CHRISTIAN KÖBEL

A RESOURCE-MANAGEMENT SYSTEM FOR TRANSMISSION ENHANCEMENT AND CHANNEL DIVERSITY EXPLOITATION IN WIRELESS MESH NETWORKS

UN SISTEMA DE GESTIÓN DE RECURSOS PARA MEJORAR LAS TRANSMISIONES Y EXPLOTAR LA DIVERSIDAD DE CANALES EN REDES MALLADAS INALÁMBRICAS

A RESOURCE-MANAGEMENT SYSTEM FOR TRANSMISSION ENHANCEMENT AND CHANNEL DIVERSITY EXPLOITATION IN WIRELESS MESH NETWORKS

UN SISTEMA DE GESTIÓN DE RECURSOS PARA MEJORAR LAS TRANSMISIONES Y EXPLOTAR LA DIVERSIDAD DE CANALES EN REDES MALLADAS INALÁMBRICAS

CHRISTIAN KÖBEL, DIPL.-ING.

A Dissertation Submitted in Partial Fulfillment of
the Requirements for the Scientific Degree of
Doctor of Engineering Science

Departamento de Telecomunicaciones
Facultad de Eléctrica
Instituto Superior Politécnico José Antonio Echeverría (ISPJAE)

Friedberg (Germany) and Havana (Cuba)
April 2015 – version 2.7

Cujae
Instituto Superior Politécnico
José Antonio Echeverría

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN
UNIVERSITY OF APPLIED SCIENCES

Christian Köbel: *A Resource-Management System for Transmission Enhancement and Channel Diversity Exploitation in Wireless Mesh Networks,* A Dissertation Submitted in Partial Fulfillment of
the Requirements for the Scientific Degree of
Doctor of Engineering Science, © April 2015

supervisors:

Prof. Dr.-Ing. Walter Baluja García

Prof. Dr.-Ing. Joachim Habermann

location:

Friedberg (Germany) and Havana (Cuba)

date:

April 2015

It is intended that the pages 44, 103, 105, 112, 115, 116, 117, 121, 122, 123, 124, 125, 126, 128, 133, 134, 136, 185, 203, 204, 206, 211 and 212 are printed out in color.

Für Maria und meine Familie…

ABSTRACT

Wireless Mesh Networks (WMN) have been in the focus of extensive research for more than one decade now. However, with a growing network size, routing behavior and performance become unpredictable and less reliant on longer multi-hop routes. The result is insufficient transmission capacity. In addition, vertical traffic lacks protection against horizontal traffic. These flaws threaten to create acceptance of this technology among professionals. Mesh nodes equipped with multiple WirelessLAN interfaces may significantly improve these conditions, along with hardware investments in an economically sustainable range; a fact which created much attention in the research community. Based on recent findings on multi-interface WMN design, the presented work describes the next steps towards an efficient resource management. If non-overlapping channels are used for communication, the system enables an optimal usage of the 802.11 spectrum on layer 2. To manage bundles of multiple WLAN links between mesh neighbors, a modified node architecture and a novel middle-layer software module have been created. Hop-to-hop load balancing in a bundle is included in each node. Packet scheduling is performed based on a set of pre-defined load balancing modes. These modes introduce awareness of current network conditions and cover a wide variety of requirements on mesh networks; from improved performance to robustness. Further inspiring technologies, like layer 2 forwarding and hop-top-hop priority queuing, have been tailored in the novel architecture. The achieved result is a flexible, multi-purpose platform, ranging from a commercially oriented mesh backbone to spontaneously set up emergency networks. A homogeneous mesh backbone is investigated and studied as the object of research, to which clients on lower levels can connect to. A set of simulator-driven measurements outline the effectivity of the multi-interface system.

RESUMEN

Desde hace más de una década han sido las redes malladas inalámbricas (Wireless Mesh Networks - WMN) foco de numerosas investigaciones. Sin embargo, con el crecimiento de la red, el comportamiento del enrutamiento y las prestaciones de la red se vuelven impredecibles y menos confiables. El resultado es una capacidad insuficiente de transmisión. Además el tráfico vertical carece de protección contra el tráfico horizontal. Este hecho amenaza la aceptación de la tecnología entre profesionales. Nodos mallados equipados de múltiples interfaces WirelessLAN podrían mejorar significativamente estas condiciones, sin significar mayores inversiones económicas, lo que hace llamar la atención sobre este tipo de investigaciones en la comunidad científica. El presente trabajo describe los pasos para una eficiente gestión de los recursos basados en recientes descubrimientos en el diseño multi-interface WMN. Este sistema permite, en el caso de una comunicación de canales no solapados, un uso óptimo del espectro 802.11 en la capa 2. Para manejar paquetes de múltiples enlaces WLAN entre mallas vecinas se ha diseñado una arquitectura de un nodo modificado y un novedoso módulo de la capa 2.5. En cada nodo está incluido el balance de carga en un paquete, de un salto a otro. La organización de paquetes se aplica basada en un set de modos predefinidos. Los modos actúan sensibles a las condiciones de la red y cubren una amplia variedad de requerimientos en redes malladas; de un mejor desempeño a una mayor robustez. Otras tecnologías que inspiran este trabajo, como el reenvío en la capa 2 y colas de prioridades, han sido incorporadas en esta novedosa arquitectura. El resultado es una plataforma flexible que puede ser utilizada para diferentes propósitos, desde una malla backbone de orientación comercial hasta una red inalámbricas de espontánea conformación. Como objeto de investigación fue estudiado un backbone homogéneo. Un conjunto de mediciones basadas en un simulador muestran la eficacia del sistema multi-interface.

No man ever steps in the same river twice,
for it's not the same river
and he's not the same man.
— Heraclitus

ACKNOWLEDGMENTS

Words of gratitude cannot express what the following people deserve. Their presence has constantly shaped my future in the most positive way. My humble thanks go to…
my wife Maria, my true light of hope and my fountain of motivation,
Walter and his family, for all his patience and the countless hours of support. In brief, for simply all he has done for me,
Prof. Dr. Habermann, for all the irreplaceable guidance, help and generous support,
my families in Germany and Cuba, for providing a place of ease, for whenever I was in need for serenity,
Alain and his family, for always giving me valuable advice in all these years,
my fellow students, Hongmei, Johannes, Yeni and all the rest,
all my friends at CUJAE, especially Javier Muñoz, Javier Quintana, Jorge, Lili, Jeffrey, Larisa, Julio and Karen,
the members of the team in the telecommunications laboratory at THM,
and Nico at Deutsche Telekom.
My special thanks go to Wolf-Rainer and Mario; without them *none* of this would have been possible.
Last but not least to all those friends not mentioned, but never forgotten.
Beyond the thesis, this episode gave me some friends for life, which is the greatest gift.
Ich danke außerdem meiner Mutter Anna, meinem Vater Erwin, meinem Bruder Timo und meinem Opa Willi und meiner Tante Anni. Außerdem Irene und Franz, die uns fern, und doch so nah sind.

CONTENTS

INTRODUCTION 1
1 CHAPTER 1: TECHNOLOGICAL ANALYSIS OF SOLUTIONS TO ENHANCE MESH COMMUNICATION, WITH A FOCUS ON CHANNEL DIVERSITY AND QOS 10
1.1 Introduction . . . 10
1.2 Wireless Mesh Networks . . . 10
1.2.1 Definition . . . 10
1.2.2 Taxonomy . . . 11
1.2.3 Deployment . . . 11
1.2.4 Interference . . . 13
1.2.5 Heterogeneous Traffic Flows . . . 16
1.2.6 Summary of Problems . . . 17
1.2.7 Routing Protocols . . . 18
1.2.8 Routing Metrics . . . 21
1.3 IEEE 802.11 Wireless LAN . . . 25
1.3.1 Medium Access Control . . . 26
1.3.2 Physical Layer . . . 28
1.4 Quality-of-Service . . . 30
1.4.1 Differentiated Services . . . 30
1.4.2 Sub-Layer 3 Forwarding . . . 31
1.4.3 Packet Queues . . . 33
1.4.4 General Packet Scheduling Mechanisms . . . 33
1.4.5 Specific Solutions for Wireless Mesh Networks . . . 34
1.5 Multi-Interface Multi-Channel Wireless Mesh Networks . . . 35
1.5.1 Management Approaches for MIMC Wireless Mesh Networks 35
1.5.2 Channel Assignment . . . 36
1.5.3 Load Balancing in MIMC Wireless Mesh Networks . . . 38

1.6 Conclusion 40
2 CHAPTER 2: A RESOURCE-MANAGEMENT SYSTEM FOR TRANSMISSION ENHANCEMENT AND CHANNEL DIVERSITY EXPLOITATION IN WIRELESS MESH NETWORKS 42
2.1 Introduction 42
2.2 Containment of Target Mesh Networks 42
2.3 Overview 43
    2.3.1 System Architecture and Functioning Overview 45
    2.3.2 Process Point of View 48
2.4 Mesh Routing Protocol 50
    2.4.1 Node Identity with Multiple Radios 50
    2.4.2 Topology Analyzer 51
    2.4.3 Link Cost 51
2.5 Traffic Analyzer 51
    2.5.1 Packet Classifier 52
    2.5.2 Flow Identifier and Monitoring 53
    2.5.3 Process Output and Algorithm 53
2.6 Channel Allocator 53
    2.6.1 Requirements 56
    2.6.2 CA Table Evaluator 58
    2.6.3 The Role of Channel Switch Transition 62
    2.6.4 Control Channel 62
2.7 Traffic Engineering Labeler 63
    2.7.1 Design Constraints 63
    2.7.2 Algorithm 64
    2.7.3 Label Format 66
    2.7.4 Label Assembly of NHF 68
    2.7.5 Label Assembly of QSF 77
    2.7.6 Commutation Table 79
2.8 Multi-Hop Radio Resource Manager 79
    2.8.1 Virtual Interface and Bundle Management 81

2.8.2 Label-based Multi-Hop Packet Commuter 86
2.8.3 Intra-Bundle Queues 86
2.8.4 TX Packet Scheduler 91
2.8.5 Design Constraints 93
2.9 Software Implementation 95
2.10 Hardware-Platform Recommendations 96
2.11 Conclusion 97
3 CHAPTER 3: EVALUATION OF THE PROPOSED SYSTEM 100
3.1 Introduction 100
3.2 Overview of Implemented Components 100
3.3 Simulations 102
3.3.1 Quality-of-Service and Priority Queueing 105
3.3.2 Vertical Traffic in a Mesh Network 111
3.3.3 Multi-Modal Load Balancing 118
3.3.4 Layer 2 Forwarding 131
3.3.5 Dynamic Channel Changes 135
3.4 Final Analysis of Results 137
3.5 Conclusion 139
CONCLUSION 142
RECOMMENDATIONS 144

BIBLIOGRAPHICAL REFERENCES 145
ACRONYMS 170
A APPENDIX 178
A.1 Further Related Work 178
A.2 Conceptual Amendments and Discussions 180
A.2.1 The Mesh Routing Protocol of Choice 180
A.2.2 Common Control Channel 180
A.2.3 Additional CA Protocol Requirements 181
A.2.4 GW Presence in a Standard OLSR Toplogy 183
A.2.5 Alternative Transport Method for M-LDM Messages 183

A.2.6 Examples for Labeled Packet Communication 184
A.2.7 Extensions to Include Receiver Bundle Feedback and Packet Reordering 186
A.2.8 Determination of the TX Probability with WFS 188
A.2.9 WFS Output 188
A.2.10 Distinctive Features of RR-Extended Mode and Alternative Parameters 189
A.3 Simulation Tools 191
A.3.1 Environment 191
A.3.2 Interference Generators 192
A.4 Main Configuration Parameters 194
A.4.1 Quality-of-Service and Priority Queueing 194
A.4.2 Vertical Traffic in a Mesh Network 195
A.4.3 Multi-Modal Load Balancing 196
A.5 Additional Measuremens 200
A.5.1 Single-Interface Radio Performance 200
A.5.2 MAC Parameters 205
A.5.3 Single-Interface Multi-Hop Environment 206
A.5.4 IP Processing as a Discrete Event 213
A.5.5 Channel Map Input for Scenario Manager 215

LIST OF FIGURES

Figure 1.1 Mesh backbone and adjacent network hierarchy levels 11
Figure 1.2 Taxonomy of multi-hop networks 11
Figure 1.3 A unifying wireless mesh network 13
Figure 1.4 Interference classes in WMNs 14
Figure 1.5 Intra-route interference with two streams 15
Figure 1.6 Relation between effects, problems and necessities in WMNs 17
Figure 1.7 802.11 virtual carrier sensing with RTS-CTS handshake enabled 27
Figure 1.8 MPLS header 31
Figure 2.1 Bottlenecks caused by insufficiently equipped gateways . . . 43
Figure 2.2 System architecture and overview 44
Figure 2.3 Label-based forwarding for packets of a GW flow 47
Figure 2.4 Block types used in flow diagrams 48
Figure 2.5 Level 1 process flow 49
Figure 2.6 Traffic analysis process 55
Figure 2.7 Evaluation of CA table process 61
Figure 2.8 Labeling of layer 3 packets process 65
Figure 2.9 De-labeling and correction of layer 2 packets process 66
Figure 2.10 Custom header 67
Figure 2.11 M-FIB maintenance process - locally generated destinations (part 1) 72
Figure 2.12 M-FIB maintenance process - locally generated destinations (part 2) 73
Figure 2.13 M-FIB maintenance process - destinations obtained from signaling input 74
Figure 2.14 M-LDM header 75
Figure 2.15 Example of flexible label treatment 80

Figure 2.16 Example of flexible label treatment (numerical example) . . . 80
Figure 2.17 LMHPC – Packet commutation process 82
Figure 2.18 The virtual interface 83
Figure 2.19 Enqueue process 87
Figure 2.20 Dequeue process 88
Figure 2.21 Round Robin scheduling with fallback extension process . . . 94
Figure 2.22 Conditional protection of vertical flows 95
Figure 3.1 Implementation of the custom node 103
Figure 3.2 Encapsulation with TEL 104
Figure 3.3 Scenario for testing middle-layer packet queues 105
Figure 3.4 Multi-hop UDP performance with two different queue weight distributions, using RR PS mode, $C = 5$ 108
Figure 3.5 Multi-hop UDP performance with two different queue weight distributions, using RR PS mode, $C = 8$ 109
Figure 3.6 Multi-hop UDP performance with two different queue weight distributions, using a single radio, $C = 8$ 110
Figure 3.7 Scenario for testing a mesh grid 112
Figure 3.8 Grid performance with varying amount of radios and 1 gateway node 115
Figure 3.9 Grid performance with varying amount of radios and 2 gateway nodes 116
Figure 3.10 Grid performance with varying amount of radios and 4 gateway nodes 117
Figure 3.11 Scenario for testing WFS PS mode 119
Figure 3.12 Scenario for testing Extended RR PS mode 120
Figure 3.13 MAC loss rate for testing Extended RR mode with 1 attached radio 120
Figure 3.14 Packet distribution for testing WFS mode, datagram size 500B, with background traffic TX rate @ 3 Mbit/s 121
Figure 3.15 Packet distribution for testing WFS mode, datagram size 500B, with background traffic TX rate @ 6 Mbit/s 122

Figure 3.16 Packet distribution for testing WFS mode, datagram size 1kB, with background traffic TX rate @ 3 Mbit/s 122

Figure 3.17 Packet distribution for testing WFS mode, datagram size 1kB, with background traffic TX rate @ 6 Mbit/s 123

Figure 3.18 Packet distribution for testing WFS mode, datagram size 1.5kB, with background traffic TX rate @ 3 Mbit/s 123

Figure 3.19 Packet distribution for testing WFS mode, datagram size 1.5kB, with background traffic TX rate @ 6 Mbit/s 124

Figure 3.20 Throughput comparison with Extended RR mode and threshold rate $R = 0.1$ 125

Figure 3.21 Throughput comparison with Extended RR mode and threshold rate $R = 0.2$ 126

Figure 3.22 Throughput comparison with Extended RR mode and threshold rate $R = 0.3$ 126

Figure 3.23 Observation of packet distribution for 1-hop case, with $R = 0.1$ 127

Figure 3.24 RX throughput of `mihost[1]` for testing Extended RR mode with multiple radios 128

Figure 3.25 Sent packets per radio for testing Extended RR mode with 3 attached radios and threshold rate $R = 0.2$ 129

Figure 3.26 Scenario for testing layer 2 forwarding 133

Figure 3.27 RTT with Ping and a set processing delay of 125µs 133

Figure 3.28 UDP throughput 134

Figure 3.29 Impact of IP layer processing delay in INETMANET 134

Figure 3.30 Influence of convergence and safe period in a dynamic switch scenario 136

Figure A.1 Example of GW presence in topology 183

Figure A.2 The generic OLSR packet 184

Figure A.3 Label operations in an example 185

Figure A.4 Example of filled M-FIB tables 187

Figure A.5 Scenario for testing PHY layer response 200

Figure A.6 Loss rate comparison with single-radio connection 203

Figure A.7 UDP and TCP throughput with single-radio connection for PHY testing 204
Figure A.8 Scenario for testing a single-radio mesh route 206
Figure A.9 UDP and TCP throughput with single-radio connection in a standard WMN 211
Figure A.10 Packet distribution of each hop when no cross-traffic is active 212
Figure A.11 Packet distribution of each hop when two cross-traffic streams are active 212
Figure A.12 TCP throughput of cross-traffic 213

LIST OF TABLES

Table 1.1 Mesh Routing Protocols and Supported Metrics 18
Table 1.2 DiffServ PHB types 31
Table 1.3 Ethernet Bonding Driver PS modes 34
Table 1.4 Distribution Table 38
Table 2.1 Used DSCP range 52
Table 2.2 Differences between MPLS and the system-specific label-based forwarding 64
Table 2.3 Usability of MPLS header fields 67
Table 2.4 Label commutation operations 68
Table 2.5 Mesh Forwarding Information Base table 69
Table 2.6 Overview of label construction guidelines for M-FIB entries . 71
Table 2.7 Queue mapping in forwarding label 78
Table 2.8 Simple Bundle Table 84
Table 2.9 Bundle Management Table 85
Table 3.1 Selection of performance indicators 104
Table 3.2 Queue weight distribution beneficial for stream 2 107

Table 3.3 Equal queue weight distribution 107
Table 3.4 Traffic constellations for the mesh grid scenario 111
Table 3.5 Comparison of sent packet amounts with single radio and WFS mode 125
Table 3.6 Time-variant channel map 135
Table A.1 Parameters for queue testing 194
Table A.2 Routing metric configuration in OLSR 195
Table A.3 Exemplary parameters for UDP basic burst and sink app . . . 195
Table A.4 Parameters for vertical traffic simulation 196
Table A.5 Exemplary parameters for TCP basic client and generic server app 197
Table A.6 Common parameters for multi-modal load balancing simulation 197
Table A.7 Parameters for evaluation of WFS PS mode 198
Table A.8 Parameters for evaluation of Extended RR PS mode 199
Table A.9 Parameters for SI Radio Performance simulation 202
Table A.10 General MAC parameters 205
Table A.11 Parameters for multi-hop route simulation 207
Table A.12 Exemplary parameters for TCP session and sink app 208
Table A.13 Exemplary parameters for UDP video stream-server and stream-client app 209
Table A.14 Exemplary parameters for ICMP Ping app 210

INTRODUCTION

PROBLEM SITUATION

The bandwidth demand in Internet Protocol (IP)-based telecommunication services is increasing constantly [1], [2]. This is due to the rising number of users and end-devices (especially wireless end-devices), cheaper hardware prices and lower rates for using access- and delivery networks. At the same time, IP-based services get more complex [3]. More and more digital media content, like voice, video, IPTV, pictures and video game streaming, is exchanged world-wide. Technologies to display and process this data require more advanced (web) standards, like Hypertext Markup Language (HTML) 5, JavaScript or Cascading Style Sheets (CSS) 3, which users quickly adapt to.

To keep up with the demand, distributed routing engines need to be able to handle many end-to-end connections on a global scale, providing each end-user with ubiquitous high-level access. This requires an efficient management by an Exterior Gateway Protocol (EGP) for large-scale networks, such as the common Border Gateway Protocol (BGP). Whereas these global backhaul networks offer quasi-redundant bandwidth capacity [4], requirements on the other (local) “end” of the network are different.

Often, last mile networks tend to become bottlenecks in the overall Internet communication structure, because they have to fulfill direct user demands, in terms of sufficient bandwidth levels and Quality-of-Service (QoS). This work concentrates on last mile *wireless* networks [5], which can be direct user-to-user networks, wireless backbones (e. g., a public city-wide network) or both, in a hybrid form [6]. Wireless and mobility become more and more important today. The spreading use introduces new challenges. Wireless spectrum and thus capacity is often limited, despite the rising spectral efficiency [7]. Maximum bit rates are stretched by Physical (PHY)-Layer improvements like Orthogonal Frequency-Division Multiplexing (OFDM), Multiple

Input Multiple Output (MIMO) and complex modulation schemes and codes. Especially popular standards, like Long Term Evolution (LTE) in the cellular world, and IEEE 802.11 Wireless Local Area Network (WLAN) as one of the most applied standards for wireless end-devices, are constantly enhanced. An infrastructure-based WLAN cannot be dynamically extended and lacks scalability with an increasing number of active devices [8]. A Wireless Mesh Network (WMN) overcomes these disadvantages [6]. Originally developed for pure military purposes, wireless mesh networks have long conquered the world of open source networking. The technology is mostly used to create economic and flexible wireless backbones, which are often maintained by communities (for a list of projects see [1]). The work targets such backbone setups. Planners of wireless consumer and industry networks have seen the various advantages and diverse applications of WMNs and have slowly begun to adapt the technology to present market solutions. However, a broad market acceptance has not been reached yet. One reason is the fact that WMNs are mostly based on *nodes* (members of an ad-hoc network) equipped with a single WLAN Interface (IF) [9]. A single-channel WMN suffers from the same risks of negative channel conditions on the PHY layer, like fading or distortion effects in a non-line-of-sight situation. Such effects ultimately turn the pure throughput of a WLAN IF into a highly conditional parameter. But in wireless multi-hop networks (e. g., WMNs), there are *other significant factors* which may drastically limit the *transmission capacity in WMNs*. WLAN is not full duplex [10], which causes a rapid performance and capacity degradation on multi-hop routes [11], [12]. Also, 802.11 Medium Access Control (MAC) is designed for shared-channel communication [13] and is partly based on random timers, making a consistent packet forwarding unreliable. Shared parts of a route are prone to congestion and unfair traffic treatment [5]. Finally, routes separated on layer 3 might still interfere on the same collision domain [5]. Apart from *various interference types* with WLAN, traffic in a WMN is often heterogeneous, because users create mostly vertical traffic [14]. This leads to *congestion* [6] near those nodes in a mesh, which serve as traffic portals to external networks or the Internet. Also, end-to-end routes pointing to an external gateway generally have to carry more

---

[1] Community WMNs, `https://personaltelco.net/wiki/WirelessCommunities`

traffic. Vertical traffic is not protected on those routes and has the same priority as intra-mesh traffic.

Initially there exist no forms to exploit multiple radios for parallel data transmission between nodes when a standard mesh routing protocol is deployed. The chosen protocol selects one radio to establish a link for communication with its neighbors. In the standard case, neither the selection of the radio, nor of the channel is managed. A common approach is to select the first radio index, regardless of its channel condition. Thus, having n multiple physically attached radios might imply that $n-1$ radios are registered in the Operating System (OS), but are not used [15]. However, one of the remaining radios could be tuned to a more suitable channel; offering less traffic congestion or a potentially higher bandwidth. *Awareness* of a node's network environment, global topology, passing traffic, its link states and its hardware resources is a crucial success factor for a Multi-Interface Multi-Channel (MIMC)-based WMN. For its design, methods need to be defined which create and exploit awareness in a sensible and dynamic way, in order to achieve the optimal capacity, or to improve certain other characteristics of the network.

All these constraints cause that standard WMNs have a limited transmission capacity. This issue moves WMNs further away from users expectations on a modern *delivery* network / backbone: High down- and uplink rates, and high accessibility and reliability.

SCIENTIFIC PROBLEM

Based on the mentioned limitations, *how to overcome the inefficient and limited use of multiple radios, in order to enhance the transmission capacity in WMNs* describes the scientific problem of this work.

The simultaneous usage of multiple orthogonal WLAN channels could become a feasible method to overcome limitations in WMNs [16]. This requires few extra investments in additional WLAN radios. Due to the broad acceptance of the standard in industry, prices for 802.11x interfaces are dropping.

However, additional network- and traffic information needs to be taken into account.

STARTING POINT AND TRANSITION OF TECHNOLOGY

There are three consecutive steps towards a fully coordinated, distributed management scheme which tackles the aforementioned limitations in WMNs and which involves all available radios [17]:

1. Single-Interface Single-Channel (SISC): Used in standard WMN setups; adapted solutions are available in research.
2. Single-Interface Multi-Channel (SIMC): Currently used channel can be switched according to its conditions, or Frequency-Division Multiple Access (FDMA) can be deployed. However, it is still single-channel communication.
3. MIMC: Multiple radios run on separate channels and all are used. Each radio can switch its channel, but FDMA is not deployed.

Numerous approaches exist in research for all three categories. The presented work roams within the third category. SISC solutions such as [18] alter medium access in favor of prioritized traffic or to decrease interference. Others in category 3 specialize on a limited, *fixed* or small number of radios, such as [19], [20]. Others such as [21] rely on commuting channels over multiple hops. Few approaches actually consider the aggregation of capacities per hop.

Exploiting channel diversity in a WMN without having *centralized* management instances is a challenging task, which can be generally split into *Channel Assignment (CA)* and *Load Balancing (LB)*. When two nodes within the same coverage range share common channels, all radios running on these channels can be used simultaneously. This temporarily shared set of radios shall be called a *bundle*. This raises the question of how to effectively schedule packets within a bundle by applying a LB scheme. Both management tasks can be performed independently and in a distributed manner, yet a CA scheme needs to be run at least once before applying LB. Also, LB only takes effect when two neighboring nodes can afford to share more than one interface for intercommunication (i. e., a bundle), either via exclusive channels or via channels shared with other neighbors. Both CA and LB can be implemented with fixed distribution schemes or by considering link states. This work follows the latter approach. Apart from traffic-sensitive LB, other methods to engineer and protect Gateway (GW) traffic need to be evaluated.

OBJECT OF RESEARCH

The described research work has been conducted in transmission quality and capacity in wireless mesh networks for a backbone or client network deployment.

FIELD OF RESEARCH

More specifically, the investigation centers in an innovative combination of radios and channels with available network input, in order to enhance the performance and efficiency of communications in WMN backbones.

PRIMARY RESEARCH GOALS

The main objective is to *design a holistic middle-layer system that combines multiples techniques in an innovative way, in order to efficiently utilize radio resources. Doing so enhances transmission capacities in WMNs, which also favors vertical traffic*. An efficient utilization of resources is achieved by distributing load over dynamically changing channels. The choice of channels and radios to favor for communication shall depend on measured environmental network parameters and pretended prioritization of network traffic. The described system ultimately improves the performance of *single-path* transmissions.

Based on the primary objective, the following results can be achieved:

- Improve capacity, bandwidth and packet delay levels in a standard WMN backbone
- Protect and prioritize gateway traffic

SECONDARY RESEARCH GOALS

The following intermediate outputs represent the secondary, partial objectives:

- To design a LB algorithm, which schedules packets in a bundle, based on a chosen *mode* or desired network characteristic

- To design a distributed algorithm which constantly assigns equipped radios to bundles and which provides a virtual WLAN interface to layer 3 and above
- To define the mesh-wide treatment of individual packets

PRELIMINARY HYPOTHESIS

The availability of methods for LB and treatment of GW traffic rises the essential challenge of how to effectively arrange them, in order to create a *holistic* MIMC node system and to respond to the scientific problem. The goal of this composition is to selectively load bundled radios, in order to improve bandwidth capacity and to regard quality-related parameters in a mesh network. So by combining multiple WLAN radios via a bundled interface, considering vertical traffic and by designing a system which enables both aspects, combining in a novel way techniques like CA, traffic engineering and others, it is possible to achieve better performance, QoS, and higher capacity levels in WMN communications, without modifying the PHY layer in the node architecture.

TASKS DURING THE RESEARCH PHASE

In the course of this work, the following tasks have been performed.

- Study of involved Wireless Access Technology (WAT) standards, mesh routing protocols and mesh routing metrics
- State-of-the art analysis of related solutions and achieved results in external works
- Analysis of related technologies, such as priority queuing and label-based packet forwarding
- Design of the overall system, including several components
- Design of the interface (address) management and the related virtual interface
- Design of a bundle management algorithm which constantly calculates load balancing

- Design of traffic-engineering components
- Definition of a set of load balancing modes
- Implementation and validation of the system in a simulation environment

METHODS OF RESEARCH

A theoretical (hypothetical-deductive) method was used to conform the hypothesis. A systemic method was applied for the overall conception of the system and the relation between its components.

A practical (experimental, empirical and heuristic) method was applied to obtain the presented results. Besides a state-of-the-art study and the elaboration of the hypothesis, simulations are realized.

SCIENTIFIC NOVELTY

For the first time, various radios are combined. This has been realized by assembling standard schemes and components in a *highly innovative manner* and by analyzing the resulting approach in detail. Involved standard technologies include mesh routing, QoS and traffic engineering, routing topology analysis, priority queuing and load balancing.

Furthermore, more general network parameters are improved, such as the overall performance and reliability of a wireless mesh backbone. Improvements are achieved in comparison with single-radio mesh networks, or unmanaged multi-radio mesh networks. To achieve this improvement, a custom middle-layer module has been designed, which processes cross-layer information.

PRINCIPAL CONTRIBUTIONS

The dissertation contributes aspects of the following types:

- Conceptual and theoretical:
    - Proposal of a node management system to combine radios in WMNs
    - Holistic / selective integration of established scheduling-, mesh routing-, traffic engineering- and QoS- methods

  - Middle-layer concept module with a focus on modularity. This concerns both module-internal features (e. g., traffic engineering and LB work independent from another) and external components, such as the chosen routing- or CA- protocol, or the 802.11x standard

- Practical and experimental:
  - Enhanced transmission capacities and resource exploitation in a WMN, under certain network-, traffic- and resource conditions
  - Transmission enhancement allows to consider the prioritization of vertical traffic and QoS-demands

- Socio-economical:
  - Except for an arbitrary amount of additional, standard 802.11x radios, theoretically there is no specialized hardware required. The system becomes an economic alternative for realizing high-performance WMNs built upon commodity hardware
  - User-defined parameters and LB modes allow to shape the network-response type (e. g., increased capacity or robustness). This allows for a multi-purpose WMN: backbones, emergency networks, *rural* community networks, Closed Circuit Television (CCTV), and so on
  - The developed middle-layer module is open for future technical extensions of interested developers and researchers

STRUCTURE OF THIS DOCUMENT

The thesis is structured as follows. Chapter 1 reflects a critical state of the art analysis of the research object, plus concentrated knowledge on such. Chapter 2 explains the core concepts of the system then. The evaluation of the system in Chapter 3 contains relevant measurement results. Finally, a conclusion of the research work is drawn and further recommendations are provided. Appendix A holds secondary discussions on the concept, complementary measurements and details on the simulation environment.

Chapter 1:

Technological Analysis of Solutions to Enhance Mesh Communication, with a Focus on Channel Diversity and QoS

1

CHAPTER 1: TECHNOLOGICAL ANALYSIS OF SOLUTIONS TO ENHANCE MESH COMMUNICATION, WITH A FOCUS ON CHANNEL DIVERSITY AND QOS

1.1 INTRODUCTION

The first Chapter of this thesis contains a critical analysis of the state-of-the-art. Furthermore, relevant knowledge required for subsequent Chapters 2 and 3 is provided in a compact form. The goal of this Chapter now is to outline characteristics of (multi-interface) wireless mesh networks, as well as challenges and unresolved topics in these networks. A WMN setup in an unmodified form, which uses a standard layer 3 routing protocol, can neither support QoS, nor consider different traffic demands; let alone exploit the capacity of multiple radios. Several solutions presented in Chapter 1 take this as a starting point and offer punctual, but less holistic improvements in this field.

1.2 WIRELESS MESH NETWORKS

This section briefly covers those aspects of WMNs, which are the most relevant for this thesis. There is an active WMN research community, and channel diversity is still one of the biggest challenges among this group.

1.2.1 *Definition*

Akyildiz and Wang [6] have provided a general classification of possible roles of mesh nodes in a broad communication network. Their variant of a WMN backbone is depicted in Fig. 1.1. A Mesh Router (MR) is a regular node with forwarding capabilities. A MR with GW depicts a gate to the Internet.

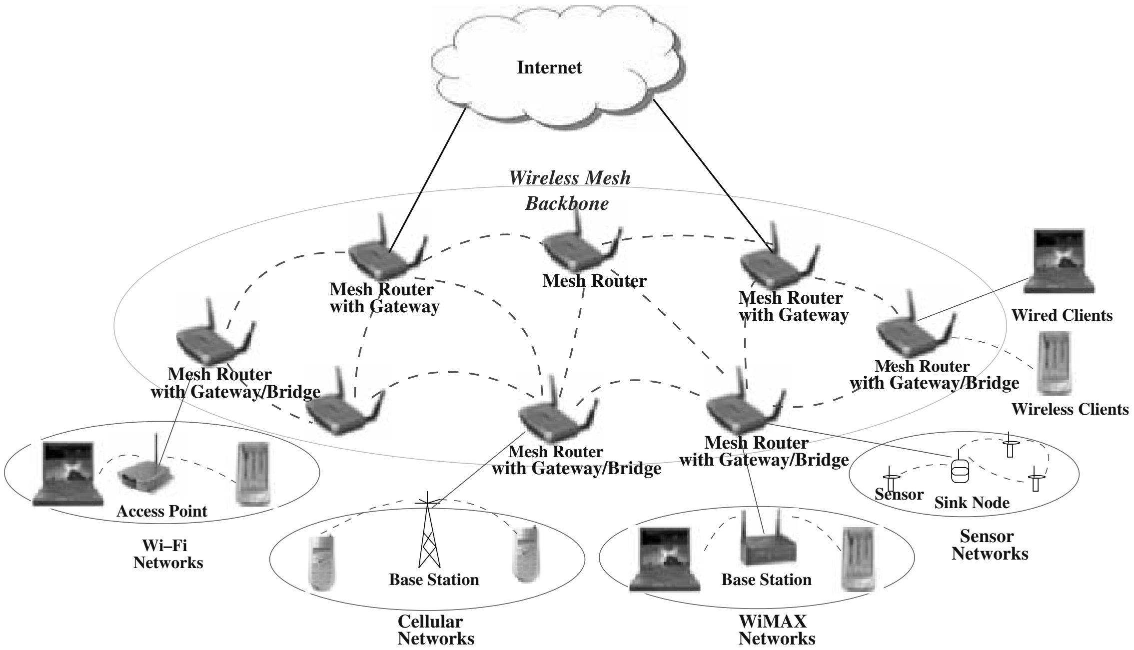


Figure 1.1: Mesh backbone and adjacent network hierarchy levels [6]

1.2.2 *Taxonomy*

WMNs as such depict a sub-category of infrastructure-based multi-hop networks, as shown in Figure 1.2.

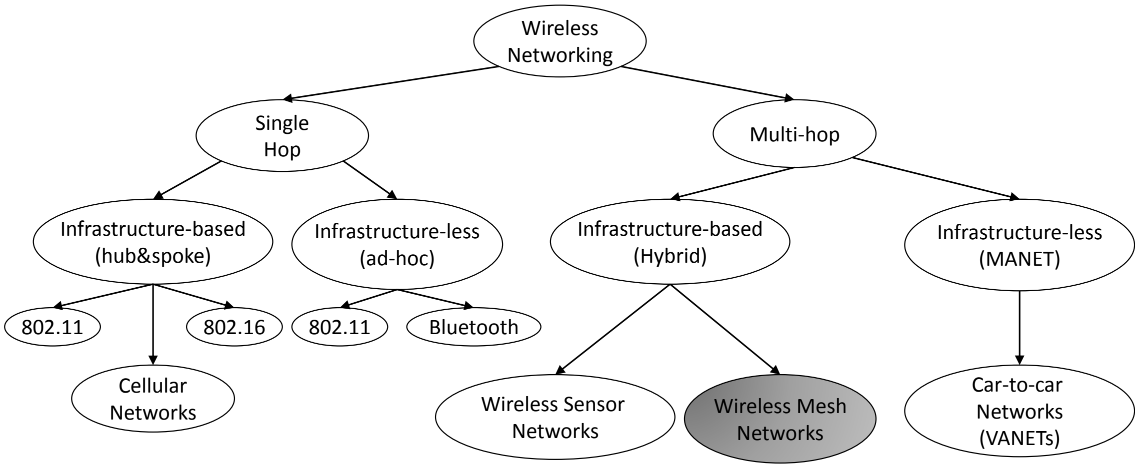


Figure 1.2: Taxonomy of multi-hop networks

1.2.3 *Deployment*

There exist a broad variety of deployment scenarios for WMNs:

- Last-hop access / metropolitan or municipal backbones
    - Mixed, dynamic traffic [22]
    - Often with a commercial background (e. g., operated by an Internet Service Provider (ISP))
- Community-driven WMNs [23]
    - Semi-professional operation and maintenance
    - Optionally carried out by Non-Governmental Organization (NGO)s
    - Seek independent of carriers, e. g.in rural zones with low expected profits
    - Mixed, dynamic traffic
- Emergency networks
    - For basic communication (e. g., Voice-over-IP (VoIP) of public or voluntary forces in disaster scenarios
    - Rapid and temporary formation
    - Require high level of robustness and reliability
- Digital Media / Carrier Grade Networks with a strong focus on QoS and Quality-of-Experience (QoE)
- Mobile Ad-Hoc Network (MANET) / Vehicular Ad-Hoc Network (VANET) for peer-to-peer, car-to-car and/or car-to-roadside communication
- Wireless Sensor Networks
    - Small amounts of data transmitted from sensors
    - Network needs to provide a long endurance

Figure 1.3 depicts a representative example of a WMN which unifies several areas of application.

Internet gateways are connected to classic wired networks and also to satellite networks (which is commonly used in rural areas and in developing countries [23]). Remote locations are directly integrated in the mesh topology via long range links.

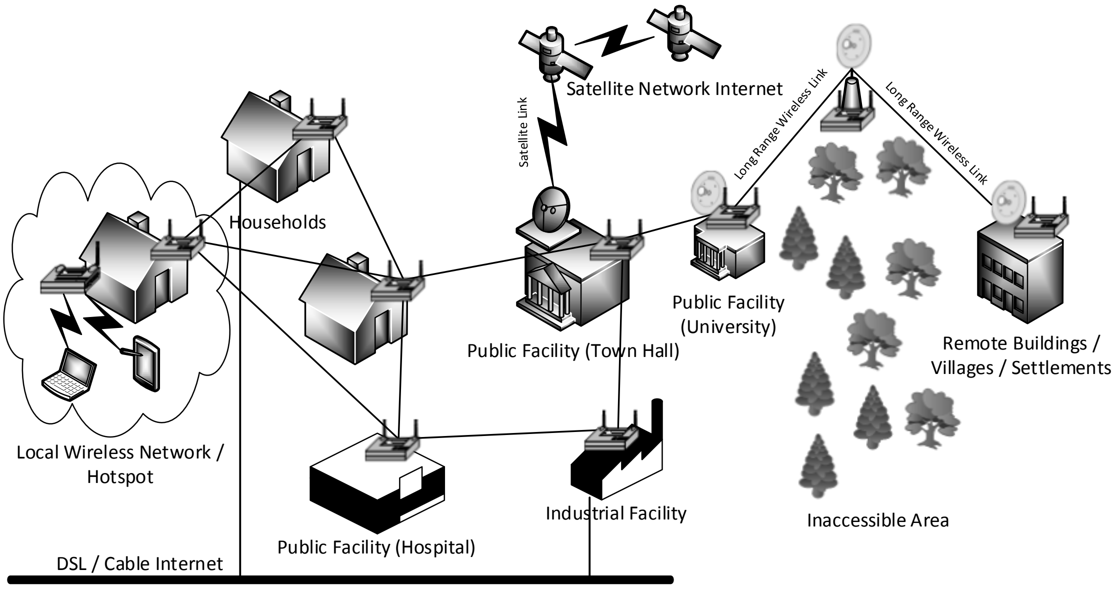


Figure 1.3: A unifying wireless mesh network

Different strategies to structure a WMN topology-wise from the scratch are intensively discussed in [24]. In the context of WMN planning, Ahmed et al. [25] focus on the optimal placement of GWs in a WMN, to provide fair Internet access to all MRs (and their clustered user groups).

1.2.4 *Interference*

Pathak and Dutta [26] describe the negative effects of interference on mesh capacity. To avoid interference is a permanent challenge in wireless networks [27]. Three substantial types of interference [15] have been identified as *relevant* for this thesis. They are listed in Fig. 1.4.

Intra-flow interference occurs within a single end-to-end flow S_1 at each intermediate hop, caused by the lack of full-duplex capability of WLAN. *Intra-route interference* occurs when two streams S_1, S_2 compete on the same route or on shared parts of a route. *Inter-route interference* may occur between two streams S_1, S_2 on different paths, sharing one or more collision domains. All interference types may arise all at once, making the wireless medium almost unpredictable on multi-hop routes. Appendix A.5.3 analyzes multi-hop performance in a SISC WMN.

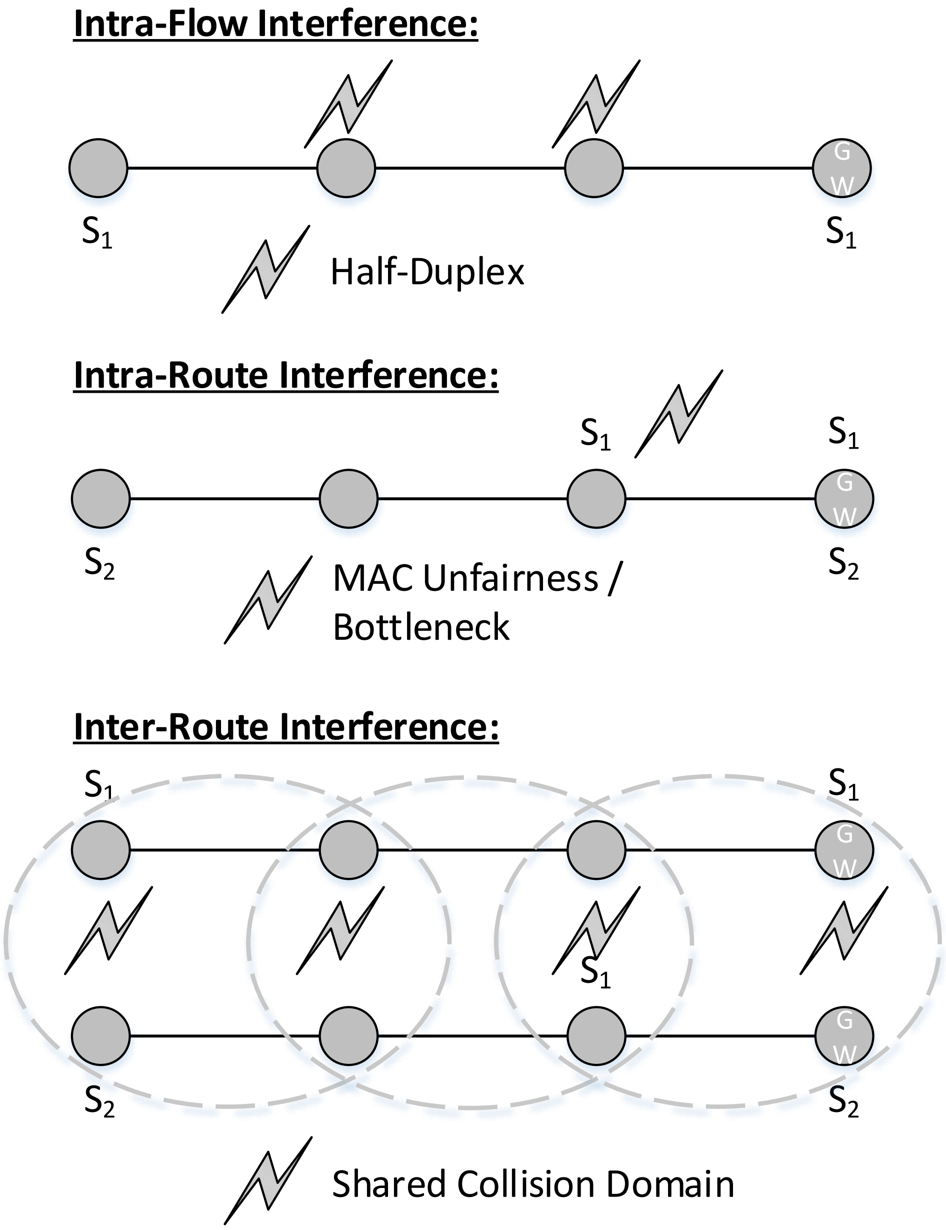


Figure 1.4: Interference classes in WMNs

1.2.4.1 *Intra-Flow Interference*

The half-duplex nature of a Wireless Network Interface Card (WNIC) [10] and the fact, that the receiving channel remains unavailable for use in the next hop, causes serious performance degradation of a single multi-hop flow [11]. A SISC node halves the capacity at each hop [12]. The longer the hop count, the lower the performance [26], [28]. Ramachandran et al. claim three basic statements in this context:

1. Longer path show worse performance than shorter paths.

2. The interference "footprint" is bigger on longer paths.

3. A transmission is more prone to failures on a longer path.

If possible, MRs should be placed close to GWs [11]. This limitation is not desired, as it works against the flexible extension of a WMN.

