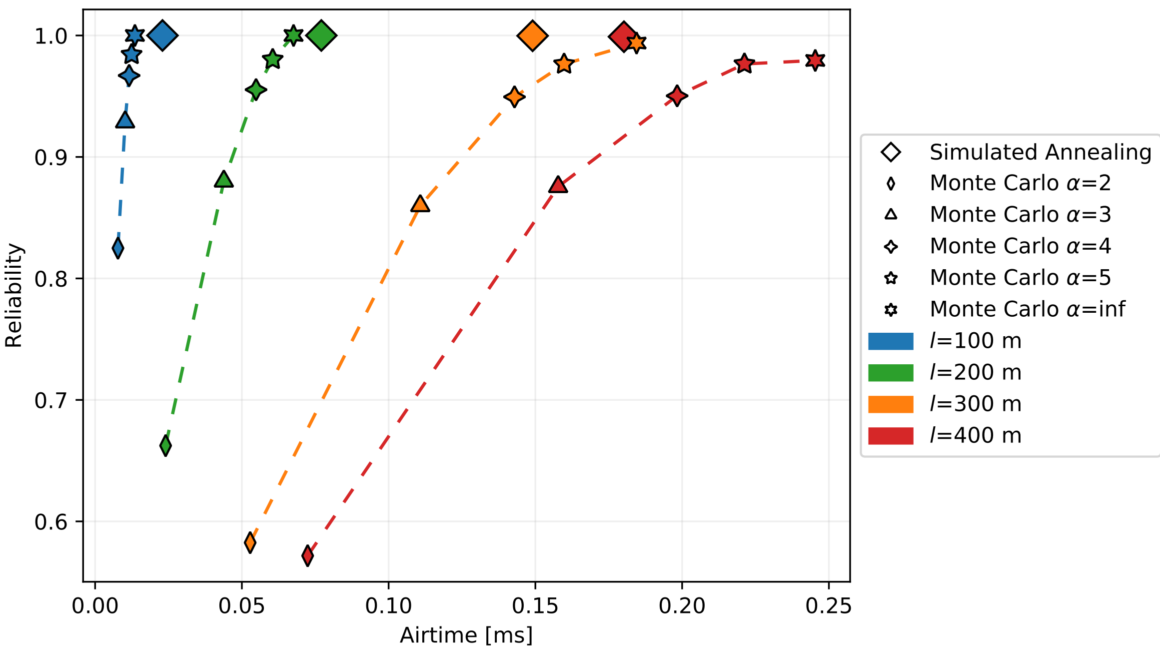



Figure 6.4: Upper performance determination for different $l$

### 6.2.1 Monte Carlo Simulation Results

The results of the monte carlo simulation for $\alpha = [2, 3, 4, 5, \text{inf}]$ show a clear tendency for each simulated network size. Regardless of the actual network size, the focus shifts with an increasing $\alpha$ from airtime to reliability, as expected according to the performance determination in equation 3.5. Evaluating the achieved statistics of each single simulation run, i.e. each random strategy, with a different performance function spreads the best accomplished results more and more apart. This showcases the overall trade-off in the performance evaluation between both contrary metrics. In the case of low losses for $l = 100$ m, reliability can be easily achieved and the trade-off is nearly linear. However this changes completely for worse link conditions with more errors and lower available rates, as every increase in reliability becomes more expensive regarding the invested airtime. For $l = 100$ m, improving the airtime by $0.070$ ms results in an increase in reliability of $0.18$. At the same time, the same airtime increment would barely increase the reliability at all for $l = 400$ m. It can be noted, that the minimum achieved airtime for $\alpha = 2$ grows with the network area from $0.08$ ms to $0.72$ ms. The larger the area is, the longer the average link between two nodes becomes, resulting in lower available data rates. This enlarges the average time for the transmission of a single packet. Similarly, the maximum possible reliability becomes worse with deteriorating network conditions. While $l = 100$ m allows for an absolute reliability of $1.0$, the achievable maximum for $l = 400$ m is $0.98$. This shows for large $l$, that even the best achievable result can not eliminate errors completely, due to the error susceptibility for the lowest MCS. The accomplished best performance for $\alpha = 5$ and $\inf$ exhibit a nearly identical reliability, while at the same time the former outperforms the latter, when evaluation the overall performance. Therefore it can be stated, that depending on the application, worse network conditions require the tolerance of errors, instead of demanding absolute reliability. If absolute reliability is demanded, although it can not possibly be achieved due to the link errors, the performance can not improve for a sufficiently large network. This holds true, if the maximum number of retransmissions is limited, to not allow compensation of every link error rate.

In general, for a larger simulated network area, the best performance decreases as expected. However, a distinct difference for the individual simulations can be observed. Between the results for $l = 200$ m and $l = 300$ m is a comparatively large gap. This is the point from which on absolute reliability can not be achieved anymore, due to the large number of links with errors for the lowest transmission rate. Apart from the natural performance decrease over the increase in $l$, this gap is further widened. This is due to the form of simulation, i.e. to select the tested strategy randomly regarding all parameters. With more links supporting partly or solely MCS0, the base rate will be selected more. Since the retransmissions are only enabled for this rate, this number now has to be randomly selected as well. This significantly increases the number of overall possible strategies with each node transmitting on MCS0. Hence a simulation with a fixed number of runs, as this work was limited to, can only cover a smaller percentage of all strategies. As a consequence, the probability to simulate a better strategy decreases, since most strategies can not approach the optimum anyway. This can be observed, as the result of the simulated annealing surpasses the monte carlo performance optimum for larger network areas. The expected result would be a relative improved airtime compared to the simulated annealing. The monte carlo simulation generally pursues a much more aggressive approach towards airtime optimization, as described in section 3.3.1. Therefore it should find, that the single optimal simulated strategy, exhibits the overall absolute possible airtime for a single broadcast attempt. If errors, i.e. a reduced reliability, can be tolerated, this bound represents the possible optimization. As the results diverge from this expectation, it must be assumed, that the executed $100000$ individual broadcast strategy simulations were not enough. However the resources and the scope of this work did not allow an improvement. Due to the increased number of overall possible strategies for $l = 300$ m and $400$ m, the probability of approaching this optimal bound with the simulated number of strategies was to low.

### 6.2.2 Simulated Annealing Results

The simulated annealing result is expected to fall behind the best performance obtained with a monte carlo simulation, as described in the previous section. It reflects the best achievable result for a reliable selection of modulation schemes. The best monte carlo simulation might have randomly succeeded against the error probability of the utilized links for this single transmission attempt. In contrast to that, the simulated annealing will always deliver an assured transmission, if the strategy was repeated.
The consideration, regarding the maximum achievable reliability in the previous section, applies to the simulated annealing as well. For deteriorating network and link conditions, errors have to be tolerated in the final performance assessment, justifying the selection of the different values for $\alpha$ in the 'cost' function. Only the selection of $\alpha = 5$ instead of $\inf$ allows the reliability to approach $1.0$ for $l = 300$ m and $400$ m. Especially for the simulated annealing, this is crucial. In the process of approaching the optimum step-by-step, a differentiation between two overall unreliable strategies is mandatory. A near-optimal strategy, with for instance only a single unserved node must be assessed as more valuable than a strategy covering only a few nodes, although both are not absolutely reliable. If this is not accounted for, the algorithm can not select the better performing strategy and therefore not approach the performance optimum.
The results show, that for $l = 300$ m and $400$ m, reliabilities of $0.9997$ and $0.9990$ respectively, can be achieved. Networks with a reduced area, as simulated with $l = 100$ m and $200$ m are able to be served with a reliability of $1.0$. The results represent the minimum airtime under the condition, that the maximum possible focus was on reliability.

In conclusion, the simulated annealing delivers an upper performance bound, that holds true irregardless of the error probabilities on the utilized links. Therefore it will be selected for the final performance evaluation and comparison. However the monte carlo results for smaller network areas offer valuable insights on the maximum possible performance, when gambling on the probabilistic error-free link deliveries. For error-tolerant applications, this indicates a potential further airtime improvement.

Simulated annealing was conducted for the scalability scenario as well. However, as there was no matching monte carlo simulation, the results are discussed later in section 6.3.2.

## 6.3 Overall Performance Comparison

Previous sections conducted the separate evaluation of the state of the art and the proposed RAIDER protocols, to determine the best available configuration. This section finally compares both to eventually evaluate the performance of RAIDER. In addition to the state of the art as a lower performance bound, the previous section also determined an upper performance bound, which will be included into the comparison.
To evaluate all protocols for their handling of different network conditions in terms of the average link quality, paired with the set of available data rates, the first scenario includes different network sizes. This again results, for a constant number of nodes, in a larger node distance with less available MCSs and for each transmission rate lower PERs. The second scenario represents the test under different network conditions in terms of the node density. With more nodes spread over the same area, not only the individual link qualities increase, but also the total amount of links. This adds routing possibilities and complexity and tests, if the protocols can handle an increased number of nodes in the network.
For both simulations, the same set of candidate protocols and configurations is used. The current state of the art is represented by flooding, variants of OLSR and of B.A.T.M.A.N., as described in table 6.1. Both OLSR and B.A.T.M.A.N. are included in a basic version, transmitting always with the fixed lowest possible rate and in a second version, utilizing only the highest available MCS, that provides complete connectivity between all nodes. As the protocols themselves do not select the data rates, this covers all possibilities and the performance can be estimated by both versions in conjunction. The simulated annealing result is used for comparison as an upper performance bound. As for the primary test subjects, the RAIDER protocol is simulated in three versions to cover all variations in the protocol parameters, i.e. the link quality threshold and the potential retransmissions.