1.2.4.2 *Intra-Route Interference*

The second case in Fig. 1.4 treats another fundamental problem. The first GW hop carries the entire traffic of S_1, as well as relayed and superimposed traffic of S_2. Fig. 1.5 [5] depicts throughput and load courses in this case. Jun and Sichitiu "define B as the throughput that can be achieved at the MAC layer in a one-hop network with infrastructure (e. g., 802.11 in infrastructure mode)". Load levels of both flows rise until a break point $B/3$, from where on flow S_1 will occupy most of the available bandwidth, and thus starve S_2. Here, WLAN MAC becomes a fairness issue [5], which does not occur in infrastructure WLANs. Measurements in appendix A.5.3.3 reveal how throughput and delay is worsened with every hop. In this context, [5] provide a method to calculate the (available) bandwidth of a WMN, when number and position of nodes are known.

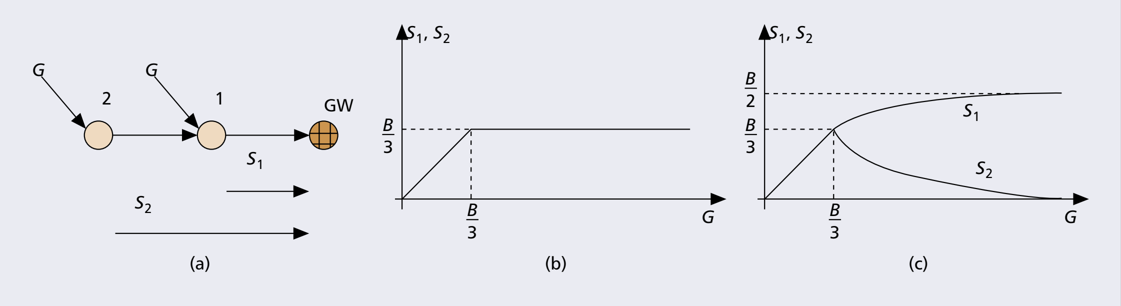


Figure 1.5: Intra-route interference with two streams [5]

1.2.4.3 *Inter-Route Interference*

In the third case in Fig. 1.4, S_1, S_2 don't interfere topology- or routing-wise with each other, but via shared wireless collision domains. Within a domain, both flows con-tend for channel bandwidth. This may cause average throughput of both streams to drop sharply. Later measurements will show that throughput of both streams will not decrease evenly, but more in an "on-and-off" fashion: one stream may consume almost the entire bandwidth, while the other stream is starving. This may randomly swap after a while, in an uncontrollable way. In extreme cases, *some nodes may never get to transmit*. Experiments with single-channel networks in this thesis have revealed, that generally the stream which has started first, has better odds to win the race for bandwidth occupancy. The same behavior can be applied to delay levels. The more collision domains a packet must pass, the less predicable becomes the performance of its carrying flow; additionally to multi-hop effects described in 1.2.4.1.

1.2.5 *Heterogeneous Traffic Flows*

Certain traffic constellations may also cause *limitations* in WMNs.

Without gateways to external networks, a node may exchange data with all available destinations (*horizontal* / intra-mesh traffic). When GWs are present, they typically span a star-shaped traffic flow map. Intermediate MRs closer to GWs have to forward more traffic than distant ones [5]. An infrastructure / backbone is the most common type of application for a WMN [29]. Mesh clients (represented or concentrated by MRs) here consume mostly Internet resources (and use back-channels to the GW, e. g., for Transmission Control Protocol (TCP) synchronization) [14], which is referred to as vertical traffic. In typical public or community mesh setups, most of the traffic is based on TCP, due to the popularity of Hypertext Transfer Protocol (HTTP), peer-to-peer and other web-related services [30], [31]. Paudel et al. [32] provide a case study of mesh traffic, with traffic ratios of popular services like browsing, Skype, and so on. As in most networks, most of the traffic is caused by a small fraction of active users [27], which further outlines the heterogeneous character of mesh traffic.

A key aspect to guarantee a flawless operation in a WMN is to prevent capacity *bottlenecks* near gateways [6]. Links in the 1- and 2-hop neighborhood of a GW are prone to high saturation and to become bottlenecks for all routes relying on this link [33], [28].

1.2.6 *Summary of Problems*

Derived from the identified problems in subsections 1.2.4 and 1.2.5, Figure 1.6 now interrelates the motives for this work, and their origins. If a node has access to more than one channel at once, the total available capacity for this node's passing transmissions rises. The achievable level of efficiency depends on how radio resources are actually used. For instance, to aggregate the capacity of a second radio with a neighbor can be disadvantageous, if this second radio is operating on an already low-performing channel. The derived motivation is thus to define methods, which *optimize* the exploitation of available resources. More total capacity positively affects all traffic. Additionally, a per packet distinction is beneficial, in order to address a heterogeneous traffic situation. The motivation is to *facilitate the transmission* of a packet of a *vertical flow* to its destined next hop by all possible means.

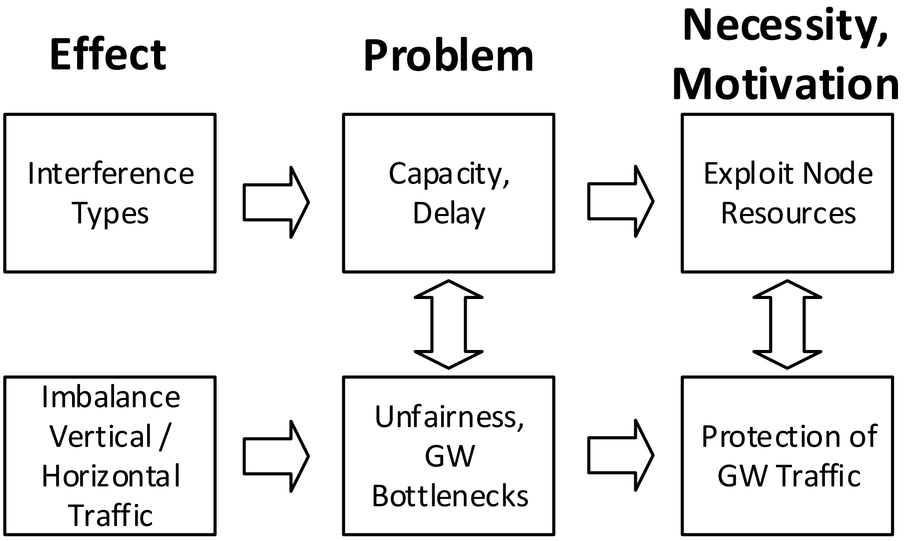


Figure 1.6: Relation between effects, problems and necessities in WMNs

1.2.7 *Routing Protocols*

Mesh protocols are often designed for a special purpose, for example deployment of a stationary wireless backbone. Campista et al. list protocols and their most prominent features in table 1.1 [34]. For a more thorough comparison, the work of Pathak and Dutta [26] is recommended. Since a mesh backbone depicts the *object of research*, the fitness of listed protocols for this environment (with regard to QoS provision, channel diversity and traffic requirements in subsection 1.2.5) is especially considered. Simultaneously, classic and established protocols were chosen (see appendix A.1 for additional ones):

Table 1.1: Mesh Routing Protocols and Supported Metrics [34]

CLASS	PROTOCOLS	METRICS
Ad-hoc based	LQSR	ETX
	SrcRR	ETX
	MR-LQSR	WCETT
Controlled flooding	LOLS	ETX or ETT
	MMRP	Not specified
	OLSR	Hop, ETX, ML, or ETT
Traffic-aware	AODV-ST	ETX or ETT
	Raniwala and Chiueh's	Hop or load-balancing metrics
Opportunistic	ExOR	Unidirectional ETX
	ROMER	Hop or delay

- Optimized Link State Routing (OLSR) [35], [36]
  - Openness for different metrics
  - OLSR supports asymmetric link sensing (i. e., the back-channel for TCP Acknowledgments (ACK) is also evaluated)
  - Topology Control (TC) messages for global link state dissemination are a product of OLSR's unique Multi-Point Relaying (MPR) [35] flooding con-

    cept, which decrease the amount of signaling packets significantly. MPR alsoscales well in all mesh backbone sizes [15]
  - Multiple Interface Declaration (MID) messages are used to inform other nodes that the disseminating node has multiple interfaces
  - Host and Network Association (HNA) messages allow GW nodes to be globally identified as such by regular MRs
  - OLSR fish-eye extension [37] can avoid routing loops by controlling unnecessary long Time to Live (TTL) values
  - To avoid Routing Table (RT) inconsistencies, due to high OLSR signaling packet loss rates, Couto et al. [38] propose to include control packet loss rates in the development of new routing metrics.
  - OLSR in an *opportunistic* protocol [34]
- Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N.)[1]
  - Each node evaluates which neighbor may offer the best path to an "originator" [39], based on Originator Messages (OGM). The choice is based on how many OGMs from a particular destination have been received through each neighbor. The best path corresponds to the one with the lowest level of utilization. Since the included metric solely depicts the OGM loss rate, it is not suitable for QoS
  - Assumes that wireless links are always unreliable, thus not suitable for loss-free networks. Although designed for 802.11, this excludes WMNs with overly homogeneous and perfect link conditions
  - Minimal hardware requirements [40], suitable for cheap router equipment
- AODV Spanning Tree (AODV-ST) [41]

---

[1] B.A.T.M.A.N. Advanced Documentation Overview, `http://www.open-mesh.org/projects/batman-adv/wiki/Doc-overview`

  - Specially designed Ad-hoc On-demand Distance Vector (AODV) version for backbones with heterogeneous traffic conditions. The classic, solely reactive AODV [42] is described in appendix A.1
  - Each GW constantly performs an AODV-style Route Request (RREQ) - Route Reply (RREP) exchange with all MRs, to *proactively* update/request routes
  - AODV-ST is effective when mostly GW flows are transported, in a (multi-) star-shaped distribution
  - Such a spanning tree approach, with GWs as signaling trunks with branches to all clients, can be combined with the signaling of layer 2 forwarding paths
  - Regular intra-node communication is based on the reactive standard procedure, which underlines the priority of vertical traffic in a mesh and the motive to reduce protocol signaling

- Hybrid Wireless Mesh Protocol (HWMP)
  - Developed within IEEE 802.11s [43], which is an important attempt to bring mesh routing closer to the consumer market and to facilitate a broader acceptance of this technology
  - Complete routing engine moved to layer 2, based on MAC addresses and transparent to layer 3. Also included for a faster layer 2 handling of packets
  - HWMP is tailored for backbone WMNs: Proactive routing to “root nodes” (gateways in the fixed network part), where a singe root node spans a distance vector tree structure, and reactive path discovery for sole intra-mesh communication in the mobile network part (not all routes between regular MRs are known here)
  - Very active research community

1.2.8 *Routing Metrics*

More sophisticated metrics are often composed of basic ones [44]. From the latter group, throughput or Round-Trip-Time (RTT) are generally considered for *QoS traffic*, while the Packet Error Rate (PER) / Packet Loss Rate (PLR) is relevant for best-effort traffic. To comprehend advanced metrics, methods to obtain basic metrics are discussed beforehand. From subsection 1.2.8.1 on, four *classic* and inspirational metrics are discussed. The Expected Transmission Count (ETX) and the Expected Transmission Time (ETT) are frequently used in research and practice (see table 1.1). When compared, the concept of ETT is closer to QoS than ETX and hop count (which merely reflects the topological distance), because it considers throughput and delay of a link. Additionally, ETT and ETX are usable out-of-the-box in the OMNeT++ [2] simulation environment. Weighted Cumulative ETT (WCETT) and the Metric of Interference and Channel Switching (MIC) represent more advanced and complex metrics, which underline the current importance in WMNs to sense different interference types and channel diversity on a path.

THROUGHPUT ESTIMATION VIA ACTIVE LINK PROBING

The "Packet-Pair Estimation" [12] method works as exemplarily described by Lavén and Hjärtquist [45]. Throughput, respectively current link capacity is then estimated with [45]:

$$B = \frac{S_L}{\min\limits_{1 \leqslant i \leqslant n} d_i} \tag{1.1}$$

Where S_L is the size of the second probe, d_i is the feedback delay and n is the number of delay samples. The packet-pair method used in ETT is based on AdHoc Probe. Its implementation details are further elaborated in [46]. Venkatesh and Wang [47] state that the dispersion of used AdHoc Probe delay samples does not accurately reflect the actual capacity. This can be improved by adapting the length of the probe train.

---

[2] OMNeT++ Network Simulation Library and Framework, `http://omnetpp.org`

MEASURED DELAY

Yuan et al. [48] apply a low pass filter method to determine the average delay:

$$d^{avg}_{(n)} = (1 - \alpha) \cdot d^{avg}_{(n-1)} + \alpha \cdot d^{new}; (0 \leqslant \alpha \leqslant 1) \tag{1.2}$$

$$d^{total} = d^{avg} - \min\{d^{new}\} \tag{1.3}$$

d^{new} is the currently measured delay. α is a tunable weight, which is used to compensate unusually high burst delay values. Yuan et al. set α to 0.2 and use the metric for intra-Multi-Protocol Label Switching (MPLS) domain Label Switched Path (LSP)-based measurements.

PACKET LOSS

MAC PLR/PER is defined by the fraction of unicast frames for which no regular ACK was received at all. A device's Receive(r) (RX) Signal-to-Noise Ratio (SNR) is a main factor for loss. Roughly spoken, a SNR above 15 dB allows acceptable PER levels [19].

1.2.8.1 *ETX and ETT*

In 2009, Lavén and Hjärtquist [45] discussed advantages of ETX and ETT over the hop count. This is an important step, since link awareness as a non-binary quality is firstly introduced. ETX and ETT are both efficient and straight-forward layer 3 metrics.

ETX represents the estimated number of required attempts to send a packet to a receiving node, and successfully receive the same non-corrupted packet in return. ETX directly processes the delivery rates of broadcast HELLO messages [45]:

$$\mathrm{ETX} = \frac{1}{d_f * d_b} \tag{1.4}$$

where

d_f forward delivery ratio, the measured probability

that a packet successfully arrives at the recipient

d_b backward delivery ratio

$d_f * d_b$ is the expected total success rate.

ETX is also not a good choice to estimate the bi-directional delivery rate for larger frames: It only considers the ratio for a relatively small packet size (e. g., with HELLO messages: 16 plus a multiple of 4 bytes, a typical size in practice would be 48 bytes). Data frames are typically much bigger. HELLO messages are only broadcasted at the lowest 802.11 data rate [49], which is not the case for data packets. Hence, their PLR characteristics differ.

ETT conquers ETX's ignorance to different PHY rates by emitting larger probes, via active throughput estimation. ETT is defined as [50]:

$$\mathrm{ETT} = \mathrm{ETX} * \frac{s}{r} \tag{1.5}$$

where

s Data packet size in Bytes

r Currently measured link rate, based on packet-pair

In a WMN with n nodes, where each node has ν neighbors, the overhead caused by ETX is $O(n)$ [50], [51]. ETT overhead is described with $O(n\nu)$, since ETT requires unicast signaling instead of broadcast.

Kim et al. [49] cast doubt on ETX being part of ETT, since the WLAN basic rate issue is inherently still present. In their general solution, they propose to consider a rate set $\mathbb{R}$, which (gradually) covers all available link rates:

$$\mathrm{ETX}^* = \arg\min_{r_i \in \mathbb{R}} \mathrm{ETX}_i \tag{1.6}$$

$$\mathrm{ETX}_i = \frac{1}{d_{f,r_i} * d_{b,r_i}} \tag{1.7}$$

$$\mathrm{ETT}^* = \arg\min_{r_i \in \mathbb{R}} \mathrm{ETX}_i * \frac{s}{r_i} \tag{1.8}$$

d_f and d_b are valid for a specific rate r_i. The best ETX value ETX_i is the minimum obtained for all measured rates. A new ETT^* becomes thus rate-sensitive.

1.2.8.2 *WCETT*

WCETT [12], [52] tackles *intra-flow interference* by putting a weight on the channel reuse along a path:

$$WCETT(p) = (1-\beta) \sum_{\text{link } l \in p} ETT_l + \beta \max_{1 \leqslant j \leqslant k} X_j \text{ , } 0 \leqslant \beta \leqslant 1 \tag{1.9}$$

k is the total number channels available in the network. The second part of the term identifies the largest X_j as the bottleneck channel: X_j is the number of times that a channel j is used on the path p. The first part of the equation is the standard ETT; β is a tunable weight. Draves et al. claim that in their testbed, WCETT outperforms ETX by 89% and hop count by 254%, in terms of TCP throughput. WCETT is a multi-channel metric and can be used to evaluate single links in a bundle as well, X_j just increases. WCETT is not able to consider inter-route interference though; respectively to treat the question how many path-external nodes deploy a certain channel.

1.2.8.3 *MIC*

This is where MIC [15], [52] steps in. It improves WCETT, since inter-route interference is regarded here. Also, MIC processes the complete topology view, therefore it fits well with global link state routing protocols. MIC for a path p comprises three parts:

$$MIC(p) = \frac{1}{N * \min(ETT)} * \sum_{\text{link } l \in p} IRU_l + \sum_{\text{node } i \in p} CSC_i \tag{1.10}$$

$$IRU_l = ETT_l * N_l \tag{1.11}$$

$$CSC_i = \begin{cases} w_1 & \text{if } CH(prev(i)) \neq CH(i) \\ w_2 & \text{if } CH(prev(i)) = CH(i) \end{cases} , 0 \leqslant w_1 \leqslant w_2 \tag{1.12}$$

At first, the smallest ETT among the total number of nodes N in the network is determined, which shall reflect the lowest transmission rate of all WNICs [15]. The Interference-aware Resource Usage (IRU) tackles inter-route interference. It includes all neighbors N_l which cause interference on a link l and thus favors paths with less aggregate channel time on l. The Channel Switching Cost (CSC) covers the intra-flow part, by seeking channel diversity on p. It distinguishes between the current channel $CH(i)$ of node i and the channel used on the previous hop with i on p. MIC captures the path length, link capacity, loss ratio and interference. A downside of MIC is that path cost estimation inflicts a large overhead.

1.2.8.4 *Need for an Average Link Cost in a Bundled Link, Using OLSR as an Example*

When multiple channels with a neighbor are available, OLSR may be able to internally register several radios [35], but for Dijkstra calculation, only one link cost per edge is needed. In the standard OLSR implementation[3], the link state of the first registered radio is used for path calculation. If this particular radio has a bad link state, but others in the same bundle show a better state, this might lead to a biased routing decision. Therefore, the average link cost per bundle / neighbor must be provided to OLSR.

1.3 IEEE 802.11 WIRELESS LAN

In this Section, relevant aspects of IEEE 802.11 Wireless Local Area Network (LAN) [13] are highlighted, as WMNs often run on conventional 802.11 b/g/a/n hardware [9].

To stretch coverage beyond local frontiers, the standard foresees the use of a Wireless Distribution System (WDS). A handover procedure is initiated when another AP with the same Extended Service Set ID (ESSID) offers a higher Received Signal Strength (RSS) to the roaming client. But the tedious re-association/ -authentication process, which is required for each handover, is time-consuming and might cause long latency and even packet losses [53]. This is fatal for real-time traffic. Roaming in a WMN is much simpler (for a node member); table-driven routing allows for IP-based roaming. As a conclusion, infrastructure WLAN (WDS) networks lack flexibility and scalability [8], in comparison to multi-hop ad-hoc networks.

---

[3] olsrd - an adhoc wireless mesh routing daemon, `http://www.olsr.org/`

1.3.1 *Medium Access Control*

The fact that in a collision domain, streams are arbitrarily served [5], makes the MAC behavior an elemental performance / bottleneck factor. Problems sum up on multi-hop routes.

While bitrates of market-ready WNICs rise constantly (currently up to 1.7 Gbit/s with 8x8 MIMO [54]), the MAC layer has not significantly evolved since the standard's general market appearance in 1999; mainly to maintain the high level of downward compatibility. MAC *amendments* like 802.11e [55] pose an exception, but are hardly relevant in standard setups.

WLAN networks are commonly based on the infrastructure-based mode. If desired (although seldomly used in practice) the centralized Point Coordination Function (PCF) is activated at the Access Point (AP), which offers a *contention-free* service. PCF is better for QoS [56], but does not create flexible networks. WMNs on the other hand require the 802.11 ad-hoc (infrastructure-less) mode, which foresees the usage of the decentralized, *contention-afflicted* Distributed Control Function (DCF). Fig. 1.7 shows its peer-to-peer-style Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) algorithm. A high power consumption and fairness issues among transmitting nodes are clear disadvantages of DCF. QoS transmissions suffer from DCF's single channel best-effort character.

SISC MAC is not well suitable for multi-hop communication [58], since a flow experiences *multiple retransmissions* along a route. In each hop, these packets need to compete anew for the medium, but still belong to the same end-to-end flow. In affect, the available flow throughput is not only limited by the raw channel capacity, but also by the forwarding loads imposed by other nodes (inter-route interference). This results in an inefficient store-wait-and-forward process. Bononi et al. [18] describe a fast-forward negotiation mode, which extends Request to Send (RTS)/Clear to Send (CTS) messages by a flow identifier and by a variable reservation period for each forwarded stream, facilitating multi-hop transmissions.

In addition, the standard DCF CSMA/CA backoff timer is based on *random time* values, so end-to-end delay and total chance of delivery of a packet become hardly predictable, especially on longer multi-hop routes. This contributes to intra-flow in-

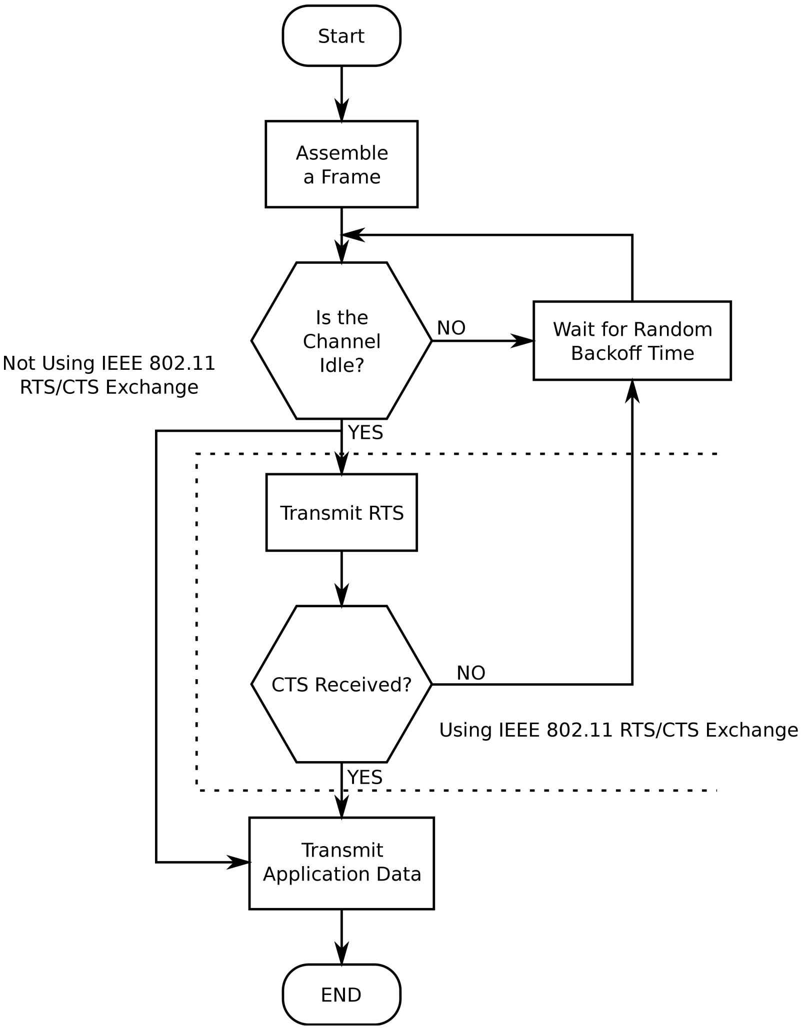


Figure 1.7: 802.11 virtual carrier sensing with RTS-CTS handshake enabled [57]

terference (degrading performance with each new hop). Pathak and Dutta state that RTS/CTS exchange is often disabled in WMNs, "because of their over-conservative nature." Hidden / exposed node problem can cause additional frame collisions and loss then. Cheng et al. [59] review the impact of MAC CSMA/CA randomness on multi-hop routes in a backhaul WMN and the resulting lower and upper bounds of achievable throughput.

Another natural limitation of SISC WMNs is given by the fact that WLAN allows only half-duplex transmission, which almost halves the Transmit(ter) (TX) rate at the MAC layer with every hop [58]. These effects are generally fatal for QoS-related traffic like Video-over-IP [60].

SELECTED MAC AMENDMENTS

QoS with WLAN can be optimized by including cross-layer signaling within the Open Systems Interconnection (OSI) protocol stack. IEEE standard committee recognized these needs by defining MAC support for QoS streaming in the 802.11aa amendment [61].

In 802.11e [55], DCF is replaced with the Hybrid Control Function (HCF), which grants a higher transmit probability to QoS-sensitive packets. 802.11e performance strongly depends on packet inspection methods in the MAC layer in order to detect protection-worthy traffic, or on the corresponding QoS signaling in IP headers (e. g., DiffServ). This denoted dependency again underlines the advances of a cross-layer QoS system. But even the QoS-friendly HCF and its Enhanced Distributed Coordination Access (EDCA) prioritization scheme cannot prevent high delays and packet drop rates of priority traffic, if there is heavy background traffic.

1.3.2 *Physical Layer*

The latest official 802.11-2012 standard [13] contains currently supported bitrates. WLAN PHY supports auto rate adaption protocols like Automatic Rate Fallback (ARF) and Adaptive Automatic Rate Fallback (AARF) [62]. 802.11n achieves up to 65 Mbit/s on the smallest channel variant (20 MHz), through improved OFDM modulation (54 Mbit/s with 802.11g). But the trend (mostly in consumer electronics) leads to spectrum exploitation. 802.11ac [54] is the latest establishment here. Spatial multiplexing with (Multi-User) MIMO, OFDM with 256 Quadrature Amplitude Modulation (QAM), antenna diversity, frame aggregation, beamforming, Reduced Inter-Frame Spacing (RIFS) and the Greenfield mode (denied legacy support for 802.11a/b/g) are all effective methods to boost bandwidth, but *channel bonding* is the most prospective feature in current PHY broadband research: 802.11n supports 20/40 MHz channels, 802.11ac additionally 80 MHz, with an optional extension to 160 MHz. The impressive capacity gain at the cost of a broad spectrum occupation is good trade-off in a single-hop WLAN. But in multi-hop communication, the limitations discussed in Section 1.2.4 would occur on all 20 MHz channels covered by a bonded channel.

The spectrum could be better used to lower interference between smaller channels. With a sensible CA protocol and bundle management, the entire spectrum can be exploited and features like MIMO, and others can still be overtaken.

MIMC ISSUES

Due to mutual effects in parallel channel usage, the PHY layer cannot offer that "using n interfaces equals n-fold performance" [63] and so only offers a limited modularity for WNIC equipment. Lasowski et al. [64] note that a 100% orthogonal channel behavior, especially on adjacent channels, is likely to occur in real setups. By assumption, two adjacent orthogonal channels don't create Adjacent Channel Interference (ACI) and can coexist interference-free. This is not fully valid in practice [65] and even affects Inter-Carrier Interference (ICI) [66]. Although 802.11a offers more non-overlapping channels than 802.11g, ACI is also an issue [67]. This is due to several factors. One is the disadvantageous alignment of WNICs on a board of the same node, where distances and angles between antennas might jeopardize performance [68]. Fuxjager et al. also confirm that on-board multi-antenna usage, with a few centimeters distance can cause frame corruptions and channel blocking, especially with multi-hop. A problem here is the "near-far effect", where a strong signal superimposes a weaker one, making it impossible to detect. This can be treated by adapted Radio Frequency (RF) filter design.

Cheng et al. [69] created an homogeneous wireless testbed for 802.11a two-hop connections. They use the newer OFDM-based 802.11a, so their conclusions can be transferred to common 2.4 GHz 802.11g/n systems (with 20 MHz channels): ACI can seriously threaten multi-hop performance. As a solution, Cheng et al. propose to increase "the distance between the two dipole antennas on a node, collinear placement of the two antennas, and larger transmit and receive channel separation". They also state that particular vendors do not always comply with the minimum transmit masks in 802.11. To effectively prevent ACI, the group recommends to always guarantee a high Signal-to-Interference Ratio (SIR), which in their testbed is a direct indicator for layer 4 performance. As a rule of thumb, a SIR of 20dB leads to an acceptable link, whereas 30dB is nearly perfect.

As a conclusion to this paragraph, CA protocols need to avoid ACI risks during the channel negotiation process. This can be achieved by considering the Received Signal Strength Indication (RSSI) of a packet, which can be converted in signal power.

1.4 QUALITY-OF-SERVICE

This Section first targets general and then mesh-specific QoS solutions (subsection 1.4.5). Further QoS threat potentials and constraints with QoS under 802.11 have been analyzed by the author in [58].

1.4.1 *Differentiated Services*

DiffServ [70] is a known platform for QoS facilitation in modern transport networks. With DiffServ, low latency can be targeted to QoS traffic, while maintaining best effort treatment for other packets in each hop. This makes it an interesting solution: So far, vertical and horizontal traffic were discussed in this Chapter. Now, *additional traffic categories in between* can be added. DiffServ defines Per Hop Behaviors (PHB) applied in *forwarding* nodes, which makes it adaptable to intra-mesh multi-hop routing. A PHB typically defines a drop-or-forward policy, in relation to other classes.

A DiffServ Code Point (DSCP) defines an *action* or treatment, which the carrying packet shall receive in the DiffServ domain. Thus, a PHB is assigned to a DSCP. The former Type of Service (TOS) field in the IPv4 header now accommodates 6 bits for the DS(CP) field and 2 bits for the Explicit Congestion Notification (ECN). IPv6 uses a Traffic Class field instead.

Table 1.2 summarizes the PHB types defined in [70].

DiffServ itself cannot avoid best-effort-style transmission. Even within a DiffServ domain, there is no guarantee of proper treatment on the route. Nevertheless, it's a common approach to mark packets. PHBs can then be freely interpreted and linked to QoS measures like queuing or shaping.

Table 1.2: DiffServ PHB types

TYPE	EXPLANATION
EF	Highest priority class, packet are not forced to wait in a queue and receive a low-drop priority. SPQ is often granted to EF PHB. EF allows low loss, latency and jitter. Note that this traffic type is also often restricted, following bandwidth and admission control.
AF	AF includes 4 sub-forwarding-classes, whereby class 4 has the highest priority. Each class further has 3 drop priority levels (high, medium, low). AF allows to offer fine-grained service levels to customers and users. Normally, a fair or WFQ scheme is applied for AF classes
Default PHB	For best effort traffic. This is the lowest class and must be always defined, as it applies to all traffic which does not fall in the former categories; including unmarked traffic
Class Selector PHB	This is a legacy class and of no importance for the system. It is kept in order to guarantee compatibility with the former IP Precedence field in the TOS byte.

1.4.2 *Sub-Layer 3 Forwarding*

MPLS [71] as a Virtual Private Network (VPN) technology is known from large transport networks. A packet is able to traverse a group of Label Switched Routers (LSR) in a MPLS domain solely on the layer 2.5. The classic IP lookup of the next hop address, which involves an analysis of the 20 byte IPv4 header and the time-consuming longest prefix match method, is avoided. The encapsulated forwarding on a (usually pre-determined) MPLS tunnel (LSP) is based on *short* and *fixed-length* labels. This requires, that the ingress router adds a header between the IP and MAC header, which is depicted in Fig. 1.8. The label part contains a Forwarding Equivalence Class (FEC), which indicates the membership to a more generalized group or type of packets. Such a group may typically be an external MPLS domain or IP subnet, or a certain

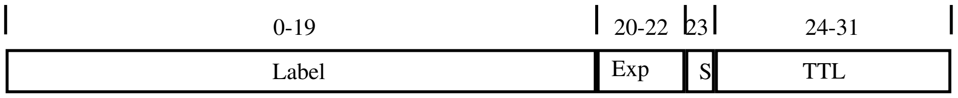


Figure 1.8: MPLS header

traffic class. Traffic engineering or QoS with MPLS allows to force or reserve (exclusive) custom paths in a connection-oriented environment. Regular IP routing is still stochastic, so in mixed networks (MPLS and IP), MPLS eventually cannot guarantee reservations or QoS levels. Also, MPLS must be supported hardware-wise by all routers. Regular routers cannot interpret the header, which makes it useless.

To signal paths and labels in a domain, there are Label Distribution Protocol (LDP) [72] and RSVP-Traffic Engineering (RSVP-TE) [73]. LDP is unlikely used to signal traffic engineered LSPs, but rather for LSPs for best effort traffic. Here, static paths are often not necessary and labels can be negotiated automatically. Thus, aspects of LDP fit well in a mesh-based approach. Labels are then stored in a LSR's Forwarding Information Base (FIB).

SELECTIVE ADVANTAGES OF PACKET COMMUTATION IN WMNS MPLS' initial speed advantage in the packet forwarding process is hardly relevant today, as modern routers include IP forwarding hardware-wise. Still, label-based packet commutation is potentially beneficial for multi-hop communication. Applied to WMN, it becomes a *non-generic* feature and the faster routing based on a commutation table is put to use again [74]. A mesh protocol, though hardware-independent, is often designed to run also on low-cost hardware, or common PC / Linux clients. Less resources from a forwarder's Central Processing Unit (CPU) are required to attend the lookup process [26], which decreases workload. This alleviates the routing engine, which is additionally occupied with the processing and dissemination of control messages.

Pathak and Dutta list further MPLS-inspired use cases in WMNs, like Data-driven Cut-through Medium Access (DCMA) [75]. To increase the multi-hop speed of the 802.11 MAC, a WNIC can combine labels in RTS frames and the lookup in a MAC label database to detect the proper next hop and thus deploy a pipe-lined forwarding. "DCMA reduces the number of channel access attempts and end-to-end latency" [75].

1.4.3 *Packet Queues*

WFQ [76], [77] allows a fair treatment of all flows while still considering their priorities. Although priority queuing with WFQ can manage flows internally in a single forwarding node, it has no link whatsoever to the optimization of multi-hop routing. This competence is still reserved for the routing protocol, which steers traffic via next-hops. Like the fast packet commutation and packet scheduling, queuing is a measure which is applied until the next hop.

Still, queuing has an impact on multi-hop routes. In [78] a mesh chain is tested with layer 3 queues at the different (dual-radio) ingress nodes. As expected, nodes with lesser hops to a GW achieve higher throughput and less delay, when no queues are deployed. Different Packet First In - First Out (PFIFO) queues were tested, in favor of streams with larger hop distances. Results generally show that delay of these streams is reduced, while bandwidth levels are unchanged and may even drop minimally.

1.4.4 *General Packet Scheduling Mechanisms*

Load balancing can be applied if multiple links to the same destination (or next hop) are available. A simple Round Robin (RR) scheme is not suitable for high speed routes with varying link qualities (especially delay) [48]. Adaptive approaches are preferred. The capacity-adaptive “cheapest pipe first” algorithm [79] selects the link with the lowest cost as the cheapest pipe. It is used for sending all packets, until its capacities are nearly saturated. In this case, load is additionally distributed to residual radios. To estimate saturation, packet pair dispersion technique [80] can be used.

Microsoft’s Mobile Access Router (MAR) [81] conceptual solution shows how different WATs in the same node are unified in a single Virtual Interface (VI), which internally distributes load, based on each interfaces availability. The exploitation of the overlap of coverage from different cellular and wireless networks allows for a more reliable, synthesized network access, which users don’t need to configure manually.

Additional improvements such as compression, caching and session-based scheduling are discussed as well.

The Linux Ethernet Bonding Driver[4] also aggregates capacities of multiple Ethernet cards into a VI. Its distinctive feature are configurable modes, listed in table 1.3. To mix this modular concept with WLAN was a partial motivation of this work.

Table 1.3: Ethernet Bonding Driver PS modes

MODE	NAME	SUMMARY
0	balance-rr	Round-Robin
1	active-backup	Only one interface is active, others are kept as fallback options
2	balance-xor	Interfaces are distributed across multiple receivers with MAC-based XOR modulo division
3	broadcast	Data is simply duplicated and broadcasted on all bonded interfaces
4	802.3ad	Bonding of interfaces with the same 802.3 [82] TX settings (speed, duplex mode)
5	balance-tlb	Adaptive transmit load balancing: one interface for each receiver, with additional policies; more complex version of balance-xor. Considers the current load of each slave, which is computed to the relative speed
6	balance-alb	Fully adaptive load balancing: balance-tlb plus receive load balancing

1.4.5 *Specific Solutions for Wireless Mesh Networks*

The layer 3 system in [9] supports QoS in WMNs. Two instances work in conjunction with OLSR. A node's "Traffic Observer" monitors services and traffic in the network. If there are streams in the same wireless collision domain which might interfere bandwidth-wise with those generated by real-time applications, the"Traffic Controller" is activated. It can instruct disturbing nearby neighbors via signaling to throttle their max. bandwidth for single best effort flows (e. g., to a maximum of 5 Mbit/s). Detailed policies and bandwidth thresholds are defined. Priorities are

---

[4] Linux Ethernet Bonding Driver, http://www.kernel.org/doc/Documentation/networking/bonding.txt

identified based on the outdated TOS field. Evaluation is done by Telekom in a real WLAN setup, named “MeshBed” [83]. The system offers an intelligent partitioning of local capacity, but *cannot actually increase it*. The only deployed method is shaping. More measures are required, in order to overcome limited capacity and QoS.

Prabhavat et al. [84] provide a comprehensive review on existing load distribution models. They claim that skewness between asymmetric routes is a major issue in multi-path load balancing. With hop-to-hop LB, skewness is of minor importance.

1.5 MULTI-INTERFACE MULTI-CHANNEL WIRELESS MESH NETWORKS

The first stage to exploit channel diversity is often an un-managed, non-LB related solution. An often used approach in a WMN backbone is to have two separate radios for edge nodes, at best using separate bands (2.4 GHz and 5 GHz). One serves local clients and the other radio is for sole backbone communication. Such examples are found in [19], [83] and [85]. A next stage denotes the use of 2 or more radios in the backbone, to minimize intra-flow interference. Within Fraunhofer’s Wireless Back-Haul (WiBACK) architecture [10] for rural and urban setups, simply 2 802.11 radios are deployed, with a gap of at least 60Mhz between 2 20Mhz channels. This avoids a throughput decrease at each hop. In [86], full-duplex communication is achieved with a dual-radio scheme. Still, additional components and increased complexity allow for a more efficient use of radios and the spectrum, as outlined in the following.

1.5.1 *Management Approaches for MIMC Wireless Mesh Networks*

A carefully designed resource allocation strategy, which matches the node-specific availability of radios and at the same time the desired network behavior, is a crucial success factor [10]. Mainly, this means introducing a distributed or centralized CA and a LB mechanism.

The CA study of Wu et al. [20] also deals with the question of how many WNICs are actually needed: It is often the case that WNICs are distributed evenly, which does not match the requirements of heterogeneous mesh traffic. It causes bottlenecks at

GWs, which are in need of more resources, while other MRs do not fully utilize their radios. This must be considered in the planning stage. Wu et al. intend to minimize the number of WNICs and recommend an absolute amount for different WMN sizes (chain and grid), based on an heuristic and an optimal approach.

A key concept depicts the abstraction of resources; for the sake of simplicity, compatibility and modularity. Adding a cross-layer design has high benefits [17]. The CARMEN architecture [87] introduces an abstraction layer, which hides particularities of WAT. An open virtual layer is also deployed in [88]. It accommodates different 802.11x interfaces and makes them independent of layer 3. For each specific interface type, a new Logical Link Control (LLC) substitute module is introduced. The underlying algorithm uses one module or another, in dependence of flow requirements. A bundling within the virtual layer is not applied. Like many other MIMC approaches, the group targets to optimize throughput and end-to-end delay as QoS parameters. A virtual layer/ interface is essential for MIMC WMNs which shall be compatible with different mesh protocols and metrics. It can also gather and reorganize all types of useful cross-layer input. A well-designed VI is further able to provide a usable platform to combine different measures to improve capacity and support heterogeneous traffic.

1.5.2 *Channel Assignment*

Channel assignment is an important feature in a MIMC system. Without it, manual channel map configuration efforts and adjustments during runtime would be unmanageable. Despite the great potential to improve capacity, CA protocols need to be carefully designed, otherwise there is a threat of unnecessary interference [89]. To achieve minimal interference depicts a list coloring problem [90]. Before network parameters are optimized, basic connectivity needs to be guaranteed. The CA approach in [21] focuses on this aspect. The centralized approach of Robitzsch et al. [91] facilitates an autonomously controlled entrance of a node in a WMN, considering ACI and ICI. Most approaches do not distinguish between orthogonal or overlapping channels. Not so in [92]; here CA is optimized for partially overlapping channels.

Receiver-based Channel Assignment (RCA) schemes [93], [94] are straight-forward, proactive, topology-considerate and easy to implement. Negotiation-based Channel Assignment (NCA) schemes perform CA on-demand and allow interference-free transmissions in most cases [93]. But their reactive nature makes them more suitable for MAC-layer approaches, where the channel is negotiated per-frame. For layer 3 packets, NCA would be too slow. Still, RCA and NCA are considered too decentralized approaches. There is no consideration of 2-hop neighbors, and WNIC resources cannot be assigned in parallel, based on the next-hop type.

To conquer these flaws, a Master thesis was tutored by the author, prior to the writing of this dissertation. The result is the Dynamic Channel Distribution based on Priorities (DCDbP) protocol [95]. DCDbP has the following characteristics:

- Distributed, node-based [93] approach
- Pro- and a reactive assignment phase: First is for connectivity, latter is for adaptive redistribution and bundling of radios
- A channel's priority is based on its relative usage by a GW, and/or 1-hop, and/or 2-hop neighbor
- An unused channel can be reactively reassigned to other nodes
- Active probing is performed, to obtain a WNIC's current throughput and thus to determine the occupancy per channel
- Extensive proactive signaling of gathered neighborhood information via "ChannelState" broadcast messages
- Dedicated, exclusive Control Channel (CC). Use cases and advantages of a CC are discussed in appendix A.2.2

These are the protocol's main advantages:

- Awareness of channel interference, connectivity and traffic
- Topology-considerate; GWs in the 1-hop neighborhood will receive the channels with the lowest utilization, since this next-hop probably belongs to one of the most used paths [96]

- Ability to determine the amount of radios per neighbor based on its priority (GW or not?), which ultimately allows bundling and load balancing

Table 1.4 contains the final output of DCDbP.

Table 1.4: Distribution Table

NEIGHBOR IP	NEIGHBOR MAC	CHANNEL
IP 1	MAC 1 of neighbor 1	Channel 1 . . . z
IP 1	MAC 2 of neighbor 1	Channel 1 . . . z
IP 1	. . .	Channel 1 . . . z
IP 1	MAC n of neighbor 1	Channel 1 . . . z
. . .	. . .	. . .
IP m	MAC 1 . . . n of neighbor m	Channel 1 . . . z

Kim et al. [97] have analyzed different aspects of CA algorithms in OMNeT++.

1.5.3 *Load Balancing in MIMC Wireless Mesh Networks*

Scheduling is the next logical step after CA. If sufficient resources are available between two adjacent nodes, bundling is able to improve the resource utilization beyond CA measures [98]. Furthermore, channel bundling can be used to reduce signaling overhead [98]. Also, the allocation of channels to a single bundle reduces computational cost, since "all the channels'in the same bundle are either available or busy simultaneously, a secondary user can sense each bundle of channels instead of each channel individually." [98]. The development of the author's own approach is discussed in [99], [100] and [101].

Kim and Ko [102] describe a VI which sits upon and controls multiple WLAN MACs. Within the virtual interface, the IF with the best link quality is chosen for transmission, on a per-packet-basis. Their approach segregates low-performing interfaces in a bonded set of IF, which wastes capacity in certain constellations. Instead of link state information derived from the applied ad-hoc routing metric, Kim and Ko opt to calculate link quality based on a combination of packet receive rate and packet error rate, which are reactively measured in short intervals. Basic channel assignment is not included, which causes additional configuration efforts for the user. A neighbor table is maintained, which holds information on the interface availability and link states in the neighborhood. To signal a node's associate IF addresses, a modification of the Address Resolution Protocol (ARP) is used.

Hu confirms that establishing channel diversity (by having single channel links) is not enough; this diversity must be *actively utilized*, in order to improve capacity. In their work [103], a system model is described, which uses multi-radio for parallel transmission between nodes. Again, a VI with a virtual MAC address is used. In their simulator testbed, two kinds of TX-oriented, scheduling algorithms are tested. Although entirely different in their behavior, both consider *hop-to-hop scheduling*. Hu defends this decision with the varying nature of the wireless medium, making multi-hop / flow coordinated scheduling too complex. First algorithm creates redundant packet copies and schedules one per selected IF. Unfortunately, RX behavior is not specified in this case. This mode aims to improve loss resilience, but has only a moderate impact on throughput, as expected. Second algorithm applies a "partition-based" scheduling, to improve throughput. A radio is randomly chosen and its TX probability is directly based on the ETT value. This design is straight forward; also, the metric is inter-changeable. Both modes can increase throughput up to 10% for TCP; second algorithm enables 90% for User Datagram Protocol (UDP). The work is one of the first TX approaches for parallel transmissions. It does neither take traffic, nor node roles into account. Hu recognize this and state that future solutions need to include full awareness of multi-hop conditions, to optimize scheduling. A first step towards this ambitious goal can be to consider layer 3 topology (i. e., next-hop type and vertical/horizontal traffic differentiation) and flow information (i. e., QoS and its destination). Still, their work already shows promising results and is identified an important step to improve WMN capacity.