### 6.3.1 Average Link Quality

To simulate different average link qualities, the simulation is repeated for $l = [100, 200, 300, 400]$ m. Each iteration combines the results for $250$ random

networks, with $15$ nodes and $100$ batches per source.
The results include a precise simulation for the state of the art performance, which has already been analyzed in chapter 4, as well as the simulated annealing results from section 6.2. The simulated annealing results are also included in the detailed performance figures.
All results follow a clear pattern. As expected, the better the network conditions are, the better is the performance overall and the closer it is to the determined optimum. Furthermore, the results show a distinct grouping of the simulated protocols. The variations of OLSR all exhibit a lower airtime, but at the same time a comparably low reliability. The reduced number of transmissions reduce the airtime, while the increased dependency on the links between the MPRs renders it more error-prone and thus less reliable. The opposite side of the spectrum are the B.A.T.M.A.N. variations, that invest much airtime to achieve high reliabilities. They are the only protocol variations achieving a consumed airtime of more than $0.6$ ms, eventually even up to $1.44$ ms. In between those two are flooding and all the variations of RAIDER. Flooding achieves a constantly average airtime between $0.39$ ms and $0.48$ ms, while the reliability quickly decreases with deterioration surrounding network conditions. Compared to the state of the art protocols, it is evident, that the variations of RAIDER offer a distinct advantage over flooding, as well as both B.A.T.M.A.N. and OLSR, since they are in general clearly closer to the upper performance bound. For each simulated network parameter $l$, one variation of RAIDER approaches the simulated annealing result the closest of all measured candidate protocols. *OLSR MAX* clearly outperforms the optimal airtime, however at a considerably worse reliability, as seen in figure 6.6. Therefore the combined performance is significantly lower.
Note that the x-axis is cropped at an airtime of $0.25$ ms. The airtimes for flooding and *B.A.T.M.A.N. BASE* are $0.48$ ms and $1.44$ ms respectively, as shown in A.3.

As all variations are specialized differently, the best performing RAIDER variation changes for different network sizes. The regular RAIDER configuration with no threshold and retransmissions in place, *RAIDER STD*, performs similar to *OLSR BASE* for all measured $l$. Although not closest of all RAIDER configurations, for good network conditions, i.e. $l = 100$ m, *RAIDER STD* approaches the upper performance bound relatively close. This is shown in figure 6.6, where the airtime as well as the reliability exceed the optimum slightly for the mean values and definitively considering the deviation. The *RAIDER* performance peaks at a reliability of $0.995$ with an airtime of $29$ µs. However, with a deteriorating average link quality the performance, especially in reliability, drops significantly to $0.73$. As the detail plots for $l = 100$ m and $400$ m demonstrate, the reliability stays in general ahead of the OLSR average, in both mean value and deviation. Compared to B.A.T.M.A.N., a reliability gain can only be achieved against *B.A.T.M.A.N. BASE* for $l = 100$ m. In terms of airtime, it is one of the two best performing protocols, with the only exception of *OLSR MAX* in the better network conditions, as it can be seen in detail in figure 6.6. Although the low airtime comes at a price of a lower reliability than flooding and *B.A.T.M.A.N. BASE*, it

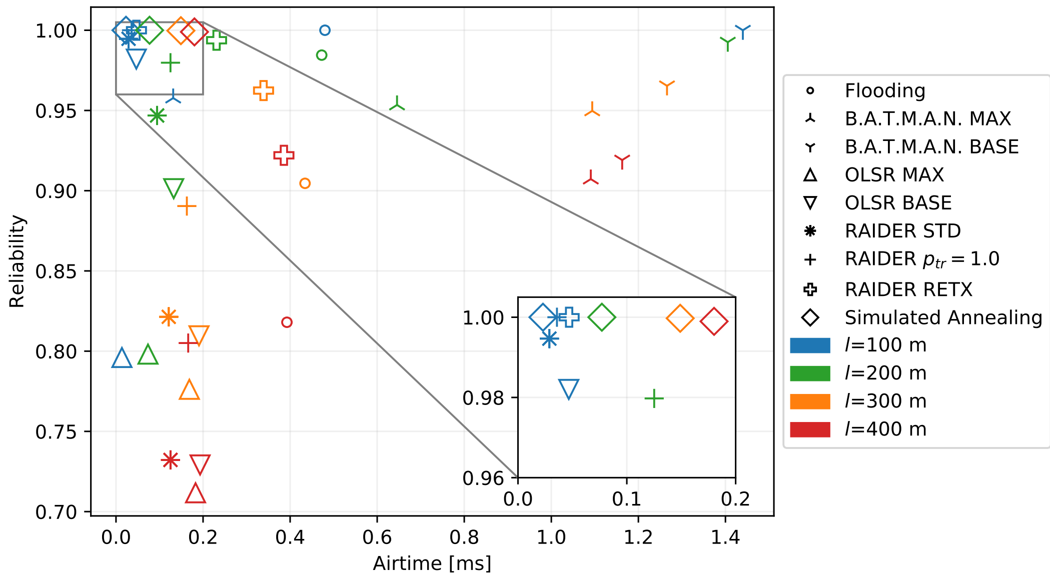



Figure 6.5: Performance overview for all protocols for different $l$

still massively outperforms *OLSR MAX* in reliability and therefore also considering the overall performance. In relation to the poor performance against the performance bound for larger network sizes, this is relevant only for $l = 100$ m.

The remaining two RAIDER implementations *RAIDER RETX* and *RAIDER* $p_{tr} = 1.0$ performed closer to the simulated annealing. Naturally, *RAIDER RETX* includes more transmissions overall and thus it also performs closer to B.A.T.M.A.N.. However it always offers immense savings in airtime, while performing equally or better in reliability, as figures 6.6 and 6.7 show. The airtime never exceeds $0.4$ ms, while the B.A.T.M.A.N. versions on average never fall below $0.9$ ms. For $l = 300$ m and $400$ m it also performs best overall among all candidate protocols, with a minimum reliability of $0.92$ for $l = 400$ m. In network conditions with unreliable links, the redundant retransmissions compensate the link errors to approach the upper performance bound as close as possible. For better network conditions with less error-prone links, the retransmissions can be omitted. Therefore *RAIDER* $p_{tr} = 1.0$ achieves the best performance results. In the smallest simulated network, it invests the small extra amount of airtime over the *RAIDER STD* minimum to guarantee absolute reliability, as shown in figure 6.6. At the same time it exceeds *RAIDER RETX*'s airtime of $47$ µs by additional $11$ µs.
In summary, all RAIDER implementations offer a clear advantage in performance over the state of the art. Compared to B.A.T.M.A.N. it can not match the reliability of *B.A.T.M.A.N. BASE*, but stays at the same time well below its consumed airtime. The opposite is true for OLSR, as RAIDER versions can not go below the airtime in any case. The additional invested airtime however, creates a distinct reliability advantage. Flooding requires more airtime, than any RAIDER implementation for all net-

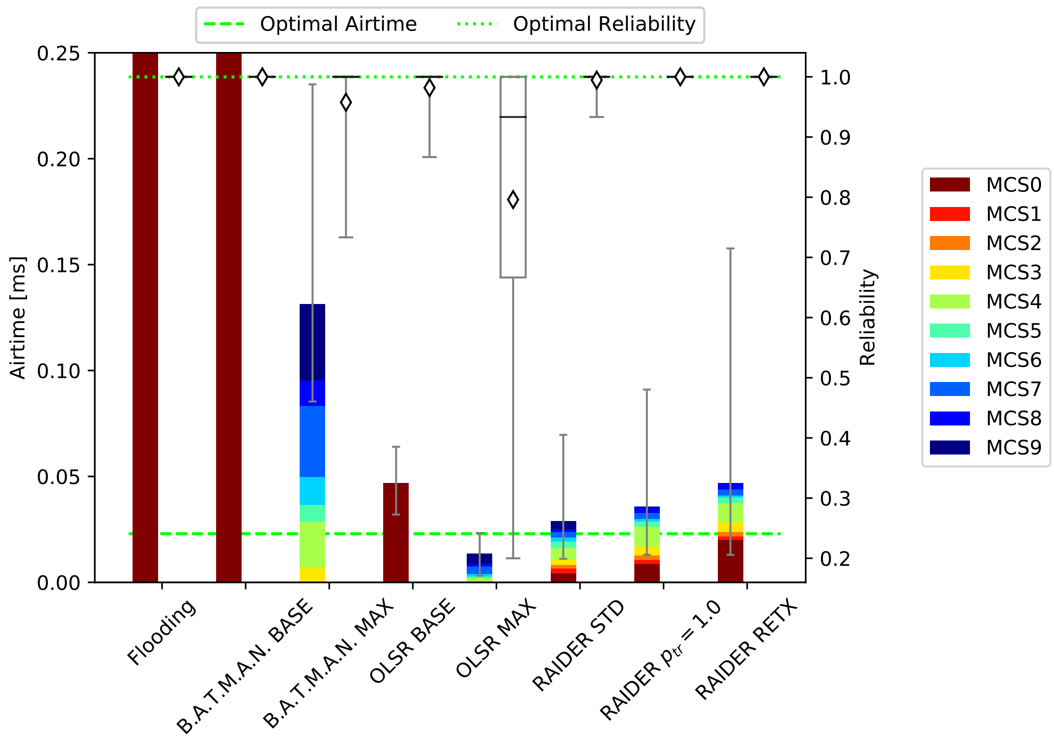



Figure 6.6: Detailed performance for all protocols for $l = 100$ m, cropped

work sizes, although it approaches *RAIDER RETX* for $l = 400$ m. In general, the state of the art can, if at all, only surpass a single RAIDER version. The overall performance compared against the upper performance bound shows as well, that a configuration of RAIDER is always the best performing protocol. Figures 6.6 and 6.7 show, that several OLSR and RAIDER versions approach or exceed the optimum. Nevertheless, this does not reflect the overall performance, since the reliability can simultaneously not match the optimal bound. However RAIDER comes closest and can as sole protocol truly approach the performance bound, although only for $l = 100$ m.

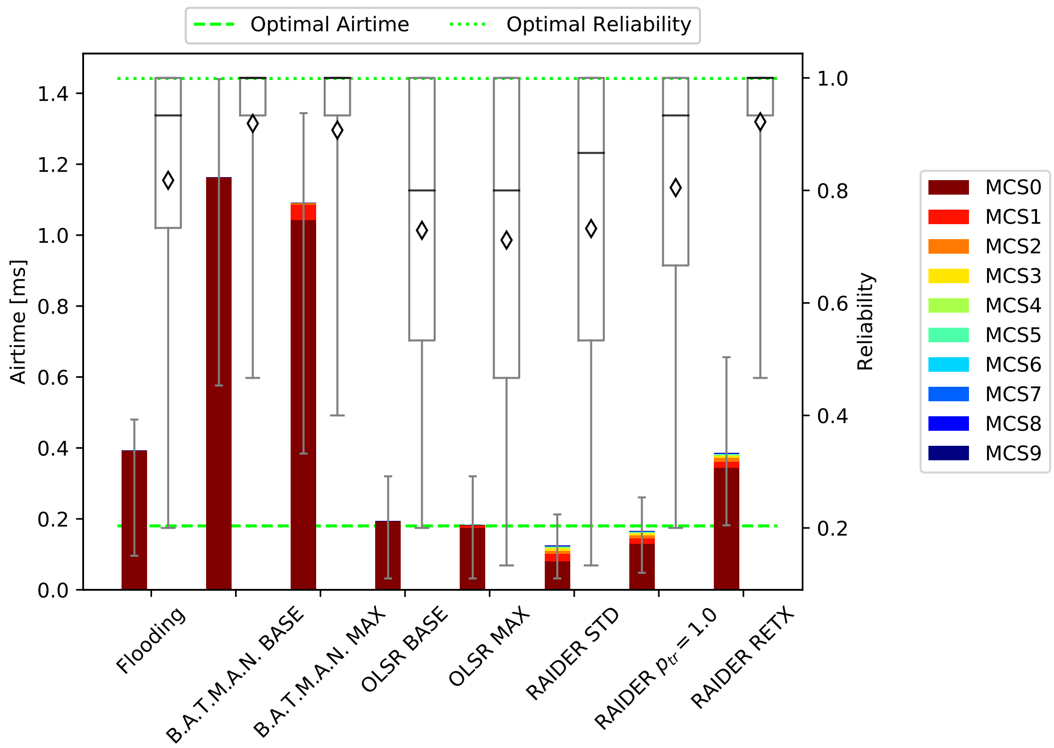



Figure 6.7: Detailed performance for all protocols for $l = 400$ m

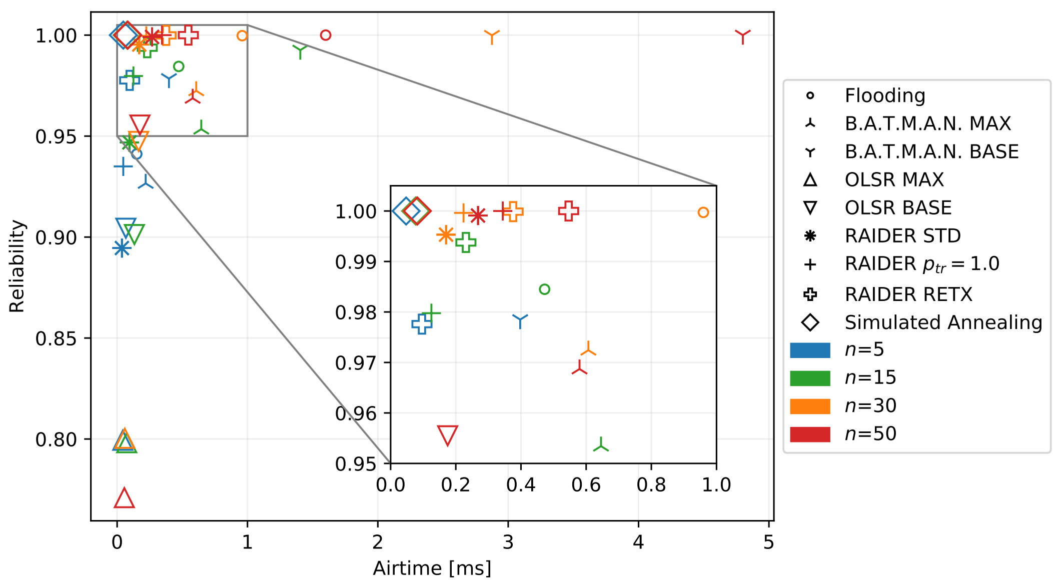



Figure 6.8: Performance overview for all protocols for different $n$

### 6.3.2 Scalability

This simulation includes a varying amount of nodes $n = [5, 15, 30, 50]$ in the network with $l = 200$ m, to create different network densities. Each iteration combines the results for $250$ random networks and $100$ batches per source.
The results of the scalability performance in figure 6.8 for flooding show a steady increase in airtime, while maintaining a reliability of more than $0.94$. Naturally, the addition in airtime raises the level of reliability for larger $l$ to finally match $1.0$. The airtime scales poorly with the number of nodes, as each node receiving the information for the first time will retransmit. The overall airtime even exceeds $1.5$ ms for $l = 400$ m. In return, this will minimize the number of unserved nodes in very dense networks. However for a sparse network, the amount of connections in between the individual nodes is not sufficient to compensate for every erroneous transmission, decreasing the overall reliability.
*OLSR MAX* delivers with the usage of the highest connecting data rate a minimum in airtime with $43$ µs. This can only be surpassed in sparse networks by *RAIDER STD* with $37$ µs and its usage of faster transmission rates, depicted in figure 6.10. Although both protocols perform better in airtime, than the simulated optimal in airtime, the optimal performance bound overall can not be beaten, due to the high unreliability. For the maximum simulated density, this transmission optimization by *OLSR MAX* culminates in a minimum airtime overall for that network density, which on the other side comes at a cost of a immense reliability decrease, as shown in figure 6.9. Note that the x-axis is cropped at an airtime of $1.5$ ms. The airtimes for flooding and *B.A.T.M.A.N. BASE* are $1.6$ ms and $4.8$ ms respectively, as shown in A.4.
In general for all $n$, the basic *OLSR BASE* version utilizing the lowest possible rate comes close to *OLSR MAX*'s performance in airtime. This can be explained, since in dense networks, a large proportion of the network can be covered with a single transmission on the lowest rate. For sparse networks, the highest possible MCS is more often MCS0, which reduces the gain in airtime. The additional airtime, compared to the previous OLSR implementation, offers a better reliability in return.

B.A.T.M.A.N. exhibits the identical weaknesses as flooding. Since the number of transmission is fixed per served node, the airtime scales poorly, even worse with three transmissions per node. No version comes close to optimal bound in regard of airtime. *B.A.T.M.A.N. BASE* is by far the only protocol, that exceeds $2$ ms. The second version, utilizing always the highest available rate, can not reach a considerably good performance in airtime and in reliability. The average in airtime never falls below $0.2$ ms, while the mean reliability never exceeds $97\%$. It fails to balance the right amount of invested airtime and achieved reliability to accomplish an efficient performance.
Regarding the results for the variations of RAIDER, figure 6.8 shows that RAIDER surpasses all state of the art protocols in combined performance. It approaches the simulated annealing result the most of all protocols. In comparison to the rest of

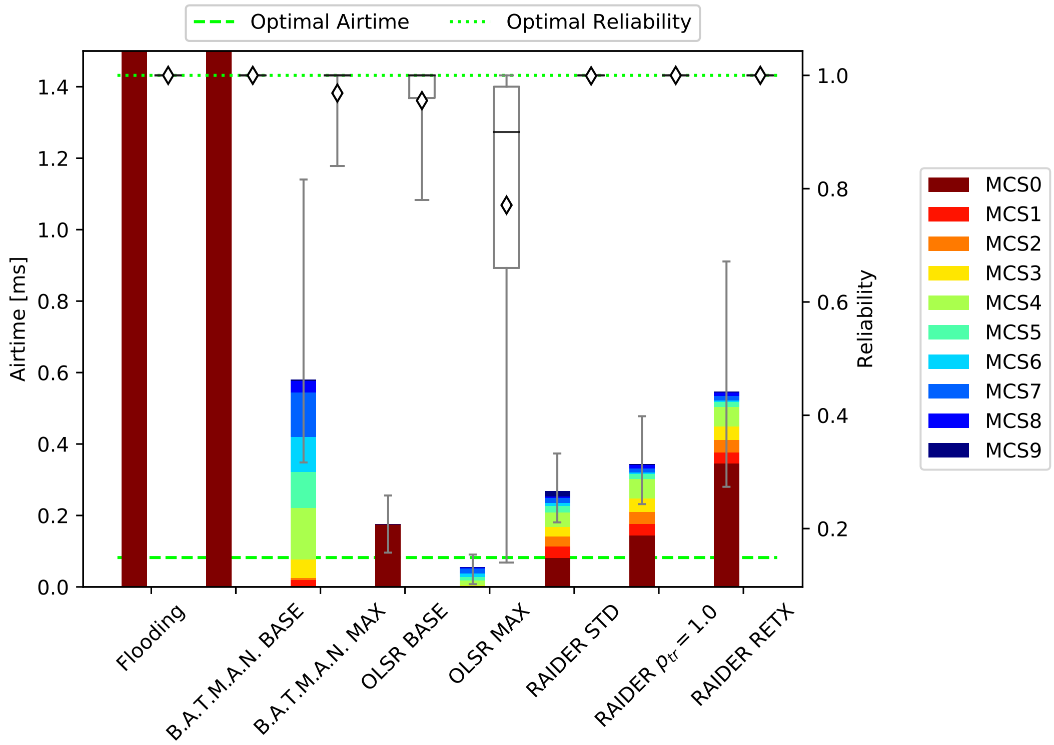



Figure 6.9: Detailed performance for all protocols for $n = 50$, cropped

the protocols, they clearly shift in performance with the number of nodes in the network. With an increasing number of nodes, the reliability constantly improves, while the airtime increases slightly. This presents a disadvantage for *RAIDER RETX* for a high network density, as the retransmissions are not necessary and only increase the airtime up to $0.55$ ms. The invested airtime, however, guarantees as the only protocol a reliability constantly above $97\%$. In sparse networks, as seen in figure 6.10, the redundant transmissions achieve a clearly better reliability also for the deviation of the results. If the network becomes populated more and more, the best performing implementation of RAIDER shifts towards the *RAIDER STD* version. For medium densities, the retransmissions can be omitted first and for even higher densities, the link quality threshold can be omitted as well. *RAIDER STD* still achieves $99.5\%$ reliability for $n = 30$, while *RAIDER* $p_{tr} = 1$ invests $50$ µs more airtime to raise the reliability to $99.96\%$. Depending on the application, one of those implementations would perform best. For $n = 50$, the airtime and reliability for *RAIDER STD*, $27$ µs and $99.91\%$, are even closer to *RAIDER* $p_{tr} = 1$ at $34$ µs and $99.999\%$. Overall, the links become sufficiently short, that errors rarely occur to marginally reduce the reliability, while the omission of the threshold keeps the protocol from selecting slower rates.
Considering the results for the simulated annealing alone, it can be stated, other than in the previous scenario, they can not really be matched or even seriously approached by any protocol version. In comparison to the average link quality simulation, the in-