1.6 CONCLUSION

The review of related work has clearly shown that WMNs suffer from multi-hop and interference limitations and that vertical traffic cannot be properly protected, without suitable measures. Researchers agree that MIMC WMNs generally have the potential to overcome these issues, and additionally to improve delay and throughput levels of multi-hop transmissions. Link state routing, channel diversity, radio resource management, load balancing, QoS techniques and CA are methods usually used in an isolated mode, which now need to be analyzed and intelligently arranged, in order to address the previously mentioned issues. The result of this process shall be a *holistic combination* of these methods, from a *systemic perspective*.

Chapter 2:

Proposal for a Resource-Management System for Transmission Enhancement and Channel Diversity Exploitation in Wireless Mesh Networks

2

CHAPTER 2: A RESOURCE-MANAGEMENT SYSTEM FOR TRANSMISSION ENHANCEMENT AND CHANNEL DIVERSITY EXPLOITATION IN WIRELESS MESH NETWORKS

2.1 INTRODUCTION

Chapter 1 has revealed that limited capacity and traffic unfairness have a negative influence on transmissions. A node cannot determine the final route of a packet, therefore the necessity to scavenge whatever capacity is available for every single next hop link was identified. This enhancement can be achieved with the deployment of MIMC nodes. A *resource management system* is necessary which exploits all radios and at the same time uses input natively provided by the link state routing protocol. Based on the previously gathered conclusions, this newly invented system is now described. It contains the system's architecture, functioning and details on all related components. All original content and concepts have been developed within the research tasks of this thesis.

2.2 CONTAINMENT OF TARGET MESH NETWORKS

Chapter 1, 1.2.5 discusses mesh networks with heterogeneous traffic distributions, which shall be investigated here. Arrow (1) in Figure 2.1 represents a GW flow. Note that in the Figure, the arrow is positioned independently from the route(s) between source and destination, because the actual path of each packet is transparent to upper layers. This work's focus lies on the optimization of this type of flows. The system does not target to optimize cross-traffic between non-gateway nodes (see arrow (2)) as it is of minor importance. In principle, the system will work in homogeneous WMNs. Without GWs, all intra-node end-to-end routes are distinguished only by DiffServ (DS)-based priorities, if available. Without DS, packet Destination (DST)s

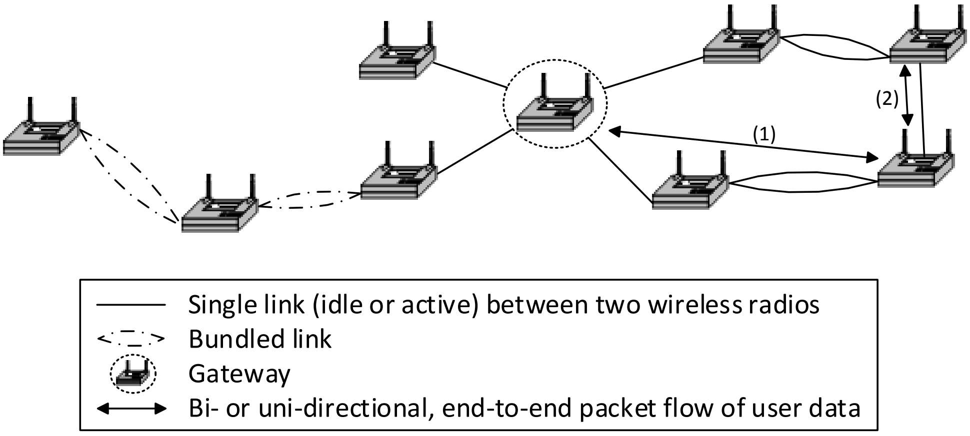


Figure 2.1: Bottlenecks caused by insufficiently equipped gateways

do not influence resource management at all. A particular investigation of such scenarios is not within the scope of this work.

A system administrator needs to equip each GW node with an exceptional amount of radios. Bundles between single nodes may contain multiple radios, but the weakest link in a chain determines its overall capacity.

2.3 OVERVIEW

This Section provides an overview of the *system components*, their *tasks and interplay* and the *compound architecture* which accommodates them. Later on, the related *process* and functionality of each component is explained. The term *module* refers to the system's location in the OSI reference model, where all novel components are nested in a separate module between layer 2 and 3. Fig. 2.2 depicts the middle-layer (2.5) solution with multi-layer input.

A NOTE ON BROADCAST AND UNICAST

All following methods refer to unicast packets; broadcast packets are not separately discussed. Their treatment is simple: When a broadcast packet is sent out, its copies will be sent over all attached radios.

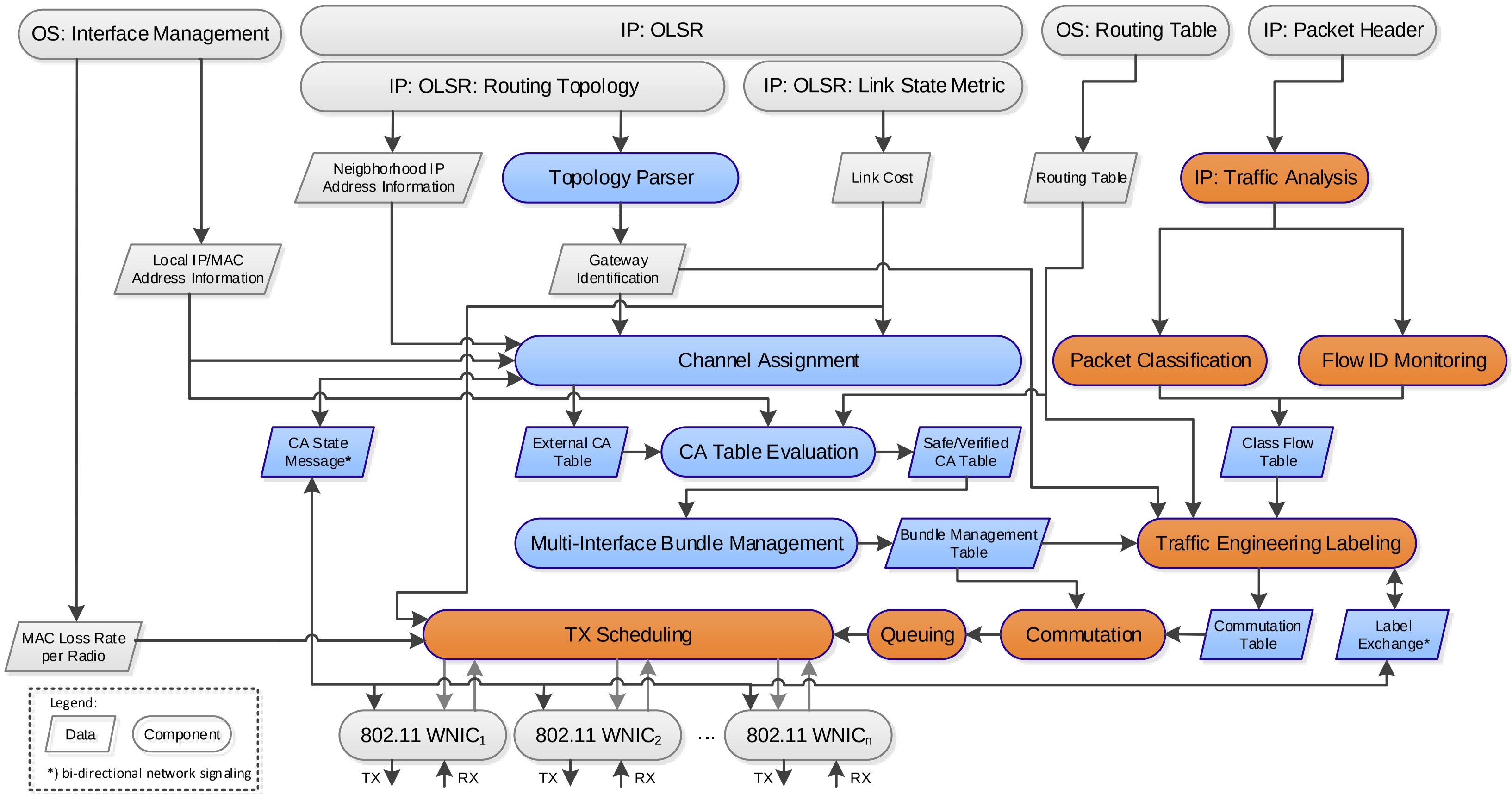


Figure 2.2: System architecture and overview

2.3.1 *System Architecture and Functioning Overview*

This paragraph explains the system functionality in brief. Figure 2.2 visualizes the basic principles and can be used for implementation purposes. An asterisk* indicates a reference to a block in Fig. 2.2. Grey blocks are unmodified existing components in mesh nodes. Blue blocks are novel components related to *resource distribution*. Orange blocks are novel components related to *packet processing*. Note that not all in- and outputs are listed in detail here.

The host system must provide the Routing Table* (filled by OLSR), Local IP/MAC Address Information*, as well as access to all radios. Three operations are performed on a WNIC*; sending, receiving and switching its channel. OLSR* provides Link Cost* values and Routing Topology* information. From the latter source, especially the Identification* of nodes with access to the Internet (gateways) and connectivity information on the 1-hop Neighborhood* is required.

Mesh SISC communication causes a variety of combined *problems* (see Fig. 1.6). This *reduces the transmission capacity*. Two major issues were defined:

2.3.1.1 *Capacity*

The *first substantial problem* is the decrease of multi-hop capacity (see Chapter 1, 1.2.4). Exploiting channel diversity can solve this, but requires the following measures: Channel Assignment* negotiates channels with neighbors via custom CA State Messages*. Criteria for this distribution is described in Section 2.6.1. CA* protocol shall be exchangeable, therefore an intermediate function is needed which proves that proposed channels actually grant neighborhood connectivity. The result of this CA Table Evaluation* is the Verified CA Table*, which also has a safe version (single channel).

The Multi-Interface Bundle Management* maintains bundle information in a central Bundle Management Table (BMT)*, which also stores load balancing statistics. The core of this table is derived from the Verified CA Table*. The system is theoretically able to handle any number of attached radios; in reality the amount is mostly limited by hardware capacities, power supply and on-device wireless interference between omni-directional antennas [104]. The Bundle Management Table* enables a flexible way to manage traffic to next hops. Before the actual load balancing within a bundle, a neighbor (and all radios linked to it) is reduced to a simple bundle index in the first place.

Load balancing is not an issue related to mesh routing, to queuing or to layer 2 forwarding and can therefore be detached from these functionalities, because packets are scheduled on a single-hop basis. Three basic, yet effective and distinctive methods have been selected to exploit the given *capacity* to a next hop, to reach a *better performance*. The simplest approach schedules packets evenly in a round-robin fashion. The second approach combines round-robin or single-radio transmission with optional fallback interfaces. Active radios are replaced with fallback radios, if the MAC Loss Rate* on this link crosses a critical limit. The third approach performs link-quality-adaptive load balancing. These three methods are referred to as Packet Scheduling (PS) modes, represented by the TX Scheduling* component in Fig. 2.2.

2.3.1.2 *Traffic Separation*

The *second substantial problem* is dealing with the necessary protection of GW- and QoS- traffic from cross- and classless traffic. Chapter 1, 1.2.5 discusses the typical traffic behavior in a mesh backbone, where mostly Internet traffic is consumed by MRs. The system grants higher priorities to GW flows and flows with QoS-demands, over horizontal traffic. The following components are required to reach this goal.

A packet is received from upper layers (IP) and leaves the system through one of the bundled radios. Every newly created packet or every packet entering the WMN is monitored in the Traffic Analysis (TA)* block in Fig. 2.2. Destination IPs are extracted from passing packets (Flow ID Monitoring*) and stored as flows in the Class Flow Table*. Entries are combined with DiffServ classification detected in packets.

When a packet passes the Traffic Engineering Labeling (TEL)* component for the first time, two labels are injected. If the carrying flow is passing on a GW route, the packet receives the first label for fast layer 2 forwarding by the TEL engine, plus a second label to put the packet in the highest priority queue in each path member (label content remains the same here). The Multi-Hop Radio Resource Management (MHRRM)* component handles the push, swap, forward, and pop operations on labels. The forwarding aspect is depicted in Fig. 2.3. To determine the affiliation to a GW flow, the

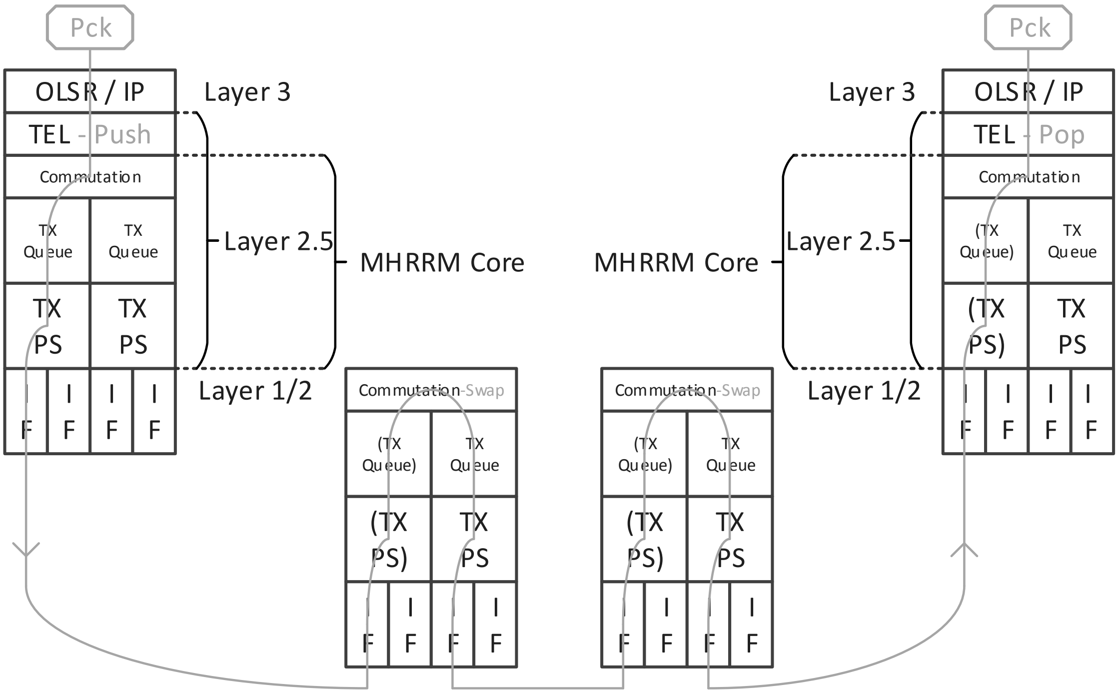


Figure 2.3: Label-based forwarding for packets of a GW flow

packet's IP DST must coincide with a mesh-external IP address. Gateway Identification* is enabled by the Topology Parser*. If a packet does not flow through, or enter via a GW, but still bears a DS classification, its DSCP value is mapped to an equivalent custom label, which allows the packet to be enqueued with a higher priority than others.

The advantage of labeling takes effect in intermediate nodes. An incoming packet can be fast forwarded, without the time- and hardware-consuming IP lookup. The next hop is still determined by OLSR, but routing information is now provided below layer 3, in the Commutation Table*. To enable a fast commutation and label swapping for layer 2 forwarded packets, constant label exchange via Label Exchange Messages* with 1-hop neighbors is required. TEL* also requires access to the Bundle Management Table*, to map non-permanent labels to static bundle indices in a node.

After the next-hop has been determined via standard IP routing or layer 2 forwarding, each packet is subject to Queuing*, also based on a label. Again, GW and DiffServ-coded packets experience a faster removal from their queue, in case multiple streams to the same next hope are polled. A fixed amount of queues is maintained per neighbor.

2.3.2 *Process Point of View*

The best way to show the details of the proposed system, its functioning and novelty is from a process point of view. The top level process is shown in Figure 2.5. Subprocesses are handled in lower levels. For a better understanding, used block types are listed in Figure 2.4 and related acronyms in appendix 6.

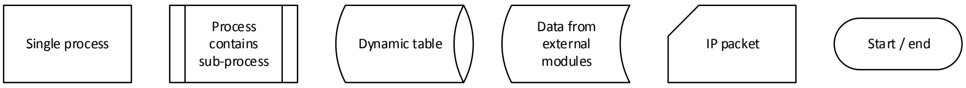


Figure 2.4: Block types used in flow diagrams

From Section 2.4 on, terms written in *italic* in subsequent Sections refer to (sub-) processes from Fig. 2.5.

External input in Fig. 2.5 is obtained from processes available in an unmodified protocol stack. For instance, the routing engine in the OS kernel provides the RT. Processes (middle column) follow a (chrono)logical order in the concept. *CA* and the *Evaluation of the CA Table* are performed apart. For the rest of the processes, following the way of a single *layer 3 packet* through the chronological process order is a recommended starting point to interpret the flow of Fig. 2.5. *TEL*, *Label-based Multi-Hop Packet Commutation (LMHPC)* and *TX PS* are executed packet-wise, with a processed, or manipulated packet as an output. Additionally, *Traffic Analysis* monitors packets from layer 3. Incoming *Layer 2 packets* are either forwarded or finally handed to upper layers. *TA* is the only process which does not manipulate header data or schedule a packet for sending, therefore it has no packet to layer 2 as an output.

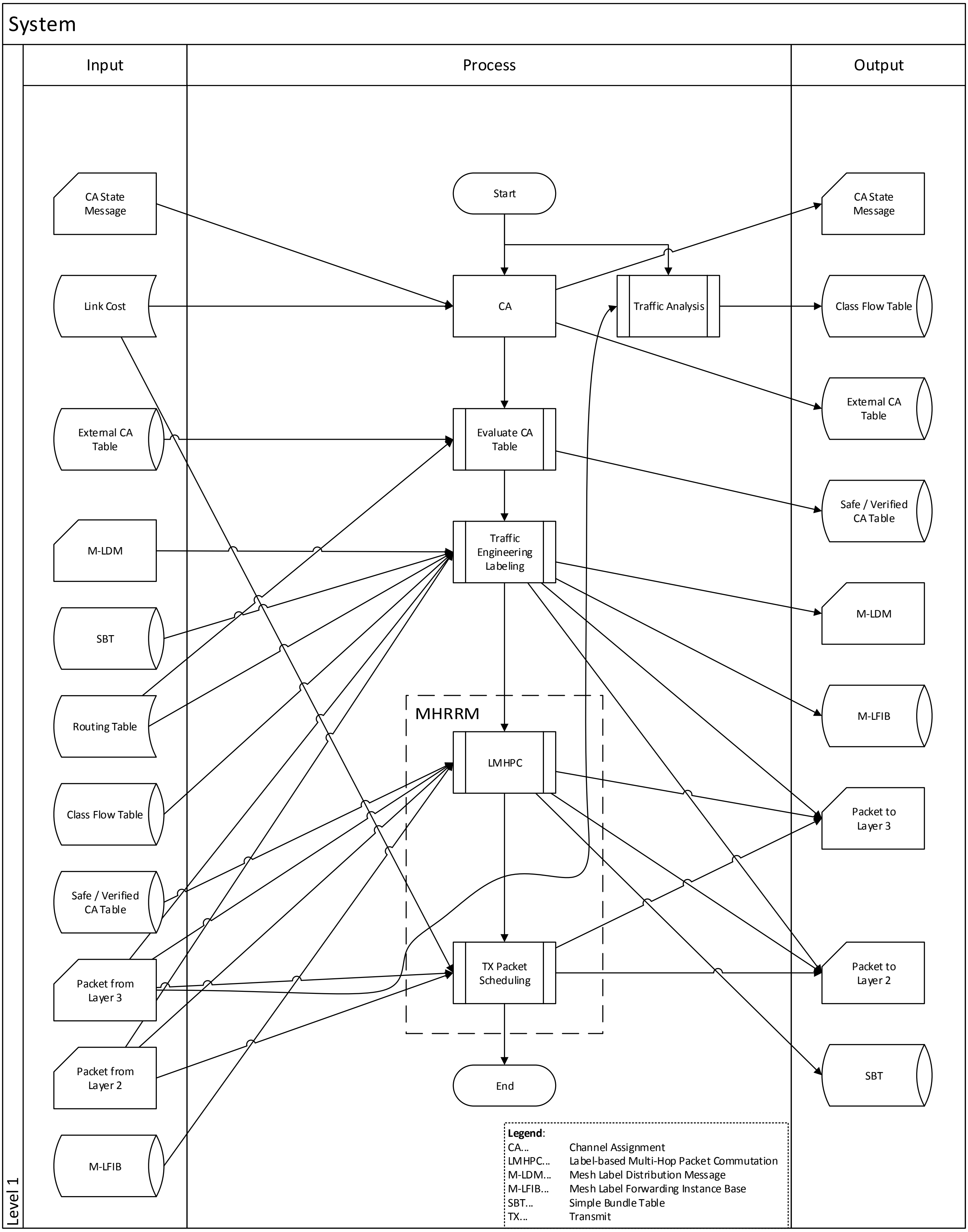


Figure 2.5: Level 1 process flow

Listing 2.1: Required information of 1-hop radios

```
class NodeEntry {
private:
        nsaddr_t currentNode; // local (main) IP / node ID
        nsaddr_t nbNode; // neighbor’s (main) IP / node ID
        nsaddr_t localAdd; // IP of local radio
        nsaddr_t nbAdd; // IP of neighbor radio
        double etx; // ETX value of neighbor radio
        double delay; // ETT value of neighbor radio
// ...
};
```

The different tables are all dynamic. Still, assigned channels and the amount of attached radios and 1-hop neighbors are not supposed to change *during process flow*. An input is only shown the first time it is used in the Figure; later on it is assumed that this input or resource will be available for all subsequent processing steps and decisions.

Figure 2.5 is a main reference for the next sections, where each process, sub-process / -algorithm and input / output data unit will be explained separately, in detail and with a visualization similar to Fig. 2.5.

2.4 MESH ROUTING PROTOCOL

Although it is not a newly designed process, the routing protocol marks the starting point for the detailed description of processes, as every following one is based on the layer 3 connectivity map. The system design requires a *proactive link state* protocol. Amongst the reviewed candidates, OLSR is recommended for the system. The decision is elaborated in appendix A.2.1. The variables in the C++ code segment in listing 2.1 are maintained in OLSR and reveal which IP layer data of a WLAN IF is important.

2.4.1 *Node Identity with Multiple Radios*

OLSR proactively maintains all possible IP destinations in the RT. This implies that a node with n attached radios *may* appear with n entries in each table. OLSR registers all radios in a node and by default picks a *main* IP among them: The IP address of the smallest index j of the totality of all local radios $\sum r_j$ identifies the node routing-wise, and represents the *node ID* in the system. Each radio has to receive a unique IP at network startup. One (less-preferred) possibility is to schedule a Dynamic Host Configuration Protocol (DHCP) discover after the protocol stack has loaded. The author recommends to use Dynamic Wireless Mesh Network Configuration Protocol (DWCP) [105]. DWCP introduces different node roles. Selected MRs contain address pools with local validity. Another distributed solution (for MANETs) is Extended Distributed DHCP (E-D2HCP) [106].

2.4.2 *Topology Analyzer*

Information about which nodes have access to external networks is a crucial input for packet labeling and is generated in the Topology Parser in Fig. 2.2.

2.4.3 *Link Cost*

A selection of routing metrics has already been discussed in Chapter 1, 1.2.8. Any metric is applicable, as long as it reflects diversity in the 1-hop environment. The chosen metric must reflect environmental influences which might alter the reliability or capacity of a single radio link. The ETT metric (see Chapter 1, 1.2.8.1) has been chosen for the system. ETT's proactive link state probing is performed for all radios. The requirement to provide an average link cost value is explained in Chapter 1, 1.2.8.4.

2.5 TRAFFIC ANALYZER

The *TA* process in Fig. 2.5 monitors packets coming from layer 3, or those generated by applications and protocols in higher layers, and extracts QoS-related information, if available. Forwarded packets are only analyzed when they receive a regular IP lookup, because it is assumed that they have been monitored already in the node of origin. *TA* serves to provide identified characteristics of passing packets to the later *TEL process*.

The system is not bound to a specific packet classification scheme and can be generally adapted to any scheme which allows to assign a class or priority to a single packet. In this work, basic DiffServ is used to consider QoS-demands of flows. Traffic analysis and flow identification could be integrated in the later *TEL process* with ease; still, it is desired to keep it as a discrete instance. The output of the TA process is a uniform table, which feeds *TEL*. Any classification scheme with any number of classes can be integrated here (for instance, when Deep Packet Inspection (DPI) tools are used and an arbitrary scheme based on DPI output is applied), as long as it can be mapped to the later described *Class Flow Table (CFT)*.

2.5.1 *Packet Classifier*

The DS Code Point is extracted from a passing packet and mapped to a fixed class code c_k in the left column of table 2.1. Only five selected DSCP encodings are used; EF and four AF PHBs. There are no AF sub-classes and the drop precedence is not regarded. This is because the drop probability of a packet is not a measured or estimated parameter in the system. There is no determined code for the Default PHB, which equals non-marked packets and those packets whose DS field cannot be parsed in the IPv4 TOS field or in the IPv6 Traffic Class field.

Table 2.1: Used DSCP range

CODE	PHB	DSCP RANGE
c_1	EF	/
c_2	AF class 4	AF41 (DSCP 34) - AF43 (DSCP 38)
c_3	AF class 3	AF31 (DSCP 26) - AF33 (DSCP 30)
c_4	AF class 2	AF21 (DSCP 18) - AF23 (DSCP 22)
c_5	AF class 1	AF11 (DSCP 10) - AF13 (DSCP 14)

Since effects of DiffServ on mesh networks are not in the focus of this work, five DSCP categories are sufficient to demonstrate DS integration. If a classification scheme with $\neq 5$ classes is used, a hash function is needed to map each class to c_{1-5}. The hash function can be freely chosen by the network manager.

As described later on, a packet is treated according to the policy mapped to its class. MRs along a route will not change DSCP values.

2.5.2 *Flow Identifier and Monitoring*

Although DS is not a flow-based service, DSCP values are combined with flow information. A flow is determined through a single packet class code c_k and a *five-tuple of IP source / destination address, source / destination port and the transport protocol (TCP / UDP)*. For the sake of simplicity, it is assumed that each packet bearing a valid DSCP value other than 0 was created by an application or protocol which is using either UDP or TCP. When a new packet from upper layers is passing for the first time, this information is stored in a new entry in the *Class Flow Table*. Since the packet class code c_k is derived from the original DSCP, flows having the same five-tuple, but different DSCPs within one of the ranges specified in table 2.1, are grouped into a single entry.

Using the CFT avoids that abstracted class information needs to be transmitted on a per-packet-basis to the *TEL* process. Maintaining a table is preferable and requires less computational processing efforts. It also serves a as base for further QoS-extensions, like statistics (e. g., per-flow performance) and management (e. g., accounting).

2.5.3 *Process Output and Algorithm*

The algorithm for both packet classification and flow monitoring is depicted in Figure 2.6. It has been adapted to use DiffServ-based classes.

2.6 CHANNEL ALLOCATOR

When no centralized instance to distribute channels is available, every node has to deploy a distributed *Channel Assignment* algorithm in order to exploit channel diversity. The ability of a node to individually perform CA underlines the decentralized character of a WMN.

The principal motivation to deploy CA is to decrease interference and congestion on a single channel. This is achieved by mapping *radios and channels* to neighbors. The way how both resources are handled has a pervasive effect in the novel node system, down towards packet scheduling.

The design of static, dynamic and hybrid CA schemes is a discrete problem in MIMC mesh design[26]. The optimized usage of bundled resources, in order to achieve a specific network behavior, describes a task independent of the choice of channels. Therefore, *CA* is considered an *external* process in Fig. 2.5, with a *static* output.

The required table output of the CA process is taken as a *recommendation* to assign radios and channels to neighbors. There is no feedback whatsoever to *CA* process. CA process is also not triggered to rerun upon certain events.

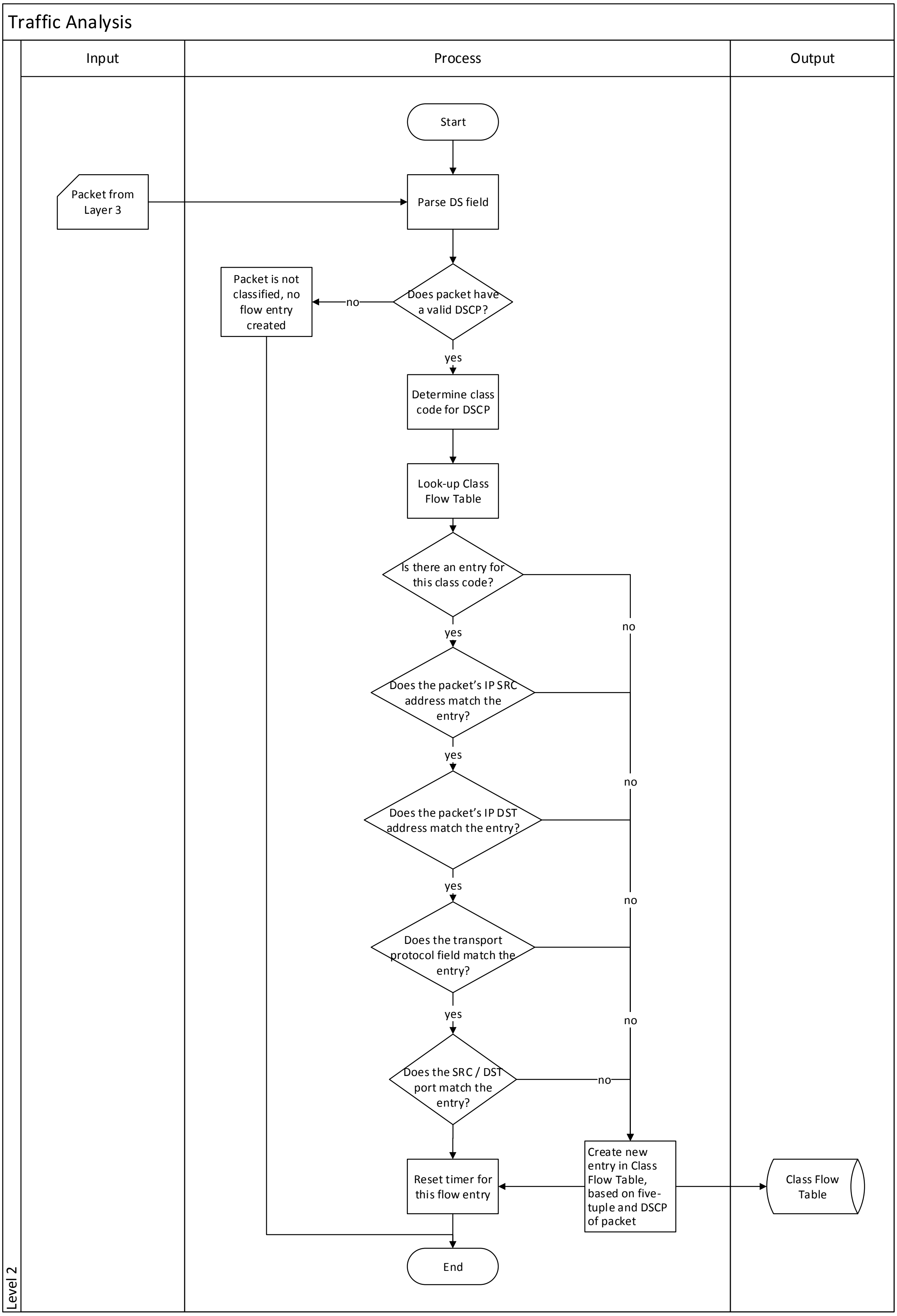


Figure 2.6: Traffic analysis process

2.6.1 *Requirements*

A list of requirements and universal aspects of the chosen CA protocol's behavior has been defined:

CHANNEL TYPE DISTINCTION:

Radios can operate on exclusive channels or on shared channels. In the first case, only a single 1-hop neighbor is reached with this channel. In the second case, more than one neighbor are reachable. Using shared channels is used when the amount of radios is less than the amount of 1-hop neighbors.

CONNECTIVITY ESTABLISHMENT:

Connectivity to all nodes in a radio's transmitting range has the highest priority in a CA algorithm.

PRO- AND REACTIVE:

All assignment tasks shall be executed in both a proactive and reactive fashion. In proactive mode, resources are assigned at network startup and in the following in a repetitive manner, if required. Proactive assignment is independent of channel quality. The proactive start phase shall allow a newly emerging MR to experience the best possible entry in a network, in terms of basic mesh connectivity and interference-avoidance. In reactive mode, a change of channel state / quality may trigger a partial or full reassignment of channels, depending on the implemented policy. Reactive assignment is not bound to proactive evaluation interval.

CHANNEL LIFE-CYCLE AND RE-USE:

If a radio is idle for a predetermined amount of time (no processed frames), it is distributed anew (proactively). The Fallback Scheduling mode maintains silent radios on purpose. These fallback radios pose an exception to this requirement.

QUANTITY OF RADIOS:

CA has to assign the quantity of radios per neighbor, along with their channels.

In principle this decision does not fall within the scope of CA. But later requirements foresee that CA distinguishes between priorities of neighbors and assigns resources based on this analysis. Since a neighbor's "importance" is evaluated anyway, it is a sensible decision to allow CA to also allocate physical radios.

SINGLE-EDGE DISTRIBUTION:

If a gateway is present in the 1- or 2-hop neighborhood, this next hop to the GW (or leading to it) is privileged; it receives more assigned radios and high-quality channels. The evaluation of neighbor-importance and allocation of resources based on importance shall be adjustable by tunable weights.

EQUAL DISTRIBUTION:

If no GW is present in the 1- or 2-hop neighborhood, channels and radios shall be equally distributed among neighbors. In a best case scenario, each neighbor is served via a unique channel. If there are more radios available than neighbors, the additional resources are randomly assigned. The general motivation behind this is to create active channel diversity among 1-hop links, so interference is proactively avoided. Thus, channel assignment with equally important neighbors becomes a distance-1 edge coloring problem [107].

RESOURCE UTILIZATION:

All equipped radios shall receive a channel.

ADAPTION TO HETEROGENEOUS ENVIRONMENTS:

CA protocol running on a node is able to negotiate channels with neighbors with a deviant amount of radios and set of used channels. This requirement is especially important for gateways, which shall receive more radios, in order to maintain multiple connections to different neighbors without causing bottlenecks.

SINGLE-RADIO CONSIDERATION:

A channel is assigned even if node has only one radio. To enable basic connectivity in a channel-diverse neighborhood, a shared channel is chosen with the purpose to reach the majority of neighbors.

FULL SPECTRUM:

CA shall take advantage of all channels a radio has access to and a node can theoretically use with its neighborhood.

CONTROL CHANNEL:

Possibility to configure a designated CC, on which all nodes may receive channel assignment information. A CC is further treated in subsection 2.6.4)

Further *secondary* requirements are defined in appendix A.2.3.

Within the frame of investigations, the novel DCDbP protocol [95] was proposed, which fulfills all requirements. Its functionality is summarized in Chapter 1, 1.5.2. The output of this CA protocol is included as the *External CA Table* in Figure 2.5. Table 1.4 specifies the output.

2.6.2 *CA Table Evaluator*

CA itself is a modular component, but its list of assigned channels is a crucial system input in the first place, and must be therefore fully reliable and feasible. It is not sufficient to consider the the unverified *External CA Table* as simply given.

The novel *CA Table Evaluation* offers an intermediary trust check between the output of any CA protocol and the system, before radios are actually utilized. It reacts to dynamic channel switches. The process is also decoupled from the input whether a next hop is, or imminently leads to a GW, or from the question if this link is adequately equipped with the best channels and a majority of radios. It is expected that this was considered in the CA protocol.

However, it is generally recommended to seek for an environment without frequent or fast switches.

CA Table Evaluation has the responsibility to decide if the proposed set of channels from the CA can be accepted. For that, it must make the following decisions:

- Is an *initially* proposed channel a valid, usable resource?

- Can it be justified to switch a channel already overtaken in the currently used CA table?
- Can it be justified to shift a radio resource to another neighbor?

The following factors need to be taken into account when evaluating a channel:

- Its *connectivity*, to make sure that the supposed neighbor is actually reachable
- Its current load and previous statistics, like its packet loss rate or SIR, to avoid an interruption of ongoing transmissions
- The current overall load of its bundle, to avoid a sudden capacity drop
- Switching frequency in a given time window, to avoid too short assignment times, or unstable, radical changes in the channel map
- Next-hop type this channel is serving (see "variable allocation stability" in requirements on CA protocols in appendix A.2.3 in this context)

Based on these criteria, the CSC is calculated in the evaluation phase. The operator must define a threshold for a maximum cost. The cost per channel can be calculated upon each change in the proposed CA table, or in a periodical manner. Optionally, CSC calculation can be part of the CA protocol itself. Such a requirement is described in appendix A.2.3.

The process fulfills the basic requirement of testing the *connectivity* per channel. For now, there is no other specific cost metric; a channel is either kept, or discarded when the neighbor is not reachable.

The output is the *Verified CA Table* (the process also adds all local MAC addresses, for the later described central BMT. In the best case, all entries from table 1.4 are overtaken. Otherwise, a reduced (but verified) set of radios is handed over to *LMHPC*. Following processes rely on the evaluation of CA table input. As a last task the process switches channels after verification.

It is crucial to avoid packet loss after switches, which might occur if instances of a distributed CA protocol fail to fully synchronize the new channel set between nodes. This makes a timer for a *safe* (or convergence) period a legitimate option. The safe period delays the actual implementation of the proposed channel set and reveals a basic inertness of the system. It also circumvents the influence of temporary, radical changes on the system after the first seconds of a switch. Once a new *External CA Table* is received, a *Safe CA Table* is created, which only contains the first MAC entry per neighbor from the verified version. It is recommended for mesh operators to configure each mesh node's first radio to a common channel. This guarantees a minimal, but fail-safe setup. During the safe period, OLSR converges with all registered radios to the new channel set and collects link states of all radios, in case ETX or ETT metrics are used. After the period, all current link states are directly available for the PS process. Also, bundles may now include all radios, instead of just one per neighbor.

2.6.2.1 *Requirements*

Additional overhead is necessary in order to test if each next hop is reachable via its corresponding channels. The "arping"[1] tool is considered suitable for this task. Unlike the Internet Control Message Protocol (ICMP)-based ping [108] tool, arping operates on layer 2, is based on ARP and enables to "ping" a MAC address. It even allows to resolve an IP address, so it can be verified if the returned IP is linked to the 1-hop neighbor (based on the RT).

2.6.2.2 *Algorithm*

The algorithm to *Evaluate the CA Table* is depicted in Fig. 2.7.

As a result, it is possible that a local radio is shared with multiple neighbors on a common channel. "Neighbor f" refers to the main IP of a node. In the algorithm there is no further treatment of unused channels. If a node is not reachable via a proposed channel, the channel is not explicitly released. Following the requirements in subsection 2.6.1, it will be reassigned by the CA later on, as it becomes idle.

The outcome of the algorithm is that only those channels are overtaken, on which the neighbor has responded. There is no renegotiation after a conflict (proposed channel is not usable).

---

[1] arping tool, Thomas Habets, `http://www.habets.pp.se/synscan/programs.php?prog=arping`

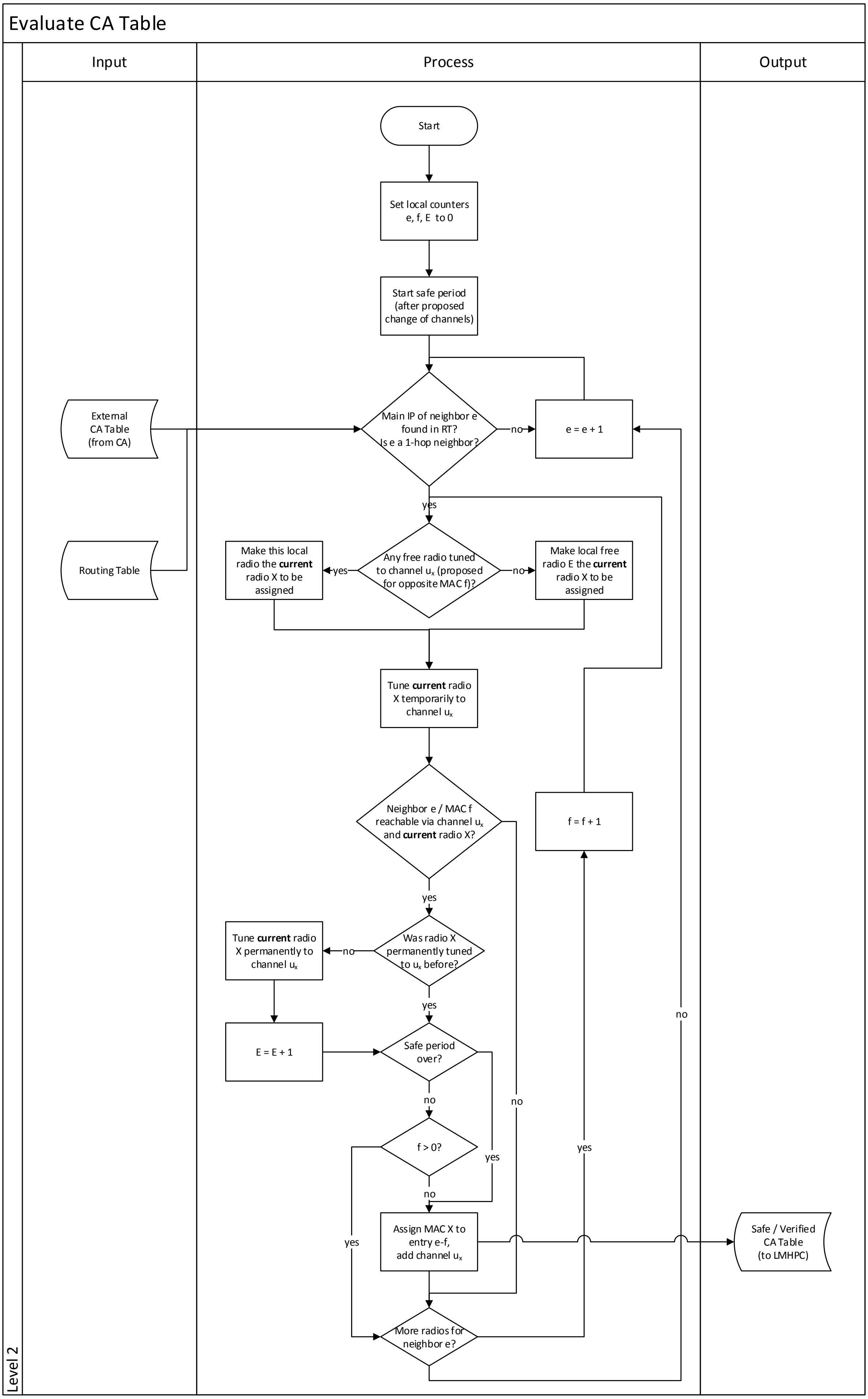


Figure 2.7: Evaluation of CA table process

2.6.3 *The Role of Channel Switch Transition*

In a multi-channel WMN, a new or disappearing neighbor may cause the *CA* protocol to reiterate channel distribution. A temporal transition phase between channel switches has no importance for *TEL* and processes within *MHRRM*. Furthermore, while these processes are executed, the *Verified CA Table* is considered quasi-static; respectively is represented by the *Safe CA Table*. Generally, optimization enabled by *TEL* and *MHRRM* processes take place within a *stationary* CA scheme, with a limited time frame. Different states and their time windows, in which a channel distribution is stable and valid, are independent from another. The transition phase between states is the safe window in which a new CA table is established and all related system components regulate themselves.

2.6.4 *Control Channel*

A control channel is optional for the presented system, but can be easily integrated. Two requirements need to be matched. First, network administration needs to select the same channel index as CC in all MRs. Since all nodes then have a radio (the first one is preferred here) tuned to a common channel, this radio will be member of all maintained bundles. Secondly, packet scheduling needs to be modified:

- Define which type of signaling packets shall be scheduled via the CC (a brief list of recommendations is provided in appendix A.2.2). At least *CA State* and *Mesh Label Distribution Message (M-LDM)* messages should be selected, since both are custom packet generated by the system
- Define if CC is used *exclusively* for signaling (depends on the number of radios; an exclusive use might waste network capacity), or in a hybrid fashion.

A CC can be also used as a back-channel for signaling of all layers (e. g., TCP Synchronize (SYN)/ACK).

2.7 TRAFFIC ENGINEERING LABELER

The *Traffic Engineering Labeling* process adds and removes *one or two custom labels* in the middle-layer module at the ends of a flow, on a per-packet-basis. The first label contains the Next-Hop Field (NHF) and is stored in a separate header. Only packets of vertical flows receive this header. The second label contains the Queue Selector Field (QSF). It is stored in all packets. Label content is composed from cross-layer system input: Mainly OLSR topology- and DS information is used for layer 2 processing of a packet. Both labels take effect in the later explained *LMHPC* process. *Forwarding and enqueuing* are predetermined by these two fields:

FORWARDING, VIA NHF:

> The system follows a *MPLS-like* routing approach. For each packet of a vertical flow which enters the mesh cloud or is generated in it, a label will be added firstly in the *TEL* process at the ingress node. It is removed by *TEL* at the destination MR or the egress GW. It may also cross the WMN between two GWs.

ENQUEUING, VIA QSF:

> Among numerous methods to shape traffic characteristics (see Chapter 1, 1.4), the applied one is to lower or increase a packet's forwarding probability, by forcing it through multiple queues along a route.