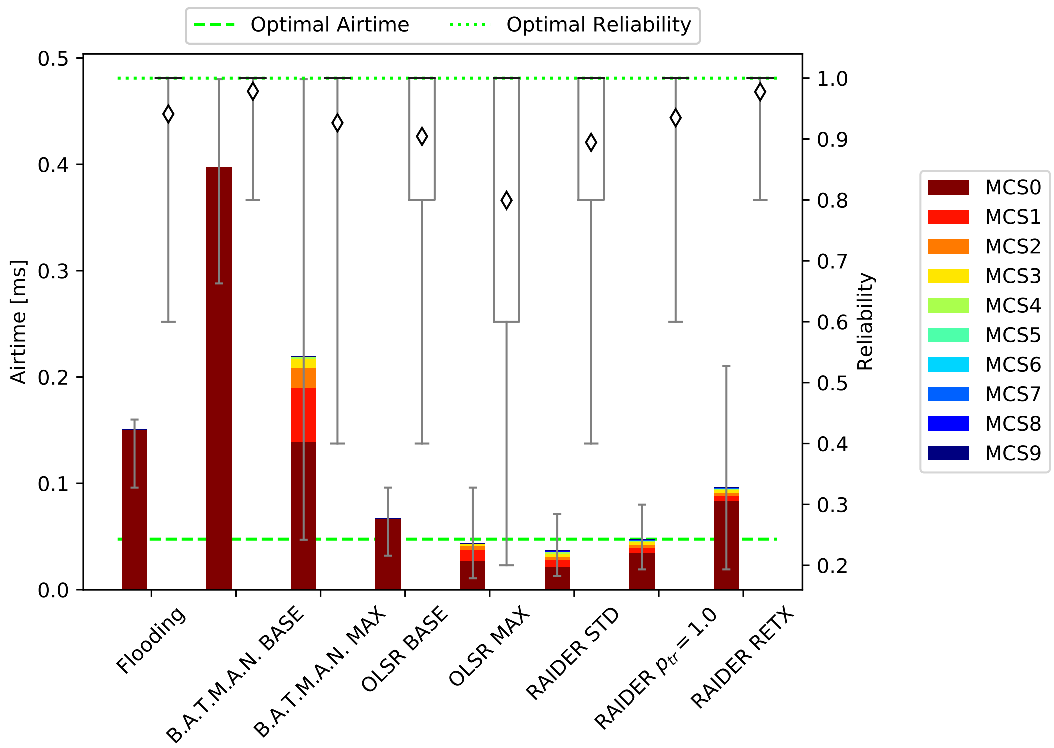



Figure 6.10: Detailed performance for all protocols for $n = 5$

dividual results for different $n$ do not differ as much from each other. They all achieve complete reliability for all four amounts of nodes in the network. Therefore, the optimization with $\alpha = \inf$ was suitable, as errors do not have to be tolerated. The optimum for $n = 5$ stands out, with a clearly reduced airtime, with all others following in ascending order. The more nodes are in the network, the more airtime is required. However, as the results are not significantly further spaced from each other with an increasing $n$, the optimum exhibits an unprecedented scalabilty, setting itself apart from all protocols clearly.
Given that both performance metrics airtime and reliability are regarded together, RAIDER exhibits the best scalability, outperforming the state of the art in flooding, B.A.T.M.A.N. and OLSR. The latter scales excellent with the increasing number of nodes in airtime, but does not manage to replicate this for reliability. B.A.T.M.A.N., in contrast, scales similarly to flooding poorly regarding the consumed airtime, which outweighs improvements in reliability.

### 6.3.3 Summary

In conclusion, the RAIDER variations deliver a distinct performance improvement compared to flooding, B.A.T.M.A.N. and OLSR. While both B.A.T.M.A.N. and OLSR have their advantages in different conditions, both are outperformed by one implementation of RAIDER in *RAIDER RETX* and *RAIDER* $p_{tr} = 1.0$ respectively. In contrast to B.A.T.M.A.N. and OLSR however, RAIDER is able to switch between the presented versions dynamically. Therefore the best performance can always be selected for the comparison. Table 6.2 showcases the exact performance gains. The determined performance column represents the performance evaluation with $\alpha = 5$, combining airtime and reliability into one metric. As discussed earlier, $\alpha = 5$ is selected, . The performance of B.A.T.M.A.N. and OLSR was averaged over both versions, taking the possibility of the usage of different transmission rates into account.

| | Protocol | Reliability | Airtime | Determined Performance |
|---|---|---|---|---|
| $l = 100$ m | B.A.T.M.A.N. average | +2% | -95% | +2326% |
| | Flooding | +0% | -93% | +1233% |
| | OLSR average | +12% | +20% | +50% |
| $l = 200$ m | B.A.T.M.A.N. average | +1% | -88% | +750% |
| | Flooding | +0% | -74% | +271% |
| | OLSR average | +15% | +22% | +68% |
| $l = 300$ m | B.A.T.M.A.N. average | +0% | -71% | +256% |
| | Flooding | +6% | -22% | +74% |
| | OLSR average | +21% | +88% | +40% |
| $l = 400$ m | B.A.T.M.A.N. average | +1% | -66% | +207% |
| | Flooding | +13% | -2% | +85% |
| | OLSR average | +1% | +105% | +67% |
| $\bar{l}$ | B.A.T.M.A.N. average | +1% | -80% | **+885%** |
| | Flooding | +5% | -47% | **+416%** |
| | OLSR average | +19% | +59% | **+56%** |

Table 6.2: Achieved reliability, airtime and performance of RAIDER compared to B.A.T.M.A.N., flooding and OLSR for different $l$

On average, RAIDER outperforms flooding by $416\%$, with an airtime decrease of $47\%$, while improving the reliability by $5\%$. Compared to OLSR, it increases the airtime by $59\%$, but at the same time also significantly increases the reliability by $19\%$ to offer an overall improvement of $56\%$. The biggest gain of RAIDER with $885\%$ was obtained against B.A.T.M.A.N., by immensely reducing the airtime by $80\%$ while also slightly increasing the reliability by one percent. Compared to the upper performance bound, RAIDER clearly offered the best performance. As shown in table 6.4, for the achieved reliability, it falls $3\%$ short of the performance bound. The airtime can be improved, as it is still $90\%$ higher than the optimum, leading to an overall deficit of $53\%$ of the optimal performance.

The scalability results, in comparison between the state of the art protocols and RAIDER, are very similar to the previous scenario, assessing the performance for

| | Protocol | Reliability | Airtime | Determined Performance |
|---|---|---|---|---|
| $n = 50$ | B.A.T.M.A.N. average | +1% | -90% | +980% |
| | Flooding | +0% | -83% | +494% |
| | OLSR average | +16% | +133% | -11% |
| $n = 30$ | B.A.T.M.A.N. average | +1% | -87% | +734% |
| | Flooding | +0% | -77% | +329% |
| | OLSR average | +14% | +100% | -2% |
| $n = 15$ | B.A.T.M.A.N. average | +1% | -88% | +750% |
| | Flooding | +0% | -74% | +271% |
| | OLSR average | +15% | +22% | +68% |
| $n = 5$ | B.A.T.M.A.N. average | -2% | -85% | +498% |
| | Flooding | -1% | -69% | +209% |
| | OLSR average | +10% | -15% | +89% |
| $\overline{n}$ | B.A.T.M.A.N. average | +0% | -87% | **+741%** |
| | Flooding | +0% | -76% | **+326%** |
| | OLSR average | +14% | +60% | **+36%** |

Table 6.3: Achieved reliability, airtime and performance of RAIDER compared to B.A.T.M.A.N., flooding and OLSR for different $n$

different link qualities. The evaluation for this scenario was also conducted with $\alpha = 5$, errors and unreliable performances across all protocols and $n$ had to be considered. However, especially in the comparison to the performance bound, this favors RAIDER, as the missing reliability can not be penalized, to assure a possibility to still compare both results. For $\alpha = \inf$, RAIDER could not have been evaluated at all, therefore a comparison would not make sense. In the comparison of RAIDER against the state of the art, this also favors protocols such as OLSR, as the missing reliability is rather tolerated and the airtime optimization weighs heavier. All in all, the average over all $n$ present very close performance statistics to the previous scenario. RAIDER outperforms all state of the art candidates by a similar margin. Compared against B.A.T.M.A.N., it could again reduce the airtime by up to $90\%$, while maintaining the same reliability. Flooding shows a similar tendency, with the sole difference, that it came with considerably less airtime in the beginning. Thus the gains in airtime are slightly smaller. Considering OLSR, a small airtime investment with an increase by $14\%$ achieved a $60\%$ airtime improvement. Overall the determined performance indicates, that RAIDER is clearly performing better, considering the scalability. In comparison to the upper bound however, RAIDER again falls well short of the performance bound. Especially for larger number of nodes in the network, it trails immensely considering the consumed airtime. Since it can operates in the scope of the 2-hop neighborhood, RAIDER can not plan the transmissions through the whole network, which leads inevitable to unnecessary transmissions. The more nodes, the more global coordination is required to achieve

| | Reliability | Airtime | Determined Performance |
|---|---|---|---|
| $l = 100$ m | +0% | +57% | -36% |
| $l = 200$ m | -2% | +62% | -44% |
| $l = 300$ m | -4% | +128% | -64% |
| $l = 400$ m | -8% | +114% | -69% |
| $\bar{l}$ | -3% | +90% | **-53%** |

Table 6.4: Achieved reliability, airtime and performance of RAIDER compared to the upper performance bound for different $l$

| | Reliability | Airtime | Determined Performance |
|---|---|---|---|
| $n = 50$ | +0% | +458% | -82% |
| $n = 30$ | +0% | +191% | -66% |
| $n = 15$ | -2% | +54% | -41% |
| $n = 5$ | -6% | -43% | +25% |
| $\bar{l}$ | -2% | +165% | **-41%** |

Table 6.5: Achieved reliability, airtime and performance of RAIDER compared to the upper performance bound for different $n$

the optimum. The expendable transmissions in RAIDER increase the airtime by up to $458\%$ compared to the performance bound. All this airtime could be saved, by increasing the scope, in which optimization and orchestration happens. Eventually RAIDER performs overall $41\%$ worse, which is mainly credited to the airtime surplus.

All in all the overall performance of RAIDER, as well as in reliability and airtime separately, exhibits the desired increase. Depending on the surrounding conditions and the planned use case, RAIDER is adaptable to match the requirements as best as possible.
In relation to the constraints in the protocol conception described in section 5.1, the protocol delivers a sufficient step towards the upper performance bound. To further decrease the gap, the prerequisites for RAIDER can be lifted and the protocol accordingly optimized and extended. However, this would create dependencies, that restrict the usage from certain scenarios, as all possible deployment scenario must match them. Considering the 2-hop neighborhood, that performance can not be optimal by design, as the topology information is not sufficient to detect loops in any case. This inevitably leads to inefficiencies, that decrease the performance compared to the optimum.

# 7 Conclusion

This work gave an overview of current approaches in broadcast schemes for WMNs. Most of the current state of the art relies on fixed transmissions rates, which is incorporated into the whole protocol concept. Slight adaptions to support an usage of a dynamic selection to utilize the multi-rate capabilities of the WLAN environment showed only small improvements, as the problematic features remained. Further approaches, that offer a dynamic rate selection themselves depended on different assumptions, that can hardly be realized in the way, that they are required. Obtaining up-to-date complete network knowledge or fulfilling the tasks of a central network entity are not generally applicable to WMN use cases, if they comply at all. From this in-depth analysis, a set of requirements and guidelines for the development of an improved broadcast scheme were derived. Following these principles and in the scope of the defined prerequisites, a new broadcast scheme RAIDER was developed. To properly put its performance in proportion, an upper performance bound was determined by probabilistic simulated annealing.
For all evaluations of the different broadcast schemes, a basic network simulator was written to obtain a performance from each candidate protocol under various circumstances and network conditions. However, the performance evaluation bears a fundamental problem in broadcast. In contrast to unicast transmissions, reliability is not binary anymore and has to be included in the performance evaluation in conjunction with a second performance metric. For the purpose of the second metric, this work used the consumed airtime. In the case, that every node only transmits once, it is similar to the latency, but in contrast to that also considers the influence of all caused transmissions in the whole broadcast process on the remaining network activities. Depending on the topology knowledge, these transmissions can exceed the time, until each node received the broadcast, increasing the airtime compared to the delay.
The overall performance determination including the achieved reliability and airtime must be adjusted to the desired use case. The performance reacts very sensitive to small changes in the calculation of a joint performance metric, evoking significant variations in the final evaluation. Therefore, this work focuses on the reliability over the airtime. To match the requirements of lower-layer network protocols, such as for instance ARP and at the same time industrial applications, that demand a very reliable and efficient delivery, the reliability was prioritized in the performance evaluation. Nevertheless, all results need to be analyzed and interpreted accordingly. For different assumptions and prerequisites, the analysis might change in terms of the achieved performance and the preferred broadcast solution. This also applies for the determined upper performance bound. A suitable performance determination is able to weigh different results against one another to determine the superior perfor-

mance.
In the end, the new broadcast scheme outperformed the existing broadcast schemes significantly. This holds true for the selected performance definition, taking the achieved airtime and reliability with a different weight into account. Furthermore it proved to be scalable as well as adaptable to match a broad range of possible applications. With the use case and environment specific adjustment of integrated protocol parameters, the sensitivity for link error rates and the ability to retransmit on the lowest rate, RAIDER maximizes performance in all conditions. This integration of link quality awareness, among further details, sets it apart from the current state of the art broadcast schemes. The usage of low-cost and reliable up-to-date information of the current 2-hop neighborhood as routing information is a sufficient base to already optimize the usage of maximum transmission rates and therefore the broadcast performance. The significant gains indicate, that a fixed rate is no longer a necessity for a constantly reliable broadcast. The open and general prerequisites enable RAIDER to be able to be deployed in all sorts of environments and applications.

## 7.1 Outlook

In future work, the performance of RAIDER can be improved to approach the possible upper performance bound even more. The usage of a 2-hop neighborhood routing information base can only maximize the performance in this limited scope. This might cause in the different scope of the complete network unnecessary transmissions, that increase the airtime. Furthermore, to optimize for the current network environments, RAIDER does not contain a fully automatic recognition of the current network conditions and an according adjustment of the protocol parameters. Therefore the balance between reliability and airtime has to be investigated in further detail, including the intended use case and its requirements. This can improve the protocol performance by combining the respectively best performing current RAIDER version for each scenario into a single fully automatic and dynamic protocol.
Finally, RAIDER has to be tested in a broad range of applications, including different traffic patterns. Further applications, such as for instance video streaming, have different demands, focused on high throughput and fast delivery. This also includes the usage of a network code, such as for instance RLNC. The ability to recover losses in the transmissions by transmitting extra redundancy allows RAIDER to operate more efficiently also with lower link quality thresholds. A further investigation of this balance could be conducted. An optional implementation of feedback could assist the performance of the coding, as well as the up-to-date topology information base.
Considering the simulation environment, the current implementation supports so far only stationary nodes. To begin the concept development of a WMN broadcast scheme, this is a requirement. However, this requirement must later be dropped, to include the natural node behaviour, that characterizes WMNs. Spontaneous node or link failures, as well as sudden or gradually changes in link metrics have influence

on the routing performance. This holds true, if the routing information base is not updated, since the change is not detected and communicated. This resilience consideration might require RAIDER to be extended, to support the more dynamic environment. So far, it includes a possible error tolerance, in that with a high link quality threshold it selects the data rates very conservatively, which should tolerate minor changes in the link error rates. However, the current broadcast might not be able to recover from a node failure in time, as forwarding assignments depend on the selected forwarding nodes to transmit accordingly.
This work includes the determination of an upper performance bound for broadcast, tailored specifically for the performance of RAIDER, considering the MCS0 retransmissions. To obtain a general performance bound, this work could be extended. Every considered use and deployment case might come with different requirements and possibilities. Therefore the optimization goal can change. However, even slight changes in the performance evaluation could drastically change the result. Hence, different applications and use cases might require a different performance bound. The ascertained results are applicable only with similar prerequisites. If the constraints for the monte carlo simulation match the desired use case, the simulation could be extended for more iterations, to obtain more precise results for larger network sizes.

Following the conceptual work on RAIDER and its mechanics, the protocol must be deployed to be tested in a realistic testbed. This will put the simulator performance in proportion and enables an actual performance evaluation. The influence of so far not regarded lower-layer functionalities, such as the actual CSMA/CA with RTS/CTS or channel transmission details, such as interference or different signal disturbances, could affect the performance as well. This evaluation will show the influence of all assumptions, utilized models and simplifications in the simulator and the protocol mechanics.