2.7.1 *Design Constraints*

Fundamental differences between mesh and MPLS routing can be identified. Within a WMN which is not driven by source-routing, a router usually does not set explicit paths (except with tunneling methods), because it cannot be guaranteed that an initially calculated route will be actually followed. An essential MPLS feature on the other hand is the possibility to explicitly specify a fixed LSP at the ingress router. An ingress router in a mesh would be either the MR generating the packet or the GW node which injects the packet in the mesh cloud. Fixed LSPs cannot be regarded in the system, as the concept of a predefined path of routers is not conformable with the philosophy of ad-hoc routing ("hop to hop" principle).

Concerning the validity scope and the function of labels, there are some similarities. Using the MPLS terminology, the system deploys the "conservative retention mode", because labels are received only from 1-hop neighbors. When a MPLS FEC is a policy valid for a group of labels, then labels correspond to mesh destinations and next hops (out-interfaces) correspond to the treatment for this group, respectively to the policy. Thus, the same label can be assigned to different packets, because they belong to the same flow. The same out-bundle can be assigned to different flows, because they require the same treatment.

Table 2.2 summarizes further basic differences.

Table 2.2: Differences between MPLS and the system-specific label-based forwarding

MPLS	TEL
Traffic engineering, override IP routing	Standard least cost path calculation
Fixed length label instead of longest prefix match	Similar
Routers announce in-label - DST tuple	Nodes announce out-label - DST tuple
Labels are valid / announced domain-wide	Labels have a 1-hop validity
Labels are not unique	Similar

2.7.2 *Algorithm*

Along with the task to label and de-label packets at the edges of a route, *TEL process* also need to generate and update the Commutation Table, which is provided to the *LMHPC* process. The table is filled proactively. The algorithm embedded in the *TEL sub-process* is depicted in two Figures. Fig. 2.8 describes the labeling of packets entering *TEL* from layer 3 (mainly labeling required) and Fig. 2.9 describes the treatment of layer 2 packets (mainly de- and re-labeling required).

Treatment of broadcast packets is not included in the visualizations of the different algorithms. QSF of a broadcasted packet is set to `0x0`. A custom header is not injected.

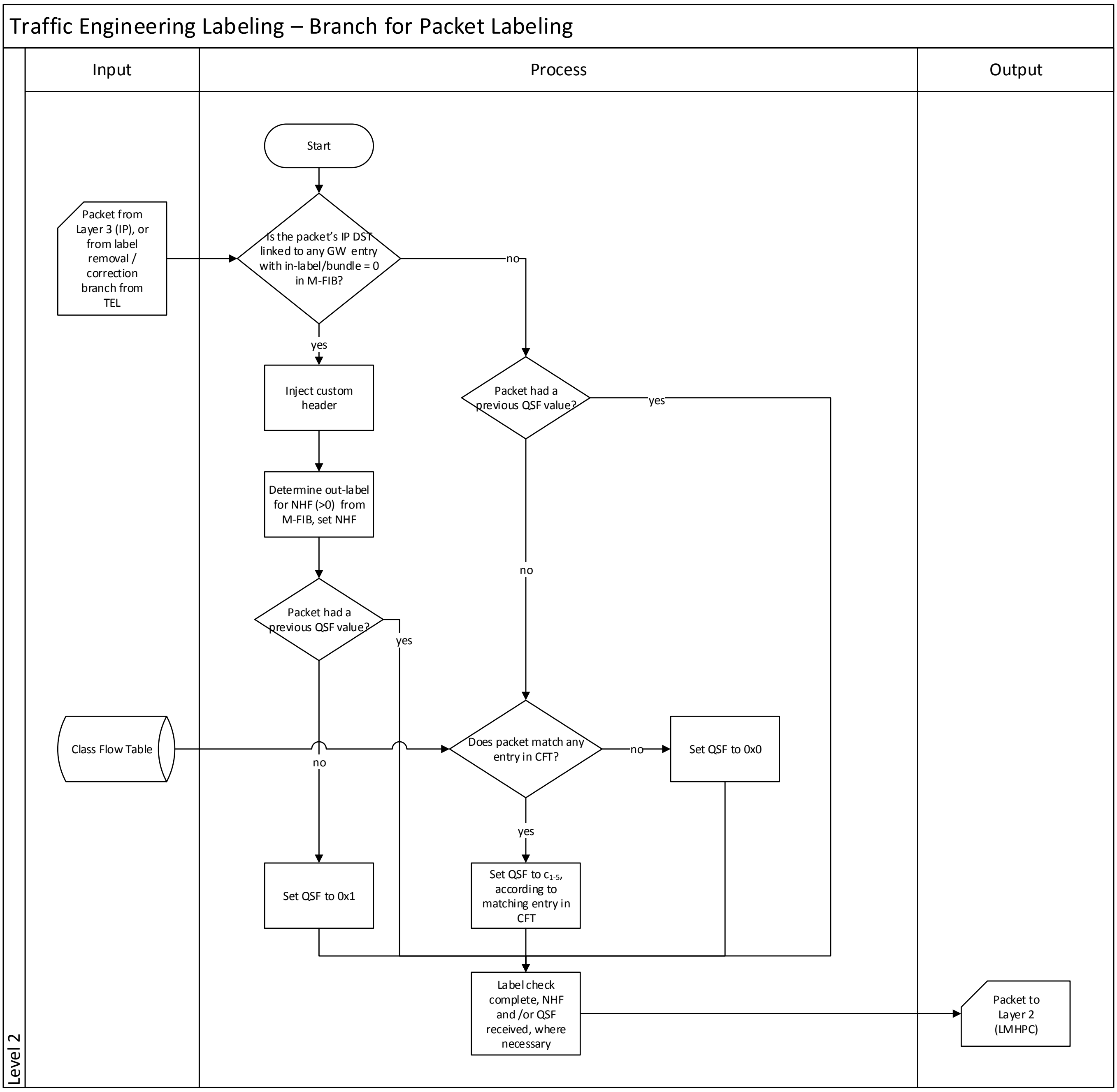


Figure 2.8: Labeling of layer 3 packets process

It is foreseen that all mesh nodes inject headers and labels accordingly, thus all nodes need to implement the described layer 2.5 module. To achieve a compatibility with regular nodes, which do not apply traffic labeling engineering, all algorithms are refined in way that forwarders can detect a missing label and trigger a correction, where necessary.

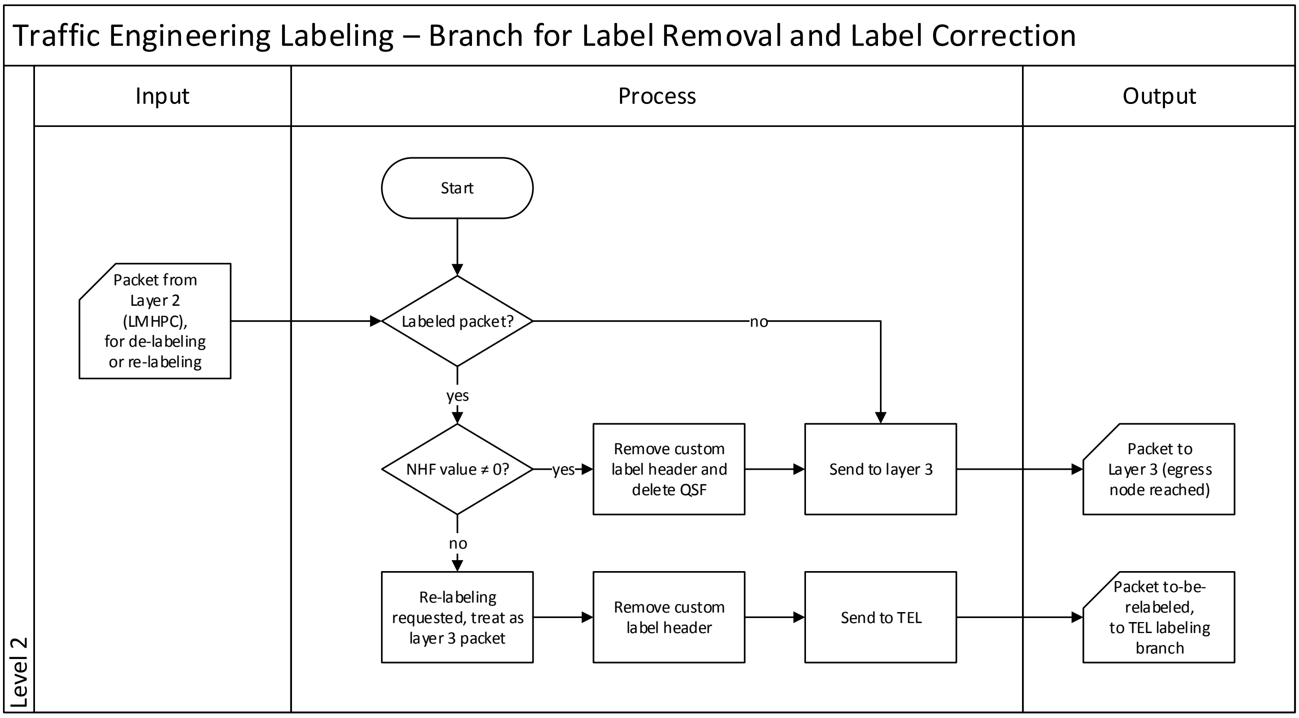


Figure 2.9: De-labeling and correction of layer 2 packets process

2.7.3 *Label Format*

The composition of NHF and QSF shall be described, as well as related design choices. Table 2.3 evaluates the usability of fields from the 32-bit MPLS header [71] (left column) for NHF/QSF (right column). The middle column identifies equivalents to the system.

The reduction of unnecessary features leads to a compact label design in *TEL*.

The two labels basically contain destination and traffic information. NHF has a length of 6 bits and QSF a length of 3 bits. In a forwarding MR, NHF is processed first to swiftly determine the next-hop node to be used, in case the packet bears the additional 6-bit header. Afterwards, QSF determines the priority queue to be used.

Figure 2.10 reveals their position in an encapsulated packet. The extra header is placed between the 14 byte MAC header and the 20+ byte IPv4 header.

Whenever tunnel- or overlay protocols such as MPLS, IP Security (IPsec) or Point-to-Point Protocol over Ethernet (PPPoE) [109] add encapsulation overhead, Data Link Layer (DLL) Maximum Transfer Unit (MTU) becomes an issue. Fragmentation (with IPv4), or packet drops (with IPv6) by MR IFs shall be avoided [51]. TCP Path MTU

Table 2.3: Usability of MPLS header fields

FIELD	EQUIVALENT	USABILITY
(Forwarding) Label Value	NHF, described in Section 2.7.4	The equivalent field is used to enable a fast lookup process of the next hop. The length of 20 bits has been reduced to 6 bits.
Exp	QSF, described in Section 2.7.5	DiffServ support has been overtaken in the system as well, although PHBs have been partly summarized. The original 6-bit IPv4 DS field has been reduced to 3 bits. ECN is not used.
S	None	The Bottom of Stack bit is not used, because only a single header is necessary for point-to-point transmissions in a single mesh cloud. This may be an option for future interdomain routing though (as it is foreseen with MPLS). Also, if the solution is combined with VPN-related headers.
TTL	None	A TTL counter is not needed, since this functionality is included in the IPv4/6 header. Also, taking care about loop prevention across several domains is not within the scope of defined tasks, as the optimization is focused on a single homogeneous mesh backbone or client network.

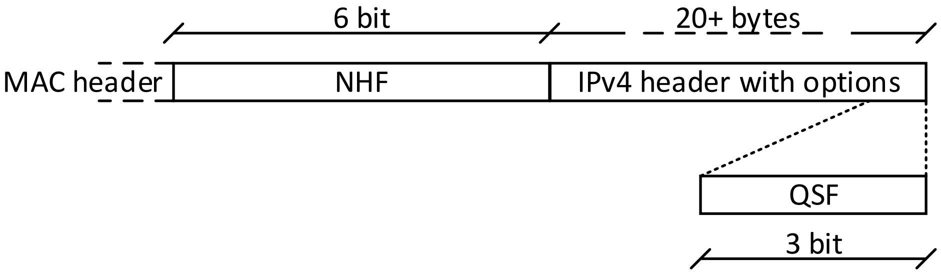


Figure 2.10: Custom header

Discovery (PMTUD) [110] with IPv4 Maximum Segment Size (MSS) adaption excludes UDP [111] and other protocols (IPv6 includes PMTUD on layer 3 though [112]). However, while a MTU of 1500B is common for Ethernet, 802.11 offers 2.272B [109]. This is sufficient for intra-mesh encapsulation. As a precaution, it is still recommended to either increase the MTU across the mesh by 2 bytes for the new header (e. g., to 1502B), or decrease the MTU at the ingress / egress MR by 2 bytes (e. g., to 1498B).

2.7.4 *Label Assembly of NHF*

The Next-Hop Field allows a packet coming from, or going to a GW to be fast-forwarded. To enable this, label commutation operations are required and listed in table 2.4.

Table 2.4: Label commutation operations

ROLE OF NODE	MPLS-EQUIVALENT	MPLS-OPERATION
Vertical traffic generator	Ingress router	Push / impose label
Forwarding node	LSR	Swap label
Vertical traffic receiver	Egress router	Pop / dispose label

2.7.4.1 *Gateway Traffic Identifier and Ingress Node Tasks*

A strong requirement of the system is that both ends of a vertical flow are able to detect themselves as such. The recognition implies that nodes do not need to specifically identify and mark GW packets outside the layer 2.5 module. With OLSR, gateway nodes typically announce their connectivity to the Internet via HNA messages. In a node's routing engine, those entries are stored separately as 0.0.0.0, along with the actual main IP of the node (see appendix A.2.4 for an example topology).

All nodes in the network may generate traffic flows, thus all can become ingress MRs. *Non-GW ingress MRs* label outgoing packets with an NHF value $v \neq 0x0$, but only if their IP DST matches an external DST outside the mesh cloud / subnet. This represents an *upload* path to a GW. The labeling process is always performed at the beginning of a route. It is not necessary to consider the IP Source (SRC) address to identify GW flows. If the DST IP of non-GW ingress MRs is not external, no header is injected.

Ingress GWs will always mark their outgoing packets with a label value $v \neq 0x0$, no matter of their destination. This represents a *download* path from a GW. The destination can be either a client in the same WMN, or another GW.

GW flow detection based on the OLSR topology is also important to define the QSF. GW packets are favored in the forwarding process via beneficial queuing policies.

2.7.4.2 *Design Constraints*

A design requirement is to facilitate multi-hop forwarding of GW flows. To keep internal tables with FIB functionality [71] and NHF label sizes small, it was decided to *only list entries of GW endpoints* in the FIB-like table. For a MR, these are IPs associated to GWs to external networks (for outgoing vertical traffic). For a GW, these are IP DST addresses of MRs or of other GWs. If applied to standard MPLS, a GW flow endpoint would represent a special, selected sub-domain, amongst the standard ones (which would equal intra-mesh flow endpoints). As it is shown in the following, these table entries allow GW packets to be selectively fast-forwarded, bypassing a possibly time-consuming IP lookup.

2.7.4.3 *Mesh Forwarding Information Base*

The Mesh Forwarding Information Base (M-FIB) is maintained at the control plane. The table lists visible *GW flow endpoints*. M-FIB is fed by both signaling input and the mesh topology: Next-hop specification per DST is directly transported from the layer 3 RT to the M-FIB. Thus, label inclusion does not affect routing behavior. M-FIB is constructed within *TEL*. But for the actual commutation operation, the reduced commutation table Mesh Label Forwarding Instance Base (M-LFIB) for the forwarding plane is produced and handed to *LMHPC*. The M-FIB is described in table 2.5.

Table 2.5: Mesh Forwarding Information Base table

IN-LABEL	IN-B. b_h	IP DST OF s_o	OUT-LABEL	OUT-B. b_h
label	b_1	IP_0	label	b_1
...	...	...	...	...
label	b_m	IP_{l-1}	label	b_m

Legend for table 2.5:

b... bundle

h... bundle index

m... current number of registered neighbors

IP DST... IP (v4) destination address of GW flow endpoint

s... GW flow endpoint

o... GW flow endpoint index

l... number of GW flow endpoints in the WMN seen from this node

To outline the M-FIB's role in labeled packet communication, two practical examples are listed in appendix A.2.6.

To enable label swapping, in-labels need to be determined. In-labels are the out-labels of all 1-hop neighbors and are obtained via signaling. In-label and in-bundle values for locally determined GW DST entries are set to `0x0`.

The in- and out-bundle are provided by the bundle-index, which is unique per 1-hop neighbor. In- and out-bundle are never the same in a table entry. For this, the Simple Bundle Table (SBT) table is used.

The out-label contains a distinct random value between 1 and $2^6 - 1$ and is put in the NHF. It is unique within an out-bundle, respectively is unique per next hop. The validity of an out-label corresponds to the uptime of the corresponding node. The binding of an out-label to a bundle is locally valid in the 1-hop neighborhood. M-FIB tables of GW MRs are special. Since a GW MR always creates outgoing GW flows, it needs a unique out-label (within each out-bundle) for every destination in the WMN. Due to the NHF length of 6 bits, a gateway can maintain $64 - 1$ GW endpoints per next-hop. In the worst case, a GW with only one next hop available in the topology could serve 63 clients. However, NHF size can be increased to upscale the amount of addressable nodes. For bigger networks, NHF length could be increased by 4-6 bits; to allow 1023 to 4095 GW endpoints per next-hop. Out-label and out-bundle are two separated columns, because multiple GW flow destinations can be mapped to the same out-bundle. The hash functionality for this mapping is provided by OLSR path calculation, which summarizes ranges of IP addresses (i. e., several destinations) to single next-hops. A key aspect to enable a correct process flow is that OLSR will always assign a discrete next-hop ($\widehat{=}$ out-bundle) for every DST.

Routing information comes directly from the RT, which is filled by the mesh protocol. To determine the proper out-bundle for an incoming packet, the next hop for the regarding GW end-point is extracted from the RT.

The M-FIB table is updated each t_{M-FIB} seconds, or whenever RT changes.

Figures 2.11 and 2.12 (2 parts) shows how the M-FIB table is maintained for locally recognized GW flow endpoints. The process which fills the M-FIB with entries received from the 1-hop neighborhood via M-LDM packets is shown in Figure 2.13.

Table 2.6 summarizes the interpretation of label contents in this work.

Table 2.6: Overview of label construction guidelines for M-FIB entries

ACTION	IN-LABEL	IN-BUNDLE	OUT-LABEL	OUT-BUNDLE
Locally detected GW flow end-point, push label	0x0	0x0	$\neq$ 0x0	$\neq$ 0x0
Packet of vertical flow forwarded	$\neq$ 0x0	$\neq$ 0x0	$\neq$ 0x0	$\neq$ 0x0
Packet of vertical flow reached GW end-point, pop label	$\neq$ 0x0	$\neq$ 0x0	$\neq$ 0x0	0x0

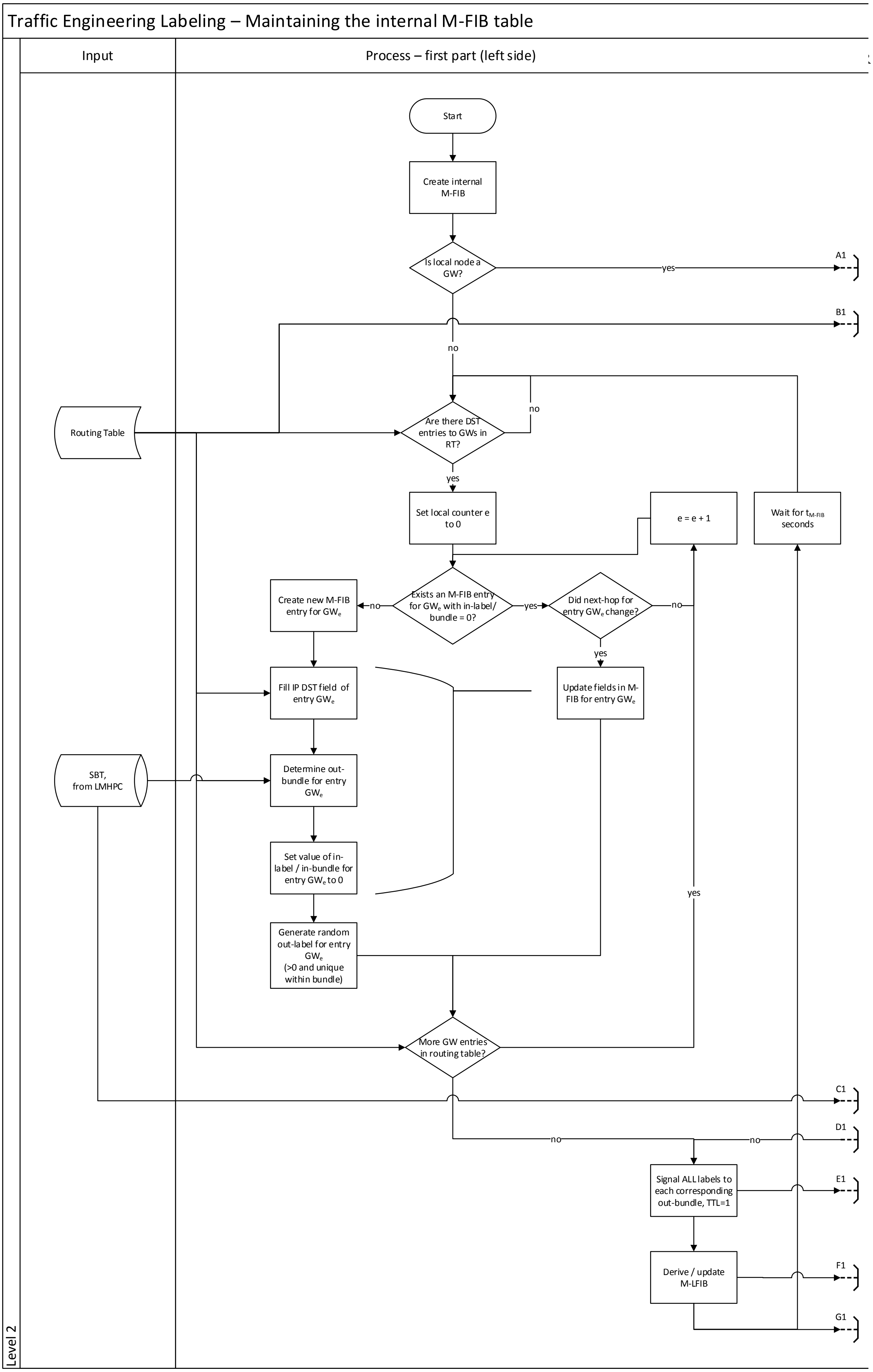


Figure 2.11: M-FIB maintenance process - locally generated destinations (part 1)

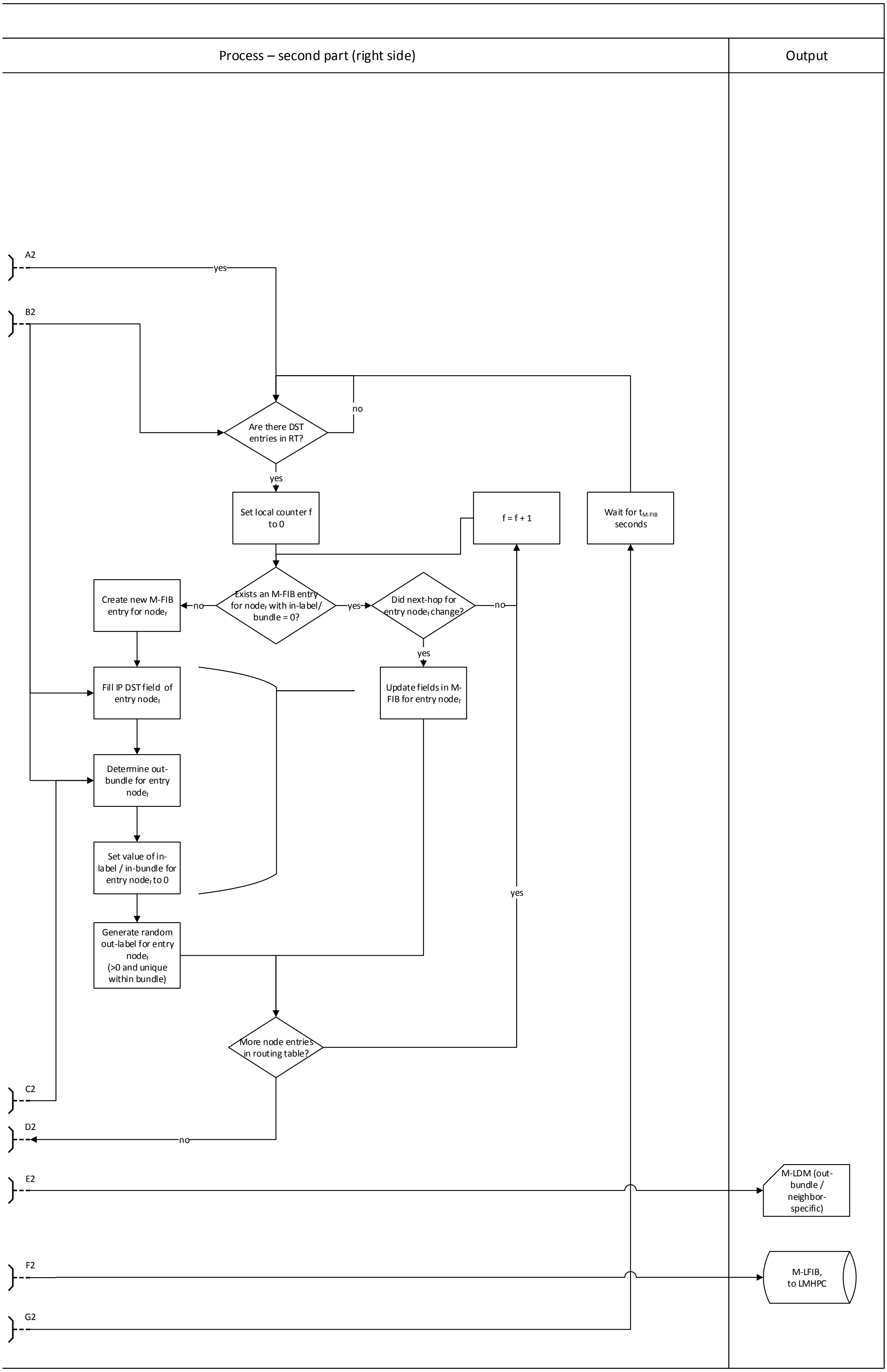


Figure 2.12: M-FIB maintenance process - locally generated destinations (part 2)

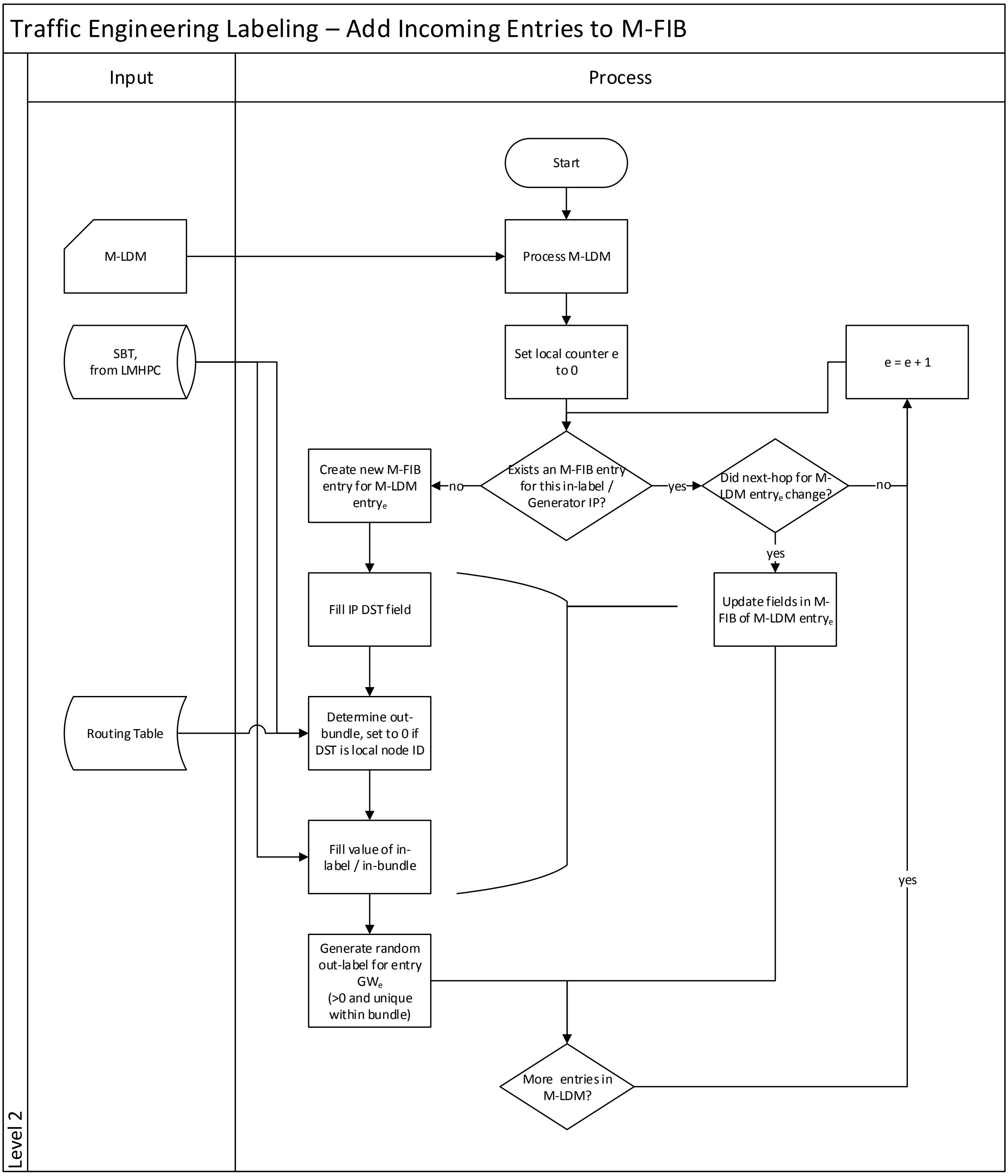


Figure 2.13: M-FIB maintenance process - destinations obtained from signaling input

2.7.4.4 *Label Exchanger*

The in-label (respectively, the previously used out-label) is needed to carry on the packet's original destination during the fast-forwarding process. Labels are not assigned on-demand [113], but determined and exchanged proactively. Mesh Label Distribution Protocol (M-LDP) is a lightweight, novel signaling protocol, which was inspired by LDP. Signaling is performed in the control plane. Each node announces its chosen out-labels for its DST entries in the M-FIB.

A *M-LDM* packet has been designed. Its header is depicted in Figure 2.14.

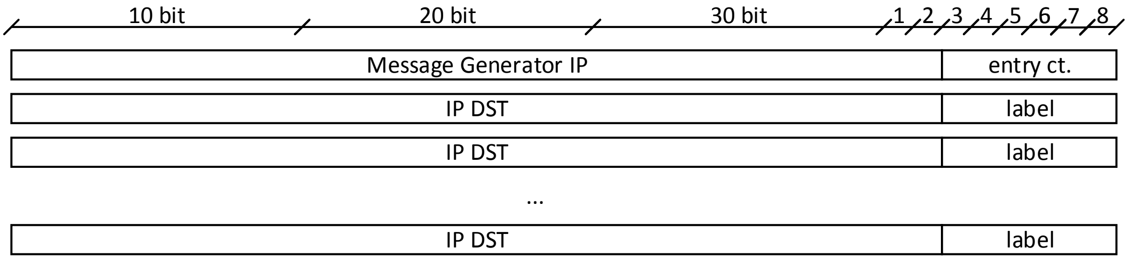


Figure 2.14: M-LDM header

The entry counter specifies the amount of DSTs listed in this message. A TTL counter is not included. TTL for this message type is always 1, comparable to OLSR HELLO messages.

Nodes announce GW flow endpoints *specifically and selectively* (unicast) to all out-bundles / neighbors. As shown in the example in appendix A.2.6, Fig. A.4, a node announces both locally created upload paths (those entries with an in-label / in-bundle value of 0x0) and those which it has received from previous hops.

The LDP-like signaling approach deploys a spanning tree signaling scheme, where *a GW endpoint depicts the trunk* and the branches depict upload paths coming from opposed endpoints. There is only a single branch available between in- and egress MR.

For each out-bundle, an individual M-LDM is created. Using an individual signaling scheme for label synchronization causes additional overhead, but also guarantees independence of the mesh protocol. However, if the system is bound to OLSR, signaling overhead can be *omitted* by fully integrating it in proprietary mechanisms: If the ETT metric is implemented as in [45], [50], OLSR unicasts packet-pair probings to each neighbor. A *M-LDM* message is then stored in a probe's payload instead[2]. OLSRv2 [36] typically offers 512B probe payload, at a default refresh rate of 1s. Another alternative signaling approach is discussed in appendix A.2.5.

Concerning signaling, M-LDP can be seen as a counter-approach to OLSR's link state principle, in a conceptual sense: Additionally to pure link quality parameters, routing related information is distributed.

A side effect with M-LDP is that a node receives information about GWs via a second way from all of its neighbors, although OLSR has placed this information in the routing table already. Hence, the signaling with M-LDM messages means storing and repeating redundant information. The related subject of RT inconsistencies between nodes is covered in Chapter 1, 1.2.7.

2.7.4.5 *Label-Switching Node Tasks*

Nodes which are solely forwarding (i. e., act as LSRs) carry out almost no tasks for a packet in the *TEL process*, respectively in the control plane. The proposed fast routing scheme strongly relies on correctly labeled (GW) packets. But a strict dependency is not given, otherwise label conflicts would lead to immediate packet loss. To implement a security strategy, *LMHPC* can request a re-labeling for an unidentifiable NHF label value (i. e., in-label is not found in the commutation table / M-LFIB. This fallback strategy is nearly similar to a regular IP lookup process, but instead of using the RT, the GW or end-point corresponding to the packet's IP DST is looked up in the M-FIB.

2.7.4.6 *Egress Node Tasks*

The receiving mesh end-point of a GW flow carries out the typical task of an egress router. *TEL process* pops the complete custom label depicted in Fig. 2.10. In comparison to MPLS standards, the "penultimate hop popping" method is not applied here. An egress MR has to register by itself if it represents the mesh end-point of an

---

[2] A corresponding parser and message handler needs to be implemented in OLSR.

incoming packet. If the commutation table lookup for a *layer 2 packet* results in an out-bundle with a value $v = 0x0$ and an out-label value $v \neq 0x0$, the custom header is removed and it is forwarded to higher layers.

2.7.5 *Label Assembly of QSF*

The treatment of QoS demands is enabled by the 3 bit QSF in Fig. 2.10. The second label ultimately determines which priority queue is used for the packet. For the sake of simplicity, an arbitrary number of 3 bits has been chosen, which would allow for 8 different classes, whereby only seven Internal Traffic Class (ITC)s are actually considered. Mesh operators can extend the size, in case more ITCs are desired.

It would be feasible to store the QSF in the same extra header introduced for the NHF and simply add it to horizontal packets as well. But this brings technical problems, because layer 3 expects an IP header in the forwarding process, before the output queue is selected. If extra-labeled packets were sent to layer 3 for classic routing, this would create compatibility problems between layer 3 and 2.5. An option would be to de- and re-encapsulate horizontal packets in *TEL* in every intermediate hop, but this potentially creates additional workload.

Therefore the second custom label, which contains the QSF, is stored within the IPv4 header. in intermediate nodes, the IP header is *parsed* for the QSF in the layer 2.5 module. QSF can be stored as one of the various options enabled by the "Options and Padding" field[3] [114]. This extends the header by at least 32 bits, but also assures compatibility on layer 3. The TOS field was not chosen to store QSF. It is not desired to overwrite the original DSCP, which may be useful for QoS across domains. An alternative storage place would be enabled by setting the "Don't fragment" flag bit. This action disables the use of other fields related to fragmentation[4], which become available then for storing the QSF. With this variant, the IPv4 header keeps its standard size of 20 bytes.

---

[3] The "IP Header Length" field must be adapted accordingly, i. e., from a value of 5 to 6

[4] such as "Identification" or "Fragment Offset"

A per-hop-behavior for an ITC is implemented by putting a packet in one of the differently prioritized queues. The distribution of ITCs to queues is depicted in table 2.7.

Table 2.7: Queue mapping in forwarding label

QUEUE	QSF VALUE	INTERNAL TRAFFIC CLASS	DS EQUIV.
q_1	0x1	Gateway traffic	/
q_2	0x2	c_1	EF
q_3	0x3	c_2	AF4
q_4	0x4	c_3	AF3
q_5	0x5	c_4	AF2
q_6	0x6	c_5	AF1
q_7	0x0	Low priority ("best effort")	Default PHB

QoS information from the *CFT* is used only once during the initial labeling process: *TEL process* checks if the packet's IP SRC/DST address, source / destination port and the transport protocol match one of the flows in the table. If so, the according class code c_k is written in the QSF.

If the packet belongs to a GW flow, it will always receive the value 0x1, independent of its actual ITC. Thus, importance of GW traffic excels any other ITC specification. Again, this follows the criterion that GW traffic must be optimized with all given resources.

It is redundant to set QSF to 0x1, because packets could be put in the highest priority queue simply by checking for a 6-bit header. However, with the proposed method queuing is independent of the deployment of layer 2 forwarding (improved modularity).

If a packet neither belongs to a GW flow, nor bears a class code, it is labeled with the value 0x0 as low-priority traffic.

2.7.6 *Commutation Table*

The *M-LFIB* is the *actual commutation / switching table* and is sent to the *LMHPC process*. It's columns are derived from the M-FIB table 2.5. Appendix A.2.6, Fig. A.3 contains a related numerical example. Thus, *M-LFIB* header contains the fields *In-label, In-bundle, Out-label and Out-bundle*.

M-LFIB is maintained separately, which theoretically allows a network operator to configure fixed paths in the forwarding plane. Comparable to the layer 2 forwarding approach in [113], the system in principle can serve as a traffic engineering platform for routing manipulation. In such a fixed LSP scenario, OLSR routing calculation would have no impact on the actual flow of packets.

2.8 MULTI-HOP RADIO RESOURCE MANAGER

The *MHRRM* is composed of the two novel sub-processes *LMHPC* and *TX Packet Scheduling*. The queue management is not specifically listed in Fig. 2.5, as it has no particular table or signaling in- or output. It has been included in *LMHPC*.

In terms of packet processing, *TEL* represents the control plane, while *MHRRM* is the forwarding / data plane. *MHRRM* is a packet distribution engine. As with the previously introduced processes, vertical / outer-mesh and horizontal / intra-mesh are separately regarded. The first category is favored in the forwarding process. *MHRRM* also manages bundles and has direct control over aggregated capacities of multiples radios. *MHRRM* is a crucial process and needs to be tightly bonded to the specific characteristics of a mesh network.

Fig. 2.15 visualizes the treatment of three representative packets, which enter a node. The first encapsulated packet belongs to a GW flow and is forwarded. In the ideal case, it is not visible to layer 3 in intermediate hops. Second and third packet are forwarded between two non-GW MRs. Fig. 2.16 contains a numerical example of the processing of NHF and QSF (both in binary notation) of two vertical and one horizontal flows. The third packet in Fig. 2.16 has arrived at the end of a GW connection.

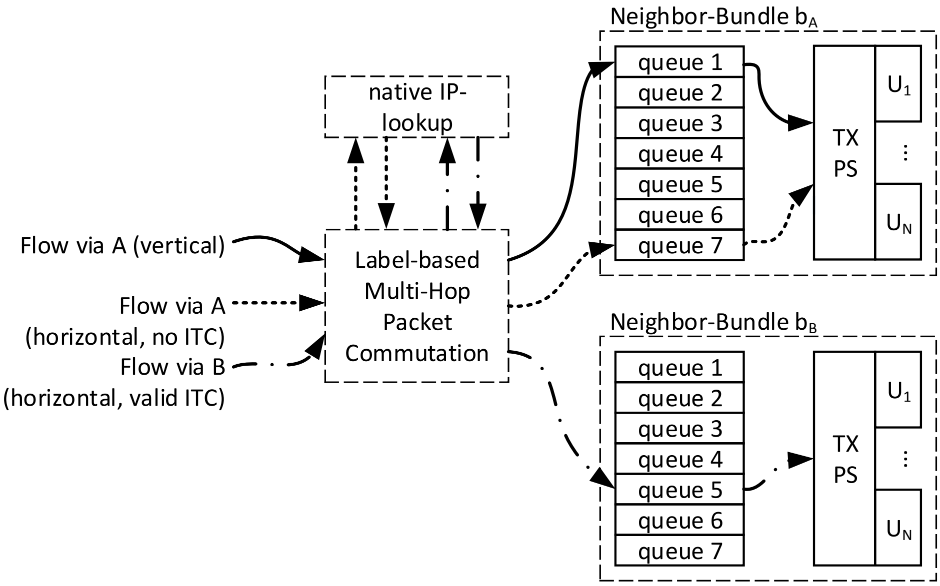


Figure 2.15: Example of flexible label treatment

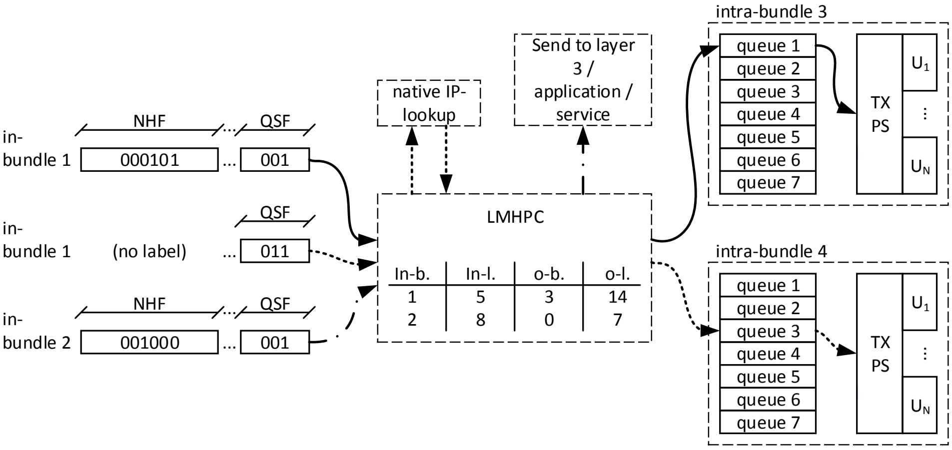


Figure 2.16: Example of flexible label treatment (numerical example)

PS and enqueuing perform different tasks on a packet. PS applies a scheme to distribute packets in a bundle. Queuing system interprets packet classes, enforces policies to incoming packets and ultimately determines their sending order. They are two separate processes. Enqueuing is executed before PS and once a packet leaves the queues its priority is not important any more in the local node. A queue regulates the removal probability of a particular packet, whereas PS determines which TX radio is chosen.

As a requirement for PS and enqueuing, interface / bundle management is described in advance.

LMHPC covers two separate processes; one for the definition of bundles and the other one for packet treatment. There is no specific algorithm for building the required tables; all aspects are included in the text. The algorithm for packet commutation is depicted in Figure 2.17.

2.8.1 *Virtual Interface and Bundle Management*

The layer 2.5 module works as an abstraction layer to upper layers, which manages activities of physical radios. Local MAC address information is extracted from the *Safe / Verified CA Table* and not from native system / OS databases, so *MHRRM* depends on the correct construction and maintenance of this table, without exception.

2.8.1.1 *Virtual Interface*

A *single* VI is provided for layer 3 and superior layers, despite the presence of multiple IP/MAC addresses. LB/MIMC is not relevant for the routing process and is managed below it, in terms of forwarding. I. e., from the IP layer point-of-view, a VI maps the TX behavior of a single IF. This is beneficial: TCP for instance foresees a single IP for SYN/ACK exchange, session control and data, because a common user will unlikely equip more than one IF [115]. But nowadays, especially mobile devices converge with heterogeneous networks (4G/5G, WiFi, and so on). While an Multipath Transmission Control Protocol (MTCP) node stack can deal with multiple IPs [116], a universal VI is preferable (although the Network Address Translation (NAT) MTCP issue [115] is not relevant in a mesh). However, addresses of multiple IFs are still registered with OLSR (see Section 2.4.1). The notional interplay between the virtual interface, bundles and radios is depicted in Fig. 2.18.