# A Additional plots

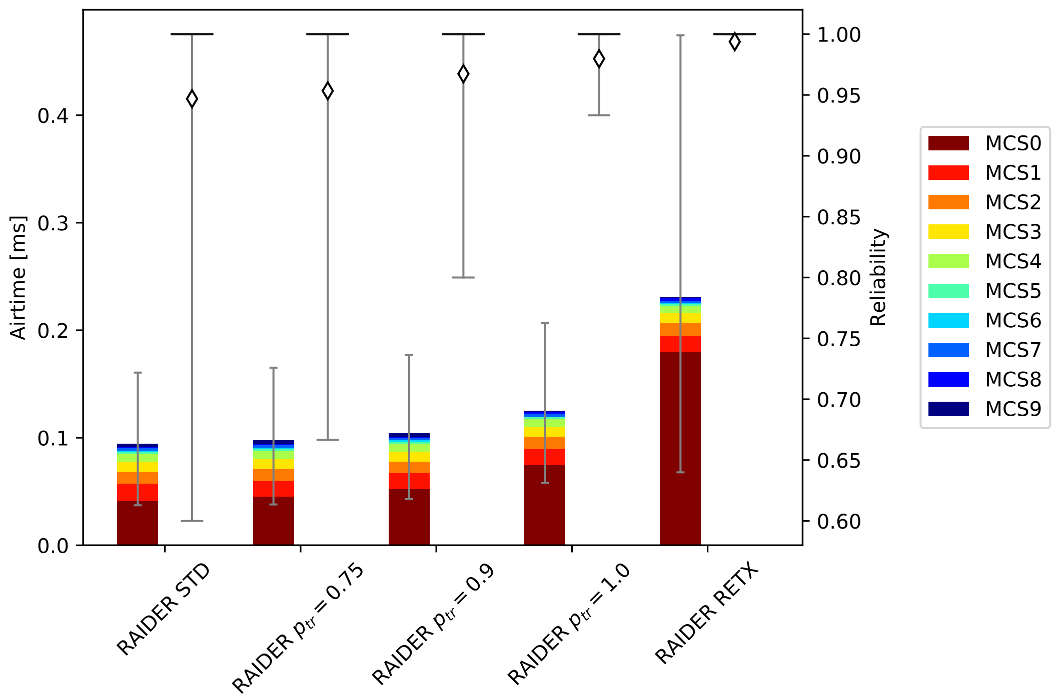



Figure A.1: Detailed performance for RAIDER variations for $l = 200$ m

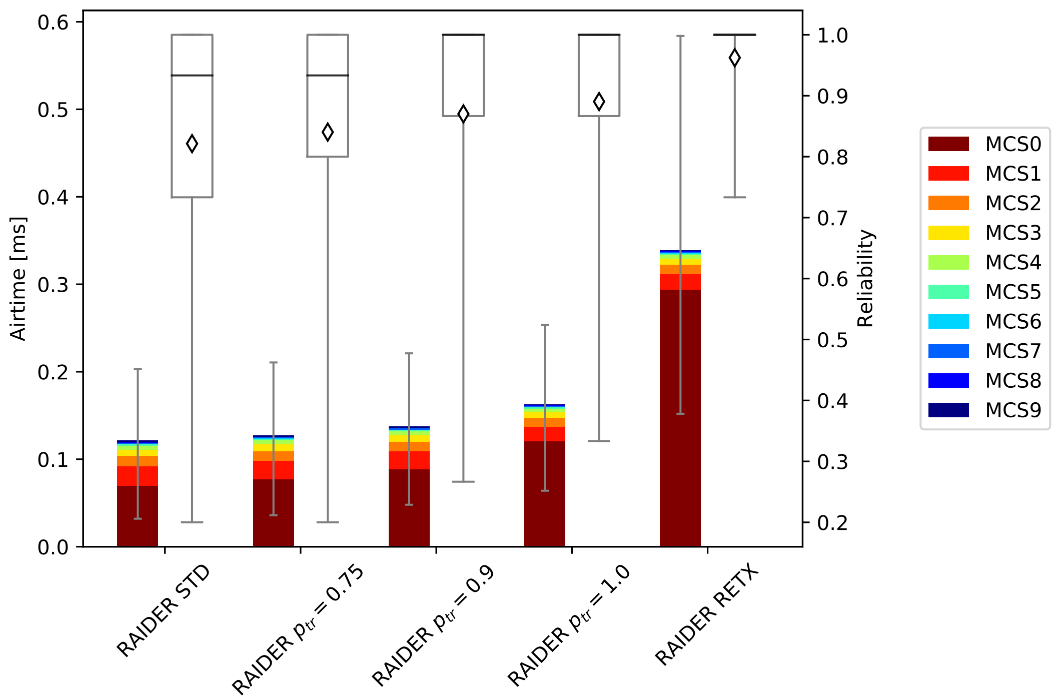



Figure A.2: Detailed performance for RAIDER variations for $l = 300$ m

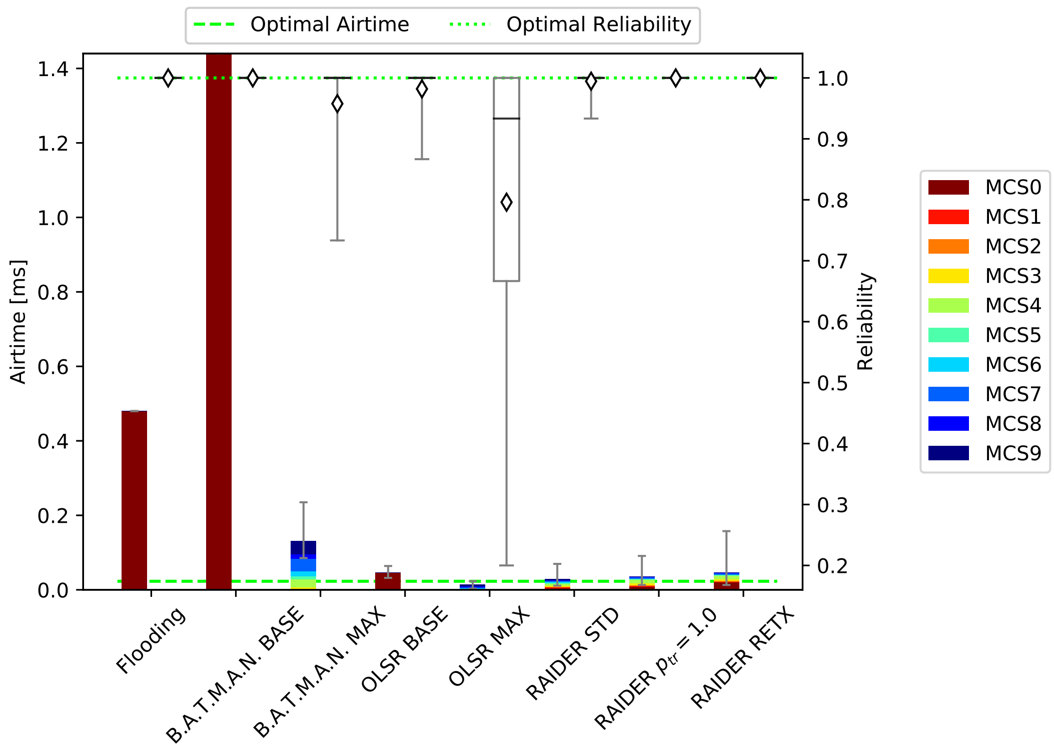



Figure A.3: Detailed performance for all protocols for $l = 100$ m

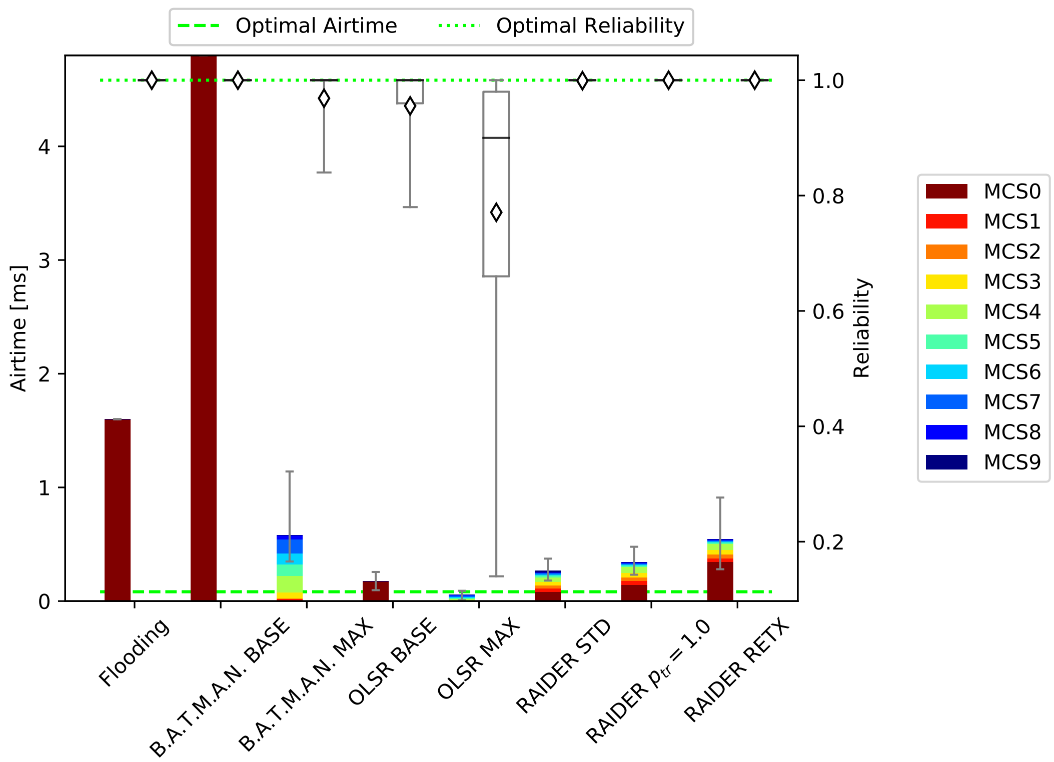



Figure A.4: Detailed performance for all protocols for $n = 50$

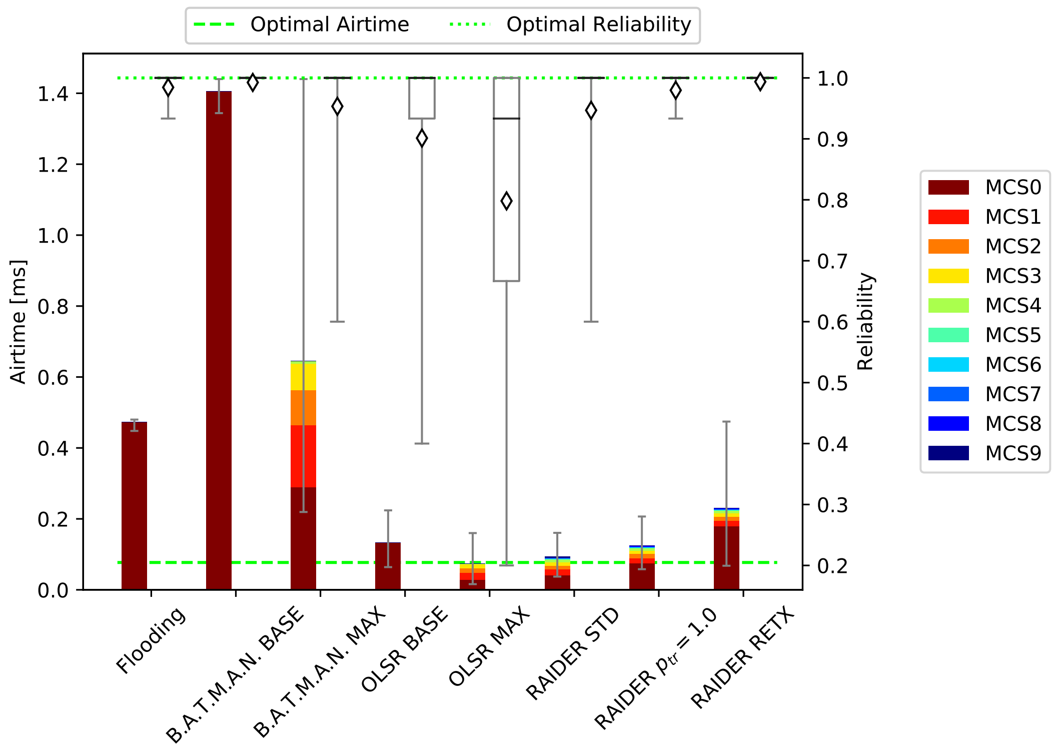



Figure A.5: Detailed performance for all protocols for $l = 200$ m

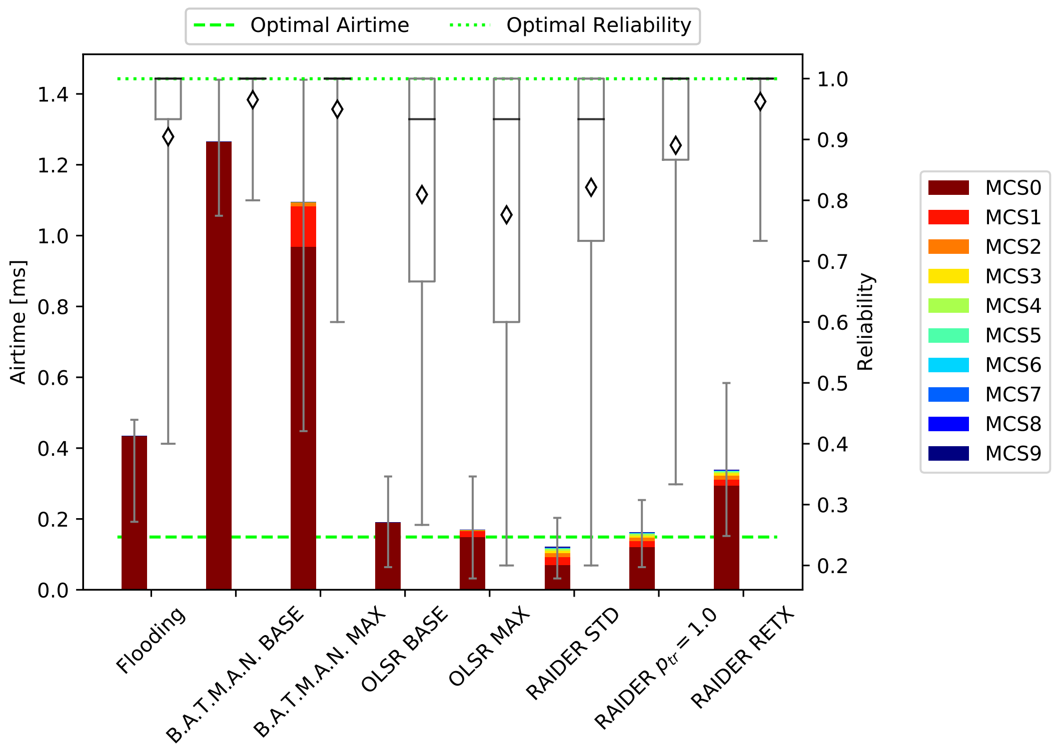



Figure A.6: Detailed performance for all protocols for $l = 300$ m

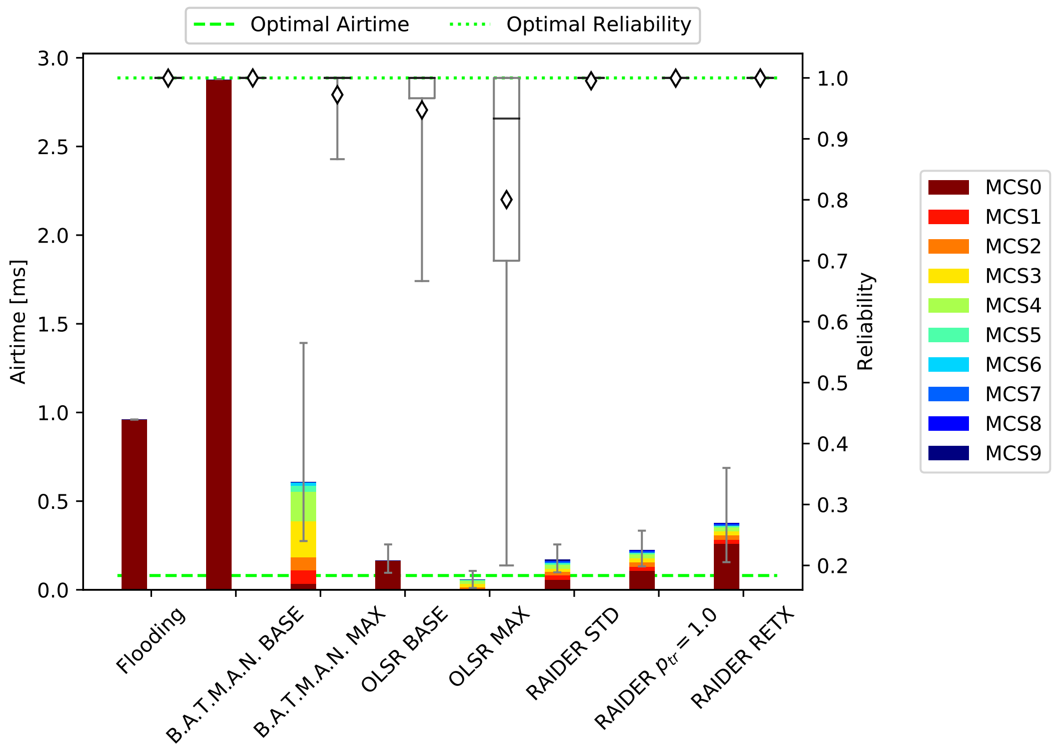



Figure A.7: Detailed performance for all protocols for $n = 30$

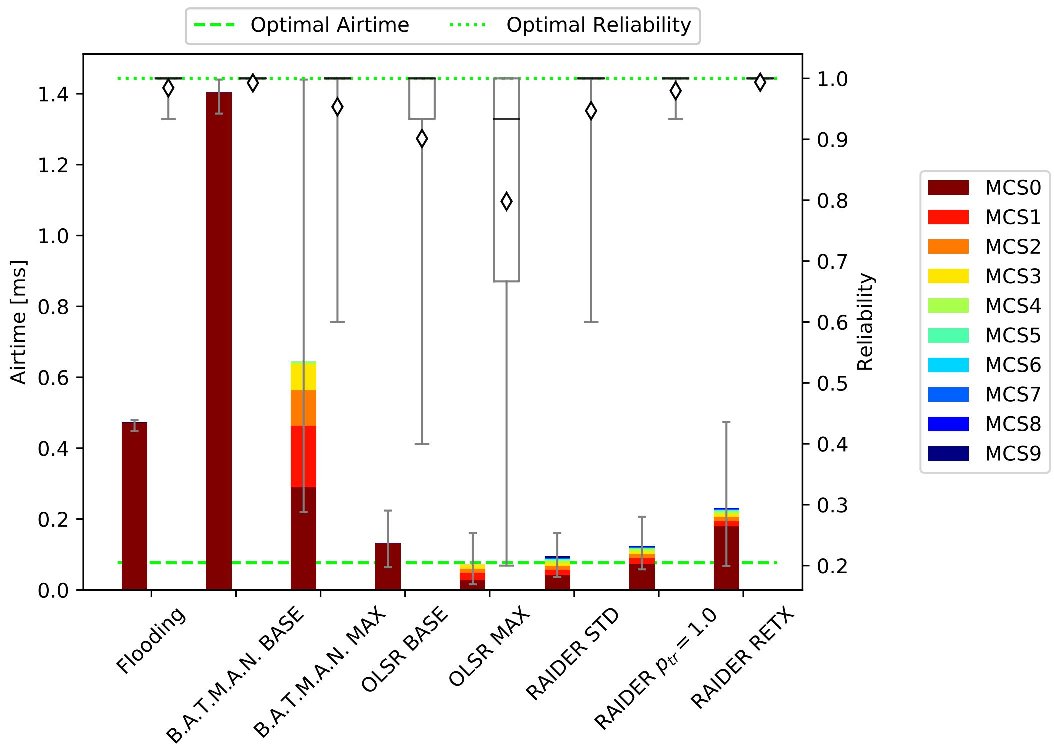



Figure A.8: Detailed performance for all protocols for $n = 15$

# Bibliography


[1] Mehran Abolhasan, Brett Hagelstein, and JC-P Wang. "Real-world performance of current proactive multi-hop mesh protocols". In: *2009 15th Asia-Pacific Conference on Communications*. IEEE. 2009, pp. 44–47.

[2] Cédric Adjih, Emmanuel Baccelli, and Philippe Jacquet. "Link state routing in wireless ad-hoc networks". In: *IEEE Military Communications Conference, 2003. MILCOM 2003.* Vol. 2. IEEE. 2003, pp. 1274–1279.

[3] M Shahwaiz Afaqui, Eduard Garcia-Villegas, and Elena Lopez-Aguilera. "IEEE 802.11 ax: Challenges and requirements for future high efficiency WiFi". In: *IEEE Wireless Communications* 24.3 (2016), pp. 130–137.

[4] Ian F Akyildiz and Xudong Wang. "A survey on wireless mesh networks". In: *IEEE Communications magazine* 43.9 (2005), S23–S30.

[5] Ian F Akyildiz and Xudong Wang. *Wireless mesh networks*. Vol. 3. John Wiley & Sons, 2009.

[6] *Allgemeinzuteilung von Frequenzen der Bundesnetzagentur, WLAN 2.4 GHz*. `https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/2013_10_WLAN_2,4GHz_pdf.pdf?__blob=publicationFile&v=5`. Accessed: 2019-10-21.

[7] *B.A.T.M.A.N.adv Protocol information*. `https://www.open-mesh.org/projects/batman-adv/wiki/BATMAN_IV`. Accessed: 2019-10-21.

[8] Dimitris Bertsimas, John Tsitsiklis, et al. "Simulated annealing". In: *Statistical science* 8.1 (1993), pp. 10–15.

[9] Azzedine Boukerche et al. "Routing protocols in ad hoc networks: A survey". In: *Computer networks* 55.13 (2011), pp. 3032–3080.

[10] Abdelali Boushaba et al. "Multi-point relay selection strategies to reduce topology control traffic for OLSR protocol in MANETs". In: *Journal of Network and Computer Applications* 53 (2015), pp. 91–102.

[11] Raffaele Bruno, Marco Conti, and Enrico Gregori. "Mesh networks: commodity multihop ad hoc networks". In: *IEEE communications magazine* 43.3 (2005), pp. 123–131.

[12] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. *Trading structure for randomness in wireless opportunistic routing*. Vol. 37. 4. ACM, 2007.

[13] Benjamin Avery Chambers. “The grid roofnet: a rooftop ad hoc wireless network”. PhD thesis. Massachusetts Institute of Technology, 2002.

[14] Chun Tung Chou, Archan Misra, and Junaid Qadir. “Low-latency broadcast in multirate wireless mesh networks”. In: *IEEE Journal on Selected Areas in Communications* 24.11 (2006), pp. 2081–2091.

[15] *Cisco Global Traffic Forecast*. `https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2020_Forecast_Highlights.pdf`. Accessed: 2019-10-21.

[16] Thomas Clausen, Christopher Dearlove, and J Dean. *Mobile ad hoc network (manet) neighborhood discovery protocol (nhdp)*. Tech. rep. 2011.

[17] Thomas Clausen, Christopher Dearlove, Philippe Jacquet, and Ulrich Herberg. *The optimized link state routing protocol version 2*. Tech. rep. 2014.

[18] Thomas Clausen and Philippe Jacquet. *Optimized link state routing protocol (OLSR)*. Tech. rep. 2003.

[19] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[20] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. “A high-throughput path metric for multi-hop wireless routing”. In: *Wireless networks* 11.4 (2005), pp. 419–434.

[21] Richard Draves, Jitendra Padhye, and Brian Zill. “Routing in multi-radio, multi-hop wireless mesh networks”. In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 114–128.

[22] Richard Draves, Jitendra Padhye, and Brian Zill. “Comparison of routing metrics for static multi-hop wireless networks”. In: *ACM SIGCOMM Computer Communication Review*. Vol. 34. 4. ACM. 2004, pp. 133–144.

[23] Ralph Droms. “Dynamic host configuration protocol”. In: (1997).

[24] V Erceg, L Schumacher, P Kyritsi, et al. “IEEE 802.11 document 03/940r4 (TGn Channel Models)”. In: *Garden Grove, California* (2004).