A physical radio takes the role of an Universal Radio Unit (URU) U_j. A PHY radio can be represented as a universal resource in several bundles. This occurs when the radio is tuned to a shared channel (communication with more than one neighbor). Then, a radio provides multiple URUs; each mapped to a different bundle b_h. N is the number of URUs in a bundle. If a radio provides only one URU, then it is tuned

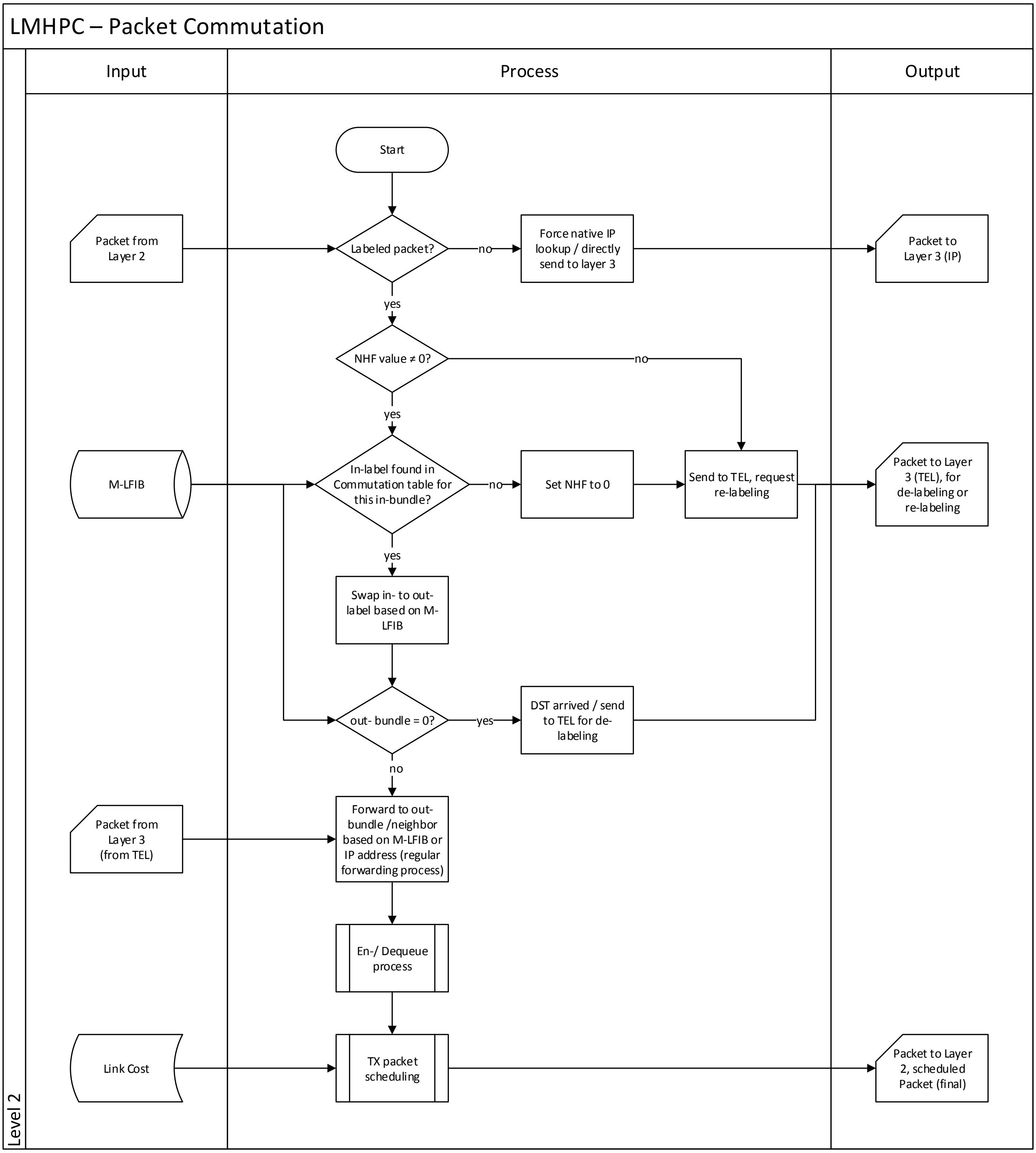


Figure 2.17: LMHPC – Packet commutation process

to an exclusive channel. Hence, radios are not necessarily exclusive to bundles in the system.

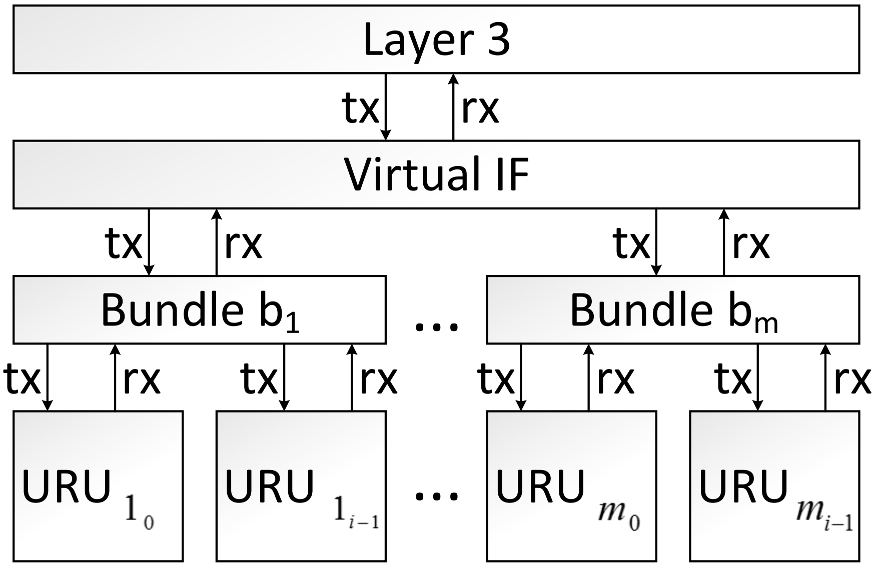


Figure 2.18: The virtual interface

The relationship between resources can be also described as:

$$VI_0 \equiv \sum_{h=1}^{m} b_h \equiv \sum_{j=0}^{n-1} r_j \qquad (2.1)$$

Where:

r. . . radio

j. . . radio index

n. . . total number of attached WLAN radios

2.8.1.2 *Design Constraints*

A *bundle* groups one or more URUs. For each 1-hop neighbor a *single* bundle is created, no matter if only a single radio is assigned to a neighbor in the *Verified CA Table*.

2.8.1.3 *Bundle Index*

The bundle index b_h , $1 \leqslant h \leqslant m$ is determined proactively once a node is initialized. Based on the incoming *Safe / Verified CA Table*, it is mapped to each 1-hop neighbor's main IP address (node ID). This mapping is used inter alia in the *Simple Bundle Table* 2.8, which is signaled to *TEL*.

Table 2.8: Simple Bundle Table

NODE ID OF NEIGHBOR t_p	BUNDLE INDEX b_h
32 bit IP of t_0	b_1
...	...
32 bit IP of t_{m-1}	b_m

Legend for table 2.8:

$t\dots$ neighbor

$p\dots$ neighbor index

To enable packet commutation, a neighbor must be assigned to a bundle index. The neighbor which receives b_1 is determined randomly and p/m is incremented henceforth. Different neighbors (and their respective group of radios) receive separated indices, while URUs connected to the same neighbor have the same bundle index (so a PHY radio may be mapped to one or more bundle indices). If a node ID entry in the *Safe / Verified CA Table* changes, a new bundle index is assigned. Value `0x0` is never used in the SBT because it is reserved for two purposes. First, an in-bundle entry in the *M-LFIB* table is set to zero by the *TEL* process when GW packets are created, or enter via the same node. Second, the out-bundle field is set to zero when GW packets have arrived at their destination / gateway.

2.8.1.4 *Bundle Management Table*

The BMT in table 2.9 in principle extends the 1-hop topology information provided by OLSR, which is passed on to *MHRRM* via the *Safe / Verified CA Table*. The SBT is included in the BMT. The BMT is managed only internally within *MHRRM* / *LMHPC*. Fields are updated dynamically, based on changes in the *Verified CA Table* and radio-specific information: Data not included in the *Verified CA Table* - i. e., the last four columns of the BMT - is related to packet scheduling. The BMT has no input or separate fields for flows maintained in the *CFT*. That is, because QoS information is already encoded in the QSF.

Table 2.9: Bundle Management Table

t_p	b_h	LOCAL MAC	u_x	NB. MAC	v_j	g_j	d_j	F_j
IP t_0	b_1	$M_{local}\ r_0$	0	$M_{t_0}\ r_0$	v_{r_0}	g_{U_0}	d_{r_0}	F_{r_0}
IP t_0	b_1	$M_{local}\ r_1$	1	$M_{t_0}\ r_1$	v_{r_1}	g_{U_1}	d_{r_1}	F_{r_1}
...	...	...	...	...	...	...	...	...
IP t_{m-1}	b_m	$M_{local}\ r_{n-1}$	y	$M_{t_{m-1}}\ r_{n-1}$	$v_{r_{n-1}}$	$g_{U_{N-1}}$	$d_{r_{n-1}}$	$F_{r_{n-1}}$

Legend for table 2.9:

IP... IP (v4) address

MAC... MAC address

u... channel

x... channel index

y... maximum number of available channels

v... link state / metric value of a radio r

g... send probability of an URU U (derived from link state)

N_{b_h}... number of URUs in a bundle b_h

d... counter for sent packets of a radio r

F... Monitor for fallback parameter of a radio (in packets) r

The enhancement of mesh transmissions via traffic engineering and multi-radio exploitation, which are discussed in the following, all require the BMT table entries t_p, b_h, local MAC, $u_x$and neighbor MAC to be static. This dogma to optimize mesh behavior within a singular, non-altering state is interrelated with Section 2.6.3.

2.8.2 *Label-based Multi-Hop Packet Commuter*

LMHPC also works as the packet commutation engine in the presented layer 2.5 module. The main task of *LMHPC* is to commute packets by swapping in-labels to out-labels or to send them to higher layers. LMHPC also forms part of the re-labeling chain. Label swapping is based on the incoming *M-LFIB* table. Mesh routing decisions were previously fed into this table in the *TEL* module. *M-LFIB* and BMT are maintained separately within *LMHPC*.

The novel *LMHPC* allows a *selective*, hop-to-hop-based fast forwarding. Information stored in the NHF enables a faster mapping of a packet to the final TX interface. In the ideal case, a packet is forwarded with complete transparency to the IP layer between the ingress and the egress MR. The lookup process in the *LMHPC* core is reduced to compare short labels with a fixed length, based on a table much simpler than a full OLSR RT, which normally contains *all* IP destinations present in the mesh. If the *M-LFIB* lookup results in a value $v \neq 0x0$ for the out-bundle, the packet is fast-forwarded. Otherwise, it has arrived at its mesh destination. For incoming horizontal packets the system exploits that OLSR always provides a native IP lookup in the MR; a fact not always given for specialized router hardware.

2.8.3 *Intra-Bundle Queues*

The QSF embodies both QoS and topology information. Queues are a practically applied concept in the system; the optimization of queue typical parameters is not foreseen. Still, priority queuing has been adopted in a novel way, to support mainly vertical traffic.

A *packet* depicted in the remaining flow diagrams of the MHRRM Section may be a layer 3-forwarded, a layer 2-fast forwarded, or a locally generated packet. The enqueuing sub-process is shown in Figure 2.19. It further hosts the dequeuing sub-process, as shown in Figure 2.20.

Priority queuing is the only measure the system can deploy in order to react to QoS demands. A guarantee that a prioritization scheme is homogeneously applied on the entire end-to-end connection cannot be provided, as the current packet owner

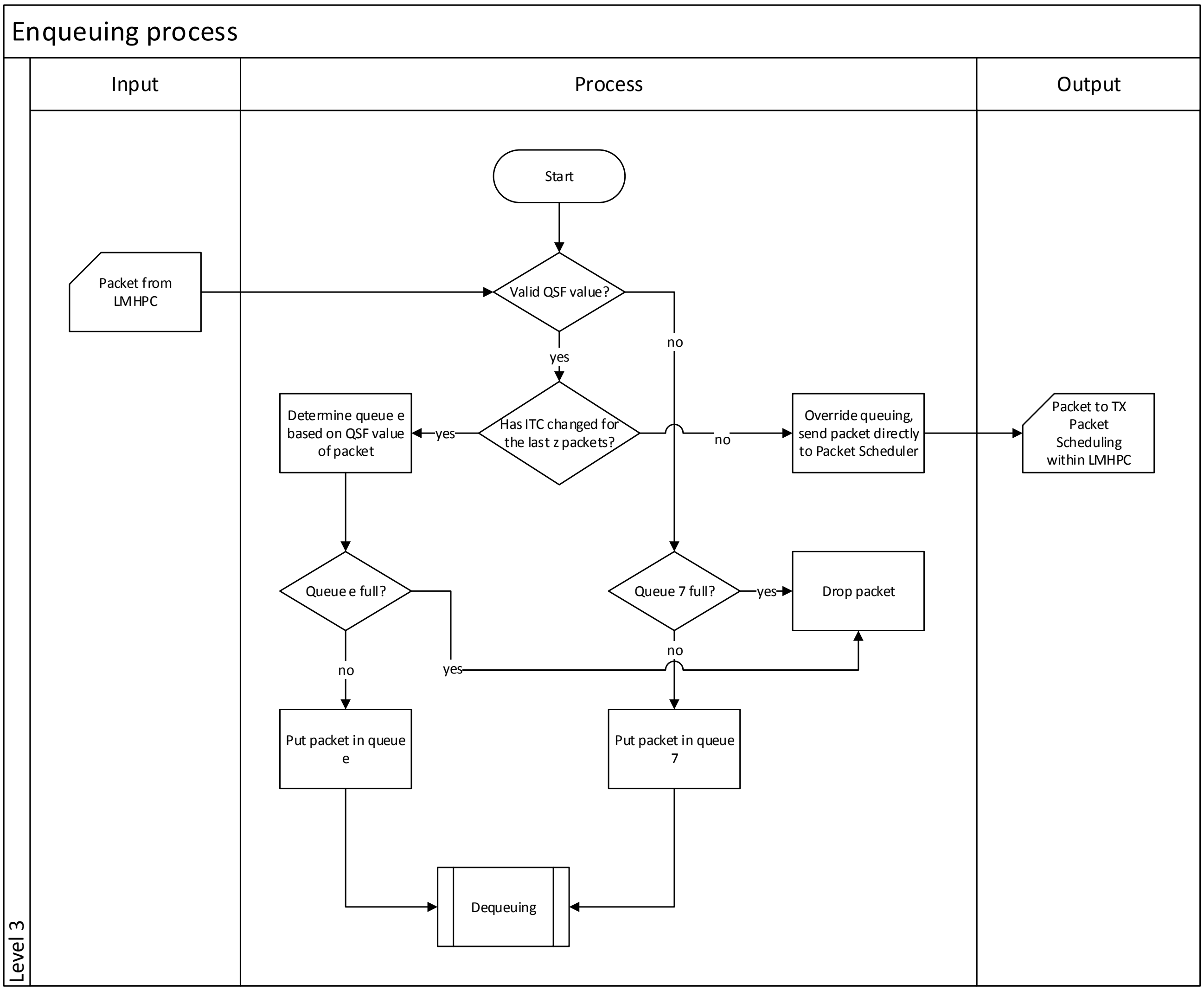


Figure 2.19: Enqueue process

cannot know if the remaining number of hops, link conditions, occurring cross-traffic and interface resources on following links will eventually remain stable.

The existing (and missing) interfaces between the layer 2.5 module and the rest of the node architecture, which have been described so far, reveal a substantial challenge when it comes to queues: *How to decide whether a packet shall be queued or not?* Link states, which are processed later on, may or may not indicate the actual congestion of a channel, depending on the chosen routing metric. Explicit data on the shared medium availability (e. g., parameters from CSMA/CA), current bitrate of a radio, probable waiting time in the MAC queue, and similar (WiFi driver) information from PHY and MAC layer are not available. Layer 2.5 module manages multiple radios and does not consider the packet's chance to be sent for each underlying MAC

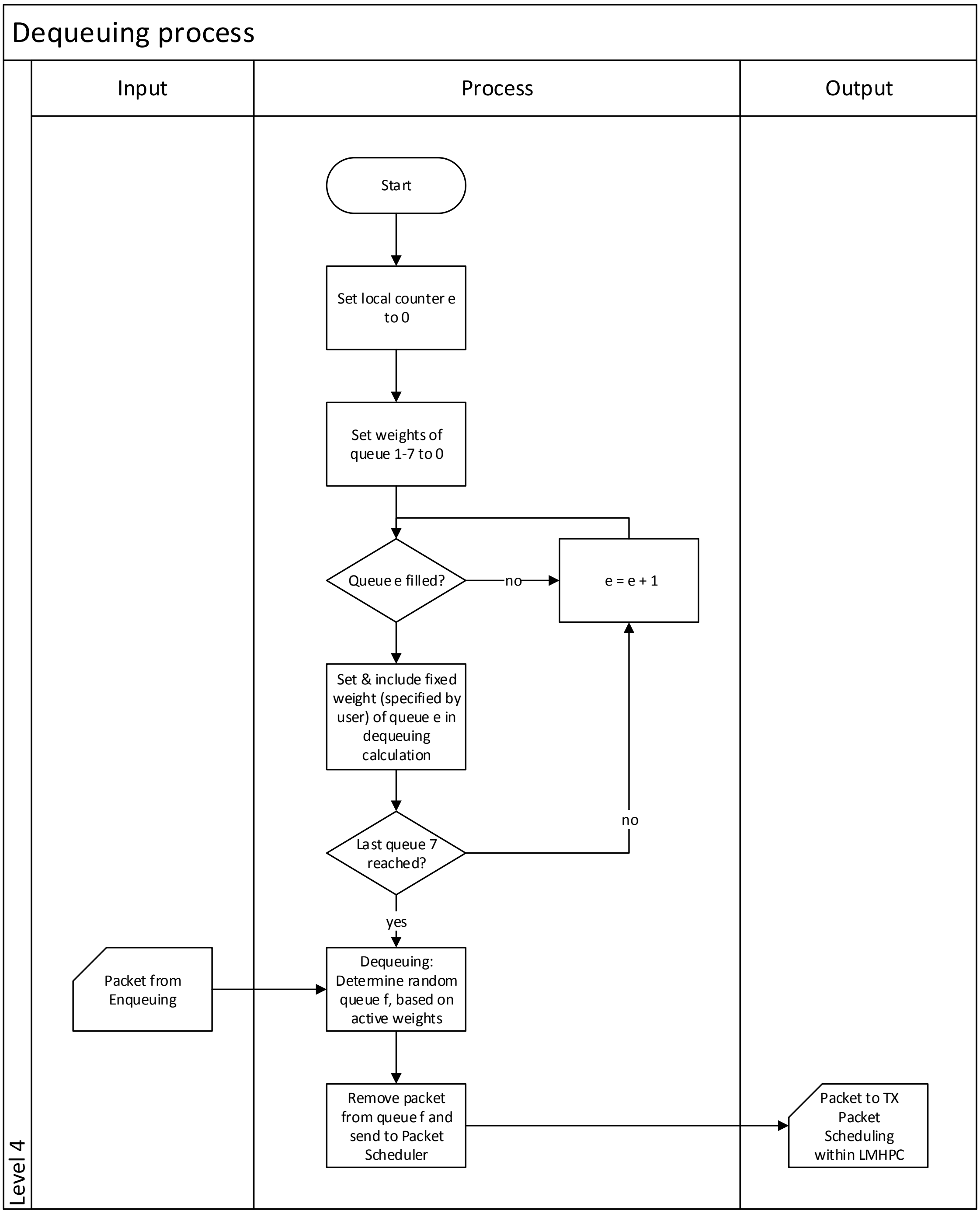


Figure 2.20: Dequeue process

component so far, due to missing cross-layer input. Thus, enqueuing a packet before MAC layer does not seem to be a feasible method at a first glance: An incoming packet with an arbitrary priority would be taken out the queue immediately by the following packet scheduling process, because there is no link quality-related criterion at hand which would make it justifiable to hold back the packet. Hence, when more than one ITC were detected, two ore more queues need to be filled up first, which implies an *artificial delay* of the packets in the middle-layer module. To avoid that all packets are forwarded evenly, despite their priority, queue process includes a mechanism, which enables queuing *only* if different flow priorities are recognized. To make this decision, queuing process introduces a local packet window with a lengths of z packets. If the last z packets had the same ITC, queuing is not applied in this bundle. The window is constantly monitored. In parallel, the same parameter z is used to define the queue space (size) in packets.

Queues per bundle bring several general advantages. Most importantly, queuing allows to rearrange the original, chronological sending order of incoming packets to a class-corresponding order. Also, a *single* vertical flow via neighbor a. will not compete with *multiple* horizontal flows via another neighbor b., which are then subject to queuing in their bundle. As said, the system is generally ready to adapt to DSCP classes, in case the user would like to deploy DS marking. Also, queuing is not tied to MIMC exploitation, which is *in line with the modular design*. Applying priority queues takes effect even if there is only a single radio attached to this bundle. But queuing further enhances packet multi-radio scheduling. A priority queue can be combined with link-state sensitive *PS* by *offering the best radios for the most important packets*.

The mapping of classes to queues is described in table 2.7. *At least two queues are needed*, to consider horizontal / intra-mesh /low(er)-priority and vertical /GW traffic. Five more queues have been included to support a simplified set of DS code points. Network administrators may adapt the amount of queues to their needs; it is not relevant for the presented mesh concept. A single queue is characterized by the following parameters:

- Fixed, tunable amount of queues, which is the same in all bundles
- Tunable queue length (in packets)
- Tail drop principle [117] within each queue
- PFIFO principle [118] within each queue

- Dequeuing policy based on WFQ
- Fixed weight w per queue, manually chosen by the user

In the later implementation, the PS process constantly demands packets and controls the removal and charging of queues. The queue process itself is passive. A request flag - which is set by default, except queues shall be filled - from PS indicates that packets can generally leave the 7 queues. Concerning the applied dequeuing algorithm, it would be obvious to use a simple SPQ-like [119] scheme. SPQ requires no configuration efforts and no definition of weights. On the downside, SPQ might imply that in cases of extreme congestion, lower-class queues might never get served and eventually experience dequeuing starvation. This leads to packet drops in the layer 2.5 module. This policy is too strict and not suitable for a multi-hop network. A packet drop along the route is fatal here, as the sender will eventually resend the packet (with TCP), causing congestion anew. Therefore, dequeuing is handled with Weighted-Fair-Queuing. It allows fair treatment of all queues, combined with the attention to a flow's priority. Compared to SPQ, WFQ lowers the drop probability of less-important queues. In the system, dequeuing is described by:

$$P_a = \frac{w_a}{\sum_{i=1}^{7} w_i} * z \tag{2.2}$$

Where:

$P \dots$ packets taken out of queue a

$a \dots$ queue index

$w \dots$ weight, $0 \leqslant w \leqslant 1$

$z \dots$ queue size / packet window length

So the weight w determines the average removal count for every z packets. In other words, a class treatment policy is realized by the weight. The actual decision of which packet is to be removed next is handled with a random access function. Only weights which correspond to ITCs recognized during the last z packets are included in the calculation of P_a. Weights of non-used queues are temporarily set to zero.

The weight influences the chance of removal accordingly. Statistically, the desired dequeuing profile per queue shall be matched.
The output of the queuing process is a single packet, which was selected based on its sending priority and the amount of other packets in the queues, and their priorities. The following PS process works independent of queue serving, so packet treatment after commutation involves two *separate* decisions: when shall a packet leave its queue and which TX radio shall be used for sending.

2.8.4 *TX Packet Scheduler*

The state-dependent [48] *TX PS* process in Fig. 2.5 determines the TX utilization per URU in a bundle. Between the two general approaches *session- and packet-based* scheduling [120], the latter was chosen. Applied to the system, this means that identified flows are not bound to single radios. Instead, a TX radio r_j for a single packet is chosen by its MAC address (see mapping in table 2.9), respectively by its index j. Appendix A.2.7 discusses additional receiver feedback and packet reordering.

2.8.4.1 *Load Balancing Modes*

A configurable set of LB modes is offered to a mesh operator. Criteria for the selection of modes are based on the motivation to define a simple parameter (mode), which has a significant impact on the behavior of the entire mesh network. The mode parameter reflects two basic network response profiles: *Capacity- and stability-oriented networks*. Both types exploit channel diversity, but for different reasons. The first category uses channel resources in parallel, whereas the second may maintain extra resources as backup options. Each mode's packet distribution scheme, and the statistical data on which it is based on, is applied separately for each bundle. A mode is applied network-wide.

2.8.4.2 *Weighted Fair Scheduler Mode*

The Weighted Fair Scheduling (WFS) mode shifts more load to links with better quality. Radios with the best link states shall bear the majority of packets. The TX radio selection scheme was inspired by the WFQ scheme, which is also used for dequeuing. A TX radio is selected randomly on a per-packet-basis. Therefore, WFS calculates a sending probability per link, to determine its usage frequency. Sending probability then alters the random selection.

WFS mode is tailored to link state mesh routing protocols. Its adaptive character has a strong positive impact on the performance of dynamic networks with *heterogeneous* link qualities. At the same time, it allows a fair treatment of interfaces with underperforming links, to prevent starvation of such.

Appendix A.2.8 specifies how a radio's link state input is converted into a TX probability in the system. WFS mode requires ETT, as it is the more accurate, QoS-related metric. In brief, the reciprocal of ETT is stored in the BMT 2.9. This weight of a radio r is listed in the BMT column ν_j. The column g contains the calculated TX probability. An exemplary output of WFS calculation for a single packet is shown in appendix A.2.9.

Applied to system parameters, sending probability is calculated with:

$$g_j = \frac{\nu_j}{\sum_{i=1}^{N_{b_h}} \nu_i} \tag{2.3}$$

With standard OLSR, ETT and ETX are calculated independent of the current activity of a link [45]. The delay, until the impact of a link state change actually takes effect in the system depends on three intervals:

- The steps described in appendix A.2.8 are performed every `periodWFS` seconds (configurable), or whenever a link state changes...
- ...whereby two relevant intervals are maintained by OLSR, which dictate the link state sample rate:
  - HELLO messages [35]
  - TC messages [35]

The safe period in the *CA Table Evaluation* covers the phase of convergence until link states of new links are stable.

2.8.4.3 *Round Robin Scheduler Mode*

RR mode evenly distributes packets in a bundle. For N URUs in a bundle b_h, each URU U_j will transmit N^{-1} of incoming packets.

2.8.4.4 *Round Robin Scheduler Mode with Fallback Extension*

This mode extends the RR method. From a given set of N URUs in a bundle b_h, a fixed number of "Fallback (FB) Radios" $B, 0 \leqslant B < N$ is reserved, in case one or more of the currently used radios fails. When $B = 0$, standard RR is applied. When $B = N - 1$, *single-interface transmission* is applied on this link, while $N - 1$ URUs remain silent. B is specified by the user, so he or she has full control over the degree of WLAN hardware utilization. This is also relevant for an alternative deployment with fully mobile nodes, where energy consumption becomes a limiting factor.
A FB threshold rate R per radio is maintained in the BMT 2.9:

$$R = \frac{F_j}{d_j} \tag{2.4}$$

R triggers the replacement of an active URU with a FB radio. This threshold is chosen by the user, has a network-wide validity and can be based on any criteria, as long as the chosen metric relates to the condition of a single radio and is measurable in frames / packets. Given the system described so far, it would be feasible to include the link quality provided by the proactive link state routing protocol. Appendix A.2.10 discusses why this is not desired, as well as alternative parameters for F.
The algorithm is depicted in Figure 2.21, where C is the amount of active / non-FB URUs. The dotted line represents the standard core functionality of RR scheduling.

2.8.5 *Design Constraints*

From a layer 3 (multi-hop) point of view, the system cannot *attempt to* protect vertical flows from congestion under the following conditions (with reference to Fig. 2.22)):

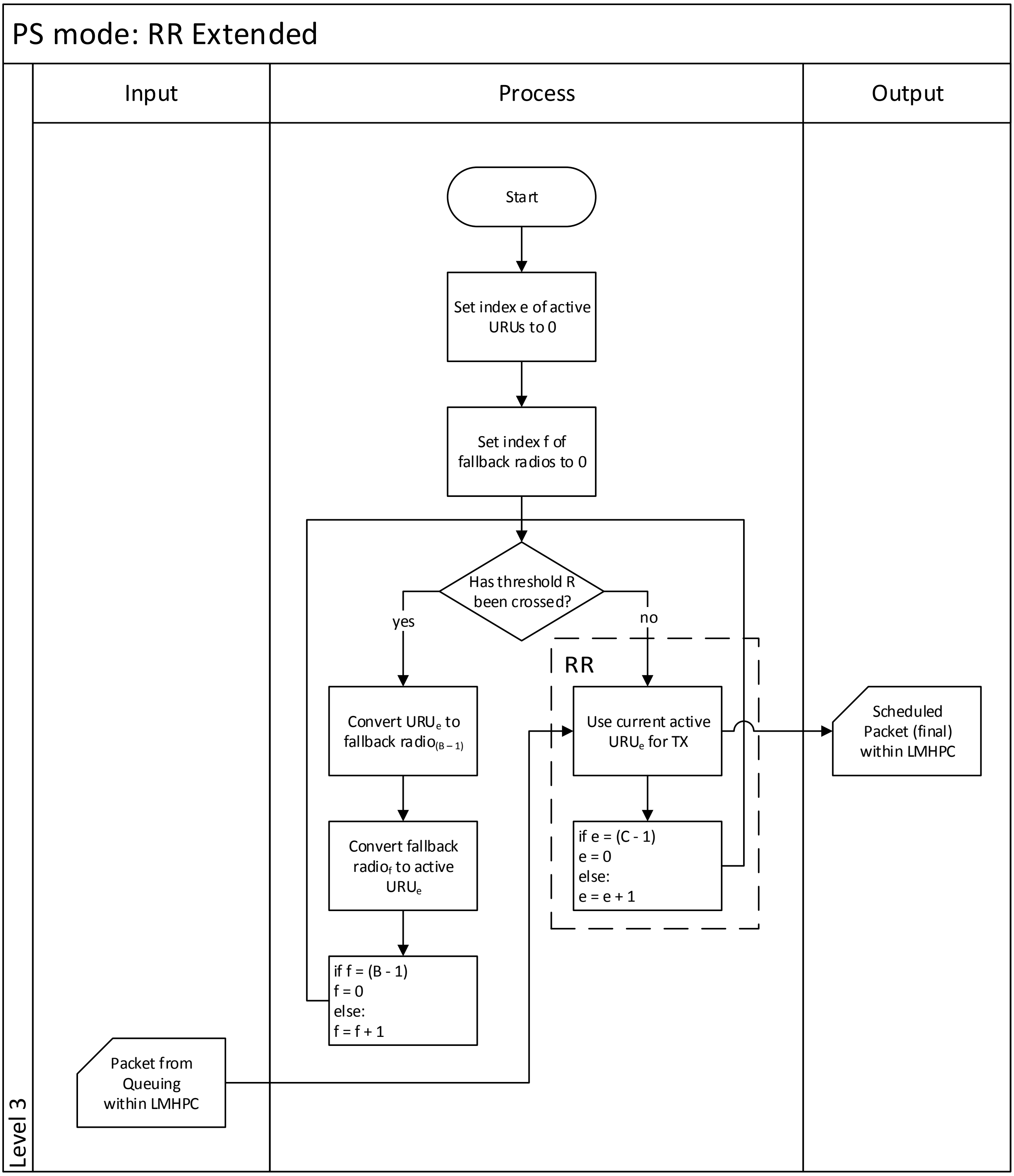


Figure 2.21: Round Robin scheduling with fallback extension process

1. A vertical flow passes through a hop, while in parallel a horizontal flow is using the same hop on a shared channel. The shared hop uses the same RX radio for both incoming flows (scenario a.)

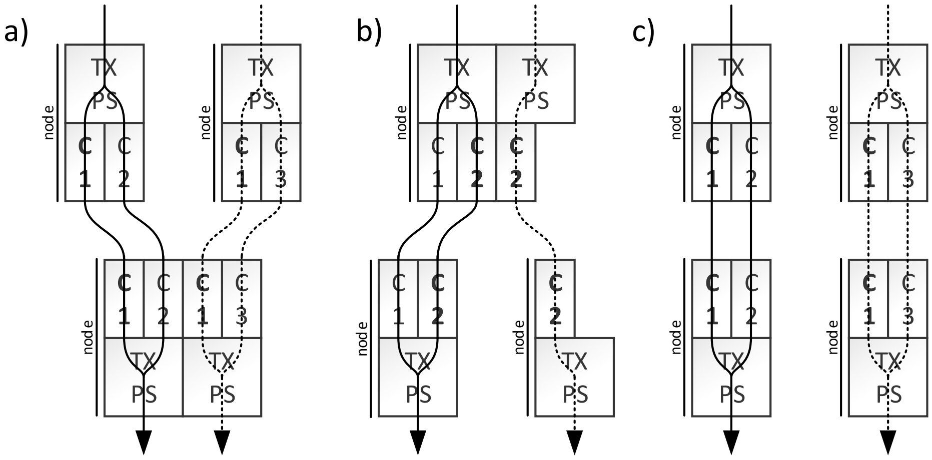


Figure 2.22: Conditional protection of vertical flows

2. A node processes a vertical and a horizontal flow in two bundles. If a shared channel is used (same radio / one URU per bundle), the vertical flow may be subject to interference (scenario b.)

3. A vertical flow is unprotected on separated routes with shared channels in the 1-hop neighborhood (inter-route interference, scenario c.)

Cases 1. and 2. can be prevented by an intra-node priorization / control of bundles, which is not foreseen. Also, deploying only exclusive channels between neighbors / per bundle resolves the issues in cases 1. and 2., which needs to be considered when setting up the WMN. The layer 2.5 module adapts to the sole of exclusive channels. Within a bundle itself, horizontal and vertical flows are separated by priority queues. Queues cannot support protection for the vertical flow in scenarios a. and c. in the upper left node in Fig. 2.22. In scenario b., the following, alternative modification would enable protection: If there would be only one queue set *per node*, the vertical flow can be prioritized over the horizontal flow in the same set. If all channels in the 1-hop neighborhood are exclusive, the current approach (one queue set per bundle) is preferable.

2.9 SOFTWARE IMPLEMENTATION

Customizable, open source firmware for routers, embedded systems or commodity PC hardware marks the starting point for development. OpenWRT or DD-WRT are suitable, economic and non-proprietary Linux distributions [121]. Chen et al. [122] compare both. Especially OpenWRT supports multiple WiFi-device sections and further delivers OLSR[5] via a native package.

To establish the designed system in the public mesh community, it is needed to:

- Port the developed C++ code to a linux networking environment as a kernel module for OpenWRT
- Packages with a similar purpose to manage and aggregate radio resources, such as "Bonding Driver" and "Ifenslave" are already selectable on OpenWRT buildroot and may serve as an inspiration. The SwanMesh node [85] also describes a dual-radio architecture in this environment
- Implement more advanced MIMC metrics, such as MIC [15]

2.10 HARDWARE-PLATFORM RECOMMENDATIONS

While Section 2.9 treats suitable firmwares, this Section now briefly outlines hardware platforms for a real-life implementation. The advantage of WMNs is that they often target commodity- [123], or non-specialized vendor hardware. Several projects in the humanitarian technology sector [23] such as Serval Mesh [124], aim to provide cost-saving solutions. Serval incorporates Android-based User Equipment (UE) in the mesh, without a separate backbone. Since WLAN range is low (approx. 100-170m outdoors and 10-30m indoors [124]), Serval offers the Linux-/multi-WAT-ready Mesh Extender platform. The hardware boasts an additional UHF packet radio device, for 10-100 times longer uni-directional links. A remarkable feature is that Mesh Extenders provide the Serval routing software for new UEs via an HTTP download, which facilitates connectivity. The drawback of including UEs as MRs is that routing software eventually requires a specific mobile OS, or even root access to enable WiFi Ad-Hoc mode (which also has higher energy requirements, compared with "managed" WiFi [124]). Thus, including the proposed system only on backbone-level MIMC hardware seems more elegant. A separate IF for local access is possible, as with a SwanMesh Node [85]. Despite from commercial architectures such as Aruba [125],

---

[5] OLSR mesh for OpenWRT, `http://wiki.openwrt.org/doc/howto/mesh.olsr`

alternative platforms are available [126], [127]. The deployed MikroTik router board (OpenWRT compatible) in [89] holds four 802.11a/g mini-PCI adapters. Raspberry Pi and Raspbian OS gained more popularity for economic WMNs [128]. The Linksys WRT54Gx router series is an outdated classic [129], but its rugged design, cheap price and openness to custom firmwares still make it interesting for low-cost networks. There exists a trade-off between uni-and omni-directional antennas [90]. The first type may decrease interference in the backbone but limit edges in the topology, while the latter offers better self-healing chances.

2.11 CONCLUSION

In Chapter 2, a system concept is proposed, whose features are innovative in most cases to a standard MIMC WMN with special conditions.

The system has the following *main characteristics*:

- Mechanisms to enhance transmission performance in WMNs have been assembled into a novel system architecture for MIMC nodes. This design is highly modular. Several distinctive, novel components are therefore arranged within a middle-layer (2.5) module, which extends a (previously analyzed) standard mesh node architecture. From the latter, cross-layer input from layer 2 and 3 is included.
- Novel components for TA, TEL, evaluation of a proposed CA scheme, bundle definition, packet commutation, priority queuing and LB were created.
- Bundling of radios with three independent PS modes, to target *capacity-* and stability-oriented mesh setups. This *solves* the pervasive issue of performance degradation due to single-radio hop-to-hop interference in transmissions.
- The protection of vertical traffic is twofold:
  - A labeling chain enables to optimize bi-directional vertical traffic under certain conditions. Two labels, combined in the same system, allow to prioritize vertical over horizontal flows in terms of forwarding / queuing and to consider Diffserv. Custom labels *solve* the problem that vertical

  and horizontal traffic is equally treated. Labeling will work with SISC nodes as well.
  - The most effective support of vertical / GW flows is achieved when priority queues are combined with WFS PS mode. This *optimal constellation* causes that GW packets experience the highest forwarding priority and at the same time will be forwarded via the best performing radios.

- The system is compatible with a predefined input of a CA protocol; a regarding protocol was proposed as well.

Chapter 3:

Evaluation of the Proposed System

3

CHAPTER 3: EVALUATION OF THE PROPOSED SYSTEM

3.1 INTRODUCTION

The final Chapter of this thesis evaluates the main features of the presented system in Chapter 2. Its impact on multi-hop capacity and on vertical traffic in a WMN is investigated. Horizontal traffic is considered more as disturbing cross- or parallel traffic. The purpose of the presented series of simulations is to provide a proof-of-concept for each feature selected for evaluation, by showing its positive effects on MIMC mesh networks, or representative parts of a WMN. The validation strategy of features is as follows. Single aspects of the system are systematically analyzed in parts *one after another*, using context-sensitive- setups and performance indicators. The structure of each subsection is similar: After an introduction of the scenario, its setup and events during runtime are discussed. Then, results are listed and discussed afterwards.

3.2 OVERVIEW OF IMPLEMENTED COMPONENTS

This Section briefly discusses implemented and evaluated components from the proposed system. It is declared which components will be individually analyzed and for what purpose. The list of components follows the top-down approach from Fig. 2.2. Each paragraph and terms in capital letters refer to a component:

CHANNEL ASSIGNMENT The implementation primarily foresees a static CA. A global channel map is defined manually in a configuration file (.ini), which is initiated one-time at simulation start-up. To test the particular WFS PS mode, channels are especially chosen to stress local wireless domains and to stimulate the exploitation of channel diversity. One special scenario in Section 3.3.5 was designed to expose the system to dynamic channel switches and test the safe period feature from the CA Table Evaluation.

TRAFFIC ANALYSIS AND TRAFFIC ENGINEERING LABELING

All packets are marked at their respective mesh origin where applicable, which can be controlled via the configuration file. This applies for a packet's ITC: The QSF is embedded in the IPv4 Options field [114]. Queue mappings q_{1-7} from table 2.7 are then stored in the QSF. Regarding the necessity of a node to identify GWs in the topology (respectively, identify itself as one), it can access a global IP list of all mesh gateways, which answers to the output of the Topology Parser component.

COMMUTATION Peculiarities of OMNeT++ need to be regarded when Commutation is tested. A separate measurement in Section 3.3.4 concentrates on the temporal impact of avoiding a per-packet processing in the network layer, which is expected when a packet is solely forwarded within the layer 2.5 module.

QUEUING Section 3.3.1 highlights the QSF functionality within the Queue component. The designed setup provokes that a high priority / vertical stream is disadvantaged in terms of hop count to the GW, in comparison to other low-priority / horizontal streams. This is achieved by creating a shared part of a route, which needs to be passed by all streams, and by setting different hop distances between senders and receivers. Although conditions naturally imply a better performance for low-priority streams here (see Chapter 1, 1.2.4.2), the focus stream may overcome these limitations with the proposed system and a queue weighting scheme in its favor. It shall be proven, that queues can support the enforcement of the chosen ITC and thus support QoS.

MULTI-INTERFACE BUNDLE MANAGEMENT It shall be confirmed that Bundle Management is able to correctly manage capacities with all 1-hops. A WMN with a variable amount of GWs is the base for Section 3.3.2. It concentrates on the magnitude and influence of attached radio quantities on the capacity of parallel vertical streams. A quantitative rise of capacity is expected.

TX SCHEDULING Section 3.3.3 evaluates the WFS and Extended RR PS modes on a multi-hop chain. The WFS mode is applied to conquer inter-route interference (see Chapter 1, 1.2.4.3) for any stream. The mode is expected to schedule more packets on unoccupied channels. Therefore, local channel disruption is added. The Extended RR mode is different, as load shifting / balancing is based solely on the MAC Loss Rate of a single radio, without taking others in the bundle into account. In Section 3.3.3 this mode is applied to lower the influence of intra-flow interference.

3.3 SIMULATIONS

In the simulation environment a broad range of parameters can be modified, combined and used in conjunction, which creates a rich platform for investigations. It is intended to keep the amount of *fixed* parameters as long as possible, so the focus lies on a few variables. Measurements are focused on the lack of capacity on multi-hop routes and the disproportionate treatment of vertical traffic, and how this is enhanced.

Standard parameters, like OLSR-specific settings or the distance between nodes, have been set in a way that the basic structure of a scenario offers homogeneous conditions. This enables that the reader can focus on the actual impact of system features and is not distracted by the complexity caused by additional variations in standard mesh parameters. Random parameters for node positions, mobility, traffic start times, traffic models or traffic burst variations are avoided as well. This allows to focus on the impact of parameters which are actually critical for system evaluation. This mainly refers to the amount of radios per node, used channels, packet flows and hop distance.

Appendix A.3.1 lists details of the used OMNeT++ environment and general measurement conditions. Figure 3.1 depicts the custom node which has been created. The conceptual blocks from the system architecture in Fig. 2.2 are accommodated in the modules `TeLabel` / `mldp`, `scheduler`, and `myqueue`. Two custom tables can be live-monitored. The parameter `numMIRadios` in the `wlan` module in Fig. 3.1 defines the amount of 802.11g interfaces.

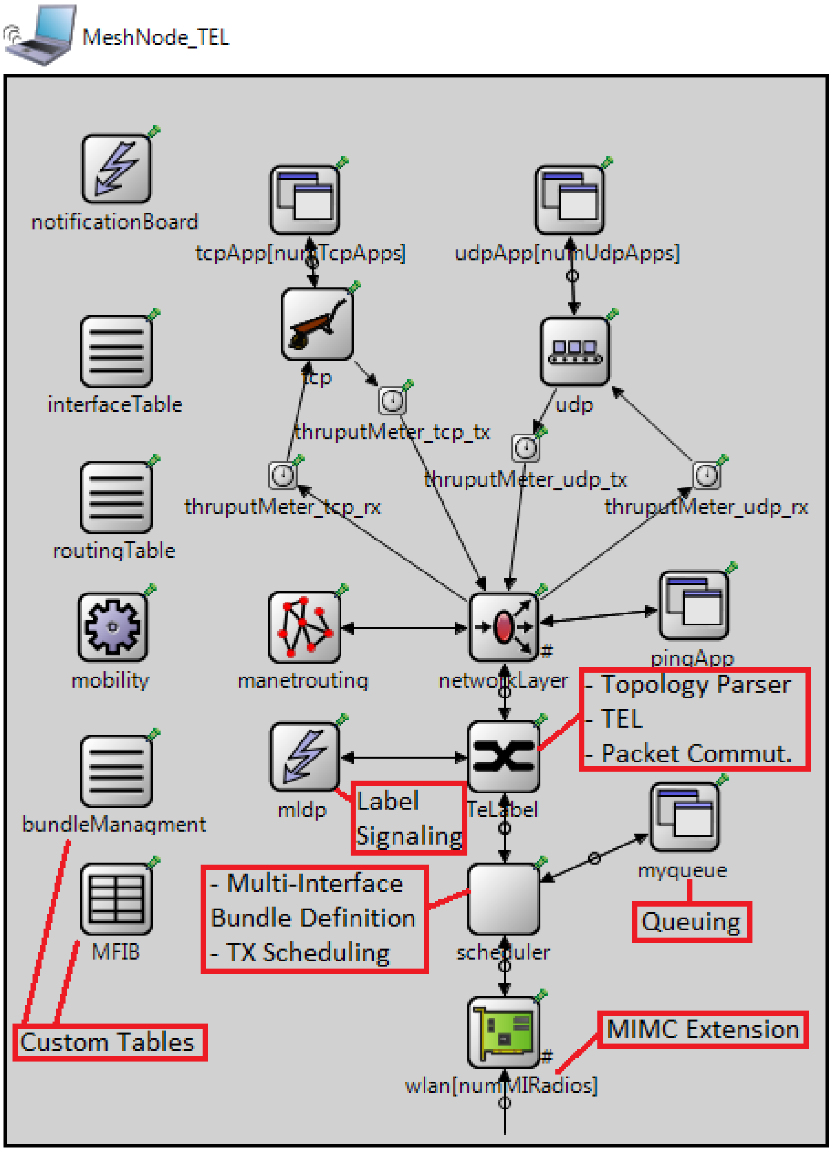
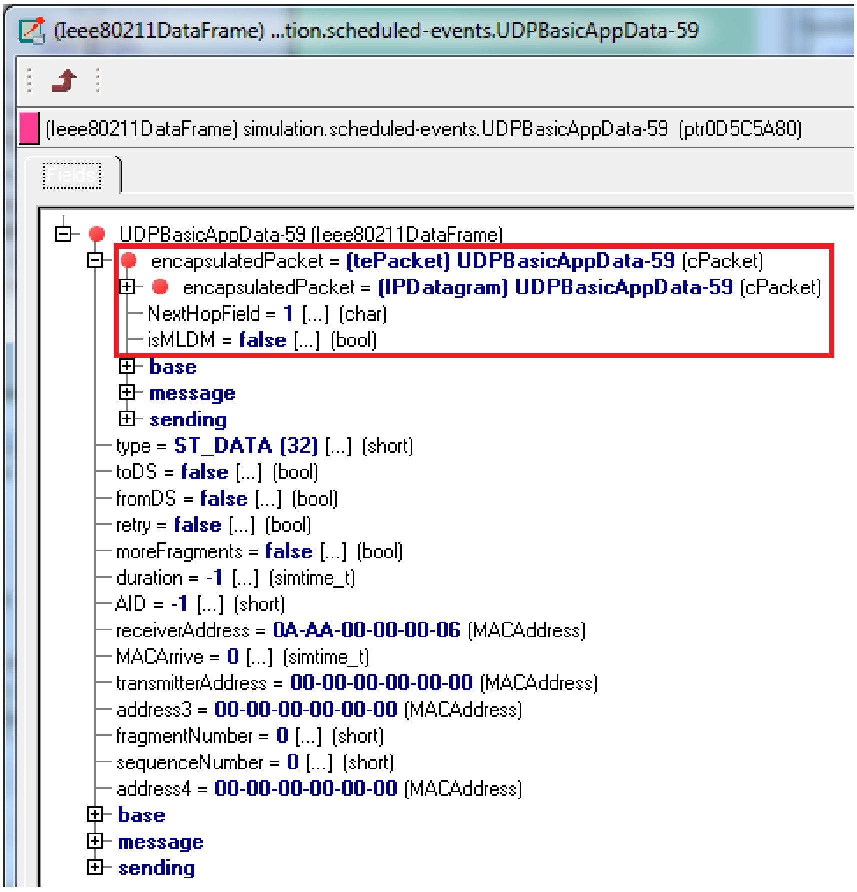


Figure 3.1: Implementation of the custom node

The square in Fig. 3.2 shows an encapsulated packet of a vertical flow.

Table 3.1 lists the author's *selection* of measurable performance indicators.

Before system components from Chapter 2 are evaluated, several measurements of standard SISC nodes have been simulated. Appendix A.5.1 treats a single 802.11



```
(Ieee80211DataFrame) simulation.scheduled-events.UDPBasicAppData-59  (ptr0D5C5A80)

UDPBasicAppData-59 (Ieee80211DataFrame)
    encapsulatedPacket = (tePacket) UDPBasicAppData-59 (cPacket)
        encapsulatedPacket = (IPDatagram) UDPBasicAppData-59 (cPacket)
        NextHopField = 1 [...] (char)
        isMLDM = false [...] (bool)
        base
        message
        sending
    type = ST_DATA (32) [...] (short)
    toDS = false [...] (bool)
    fromDS = false [...] (bool)
    retry = false [...] (bool)
    moreFragments = false [...] (bool)
    duration = -1 [...] (simtime_t)
    AID = -1 [...] (short)
    receiverAddress = 0A-AA-00-00-00-06 (MACAddress)
    MACArrive = 0 [...] (simtime_t)
    transmitterAddress = 00-00-00-00-00-00 (MACAddress)
    address3 = 00-00-00-00-00-00 (MACAddress)
    fragmentNumber = 0 [...] (short)
    sequenceNumber = 0 [...] (short)
    address4 = 00-00-00-00-00-00 (MACAddress)
    base
    message
    sending
```

Figure 3.2: Encapsulation with TEL

Table 3.1: Selection of performance indicators

TYPE	MODULE	PARAMETER
Vector	Radio	MAC loss rate SNIR in dB
	TCP	RX/TX Throughput including Overhead RTT (smoothed)
	UDP	RX/TX Throughput including Overhead
	Ping app	RTT
	Cross-layer	Number of sent frames per radio
Scalar	MAC	Number of sent frames Number of sent frames w/o retries Number of received frames Number of frame TX retries
	UDP (app)	Mean end-to-end delay
	Ping app	Various delay statistics

WirelessLAN link in INETMANET. There, the frame loss rate tapped between PHY and DLL is measured as the representative performance indicator of the PHY layer (and all underlying methods (e. g., modulation scheme). This MAC loss rate also serves as the implemented fallback parameter (see Chapter 2, 2.8.4.4). Measurements in appendix A.5.1 further serve as *references* for throughput and delay of a WLAN link. Appendix A.5.2 lists standard MAC parameters. Appendix A.5.3 then deals with issues of common SISC multi-hop chains: Apart from the pervasive performance degradation, the influence of multiple cross-traffic streams adds up here, which is also a common phenomenon in WMNs. Tables A.9, A.10, A.11, A.12, A.13 and A.14 in appendices A.5.1, A.5.2 and A.5.3 also explain standard parameters, which are valid in the following simulations.

3.3.1 *Quality-of-Service and Priority Queueing*

The purpose of queues here is to be able to alter the sending order of multiple incoming flows, to protect vertical traffic. The concept is laid down in Chapter 2, 2.8.3.

So here, the impact of per-hop queues is investigated, by applying different weighting schemes. The used network setup is shown in Figure 3.3. The links between

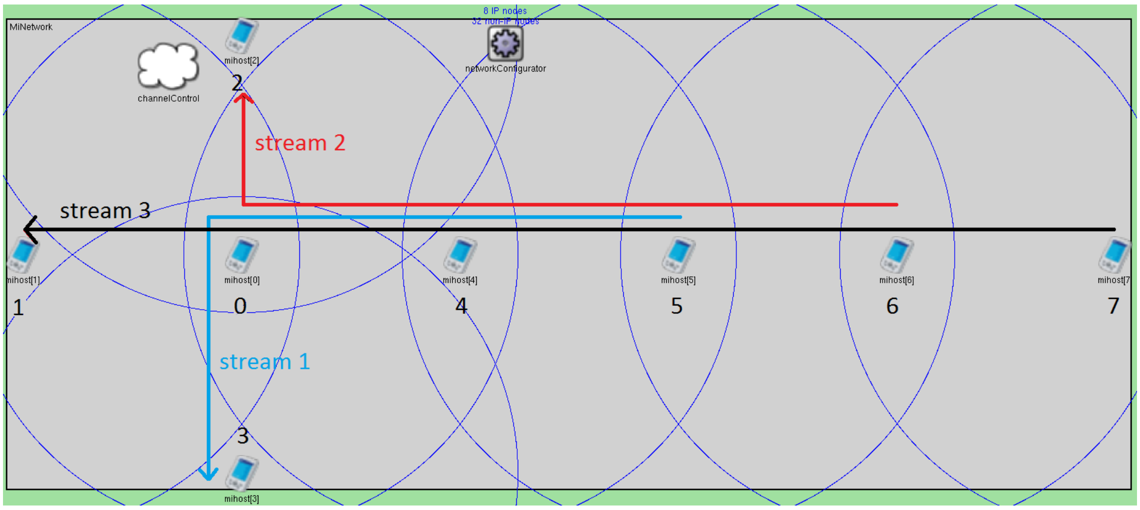


Figure 3.3: Scenario for testing middle-layer packet queues

`mihost[0]` and `mihost[5]` are simultaneously used by three UDP streams:

HORIZONTAL STREAM 1: From `mihost[5]` to `mihost[3]` (blue arrow)

VERTICAL STREAM 2: From `mihost[6]` to the *GW* at `mihost[2]` (red arrow)

HORIZONTAL STREAM 3: From `mihost[7]` to `mihost[1]` (black arrow)

First, the selected scenario allows to show that the horizontal stream 1 will presumably offer a performance superior to stream 2 and 3, due to its smallest hop distance. Secondly, when applying a weighting scheme which, clearly favors a vertical upload to a GW with stream 2, the effect should be noticeable: in every forwarding hop, sending order of stream 2 packets are beneficially changed, in order to improve the QoS characteristics of stream 2.

3.3.1.1 *Configuration*

Table A.1 in appendix A.4.1 summarizes the most relevant parameters for queue testing.

The UDP "Basic Burst" app [130] is used here. It sends single bursts with a controllable burst length, with fixed gaps between bursts. Thus, a precise Constant Bit Rate (CBR) can be configured. Its counterpart is the UDP "Sink" app [131]. Parameters of both apps are listed in table A.3.

VARIABLES AND EVENT TIMELINE

`mihost[5],[6],[7]` start UDP CBR streams. Sinks are placed in `mihost[1],[2],[3]`. Stream 2 shall be protected. Therefore, two different weight distributions are tested: Table 3.2 benefits stream 2 by putting its packets in the highest priority queue, which is weighted with 0.7. The packet class is injected at the origin (see column *Node*). Packets of stream 1 and 3 fall in lower priority queues, each with a weight of 0.1. Each queue can store 5 or 8 packets. `weightqueue7` is always required to have an assigned weight (0.1), otherwise broadcast packets will not be processed. The second scheme in table 3.3 is used as a comparison. Here, all three streams have the same priority when forwarded. All streams start at t=30s. Stream 1 and 3 have a TX rate of

Table 3.2: Queue weight distribution beneficial for stream 2

MODULE	NODE	PARAMETER	TP.	VALUE
(custom)		`queueModule`	S.	myQoSQueue
		`frameCapacity`	C.	*Variable*
		`maxBufferSize`	C.	*Variable*
		`weightqueue1`	C.	0.7
		`weightqueue2`	C.	0.0
		`weightqueue3`	C.	0.0
		`weightqueue4`	C.	0.0
		`weightqueue5`	C.	0.1
		`weightqueue6`	C.	0.1
		`weightqueue7`	C.	0.1
	6	`setunicastprio`	C.	0
	5	`setunicastprio`	C.	5
	7	`setunicastprio`	C.	4

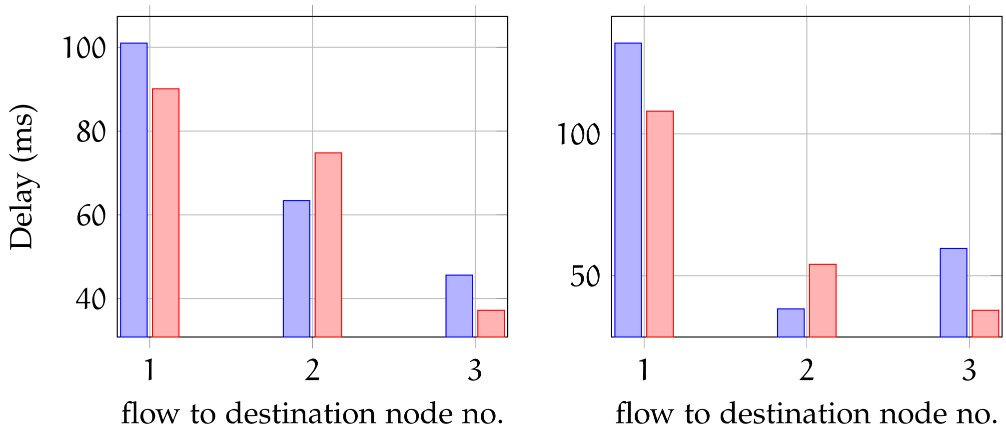
Table 3.3: Equal queue weight distribution

MODULE	NODE	PARAMETER	TP.	VALUE
(custom)		`weightqueue1`	C.	0.3
		`weightqueue2`	C.	0.0
		`weightqueue3`	C.	0.0
		`weightqueue4`	C.	0.0
		`weightqueue5`	C.	0.3
		`weightqueue6`	C.	0.3
		`weightqueue7`	C.	0.1
	6	`setunicastprio`	C.	0
	5	`setunicastprio`	C.	5
	7	`setunicastprio`	C.	4

1 Mbit/s, whereas stream 2 sends at 2 Mbit/s. Rates have been chosen at low levels, to not provoke unnecessary packet losses due to hop distances, which might lead to erratic results. Scenario has been tested with using either just one or both radios; the latter by applying RR PS mode. Additionally, the effect of a changing datagram size is tested.

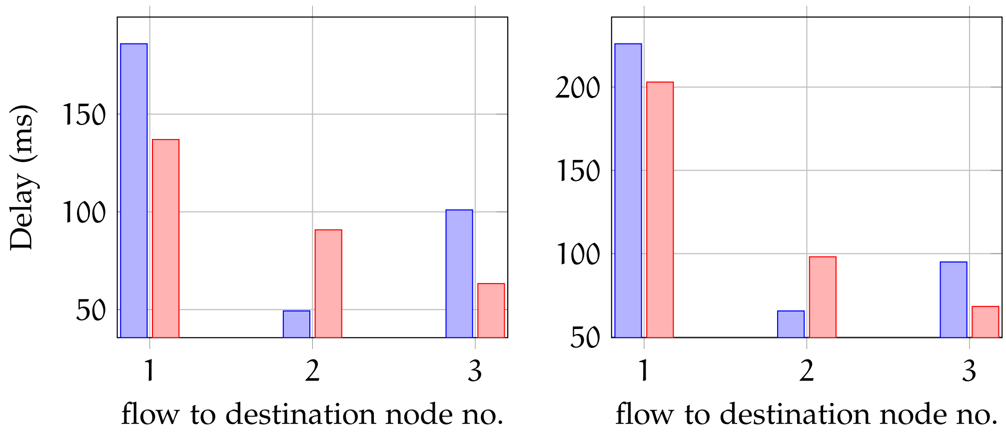
3.3.1.2 *Results*

Figures in this Section compare end-to-end delays from UDP basic burst to sink app (from table 3.1). Figures 3.4 (C=5) and 3.5 (C=8) treat the case with 2 active radios using RR PS mode, whereas Fig. 3.6 (C=8) treats the single-radio case.



(a) UDP thrpt., dt.gr. size 250B

(b) UDP thrpt., dt.gr. size 500B



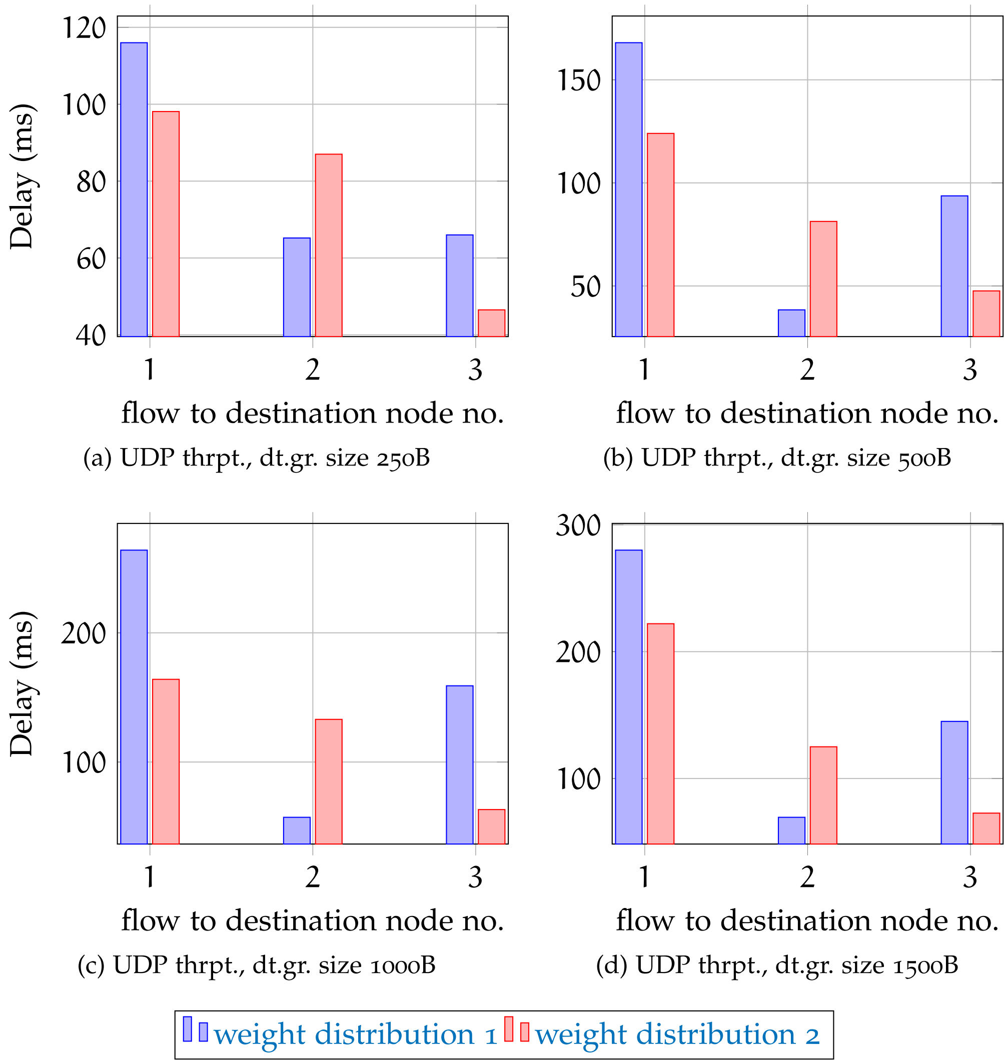
(c) UDP thrpt., dt.gr. size 1000B

(d) UDP thrpt., dt.gr. size 1500B

▯ weight distribution 1 ▯ weight distribution 2

Figure 3.4: Multi-hop UDP performance with two different queue weight distributions, using RR PS mode, $C = 5$

Stream 1 to `mihost[3]` always offers the shortest delays with the balanced weight scheme from table 3.3, especially when more network capacity is granted by using two radios. This is due to the closer hop distance between `mihost[5]` and `mihost[3]`, as this is a deciding performance factor in standard WMNs [78]. Long end-to-end



(a) UDP thrpt., dt.gr. size 250B
(b) UDP thrpt., dt.gr. size 500B
(c) UDP thrpt., dt.gr. size 1000B
(d) UDP thrpt., dt.gr. size 1500B

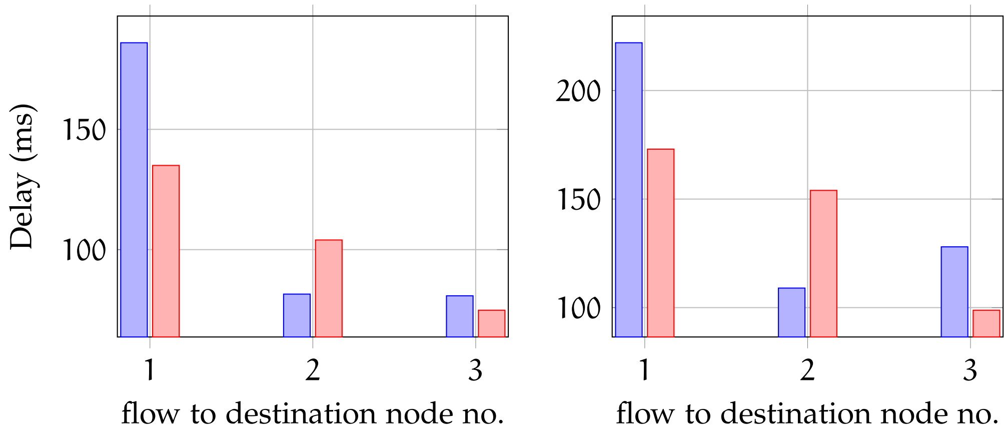
Figure 3.5: Multi-hop UDP performance with two different queue weight distributions, using RR PS mode, $C = 8$

delays can be harmful for media transmission quality [9] and thus are QoS-critical parameters.

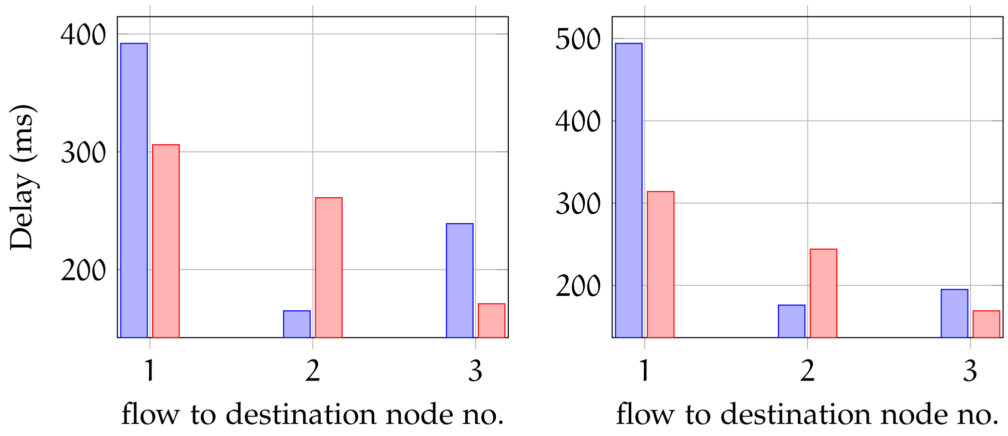
An experimental setup with a disproportionate queue size of 20 packets has led to volatile delay levels and increased congestion in the network and is not further regarded. C=8 has shown to be ideal for this scenario.

3.3.1.3 *Evaluation*

The Queue component affects delay levels *in all results*, in favor of the vertical stream 2. With the beneficial weight distribution scheme 1 in table 3.2, delay levels of stream 2 are drastically improved, in comparison to scheme 2. In Figures 3.4, 3.5 and 3.6



(a) UDP thrpt., dt.gr. size 250B (b) UDP thrpt., dt.gr. size 500B



(c) UDP thrpt., dt.gr. size 1000B (d) UDP thrpt., dt.gr. size 1500B

🟦 weight distribution 1 🟥 weight distribution 2

Figure 3.6: Multi-hop UDP performance with two different queue weight distributions, using a single radio, $C = 8$

stream 2 delay values are shortened; partially by up to 80 milliseconds. This also decreases the overall air time of stream 2 packets. In parallel, delays of stream 1 and 3 are worsened by the unequal weight distribution. Assuming that stream 2 represents a bi-directional GW flow, the queue system is proven to be an effective way to reduce delay. Delay-sensitive applications like VoIP will experience a better quality when transferred over stream 2, due to lower delays [87]. Another advantage observed for stream 2 is that the queue system potentially overcomes the unfairness issue between streams 1 and 2 in terms of delay, which is caused by intra-route interference (described in Chapter 1, 1.2.4.2). Thus, an active prioritization of the (GW) stream 2 can be realized.

The datagram size apparently increases the delay here, whereas channel diversity generally decreases it. With MIMC nodes, delay values below 200ms are possible for smaller sizes 250B and 500B, which is not given with SISC. Nevertheless, the queue system brings even when only a single radio is used.

Delay levels of stream 2 are notably lower than of those of stream 1 with scheme 1, except in Fig. 3.4a. This is improved with a higher C and suggests that C must be customized for each individual setup.

3.3.2 *Vertical Traffic in a Mesh Network*

In this setup, a broad mesh network is investigated. The purpose is to evaluate mesh capacity in a MIMC environment. The simple RR PS mode, which is explained in Chapter 2, Section 2.8.4.3, is used as a starting point here. RR enables an even, non-biased distribution of network capacity. As outlined in Chapter 2, 2.8.1 bundle definition capabilities in each node are required as well. 37 nodes have to cope with only a single, or up to 7 active GW flows. A representative set of flows has been defined. These end-to-end connections have varying hop distances. Also, the GW saturation / amount of GWs is increased in three stages. Figure 3.7 depicts the grid. Table 3.4 provides an overview of all three traffic constellations. All traffic is vertical here. Clients in Fig. 3.7 (marked blue) initiate downloads from GWs (marked red). In a real OLSR WMN, it is untypical to specify the corresponding GWs. But since the hop

Table 3.4: Traffic constellations for the mesh grid scenario

CONFIG.	CLIENT	GATEWAY
1	`mihost[14],[9],[26],[0],[5],[35],[30]`	`mihost[36]`
2	`mihost[9],[5],[35]`	`mihost[16]`
	`mihost[14],[26],[0],[30]`	`mihost[19]`
3	`mihost[14],[0]`	`mihost[7]`
	`mihost[9],[5]`	`mihost[10]`
	`mihost[26],[30]`	`mihost[25]`
	`mihost[35]`	`mihost[28]`

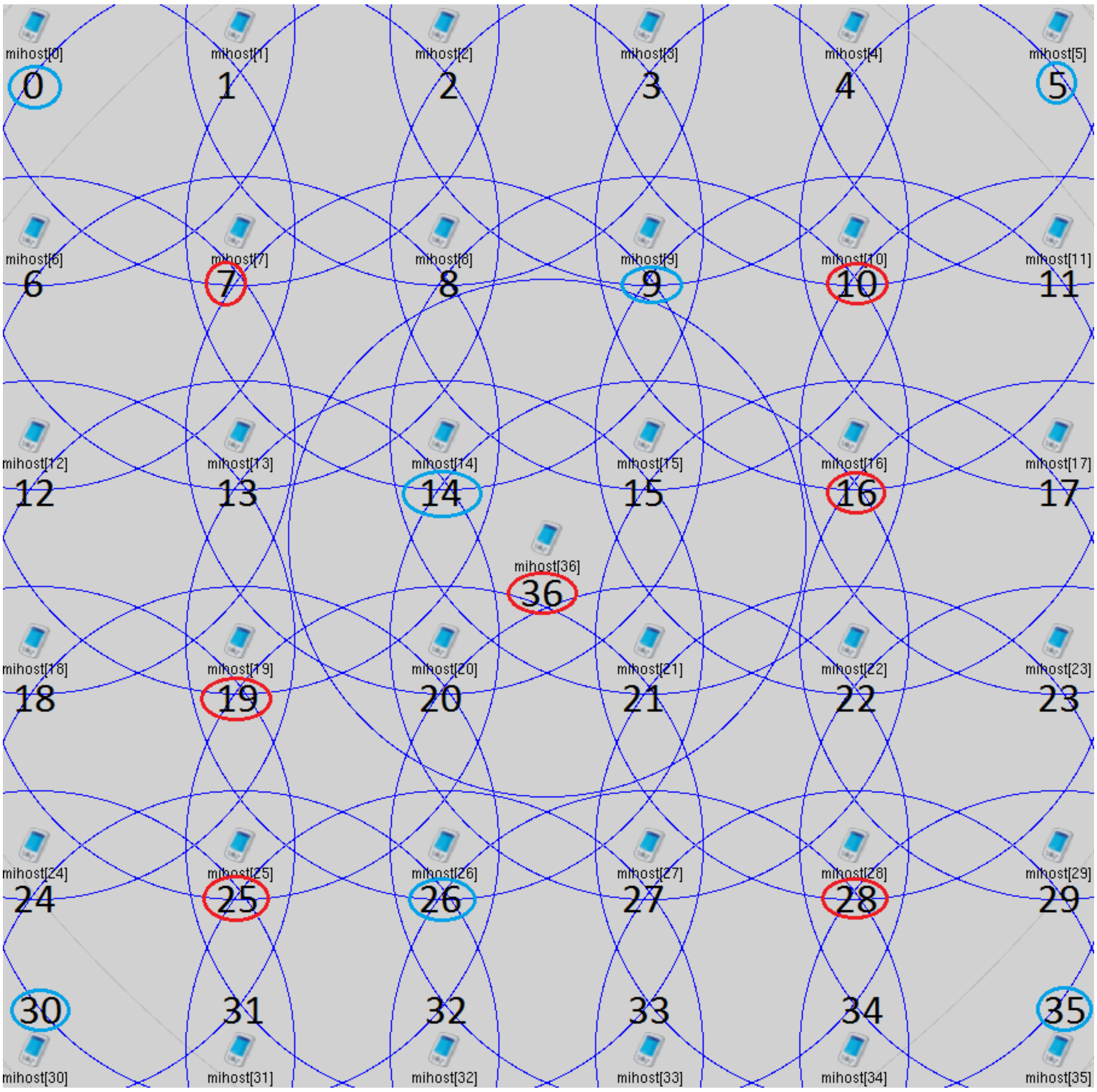


Figure 3.7: Scenario for testing a mesh grid

count metric is used in this *particular* setup, the estimated route calculation to the best GW can be easily done by choosing the shortest distance, which represents the cheapest path. Looking at the different positions of clients and GWs, it becomes clear that shorter client–GW routes will eventually also bear traffic to farther away destinations. As an example, the connection `mihost[36]` to `mihost[14]` will presumably carry traffic to `mihost[0]`, when both streams are active. The shared link becomes the bottleneck for `mihost[0]`. This shortage of resources close to a GW maps a common (problem) situation in a mesh network.

The simulation shall confirm typical behaviors of WMNs. First, as in previous measurements, the hop distance will play an important role in the degradation of performance. Secondly, the increasing amount of radios per node will raise the overall capacity, which is enabled by the system. The standard SISC case is confronted with the case where nodes deploy up to 6 individual WLAN radios. Thirdly, adding more gateways will avoid longer routes and thus improve performance of streams to destinations on the edges of the grid.

3.3.2.1 *Configuration*

Table A.4 in appendix A.4.2 highlights the most important settings for this simulation. The amount of clients is fixed to 7. They are served by 1,2, respectively 4 gateways. Since an entire mesh is simulated, the used traffic model shall be based on a commonly used application: File transfer via File Transfer Protocol (FTP)/TCP. FTP sessions usually maintain longer data sessions (unlike HTTP request/reply pairs where small chunks of data are exchanged), which allows for more comprehensible results. Used parameters are listed in table A.5 in appendix A.4.2. Gateway nodes run the TCP "Generic Server" app [132] (same table). It can manage any number of incoming TCP requests. Reply lengths (download) depend on the desired length in the request (`replyLength`). Requests are generated at the clients, which run the TCP "Basic Client" app [133]. The app serves to simulate a rough model of a FTP client. It opens consecutive sessions to the server, with multiple requests within the same session. One request / reply pair is processed after the other. `thinkTime` specifies the gap between requests and `idleInterval` the gap between sessions. The `reconnectInterval` between sessions is set short. Parametrization for session/request values is based on the recommended settings for FTP [133].

VARIABLES AND EVENT TIMELINE

All FTP sessions are initiated during the same time window, between 30s and 31s. In the course of simulation runs, always one more active stream is added to the previous one(s), in the following order: `mihost[14], [9], [26], [0], [5], [35], [30]`.

3.3.2.2 *Results*

Presented results concentrate on the development of network capacity, in terms of TCP/FTP throughput. Figures 3.8, 3.9 and 3.10 depict the impact of 1, 2 and 4 GWs. The x-axis shows the amount of simultaneously active flows.

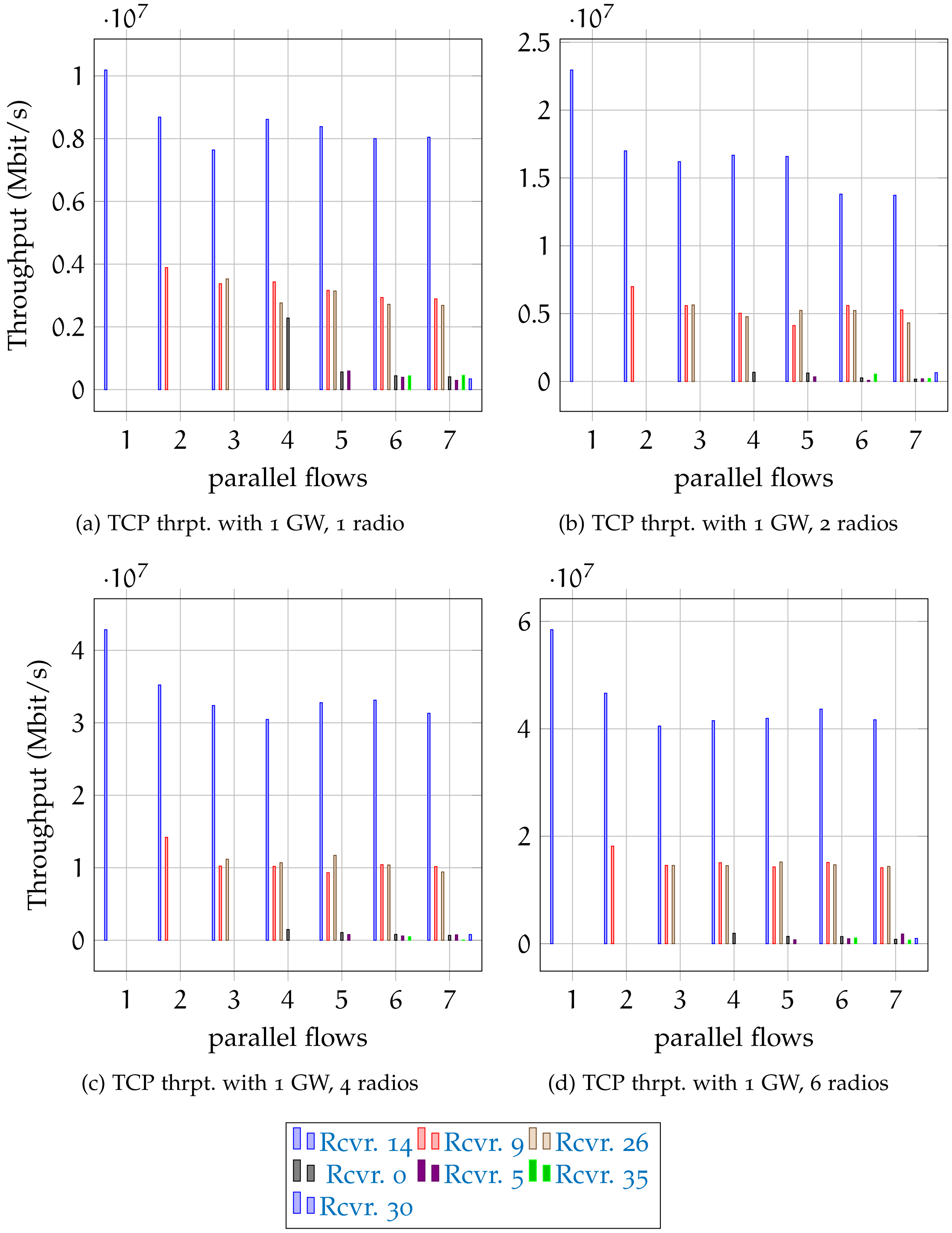
3.3.2.3 *Evaluation*

The single-channel / single-GW case in Fig. 3.8a is discussed first. From the limitations of a standard WMN, *intra-flow* and *intra-route* interference are identified as limiting capacity factors. They occur simultaneously here; the first through longer routes and the latter through shared links in a 1-hop distance to the GW. Receiver `mihost[14]` is able to maintain constantly high bandwidth levels (in comparison to other streams with a hop count higher than 1) and consumes most of the capacity. The typical 10 Mbit/s (only one flow active) is merely reduced to approx. 8 Mbit/s, leaving less capacity for the remaining 2-6 flows. From a topological point of view, flow to `mihost[14]` not only partially congests the route to `mihost[0]` (this problem can be mapped to other streams in Fig. 3.8a), but generally occupies the wireless channel(s) used by the GW at `mihost[36]`. `mihost[36]` becomes a *bottleneck* as the first hop towards `mihost[9], [26], [0], [5], [35], [30]`. In the end, streams to destinations `mihost[0], [5], [35], [30]` suffer the most from drastic performance loss, due to the *longest* hop counts.

When more GWs are available, the network has to bear *less load*, as shown in Figures 3.9 and 3.10. When 4-7 streams are active, fairness in capacity distribution is improved. With 4 GWs, `mihost[9]` and `mihost[26]` notably benefit from a better proximity to a GW.

Still, the negative influence of intra-route interference (e. g., `mihost[0]` RX throughput starves due to the greediness of `mihost[14]`) peaks in the SISC case in Fig. 3.8a and installing multiple GWs is not always an option in real-life setups.

The system's VI and BMT work properly. In conjunction with the RR PS mode, the *system permits more overall capacity*. In the 1 GW scenario, when looking at the flow to receiver `mihost[14]` (as the only active stream, with the shortest hop count) through-



(a) TCP thrpt. with 1 GW, 1 radio

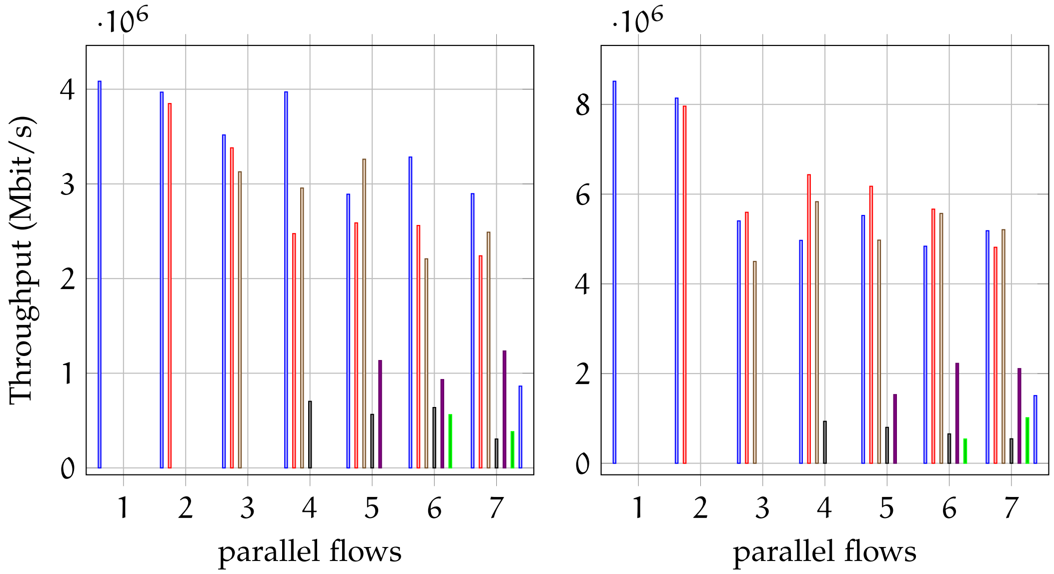
(b) TCP thrpt. with 1 GW, 2 radios

(c) TCP thrpt. with 1 GW, 4 radios

(d) TCP thrpt. with 1 GW, 6 radios

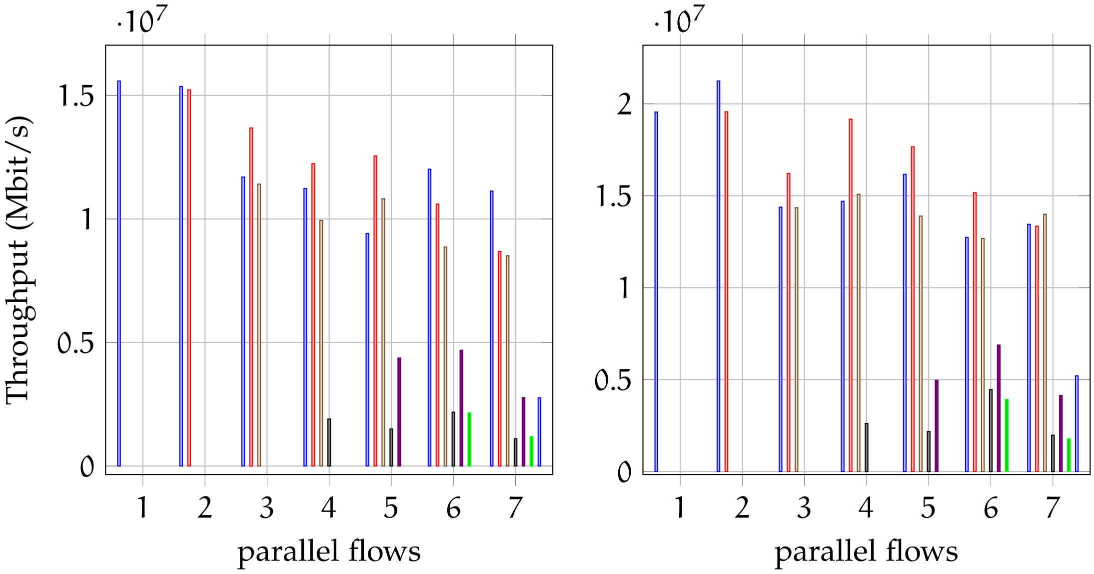
Figure 3.8: Grid performance with varying amount of radios and 1 gateway node

put is increased roughly by a factor of 6, as expected (compare Figures 3.8a and 3.8d). This linear increase applies only to 1 hop. If the hop count is 2 (e. g., flow from `mihost[7]` to `mihost[14]` with 2 GWs), throughput is increased by a factor of approx. 5 (Fig. 3.10).



(a) TCP thrpt. with 2 GW, 1 radio

(b) TCP thrpt. with 2 GW, 2 radios



(c) TCP thrpt. with 2 GW, 4 radios

(d) TCP thrpt. with 2 GW, 6 radios

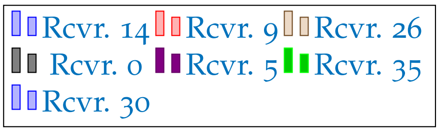
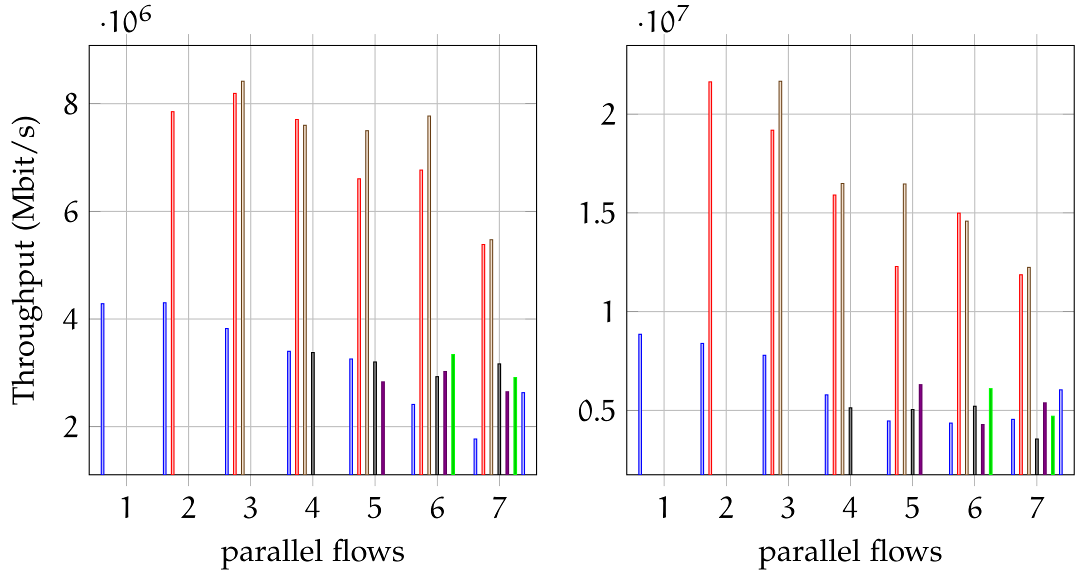


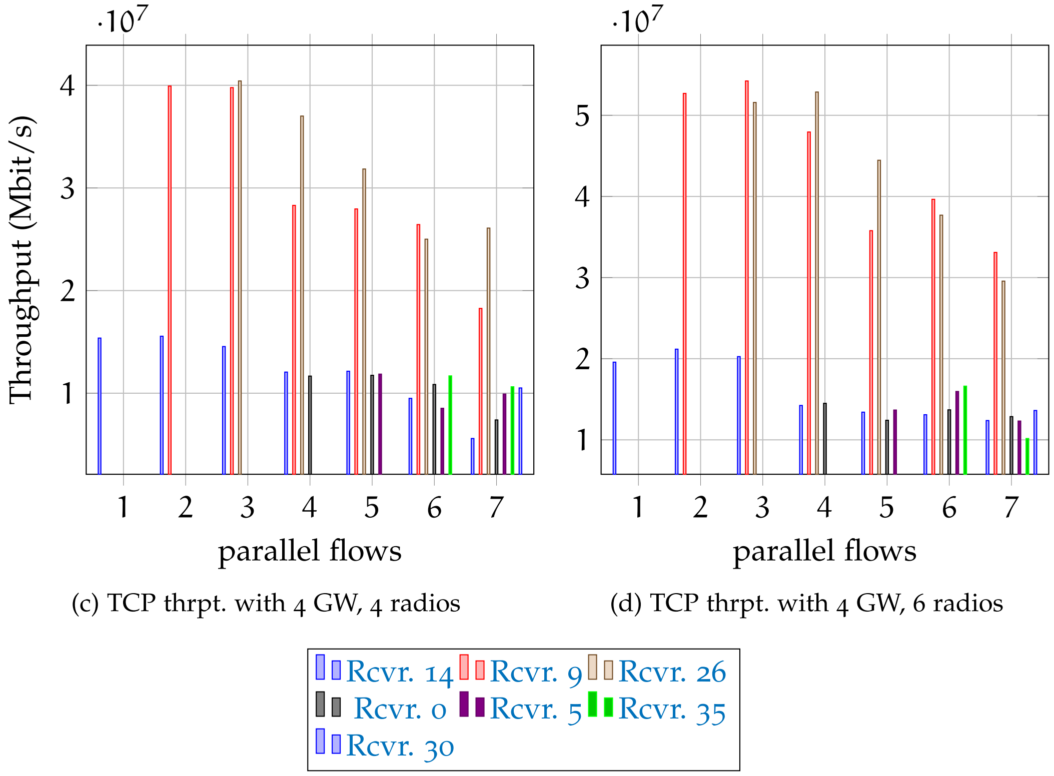
Figure 3.9: Grid performance with varying amount of radios and 2 gateway nodes

A solution to the unfairness problem has been discussed before in subsection 3.3.1.3, where a vertical flow competes with horizontal ones. In the present grid scenario and the RR PS mode, this improvement can only be achieved if different flow priorities *among the GW flows* can be identified on a shared route. To improve the situation,



(a) TCP thrpt. with 4 GW, 1 radio

(b) TCP thrpt. with 4 GW, 2 radios



(c) TCP thrpt. with 4 GW, 4 radios

(d) TCP thrpt. with 4 GW, 6 radios

Figure 3.10: Grid performance with varying amount of radios and 4 gateway nodes

the adaptive WFS favors less congested links specific to each hop on a path. The WFS, as well as the Extended RR PS modes shall be investigated in the following.