[25] *Ethenet Powerlink Standardization Group*. `https://www.ethernet-powerlink.org/index.php`. Accessed: 2019-10-21.

[26] *ExtremeMobility AP50 5i Data Sheet*. `https://cloud.kapostcontent.net/pub/9dfc1be3-973c-4cf9-a4fe-0f06589991f4/ap505-data-sheet?kui=kRzi2Gl8reN5jGiB8oMusA`. Accessed: 2019-10-21.

[27] Elahe Fazeldehkordi, Iraj Sadegh Amiri, and Oluwatobi Ayodeji Akanbi. *A study of black hole attack solutions: On aodv routing protocol in manet*. Syngress, 2015.

[28] Sandip Gangakhedkar et al. “Use cases, requirements and challenges of 5G communication for industrial automation”. In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE. 2018, pp. 1–6.

[29] Matthew Gast. *802.11 wireless networks: the definitive guide*. " O'Reilly Media, Inc.", 2005.

[30] Mikael Gidlund, Tomas Lennvall, and Johan Åkerberg. "Will 5G become yet another wireless technology for industrial automation?" In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2017, pp. 1319–1324.

[31] Tracey Ho et al. "The benefits of coding over routing in a randomized setting". In: (2003).

[32] Ekram Hossain and Kin K Leung. *Wireless mesh networks: architectures and protocols*. Springer, 2007.

[33] Chin-Kai Hsu, Chien Chen, Hsien-Kang Wang, et al. "DISCOUNT: A hybrid probability-based broadcast scheme for wireless ad hoc networks". In: *VTC2005-FALL: 2005 IEEE 62ND VEHICULAR TECHNOLOGY CONFERENCE, 1-4, PROCEEDINGS*. 2005, pp. 2706–2710.

[34] Mohsen Jahanshahi and Alireza Talebi Barmi. "Multicast routing protocols in wireless mesh networks: a survey". In: *Computing* 96.11 (2014), pp. 1029–1057.

[35] Aakash Jasper. "Comparison of Local Area Network Technologies: Ethernet (IEEE 802.3), ATM and WLAN/WiFi (IEEE 802.11 g)". In: *International Journal of Current Engineering and Technology* 5.1 (2015), pp. 503–506.

[36] David Johnson, Ntsibane Ntlatlapa, and Corinna Aichele. "Simple pragmatic approach to mesh routing using BATMAN". In: (2008).

[37] Luo Junhai, Ye Danxia, Xue Liu, and Fan Mingyu. "A survey of multicast routing protocols for mobile ad-hoc networks". In: *IEEE communications surveys & tutorials* 11.1 (2009), pp. 78–91.

[38] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. "Optimization by simulated annealing". In: *science* 220.4598 (1983), pp. 671–680.

[39] Karol Kowalik and Mark Davis. "Why are there so many routing protocols for wireless mesh networks". In: *Irish Signal and Systems Conference*. 2006, pp. 1–5.

[40] G Vijaya Kumar, Y Vasudeva Reddyr, and Dr M Nagendra. "Current research work on routing protocols for MANET: a literature survey". In: *international Journal on computer Science and Engineering* 2.03 (2010), pp. 706–713.

[41] Ou Liang, Y Ahmet Sekercioglu, and Nallasamy Mani. "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks". In: *IEEE Communications Surveys & Tutorials* 8.4 (2006), pp. 30–46.

[42] Hyojun Lim and Chongkwon Kim. "Multicast tree construction and flooding in wireless ad hoc networks". In: *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. ACM. 2000, pp. 61–68.

[43] Henrik Lundgren. "Implementation and real-world evaluation of routing protocols for wireless ad hoc networks". PhD thesis. Uppsala University, 2002.

[44] Madhav V Marathe et al. "Simple heuristics for unit disk graphs". In: *Networks* 25.2 (1995), pp. 59–68.

[45] Bratislav Milic and Miroslaw Malek. "Analyzing large scale real-world wireless multihop network". In: *IEEE Communications Letters* 11.7 (2007), pp. 580–582.

[46] Sudip Misra, Subhas Chandra Misra, and Isaac Woungang. *Guide to wireless mesh networks*. Springer, 2009.

[47] *Modbus Application Protocol Specification V1.1b3*. `http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf`. Accessed: 2019-10-21.

[48] Aminu Mohammed, Mohamed Ould-Khaoua, and Lewis Mackenzie. "An efficient counter-based broadcast scheme for mobile ad hoc networks". In: *European Performance Engineering Workshop*. Springer. 2007, pp. 275–283.

[49] Axel Neumann, Corinna Aichele, Marek Lindner, and Simon Wunderlich. "Better approach to mobile ad-hoc networking (BATMAN)". In: *IETF draft* (2008), pp. 1–24.

[50] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. "The broadcast storm problem in a mobile ad hoc network". In: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM. 1999, pp. 151–162.

[51] *Optimized Link State Routing Protocol Documentation*. `http://www.olsr.org/docs/report_html/node28.html`. Accessed: 2019-10-21.

[52] Diego Passos et al. "Mesh network performance measurements". In: *5th International Information and Telecommunication Technologies Symposium*. Vol. 2006. 2006.

[53] Wei Peng and Xicheng Lu. "AHBP: An efficient broadcast protocol for mobile ad hoc networks". In: *Journal of computer science and technology* 16.2 (2001), pp. 114–125.

[54] Gregers Petersen. "Freifunk: When technology and politics assemble into subversion". In: *Subversion, Conversion, Development: Cross-Cultural Knowledge Exchange and the Politics of Design* (2014), pp. 39–56.

[55] David Plummer. "An ethernet address resolution protocol: or converting network protocol addresses to 48. bit ethernet address for transmission on ethernet hardware". In: (1982).

[56] Ron Porat et al. "11ax evaluation methodology". In: *IEEE802* (2016), pp. 11–14.

[57] *Profibus & Profinet International*. `https://www.profibus.com/technology/profinet/overview/`. Accessed: 2019-10-21.

[58] Junaid Qadir, Chun Tung Chou, Archan Misra, and Joo Ghee Lim. "Localized minimum-latency broadcasting in multi-rate wireless mesh networks". In: *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE. 2007, pp. 1–8.

[59] Amir Qayyum, Laurent Viennot, and Anis Laouiti. "Multipoint relaying: An efficient technique for flooding in mobile wireless networks". PhD thesis. INRIA, 2000.

[60] DG Reina, SL Toral, Princy Johnson, and Federico Barrero. "A survey on probabilistic broadcast schemes for wireless ad hoc networks". In: *Ad Hoc Networks* 25 (2015), pp. 263–292.

[61] Yoav Sasson, David Cavin, and André Schiper. "Probabilistic broadcast for flooding in wireless mobile ad hoc networks". In: *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.* Vol. 2. IEEE. 2003, pp. 1124–1130.

[62] Shlomo S Sawilowsky. "You think you've got trivials?" In: *Journal of Modern Applied Statistical Methods* 2.1 (2003), p. 21.

[63] Daniel Seither, André König, and Matthias Hollick. "Routing performance of Wireless Mesh Networks: A practical evaluation of BATMAN advanced". In: *2011 IEEE 36th Conference on Local Computer Networks*. IEEE. 2011, pp. 897–904.

[64] John S Seybold. *Introduction to RF propagation*. John Wiley & Sons, 2005.

[65] Kamil Staniec and Michal Kowal. "Measurement evaluation of the TGN radio channel models usefulness in predicting WLAN performance". In: *Progress In Electromagnetics Research* 137 (2013), pp. 311–333.

[66] Jayashree Subramanian, Robert Morris, and Hari Balakrishnan. "UFlood: High-throughput flooding over wireless mesh networks". In: *2012 Proceedings IEEE INFOCOM*. IEEE. 2012, pp. 82–90.

[67] Mohsen Nader Tehrani, Murat Uysal, and Halim Yanikomeroglu. "Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions". In: *IEEE Communications Magazine* 52.5 (2014), pp. 86–92.

[68] Ievgenii Tsokalo. "Opportunistic routing with network coding in powerline communications". PhD thesis. Technische Universität Dresden, 2017.

[69] *Ubiquiti SR71-E Data Sheet*. `http://site.microcom.us/sr71_e_ds.pdf`. Accessed: 2019-10-21.

[70] Tai Wang, Bo Li, Zongkai Yang, and Wenqing Cheng. "A minimized latency broadcast in multi-rate wireless mesh networks: Distributed formulation and rate first algorithm". In: *2007 IEEE International Conference on Multimedia and Expo*. IEEE. 2007, pp. 1786–1789.

[71] Yang WenZhong et al. "A reliable multicast for MANETs based on opportunistic routing and network coding". In: *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*. IEEE. 2010, pp. 540–545.

[72] Jeffrey E Wieselthier, Gam D Nguyen, and Anthony Ephremides. "Energy-efficient broadcast and multicast trees in wireless networks". In: *Mobile networks and applications* 7.6 (2002), pp. 481–492.

[73] Andreas Willig, Kirsten Matheus, and Adam Wolisz. "Wireless technology in industrial networks". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1130–1151.

[74] *Wireless LAN Design Guide for High Density Client Environments in Higher Education*. https://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1250-series/design_guide_c07-693245.html. Accessed: 2019-10-21.

[75] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". In: *IEEE Std. 802.11ax* (1999).

[76] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment Enhancements for High Efficiency WLAN". In: *IEEE Draft P802.11ax/D4.0* (2019).

[77] Saleh Yousefi, Mahmood Fathy, and Saeed Bastani. "Vehicular Ad Hoc Networks: Current Issues and Future Challenges". In: *Mobile Ad Hoc Networks: Current Status and Future Trends* (2012), pp. 329–377.

[78] Chih-Min Yu, Shao-Kai Hung, and Yu-Chih Chen. "Forming mesh topology for Bluetooth ad hoc networks". In: *2013 IEEE International Symposium on Consumer Electronics (ISCE)*. IEEE. 2013, pp. 123–124.

[79] Yuan Yuan et al. "ROMER: Resilient opportunistic mesh routing for wireless mesh networks". In: *IEEE workshop on wireless mesh networks (WiMesh)*. Vol. 12. 2005.

[80] Qi Zhang and Dharma P Agrawal. "Dynamic probabilistic broadcasting in MANETs". In: *Journal of parallel and Distributed Computing* 65.2 (2005), pp. 220–233.

[81] Xin Zhao et al. "A high-throughput routing metric for reliable multicast in multi-rate wireless mesh networks". In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 2042–2050.