It has to be pointed out anew that hop count metric was applied here only for the sake of clear arrangement. In a real network (or in simulations of such), when

OLSR implementation allows to use link-state-sensitive metrics such as ETX or ETT, their application is preferred. In future measurements in the "Vertical Traffic" setup, enabling ETT would have caused the selection of alternative routes in most routing cases, to avoid hops on heavily congested routes.

3.3.3 *Multi-Modal Load Balancing*

This Section now demonstrates the effects of the remaining PS modes: The impact of WFS and extended RR modes is covered, whose concepts were elaborated within the TX Scheduling component in Chapter 2, sections 2.8.4.2 and 2.8.4.4. Two separate scenarios have been designed. For testing the RR with FB option PS mode, intra-flow interference is provoked, which threatens the transmission performance on this route. When compared with a single-radio node, the novel node with multiple fallback radios is less prone to this interference type To include the mesh-typical performance degradation of multi-hop routes, a chain topology is used.
One requirement to evaluate WFS PS mode is that selective channels are congested, so that load can be shifted to less congested ones. The system shall enable that free channels will bear more packets, taking the *current* link states in a bundle into account. WFS requires ETT metric to be enabled. Again, a chain is used, but with neighboring nodes which cause local inter-route interference. This time, the two disturbing streams shall start at different times, to confront the WFS process with time-variant interference.

3.3.3.1 *Configuration*

Table A.6 in appendix A.4.3 lists settings which apply for the testing of both scheduling modes. Table A.7 in the same appendix contains settings specific for the WFS PS mode and table A.8 for the Extended RR PS mode.

WFS PS MODE

The scenario is depicted in Figure 3.11. A chain topology is used. The connection between `mihost[0]-[4]` represents a route to a GW. The extra node pairs `mihost[5]-[6]`

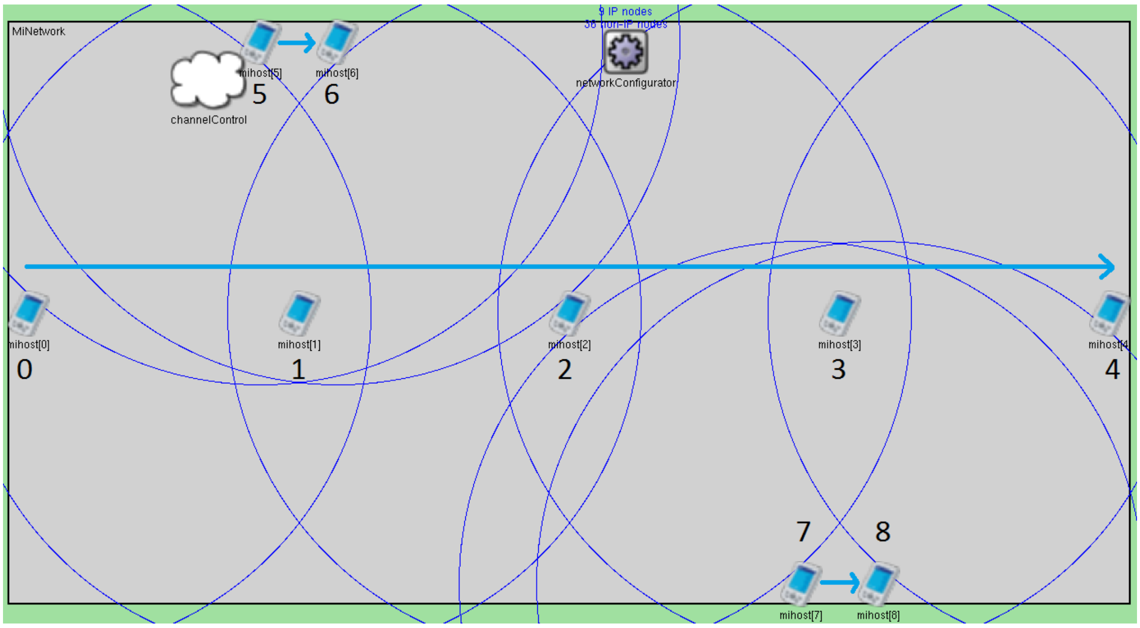


Figure 3.11: Scenario for testing WFS PS mode

and `mihost[7]-[8]` create horizontal background traffic. Each node in the chain has three radios each, so channel 2 (third radio) is not congested. A UDP basic burst / sink app pair is used for all connections (compare exemplary settings in table A.3).

VARIABLES AND EVENT TIMELINE

Streams are set up as follows:

FOCUS / VERTICAL STREAM: From `mihost[0]` to `mihost[4]`, starts at t=30s with a TX rate of 5 Mbit/s. Its performance shall be investigated (blue arrow).

BACKGROUND / HORIZONTAL STREAM 1: From `mihost[5]` to `mihost[6]`, starts at t=40s, affects shared channel 0 of `mihost[1]`.

BACKGROUND / HORIZONTAL STREAM 2: From `mihost[7]` to `mihost[8]`, starts at t=60s, affects shared channel 1 of `mihost[3]`.

Three different UDP datagram sizes (0.5kB, 1kB, 1.5kB) were tested with the focus stream. Background streams use three different TX rates: 0.1 kbit/s (i. e., almost no activity), 3 Mbit/s and 6 Mbit/s.

EXTENDED RR PS MODE

The network topology is depicted in Figure 3.12. As in appendix A.5.3, the TCP

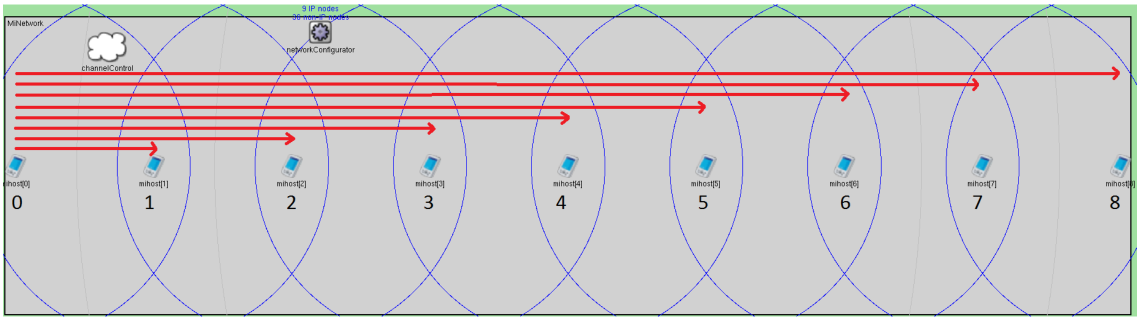


Figure 3.12: Scenario for testing Extended RR PS mode

session / sink app combination is used for the focus stream (compare table A.12).

VARIABLES AND EVENT TIMELINE

`mihost[0]` establishes a single TCP stream to all remaining destinations in the chain (red arrows, max. 8 hops). This can be either a horizontal or vertical stream; the performance shall be enhanced with the novel PS mode. Three FB threshold rates R are considered: A MAC loss rate of 10%, 20% and 30%. A snapshot in Fig. 3.13 provides a typical MAC loss rate level of a sending SISC node, measured over time. The amount of radios also varies, from 1 to 4. When n radios are equipped, the

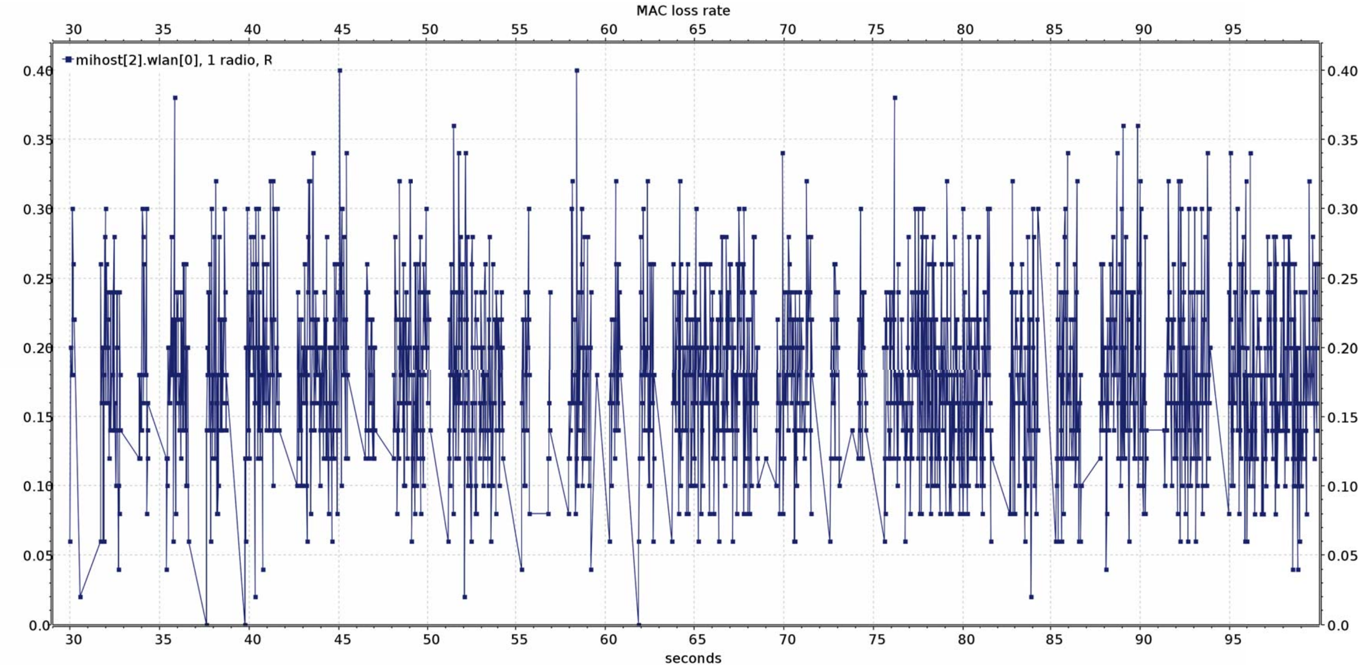


Figure 3.13: MAC loss rate for testing Extended RR mode with 1 attached radio

amount of fallback radios is $B = n - 1$.

3.3.3.2 *Results*

WFS PS MODE

Packet TX statistics are recorded per node and per radio, to confirm that scheduling works properly. Figures 3.14and 3.15 show the packet distribution per node, for a datagram size of 500B and for the background traffic rates 3 Mbit/s and 6 Mbit/s. `mihost[4]` is not considered in the visualization, as it merely serves as a uni-directional UDP sink.

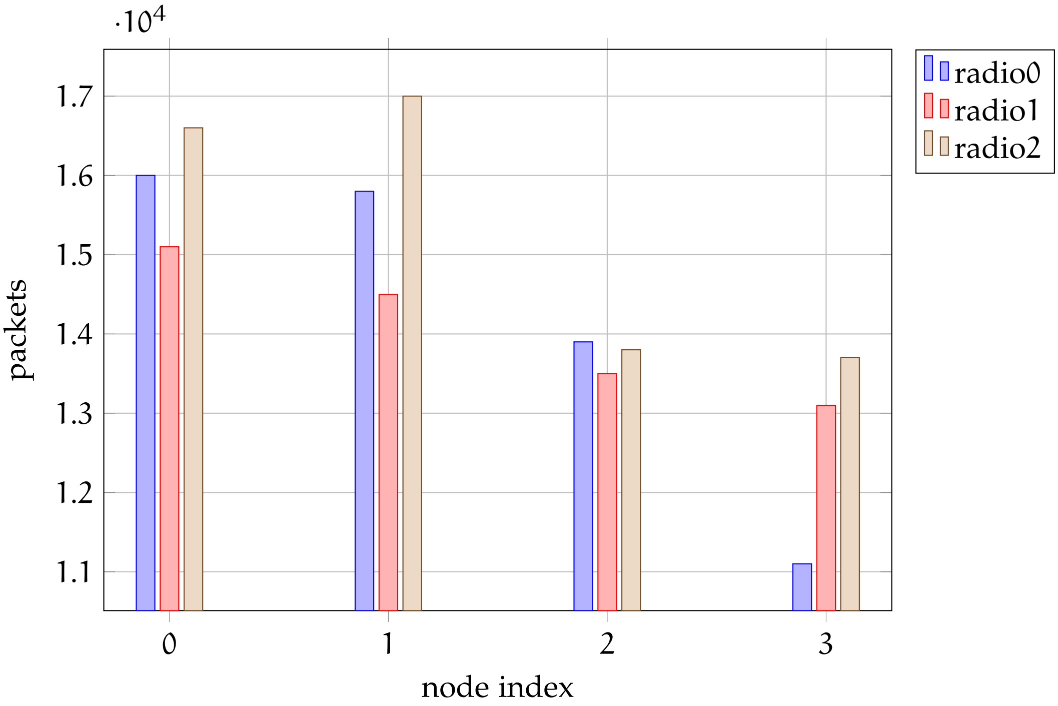


Figure 3.14: Packet distribution for testing WFS mode, datagram size 500B, with background traffic TX rate @ 3 Mbit/s

Figures 3.16 and 3.17 shows the packet distribution for a datagram size of 1000B. Figures 3.18 and 3.19 shows the packet distribution for a datagram size of 1500B.

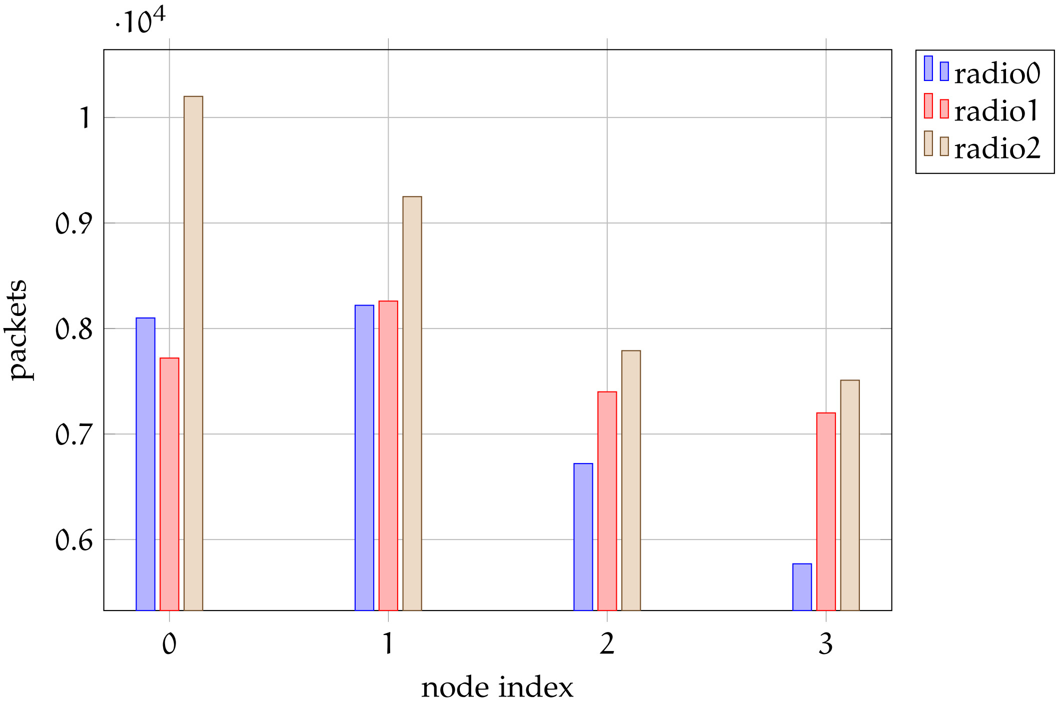


Figure 3.15: Packet distribution for testing WFS mode, datagram size 500B, with background traffic TX rate @ 6 Mbit/s

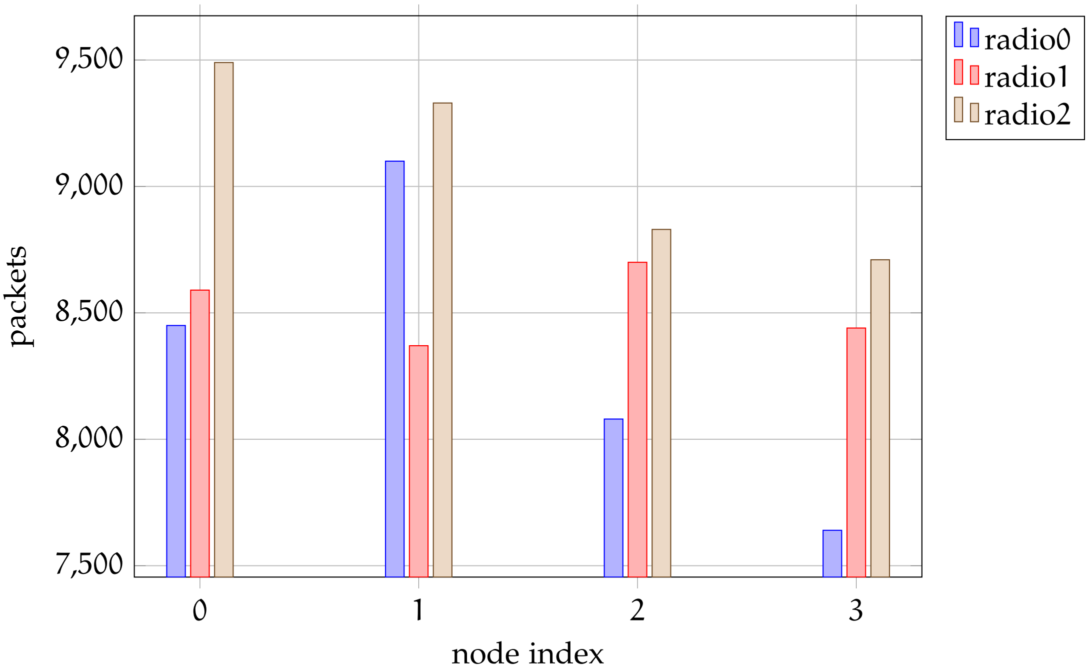


Figure 3.16: Packet distribution for testing WFS mode, datagram size 1kB, with background traffic TX rate @ 3 Mbit/s

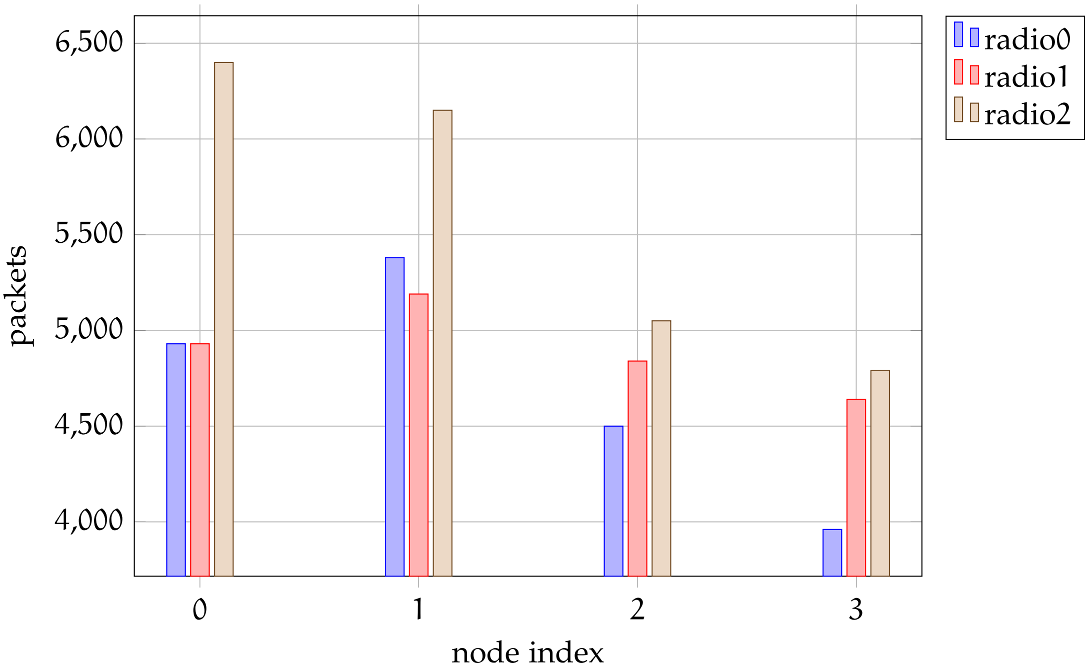


Figure 3.17: Packet distribution for testing WFS mode, datagram size 1kB, with background traffic TX rate @ 6 Mbit/s

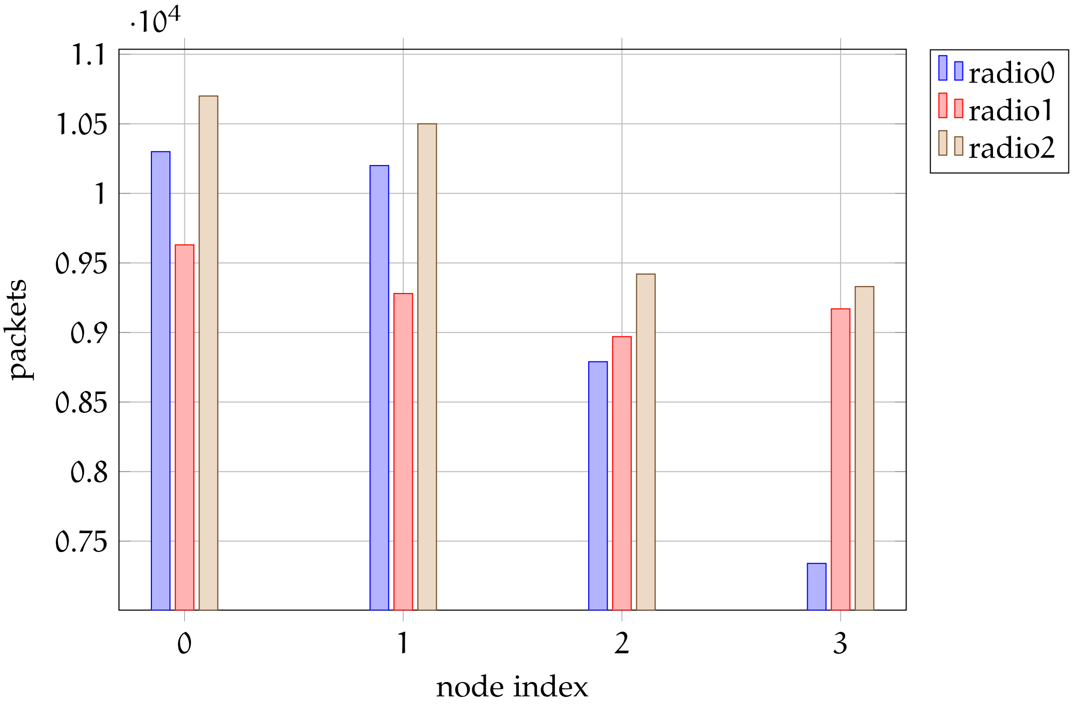


Figure 3.18: Packet distribution for testing WFS mode, datagram size 1.5kB, with background traffic TX rate @ 3 Mbit/s

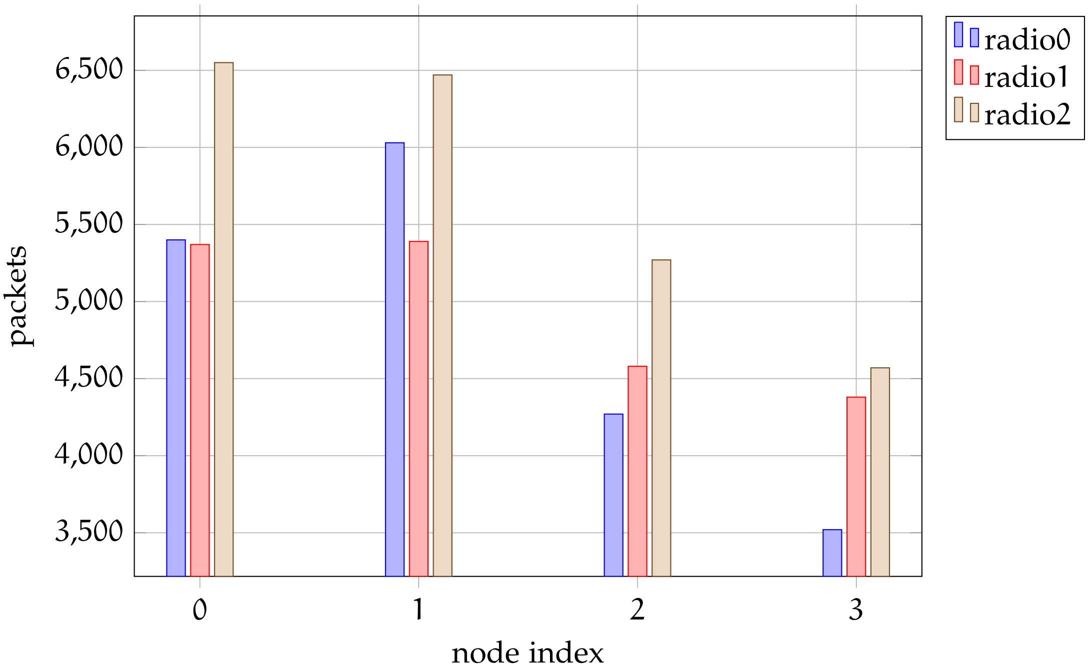


Figure 3.19: Packet distribution for testing WFS mode, datagram size 1.5kB, with background traffic TX rate @ 6 Mbit/s

Diagrams illustrate the total amount of sent packets at the end of the simulation. The packet summary cannot reveal the time-dependent behavior of each node. For instance, there is a time window between 30s and 40s where focus stream is running freely, without disturbing background traffic. In this window, only intra-route interference might occur, which is treated by the WFS scheduler of each node individually.

To show how WFS PS statistically outperforms SISC scheduling, an additional measurement series (20 runs) was added. Datagram sizes were set to 500B and 1000B, and background traffic is set to 6 Mbit/s. The total amount of sent packets at the end of the simulation is compared in table 3.5.

EXTENDED RR PS MODE

Next, results of the Extended RR PS mode are discussed. In Figures 3.20, 3.21 and 3.22 the TCP throughput with each receiving hop is compared, with different variations of R.

Table 3.5: Comparison of sent packet amounts with single radio and WFS mode

		TX packets ($\times 10^3$) per PS mode			
		Single Radio		WFS	
		500B	1000B	500B	1000B
mihost[0]	wlan0	6.27	3.15	8.05	5.34
	wlan1	0	0	8.18	4.86
	wlan2	0	0	10.4	6.04
mihost[1]	wlan0	5.92	3.08	8.65	5.38
	wlan1	0	0	8.76	5.47
	wlan2	0	0	9.70	6.26
mihost[2]	wlan0	4.82	2.75	6.32	4.49
	wlan1	0	0	7.83	4.69
	wlan2	0	0	8.38	5.33
mihost[3]	wlan0	4.58	2.64	6.18	3.78
	wlan1	0	0	7.07	4.45
	wlan2	0	0	7.17	4.80
mihost[5]	wlan0	4.39	4.39	4.39	4.39
mihost[7]	wlan1	2.93	2.93	2.93	2.93

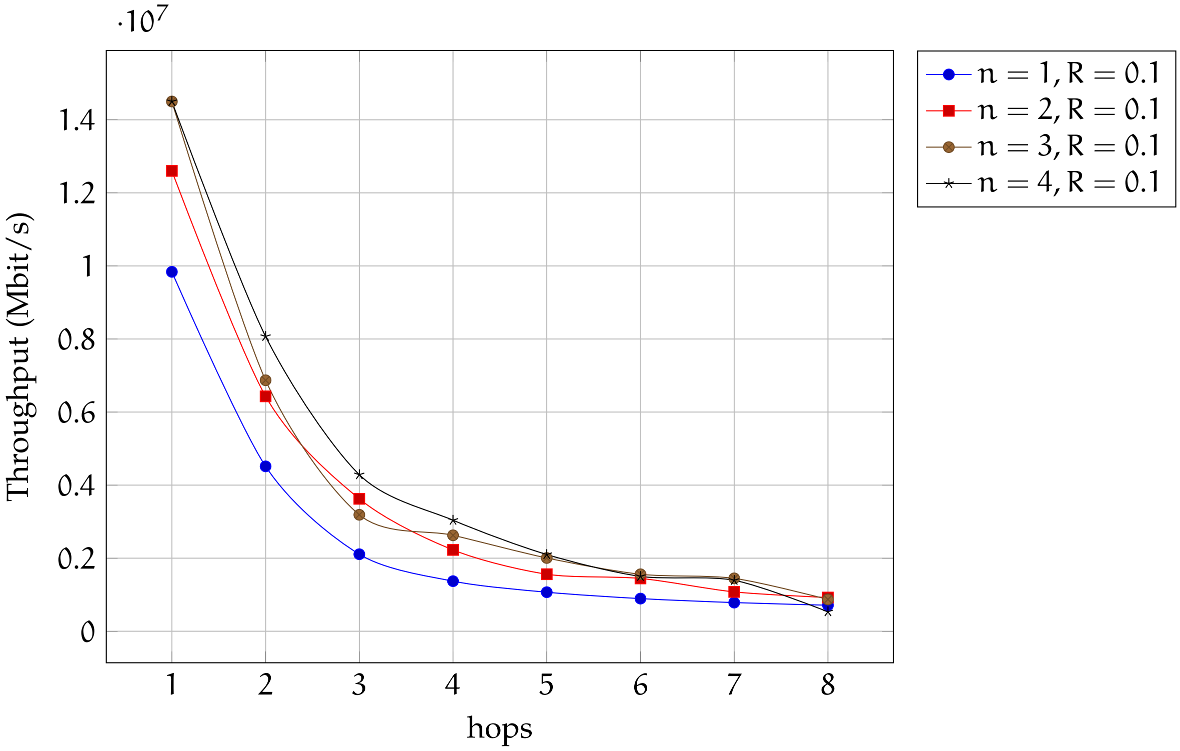


Figure 3.20: Throughput comparison with Extended RR mode and threshold rate $R = 0.1$

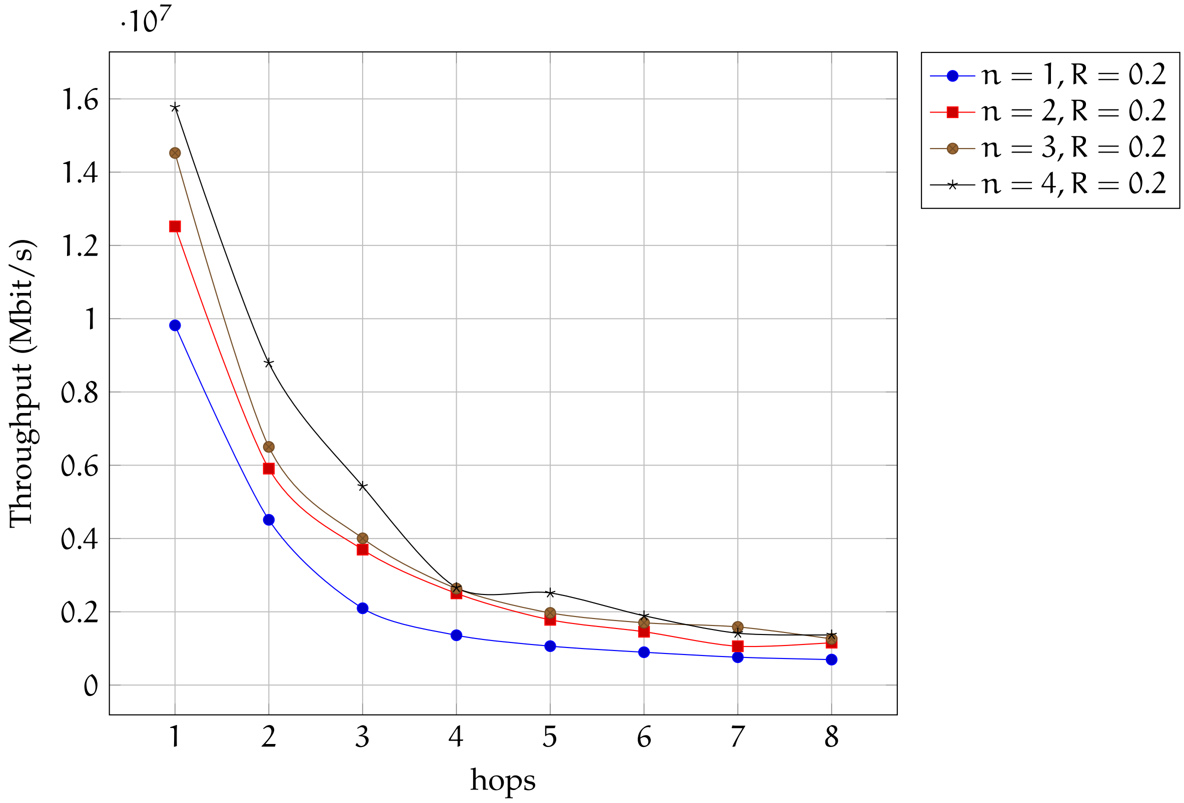


Figure 3.21: Throughput comparison with Extended RR mode and threshold rate $R = 0.2$

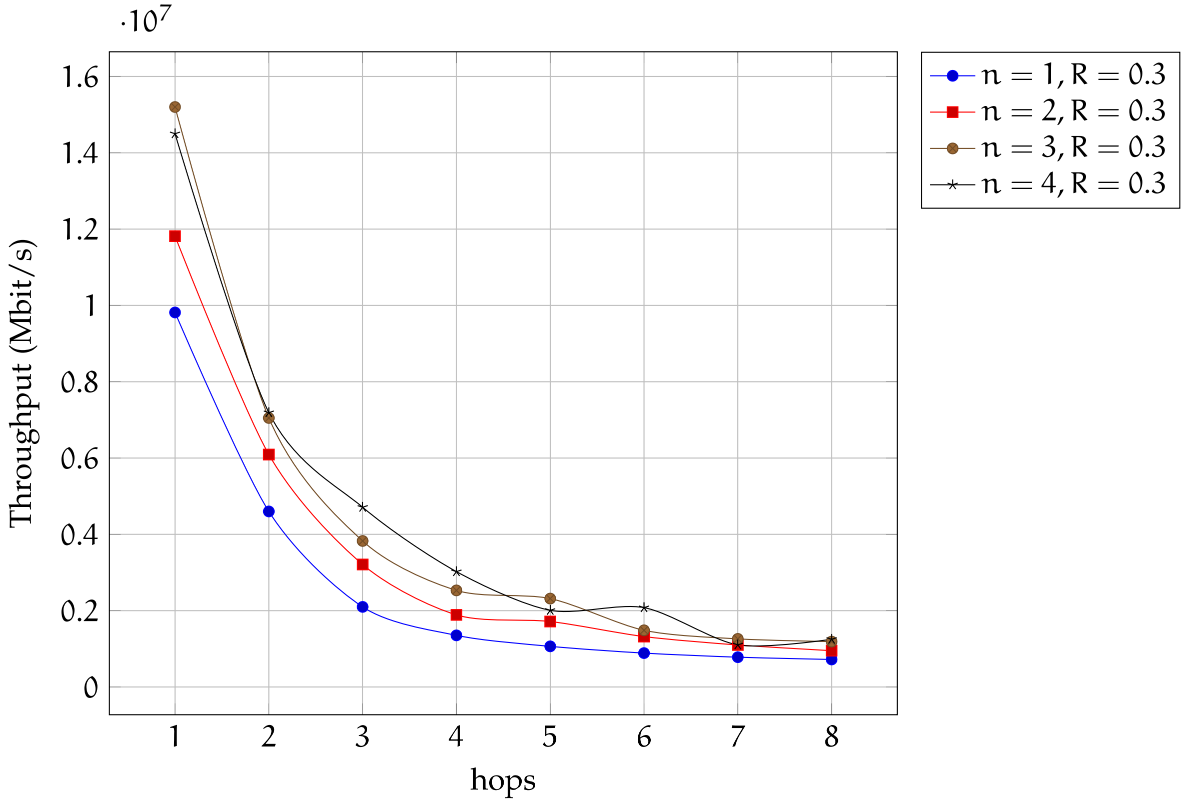
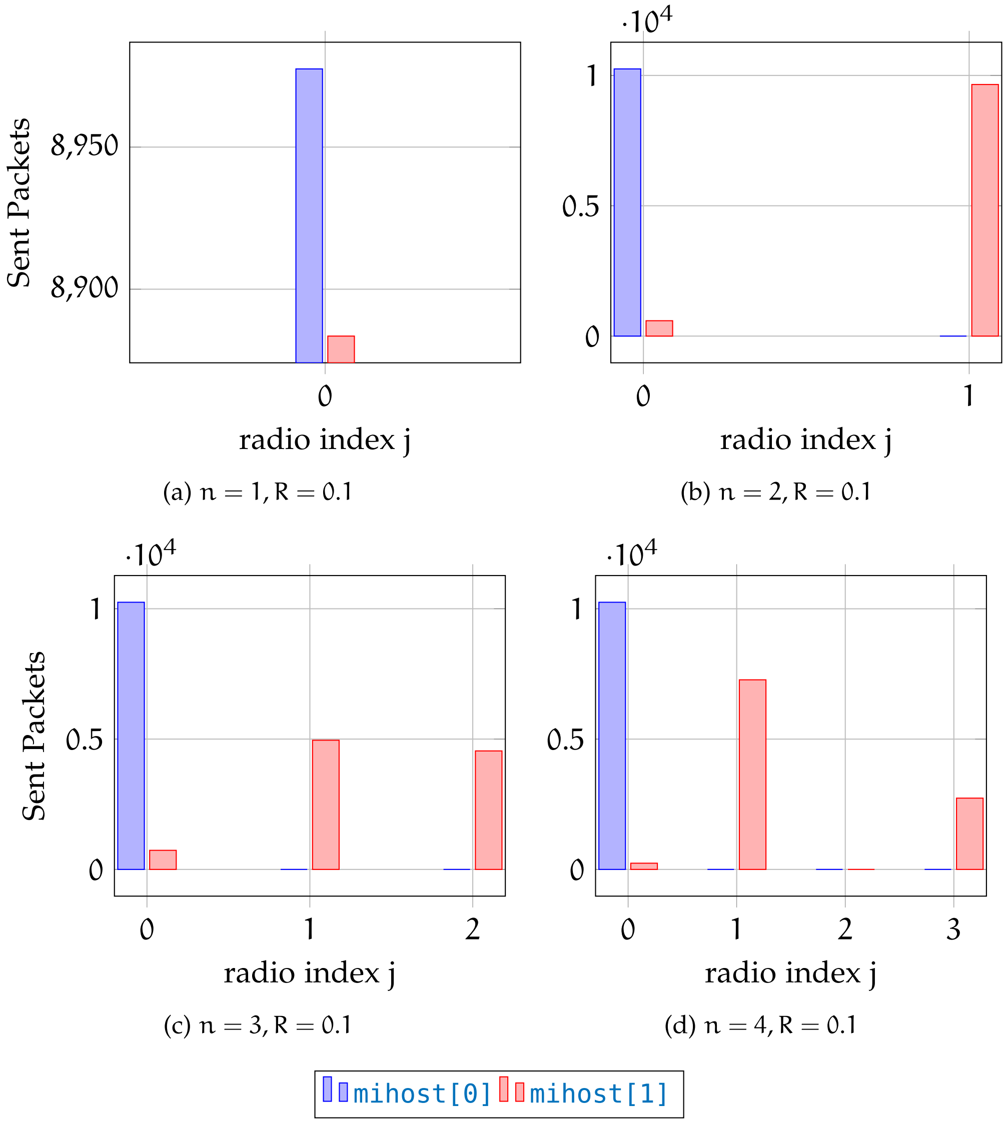


Figure 3.22: Throughput comparison with Extended RR mode and threshold rate $R = 0.3$

Curiously, throughput levels for $n = 2, 3, 4$ at the first hop roam above the expected 11 Mbit/s. A snapshot of a single run, where the average throughput peaks at 17 Mbit/s, is closer investigated. Fig. 3.23 shows the the packet distribution per radio.



(a) $n = 1, R = 0.1$

(b) $n = 2, R = 0.1$

(c) $n = 3, R = 0.1$

(d) $n = 4, R = 0.1$

Figure 3.23: Observation of packet distribution for 1-hop case, with $R = 0.1$

`mihost[0]` unicasts approx 10.000 packets here for $n = 1, 2, 3, 4$, and receives approx. the same amount in return, in form of TCP ACKs. The packet distribution in Fig. 3.23 reveals that with one radio, both nodes exchange approx. the same amount of packets (see Fig. 3.23a), as expected. `mihost[0]` sends always on channel 0, despite of the availability of fallback options in figures 3.23b, 3.23c and 3.23d. In Fig. 3.23a, channel 0 is used for both flow directions. For B = 1, 2, 3, `mihost[1]` then uses fallback radios to transmit TCP ACKs over backward channels. Figure 3.24 shows a snapshot of the the TCP RX throughput of `mihost[1]`. Mainly data is transmitted by `mihost[0]`.

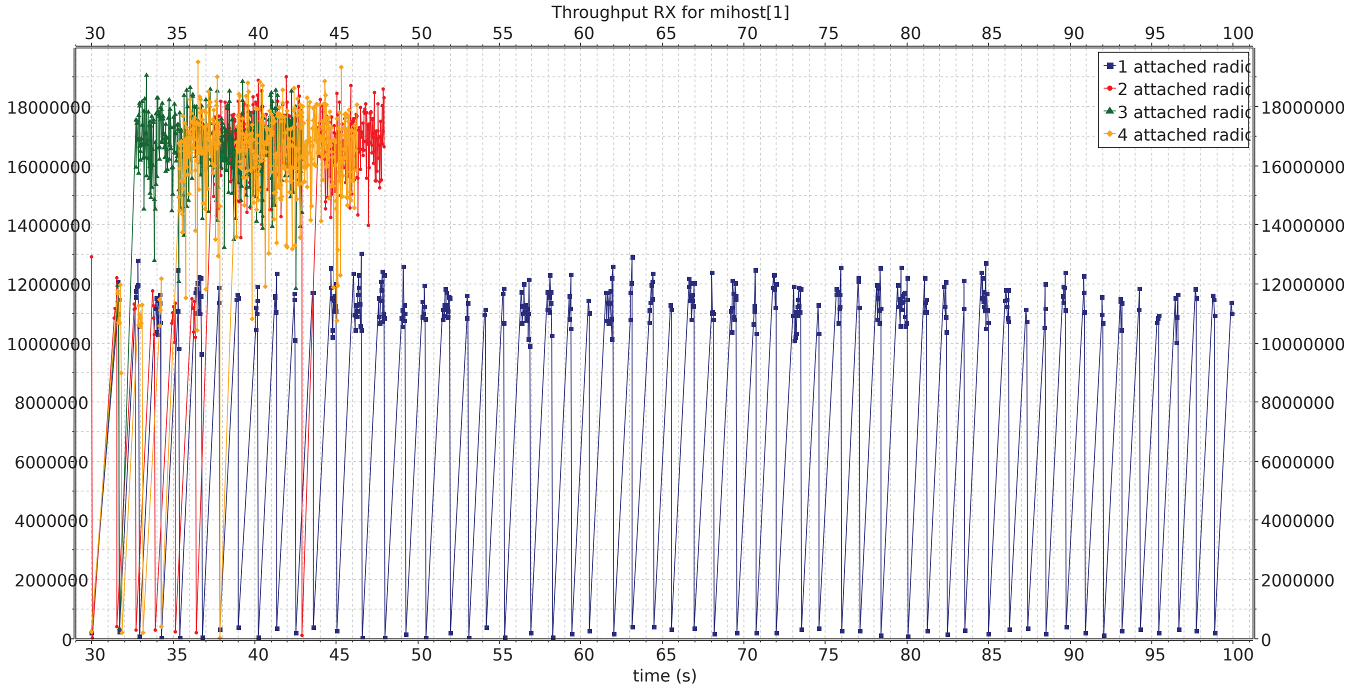


Figure 3.24: RX throughput of `mihost[1]` for testing Extended RR mode with multiple radios

The RX counterpart at `mihost[0]` behaves similar, but receives mainly TCP control messages from `mihost[1]`. The separation of TCP segments and control packets has a significant effect on TCP throughput in Fig. 3.24, although technically no load balancing is applied. Both nodes constantly sense the current loss rate. Measurements suggest that the asynchronous connection (separation of TX and RX traffic) via a data and backward channel(s) behaves self-regulating in that case.

In Fig. 3.25 (snapshot), `mihost[2]` is then equipped with 2 additional fallback radios. The PS behavior for this particular case reveals that the chosen TX radio switches from wlan[0] to wlan[2] after approx. 1s (red graph at 30s). R is set to 30% here.

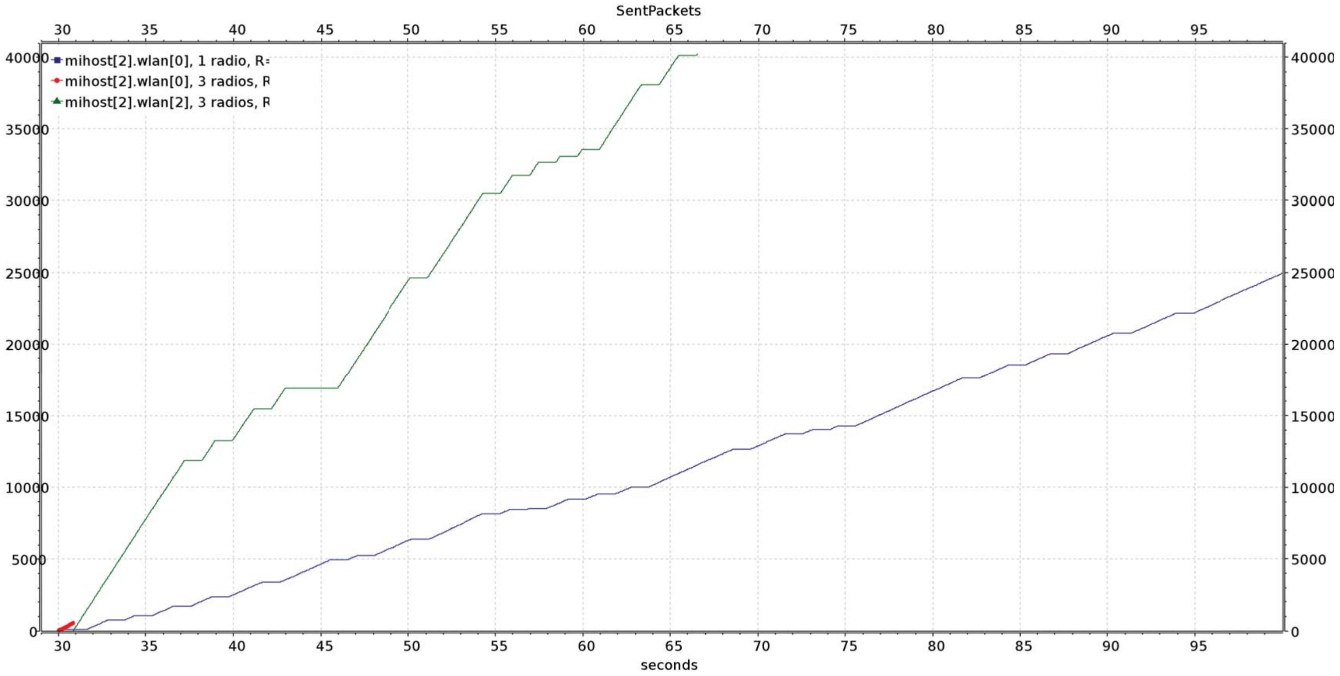


Figure 3.25: Sent packets per radio for testing Extended RR mode with 3 attached radios and threshold rate $R = 0.2$

3.3.3.3 *Evaluation*

WFS PS MODE

Generally it can be observed that the higher the background TX rate, the lower is the amount of packets scheduled on the focus route. This Parallel traffic, which blocks local channel resources has a significant impact on the performance of the vertical flow. It was observed that more load is shifted to radio 2 when the background traffic bandwidth increases. This implies less retransmissions on congested channels 0 and 1. Measurements have shown that beginning with a background TX rate of 3 Mbit/s, this desired effect is most visible. For 0.1 kbit/s link costs do not differ enough and WFS shows no effect. The results match the expected beneficial behavior of WFS PS mode, since the non-congested channel 2 is statistically used the most with significant background rates.

It is shown how the system adapts to the widely deployed ETT metric. However, it is expected that the scenario can be further improved by using an alternative metric with a more diversified output, to further separate load.

The numerical insight into the LB behavior of a forwarding bundle in table 3.5 outlines the advantage of WFS PS over the single-radio case. For instance when `mihost[0]` is observed for 500B. The single-radio version managed to send only approx. 6270 packets at the end of the simulation. While wlan[0] and wlan[1] with WFS have sent approx. 8050, respectively 8180 packets, wlan[2] has carried approx. 10400 packets. WFS outperforms the SISC case by a) exploiting all channels and b) adapting to selective congestion.

The advantage, when the WFS concept is combined with link-state driven OLSR networks is, that a performance degradation due to the use of under-performing links *and* next hops is prevented. This takes place in two steps: Without any modification or additional layer 2.5 module, a standard node already avoids congested links via *rerouting*. The ability of rerouting upon the presence of unsuitable links applies in MIMC WMNs as well: For path calculation, a single cost value per neighbor / edge is required, which can be the average quality of all bundled links. But in some cases, rerouting can cause higher path costs, due to a larger hop count. So the routing engine may still appoint an undesired neighbor as next hop, despite the presence of under-performing links [6]. In this case, a solution more fine-grained than rerouting is required. Link-state adaptive scheduling within a bundle fills this gap. Both measures rely on the same routing metric input; hence WFS PS mode can be easily embedded and perfectly complements mesh routing. Also, both measures can be combined to work in conjunction: A next hop decision is always proactive. WFS scheduling relies on proactive, traffic-independent probing, but is calculated reactively. Results in table 3.5 confirm that channels 0 and 1 with high link costs are gradually avoided.

EXTENDED RR PS MODE

Results in figures 3.20, 3.21 and 3.22 outline that among the three chosen rates for R, 20% offers slightly better results. Throughput levels in the hop distance range between 1 to 5 are continually improved with this value. In this hop range it can be also claimed that the more radios are available as fallback options, the higher is the achieved throughput with all settings for R. When the SISC case (blue graph) is compared with others, a consistent improvement was monitored with the fallback method, although it technically remains a *single-radio transmission*. Transferred to mesh communication, this means that throughput levels can be continuously increased, independent of the actual hop distance (affirmed for up to 8 hops). The described positive effects occur due to the lowered chance for intra-flow interference. Local channels with a disadvantageous loss condition are avoided completely. In comparison to the WFS mode, this absolute load shift to another radio is more radical. If R is assigned a less sensitive value than 0.3, results nearly converge with the ones of a single-radio node (or a MIMC node with $R \approx 1$).

Still, hop distance takes an enormous influence: After 5 hops, multiple FB radios (and with $R = 0.1$) have a less significant impact on performance, as throughput levels off to approx. (1 ± 0.5) Mbit/s.

The RR extension also leverages MIMC to increase *single hop* capacity beyond its regular limits. In the 1-hop case in Fig. 3.21, throughput is raised by approx. 2-6 Mbit/s. Fig. 3.24 also shows that the interference time is then drastically diminished: In the SISC case, the maximum simulation time of 100s is not sufficient to transfer 20MB. In the MIMC case, approx. 48s are needed in total. A shorter period of channel congestion hence enables more network capacity, with more uncongested air time on channel 0. The occupied time period is lessened on the back channel 1 in the same way, although the required bandwidth for TCP control traffic is already low (approx. 320 kbit/s). The small investment of just one additional radio per node offers a good trade-off in terms of improved interference levels.

Fig. 3.25 confirms that in comparison with the single-radio setup, the throughput level via the unspoiled channel is higher.

3.3.4 *Layer 2 Forwarding*

Simulations were carried out to evaluate the impact on transmissions of the proposed layer 2 forwarding within LMHPC. It foresees to avoid an IP layer processing of each packet in intermediate nodes.

Several protocols with layer 2 forwarding methods are compared in [134] in real testbeds. Especially with WDS [13] lower end-to-end delays are measured, as when layer 3 would be included in the deployed multi-hop chain in [134]. This is due to a faster lookup of next-hop MAC addresses and a reduced forwarding table size, which is also aimed at with the M-LFIB table.

Ariza-Quintana et al. [113] also aim to check the performance of layer 2 and 3 routing. Their proposed architecture with layer 2 components is implemented in OMNeT++. The group agrees that routing at mac layer reduces the "economical and computational" cost of mesh deployment. They compare the packet forwarding in layer 3 (with OLSR) and layer 2 (next-hop MAC based and label based). They discover that the simulation speed is reduced with layer 2 forwarding, from which can be concluded that less computational effort is needed when layer 3 processing is omitted. However, they achieve "very similar results (in terms of packet delivery ratio and packet delay) to those obtained with classical IP routing" [113].

The reason lies within OMNeT++. The simulator does not consider the processing delay in the IP layer: "Consequently in an actual scenario (with real routers) the lookup process at the IP tables would introduce an additional component in the delay of IP routing" [113]. OMNeT++ is a *discrete event* simulator and therefore has limitations in that context. It partly offers an abstraction of aspects of real hardware via representative values, such as error rates (e. g., in the Trivellato table in appendix A.5.1). OMNeT++ further introduces artificial and synthetic delays. `procDelay` represents the lookup of IP tables and the general processing time for each incoming packet in the IP layer. A particular micro-observation in appendix A.5.4 has shown that the IP lookup process does not consume time, when `procDelay` is (by default) set to `0`s. Ariza-Quintana et al. do not consider this in their measurements.

The per packet processing time was also not included in simulations so far, but it exists as a metric in a real life environment. In dedicated LAN router hardware it may roam approximately between 10 to 60 µs [135]. It is expected to be higher in older low-cost, standard, or commodity hardware [113] and thus will have a bigger impact on layer 2 forwarding. All subsequent delays, especially caused by collision avoidance in the MAC are added, as in simulations before.

The chosen setup is depicted in Fig. 3.26. `mihost[0]` is unicasting packets to `mihost[14]` over 15 hops. Each node has 5 radios. This causes that RTs have up to 70 entries. The simulation time is 150s. Traffic is generated only in the last 50s, due to the chosen

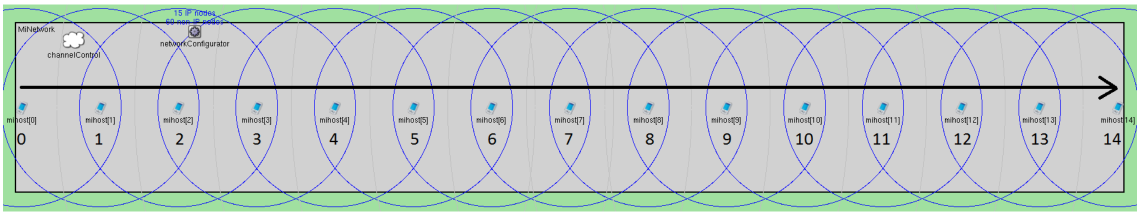


Figure 3.26: Scenario for testing layer 2 forwarding

HELLO interval and the resulting convergence time (for a study on the relationship between both see [136]). Three series (30 runs each) were measured:

1. Ping (ICMP payload of 56 Bytes) every 500ms. Fig. 3.27 contains a single snapshot. The red curve depicts the computed mean values of the course of blue values
2. UDP stream with a TX rate of 500 kbit/s. An example snapshot is given in Fig. 3.28 (the blue curve is the TX and the red curve the RX rate. Datagram size is 250B
3. Same UDP stream with a datagram size of 500B

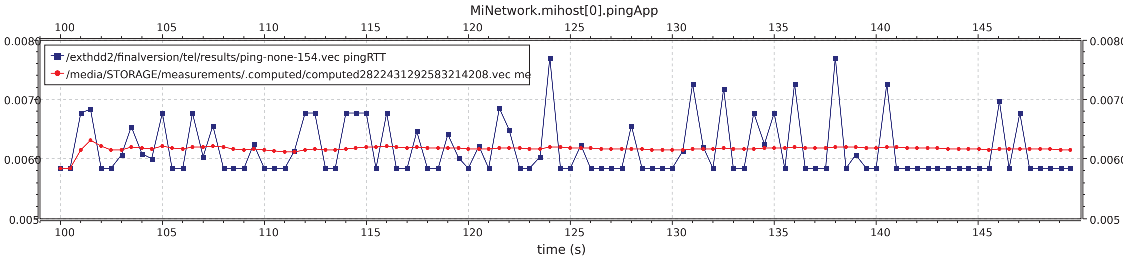


Figure 3.27: RTT with Ping and a set processing delay of 125µs

Each node will send on a singe channel. The five available channels are evenly distributed across the chain, to minimize interference. The IP `procDelay` parameter in intermediate nodes is included on top of other delays and ranges from 0s (corresponds to no treatment in the IP layer) to 200µs, with steps of 25µs. Fig. 3.29 shows what can be expected for using layer 2 forwarding in the simulator.

While UDP graphs show the one-way delay, the used ping application outputs the RTT.

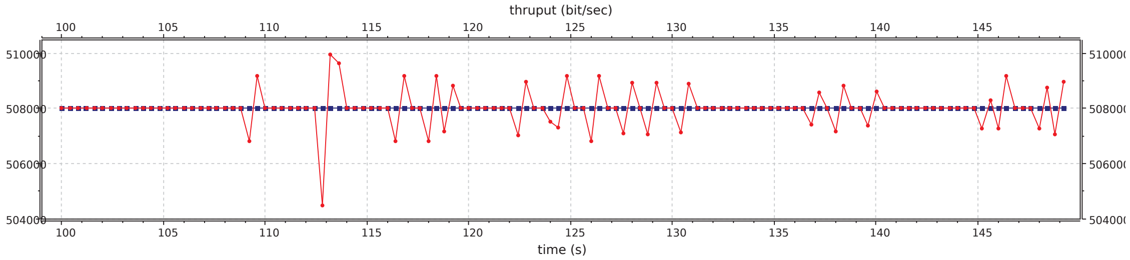


Figure 3.28: UDP throughput

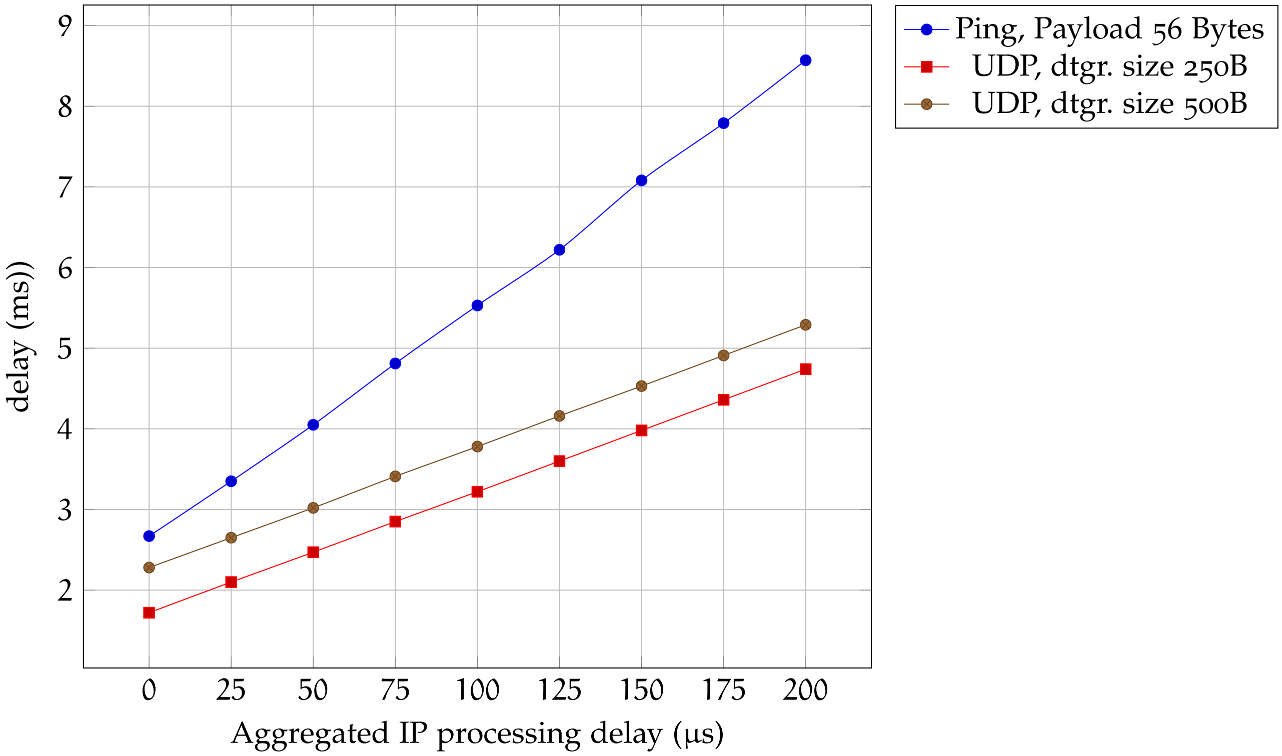


Figure 3.29: Impact of IP layer processing delay in INETMANET

The detected gains of an end-to-end delay reduction per packet in the millisecond range, which are *expected* when layer 3 processing is avoided in the simulation, lead to the following *recommended conditions*, in which commutation might be most efficient:

- Setups with a low basic level of end-to-end delays (i. e., low congestion). This was achieved with channel diversity in Fig. 3.26
- Setups with large hop distances and long RTs, as the effect adds up
- Although trends such as "IP-on-a-chip" and specialized high-performance routers will possibly minimize the effect of fast layer 2 treatment in non-

simulator setups, there is an expected margin of benefit with cheap *commodity* hardware, in which system resources may be actively relieved (see Chapter 1, 1.4.2). Open source developer communities of popular protocols such as OLSR or B.A.T.M.A.N. focus on low hardware workload and compatibility with old PCs and routers. Economic mesh installations (especially in rural areas or developing countries [23]) based on this technologies might benefit here by the proposed LMHPC component.

3.3.5 *Dynamic Channel Changes*

The system adapts to a change in the channel map within a safe period. This period from the system concept was designed to absorb negative effects from channel switches. Its influence is now tested in a single case scenario with an increase and later decrease of capacity on a route. It is expected, that the system can dynamically handle the transition of time-variant capacities, in order to always reach the highest possible performance.

The chosen setup contains 3 nodes, lined up in a chain. `mihost[0]` is unicasting packets to `mihost[2]` via `mihost[1]`. Each node has 3 radios. Table 3.6 lists the channel map. Appendix A.5.5 specifies this configuration.

Table 3.6: Time-variant channel map

		channels		
node	radio	$0 \leqslant t < 30s$	$30s \leqslant t < 70s$	$70s \leqslant t < 90s$
`mihost[0]`	wlan[0]	0	0	0
	wlan[1]	1	1	1
	wlan[2]	2	2	2
`mihost[1]`	wlan[0]	0	0	0
	wlan[1]	3	1	3
	wlan[2]	4	2	4
`mihost[2]`	wlan[0]	0	0	0
	wlan[1]	1	1	1
	wlan[2]	2	2	2

Table 3.6 reveals that `mihost[1]` controls the available end-to-end capacity. In the first and last phase it works as a bottleneck, when the unusable channels 3 and 4 are used and only channel 0 permits multi-hop communication. The simulation time is 150s. The UDP stream is set to a TX rate of 25 Mbit/s (1000B datagram size). Three PS mode variants were compared: RR and WFS *without* a safe period and RR *with* a safe period (21s). Where enabled, the safe period applies for a capacity increase. For a decrease (occurs at $t = 70s$) it is always enabled, to prevent packet loss. Results are shown in Fig. 3.30. As in previous simulations, `Hello_ival` is set to 2s and `Tc_ival` to 5s. HELLO and TC messages are OLSR-internal [35] intervals. `periodWFS`, as a novel system LB interval, is set to 1s. All three take direct influence on the convergence time between channel switches. A moving average filter ($\alpha = 0.1$) is applied to the throughput graphs in Fig. 3.30 for an improved visibility of the results.

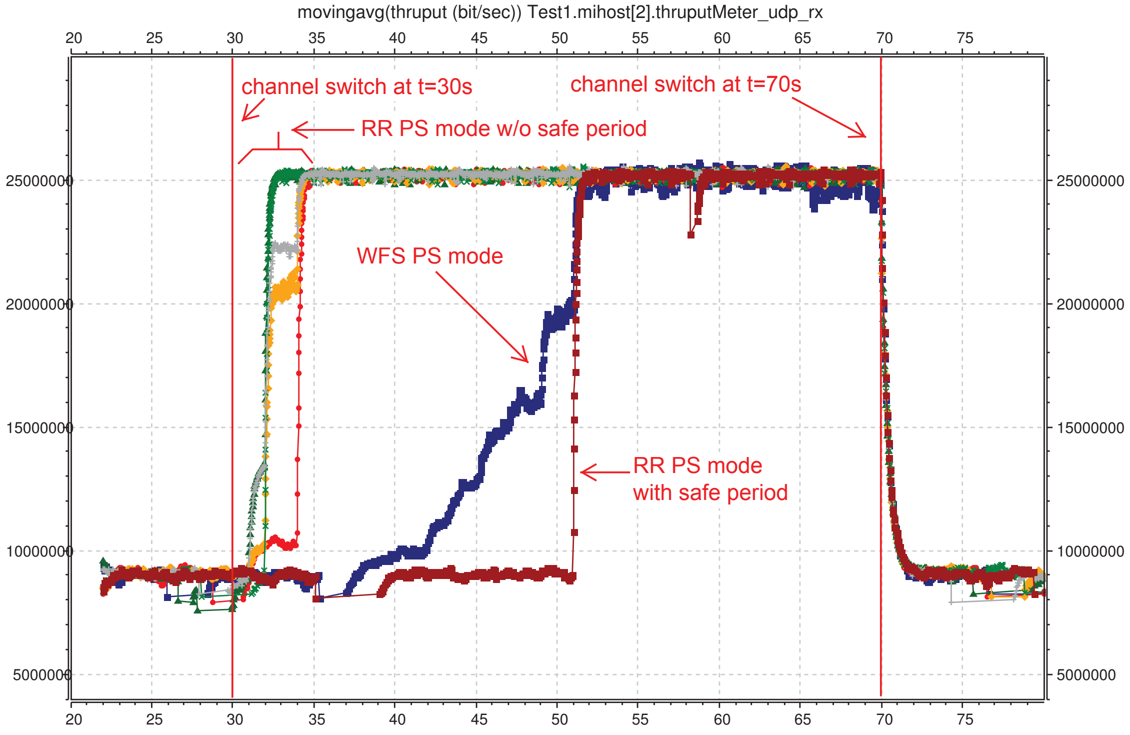


Figure 3.30: Influence of convergence and safe period in a dynamic switch scenario

The blue graph (WFS PS, snapshot) reveals that it takes approx. 21s until the capacity of all 3 radios is exploited (from approx. 8 Mbit/s to approx. 25 Mbit/s). This behavior influenced the configuration of the crimson graph (RR PS, snapshot), in which a fixed safe period of 21s was overtaken. The remaining graphs also depict the RR PS mode: It was opted to show 5 runs here.

WFS has the longest *convergence* time, because the three mentioned intervals superimpose each other at `mihost[0]` and `mihost[1]` (TX nodes) and take effect individually for each setup. For a capacity increase in a bundle, LB with WFS PS is inert. The two new radios are added to the bundle after 30s and are steadily considered for PS. The transition time can be decreased by using shorter intervals for OLSR signaling. A trade-off with increased signaling overhead needs to be considered then. However, OLSR will always show a minimum dynamic, because new or missing neighbors / radios need to be registered first. With RR PS (w/o safe period), new radios are directly used when they are registered. For the 5 runs, this may take up to 5s. The Safe CA Table within the safe period however takes immediate effect for the capacity decrease at $t = 70s$, to prevent packet loss due to non-existing radios.

3.4 FINAL ANALYSIS OF RESULTS

The Sections 3.3.1 to 3.3.5 outline the advantages of the system in a MIMC environment. These particular benefits, plus additional findings, shall be *summarized*.

The exploitation of channel diversity is considered a key solution to tackle the identified limitations of WMNs. The system adds a novel chain of components for packet treatment (Commutation, Queuing, TX Scheduling) to a standard mesh node. Better performance and higher capacity levels are reached especially in the last component, which was confirmed by testing the three implemented PS modes. Once bundles to neighbors are defined by the Multi-Interface Bundle Management component, each regarding mode leads to different benefits. If ETT-based delay probing results drop on a link in a bundle, the adaptive WFS PS mechanism assigns less load on it, with the following effects:

- WFS allows *multi-hop* vertical streams (of 4 hops in Fig. 3.11) to exchange more packets in total, despite local congestion caused by other traffic
- It potentially decreases congestion of *all* channels which are used by different streams and facilitates their coexistence

Hence, the WFS mode improves the performance of a standard WMN backbone through channel diversity and the inclusion of mesh parameters. Also, Weighted-fair-based scheduling is useful when a rerouting option is considered too cost-expensive by OLSR. In other terms, capacities on the reputed bad next-hop link can still be optimized with WFS. Load shifting with the Extended RR mode is more radical, because it forces a complete channel switch. The RR fallback extension requires the user to configure the amount of backup radios and a sensitivity threshold. Both settings can have an extensive effect, as the measurements have shown: Although technically single-radio communication is deployed, air time of packets was reduced, network stability was increased and throughput levels were raised constantly up to 8 hops. Less air time also means less congestion, which raises the overall mesh capacity and contributes to the scientific problem. The impact of intra-flow interference on a single flow, which might be both vertical or horizontal here, is decreased. The best transmission performance between 1 to 5 hops has been achieved with a threshold of 20%. Users in a WMN may access Internet via a GW in a single-hop distance. The configuration of fallback resources for this specific case also leverages the bandwidth and capacity on this link, by separating data and signaling for TCP. Thus by investing in a single extra radio, throughput within the GW neighborhood can be increased. The sole usage of RR PS enhances the transmission capacity of most tested vertical flows, but cannot solve fairness issues, since all radios are loaded evenly. Still, if the hop count to GWs can be kept short (1-3 hops) in a WMN, RR becomes an attractive and yet simplistic scheme to improve throughput in large mesh backbones.

The TX Scheduling component directly contributes to improved capacity, however the combination with the previously active components in the packet chain further outlines the prioritization of GW traffic, as the second primary research goal. The simulation to investigate in layer 2 fast forwarding, which in the system is prepared by TEL and executed by the Commutation component, has shown that by avoiding additional processing delays in the IP layer, the end-to-end delay of vertical multi-hop flows can be improved under certain conditions. In the following step, queues improve the performance of vertical- and QoS-related flows in mesh scenarios, where those flows are disadvantaged by long hop counts and competing horizontal traffic. The end-to-end delay represents a crucial QoS parameter and can be selectively decreased with the component. This resolves unfairness due to intra-route interference. Queuing even has the potential to favor QoS- or GW streams on single-channel paths; depending on the weighting scheme.

Finally, the introduction of time-variant channel switches has shown that the safe period must be adapted to external intervals of the routing protocol.

The selected measurements have shown that it is necessary to amplify a node's standard architecture with the tested components and the wrapping system, to achieve better performance, QoS, and higher capacity levels in the tested environments.

3.5 CONCLUSION

This Chapter contains the proof-of-concept of the presented system for multi-radio mesh nodes. Different techniques, which have been selected and composed in a holistic system in Chapter 2, were tested and their impact was discussed.

The selected scenarios have shown that the implemented system is able to:

- Increase the QoS performance of the network, making use of the availability of multiple radios
- Conquer / prevent interference by link-quality-aware scheduling within a bundle, with an interchangeable routing metric
- Provide a tool to facilitate QoS policies and the transmission of vertical traffic, which lead to a selective end-to-end delay improvement on multi-hop paths
- Decrease air time of packets
- Handle any radio/channel constellation in a node; whether it is set up manually or provided by a CA protocol
- Lower end-to-end delays through layer 2 forwarding are expected when commodity hardware is used

- A safe period covers convergence effects of RR and WFS PS modes after a capacity increase.

Conclusions and Recommendations

CONCLUSION

The main objective from the thesis introduction, to *design* a holistic node management system, has been accomplished. This applies as well to secondary research goals. The preliminary hypothesis can be confirmed, because the initially envisioned bundling approach now generally improves performance in WMNs and allows to achieve a higher quality of selected transmissions.

The following goals and contributions to mesh networking have been accomplished, sorted by their fulfilled tasks:

- Tasks regarding *transmission enhancement* in WMNs (as a chain of processes):
  1. Processing of pre-defined CA protocol input
  2. Traffic awareness / analysis and packet classification via DiffServ
  3. Five-tuple flow identification
  4. Consideration of GW presence in the topology, to protect its traffic
  5. Fast packet forwarding based on fixed-length labeling, to enforce the results of vertical traffic identification. Seamless and transparent to upper layers
  6. Priority queuing to enforce the results of traffic analysis
  7. Bundle definition allows to manage and virtualize combined radios, including IP and MAC management and without demanding user attention
  8. Bandwidth aggregation and scheduling in a bundle, based on dynamic link states (metrics) or a radio-specific parameter
  9. Proposed PS modes are simple, but offer a substantial diversity for different mesh scenarios
- Tasks regarding *performance improvements* over a standard WMN backbone:

  1. System evaluation has shown that multiple links to different neighbors can be fully exploited, along with the consideration of heterogeneous link conditions and different traffic classes
  2. Overall mesh capacity is improved. This is already achieved with a simple RR scheme. A fallback extension of this mode even allows to almost completely avoid unsuitable radios (suitability can be determined by a custom factor) and thus diminishes intra-flow interference. The fallback extension allows also a slower capacity degradation with an increasing hop distance and shorter packet air times
  3. Weighted Fair Scheduling adapts to regional link conditions
  4. Middle-layer queues help to overcome intra-route interference of disadvantaged flows by reducing their delay

- Tasks and results regarding the *design* of the system:
  1. Design and definition of a *holistic* multi-component system to improve wireless mesh networks, by enabling more capacity and the prioritization of vertical traffic. The heart is a middle-layer 2.5 module, whose result is to treat a packet appropriate to its class in the forwarding process and to schedule it over a suitable interface
  2. Various methods like label-switched routing, or priority queuing have been an inspiration. Their features relevant for the desired behavior of a mesh network were adapted. The result is a system which benefits from synergies of various network technologies
  3. The system is ready to work with any proactive link state routing protocol and quality-related metric
  4. The system is CA protocol independent. However, several requirements have been posed on a CA protocol
  5. The system is modular; an administrator can enable/ disable class-based queuing or label-based forwarding without affecting the exploitation of multiple radios

RECOMMENDATIONS

Future extensions of the system may include:

1. Combine the current single-path optimization with multi-path solutions
2. Fairness concepts for parallel vertical flows with different end-to-end hop distances, which take the same route or next-hop
3. Queuing process is not aware whether it is actually necessary or not to enqueue packets. Including layer 2 information on the utilized capacity of the medium, in order to refine (de)queuing decisions, might further improve treatment of ITCs
4. Investigations on the usability of awareness factors for entire routes. This may include the input which PS modes are used along the route and how the load is balanced in each bundle. Route-wide signaling (e. g., with Resource Reservation Protocol (RSVP)-like solutions) may further enrich the concept

Future measurements may consider alternative WMN characteristics, such as:

1. Low mobility of mesh routers, including predetermined or random movement patterns, if possible around physical obstacles
2. Mixed traffic in a large-scale, dynamic WMN: More complex topology, presence of more horizontal and vertical flows, randomized traffic / user behavior, a more heterogeneous landscape, more users and GWs and up- and down-coming devices

Finally, it is recommended to subsequently follow a real-life Linux implementation, as discussed in Chapter 2, Sections 2.9 and 2.10.

BIBLIOGRAPHICAL REFERENCES

[1] J.P.R. Pereira and P. Ferreira. Access networks for mobility a techno-economic model for broadband access technologies. In *5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009*, pages 1–7. doi: 10.1109/TRIDENTCOM.2009.4976221. (Cited on page 1.)

[2] Hequan Wu. Some thoughts on the transformation of information and communication technologies. In *2011 IEEE Technology Time Machine Symposium on Technologies Beyond 2020 (TTM)*, pages 1–1. doi: 10.1109/TTM.2011.6005152. (Cited on page 1.)

[3] S. Dustdar. Research challenges for building complex service-oriented systems for the future internet. In *18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09*, pages 1–1. doi: 10.1109/WETICE.2009.70. (Cited on page 1.)

[4] Yangyang Wang, Jun Bi, and Jianping Wang. Towards an aggregation-aware internet routing. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7, . doi: 10.1109/ICCCN.2012.6289241. (Cited on page 1.)

[5] Jangeun Jun and M.L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, 10(5):8–14, 2003. ISSN 1536-1284. doi: 10.1109/MWC.2003.1241089. (Cited on pages 1, 2, 15, 16, and 26.)

[6] Ian F. Akyildiz and Xudong Wang. Wireless mesh networks. In *Wireless Mesh Networks*. John Wiley & Sons, Ltd, 2009. ISBN 9780470059616. URL `http://onlinelibrary.wiley.com/doi/10.1002/9780470059616.fmatter/summary`. (Cited on pages 1, 2, 10, 11, 16, and 130.)

[7] Lee Garber. Wi-fi races into a faster future. 45(3):13–16. ISSN 0018-9162. doi: 10.1109/MC.2012.104. (Cited on page 1.)

[8] S. Maurina, J. Fitzpatrick, L. Trifan, and L. Murphy. An enhanced bridged-based multi-hop wireless network implementation. In *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, pages 1–9, March 2010. (Cited on pages 2 and 25.)

[9] D. Hock, N. Bayer, R. Pries, M. Siebert, D. Staehle, V. Rakocevic, and B. Xu. QoS provisioning in WLAN mesh networks using dynamic bandwidth control. In *Wireless Conference, 2008. EW 2008. 14th European*, pages 1 –7, June 2008. doi: 10.1109/EW.2008.4623896. (Cited on pages 2, 25, 34, and 109.)

[10] M. Kretschmer, C. Niephaus, D. Henkel, and G. Ghinea. QoS-aware wireless back-haul network for rural areas with support for broadcast services in practice. In *2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 758 –764, October 2011. doi: 10.1109/MASS.2011.84. (Cited on pages 2, 14, and 35.)

[11] K.N. Ramachandran, E.M. Belding, K.C. Almeroth, and M.M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006. doi: 10.1109/INFOCOM.2006.177. (Cited on pages 2, 14, and 15.)

[12] 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*, MobiCom '04, page 114–128, New York, NY, USA, 2004. ACM. ISBN 1-58113-868-7. doi: 10.1145/1023720.1023732. URL `http://doi.acm.org/10.1145/1023720.1023732`. (Cited on pages 2, 14, 21, 24, and 178.)

[13] IEEE standard for information technology–Telecommunications and information exchange between systems local and metropolitan area networks–Specific

requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012. doi: 10.1109/IEEESTD.2012.6178212. (Cited on pages 2, 25, 28, 131, 190, and 192.)

[14] Juan J. Gálvez, Pedro M. Ruiz, and Antonio FG Skarmeta. A distributed algorithm for gateway load-balancing in wireless mesh networks. In *Wireless Days, 2008. WD'08. 1st IFIP*, page 1–5. IEEE, 2008. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4812861`. (Cited on pages 2 and 16.)

[15] Jungfang Wang, Bin Xie, and Dharma P. Agrawal. Guide to wireless mesh networks. In Sudip Misra, Subhas Chandra Misra, and Isaac Woungang, editors, *Guide to Wireless Mesh Networks*, Computer Communications and Networks, pages 1–30. Springer London, January 2009. ISBN 978-1-84800-908-0, 978-1-84800-909-7. URL `http://link.springer.com/chapter/10.1007/978-1-84800-909-7_1`. (Cited on pages 3, 13, 19, 24, 96, and 208.)

[16] Navneet Kaur and Jatinder Singh Saini. Performance enhancement of 802.11 based wireless mesh network by using multi- radio multi-channel. In *2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*, pages 71–76, December 2013. doi: 10.1109/ICGCE.2013.6823402. (Cited on page 3.)

[17] P. Kyasanur, J. So, C. Chereddi, and N.F. Vaidya. Multichannel mesh networks: challenges and protocols. *IEEE Wireless Communications*, 13(2):30–36, April 2006. ISSN 1536-1284. doi: 10.1109/MWC.2006.1632478. (Cited on pages 4 and 36.)

[18] Luciano Bononi, Marco Di Felice, Antonella Molinaro, and Sara Pizzi. Enhancing multi-hop communication over multi-radio multi-channel wireless mesh networks: A cross-layer approach. *Computer Communications*, 34(13):1559–1569, August 2011. ISSN 0140-3664. doi: 10.1016/j.comcom.2010.12.002. URL `http://www.sciencedirect.com/science/article/pii/S014036641000486X`. (Cited on pages 4 and 26.)

[19] Vladimir Brik, Shravan Rayanchu, Sharad Saha, Sayandeep Sen, Vivek Shrivastava, and Suman Banerjee. A measurement study of a commercial-grade urban wifi mesh. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, page 111–124, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-334-1. doi: 10.1145/1452520.1452534. URL `http://doi.acm.org/10.1145/1452520.1452534`. (Cited on pages 4, 22, and 35.)

[20] Wenjia Wu, Junzhou Luo, Ming Yang, and L.T. Yang. Joint interface placement and channel assignment in multi-channel wireless mesh networks. In *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, pages 395–402, 2012. doi: 10.1109/ISPA.2012.58. (Cited on pages 4, 35, and 36.)

[21] C.T. De Oliveira, F. Theoleyre, and A. Duda. Connectivity in multi-channel multi-interface wireless mesh networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 35 –40, July 2011. doi: 10.1109/IWCMC.2011.5982503. (Cited on pages 4 and 36.)

[22] Huaizhou Shi, R.V. Prasad, E. Onur, and I.G.M.M. Niemegeers. Fairness in wireless Networks:Issues, measures and challenges. *IEEE Communications Surveys Tutorials*, 16(1):5–24, 2014. ISSN 1553-877X. doi: 10.1109/SURV.2013.050113.00015. (Cited on page 12.)

[23] Christian Köbel, Walter Baluja Garcia, and Joachim Habermann. A survey on wireless mesh network applications in rural areas and emerging countries. In *2013 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 389–394, October 2013. doi: 10.1109/GHTC.2013.6713717. (Cited on pages 12, 96, and 135.)

[24] D. Benyamina, A. Hafid, and M. Gendreau. Wireless mesh networks design #x2014; a survey. *IEEE Communications Surveys Tutorials*, 14(2):299–310, 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.042711.00007. (Cited on page 13.)

[25] A.M. Ahmed, A.H. Abdalla, and I. El-Azhary. Gateway placement approaches in wireless mesh network: Study survey. In *2013 International Conference on*

Computing, Electrical and Electronics Engineering (ICCEEE), pages 545–547, August 2013. doi: 10.1109/ICCEEE.2013.6633998. (Cited on page 13.)

[26] H. Pathak and R. Dutta. A survey of network design problems and joint design approaches in wireless mesh networks. *IEEE Communications Surveys & Tutorials*, PP(99):1–33, January 2010. ISSN 1553-877X. doi: 10.1109/SURV.2011.060710.00062. (Cited on pages 13, 14, 18, 27, 32, 54, and 182.)

[27] S. Vural, Dali Wei, and K. Moessner. Survey of experimental evaluation studies for wireless mesh network deployments in urban areas towards ubiquitous internet. *IEEE Communications Surveys Tutorials*, 15(1):223–239, 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.021312.00018. (Cited on pages 13, 16, and 192.)

[28] Cheong Loong Chan, Sheng Chyan Lee, Kee Choon Yeong, and V. Jeewa. Innovations to improve wireless mesh network performance: A survey. In *2013 IEEE Symposium on Wireless Technology and Applications (ISWTA)*, pages 80–84, September 2013. doi: 10.1109/ISWTA.2013.6688822. (Cited on pages 14 and 17.)

[29] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005. ISSN 13891286. doi: 10.1016/j.comnet.2004.12.001. URL `http://linkinghub.elsevier.com/retrieve/pii/S1389128604003457`. (Cited on page 16.)

[30] M. Afanasyev, Tsuwei Chen, G.M. Voelker, and A.C. Snoeren. Usage patterns in an urban WiFi network. *Networking, IEEE/ACM Transactions on*, 18(5):1359 –1372, October 2010. ISSN 1063-6692. doi: 10.1109/TNET.2010.2040087. (Cited on page 16.)

[31] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11B mesh network. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, MobiCom '05, page 31–42, New York, NY, USA, 2005. ACM. ISBN 1-59593-020-5. doi: 10.1145/1080829.1080833. URL `http://doi.acm.org/10.1145/1080829.1080833`. (Cited on page 16.)

[32] Dambar Raj Paudel, Kazuhiko Sato, B. P. Gautam, and D. Shrestha. Design and implementation of partial mesh community wireless network in himalayan region. In *Internet (AH-ICI), 2012 Third Asian Himalayas International Conference on*, page 1–6. IEEE, 2012. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6408447`. (Cited on page 16.)

[33] D.G. Narayan, K. Sugnani, A. Raichur, and M. Uma. A cross layer routing metric for gateway aware routing in wireless mesh network. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–6, July 2013. doi: 10.1109/ICCCNT.2013.6726735. (Cited on page 17.)

[34] M.E.M. Campista, P.M. Esposito, I.M. Moraes, L.H.M.K. Costa, O.C.M.B. Duarte, D.G. Passos, C.V.N. de Albuquerque, D.C.M. Saade, and M.G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1):6–12, January 2008. ISSN 0890-8044. doi: 10.1109/MNET.2008.4435897. (Cited on pages 18 and 19.)

[35] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003. URL `http://www.ietf.org/rfc/rfc3626.txt`. (Cited on pages 18, 25, 92, 136, and 183.)

[36] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg. The Optimized Link State Routing Protocol Version 2. RFC 7181 (Proposed Standard), April 2014. URL `http://www.ietf.org/rfc/rfc7181.txt`. Updated by RFCs 7183, 7187, 7188. (Cited on pages 18 and 76.)

[37] Y. Faheem and J.L. Rougier. Loop avoidance for fish-eye OLSR in sparse wireless mesh networks. In *Sixth International Conference on Wireless On-Demand Network Systems and Services, 2009. WONS 2009*, pages 231–234, 2009. doi: 10.1109/WONS.2009.4801855. (Cited on page 19.)

[38] R. S Couto, M. E.M Campista, L. H.M.K Costa, and O. C.M.B Duarte. An experimental analysis of routing inconsistency in indoor wireless mesh networks. In *2011 IEEE Symposium on Computers and Communications (ISCC)*,

pages 609–614. IEEE, June 2011. ISBN 978-1-4577-0680-6. doi: 10.1109/ISCC.2011.5983905. (Cited on page 19.)

[39] A. Klein, L. Braun, and F. Oehlmann. Performance study of the better approach to mobile adhoc networking (B.A.T.M.A.N.) protocol in the context of asymmetric links. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–7, June 2012. doi: 10.1109/WoWMoM.2012.6263783. (Cited on page 19.)

[40] M. Abolhasan, B. Hagelstein, and J. C P Wang. Real-world performance of current proactive multi-hop mesh protocols. In *15th Asia-Pacific Conference on Communications, 2009. APCC 2009*, pages 44–47, October 2009. doi: 10.1109/APCC.2009.5375690. (Cited on pages 19 and 178.)

[41] Krishna N. Ramach, Ran Milind M. Buddhikot, Girish Ch, Scott Miller, Elizabeth M. Belding-royer, and Kevin C. Almeroth. On the design and implementation of infrastructure mesh networks. Technical report, in IEEE Workshop on Wireless Mesh Networks (WiMesh, 2005. (Cited on page 19.)

[42] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003. URL `http://www.ietf.org/rfc/rfc3561.txt`. (Cited on pages 20 and 178.)

[43] IEEE standard for information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 10: Mesh networking. *IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11w-2009, IEEE 802.11n-2009, IEEE 802.11p-2010, IEEE 802.11z-2010, IEEE 802.11v-2011, and IEEE 802.11u-2011)*, pages 1–372, September 2011. doi: 10.1109/IEEESTD.2011.6018236. (Cited on page 20.)

[44] W. Cordeiro, E. Aguiar, W. Moreira, A. Abelem, and M. Stanton. Providing quality of service for mesh networks using link delay measurements. In *Pro-*

ceedings of 16th International Conference on Computer Communications and Networks, 2007. ICCCN 2007, pages 991–996, August 2007. doi: 10.1109/ICCCN.2007.4317947. (Cited on page 21.)

[45] Andreas Lavén and Peter Hjärtquist. Multimetric olsr and ett. In *5th OLSR Interop & Workshop*, 2009. URL `http://interop.thomasclausen.org/Interop09/Papers/Papers/MultimetricOLSR_and_ETT-1.pdf`. (Cited on pages 21, 22, 76, and 92.)

[46] Ling-Jyh Chen, Tony Sun, Guang Yang, M. Y. Sanadidi, and Mario Gerla. AdHoc probe: end-to-end capacity probing in wireless ad hoc networks. *Wireless Networks*, 15(1):111–126, January 2009. ISSN 1022-0038, 1572-8196. doi: 10.1007/s11276-007-0047-4. URL `http://link.springer.com/article/10.1007/s11276-007-0047-4`. (Cited on page 21.)

[47] G. Venkatesh and Kuang-Ching Wang. Estimation of maximum achievable end-to-end throughput in IEEE 802.11 based wireless mesh networks. In *IEEE 34th Conference on Local Computer Networks, 2009. LCN 2009*, pages 1110–1117, October 2009. doi: 10.1109/LCN.2009.5355214. (Cited on page 21.)

[48] Gang Yuan, Yunqing Chen, Yueming Wei, and Shizhong Nie. A distributable traffic - based MPLS dynamic load balancing scheme. In *2005 Asia-Pacific Conference on Communications*, pages 684–689. IEEE, October 2005. ISBN 0-7803-9132-2. doi: 10.1109/APCC.2005.1554149. (Cited on pages 22, 33, and 91.)

[49] Seongkwan Kim, Okhwan Lee, Sunghyun Choi, and Sung-Ju Lee. Comparative analysis of link quality metrics and routing protocols for optimal route construction in wireless mesh networks. *Ad Hoc Networks*, 9(7):1343–1358, September 2011. ISSN 1570-8705. doi: 10.1016/j.adhoc.2011.03.011. URL `http://www.sciencedirect.com/science/article/pii/S1570870511000746`. (Cited on page 23.)

[50] P. M Esposito, M. Campista, I. M Moraes, L. Costa, O. Duarte, and M. G Rubinstein. Implementing the expected transmission time metric for OLSR wireless mesh networks. In *Wireless Days, 2008. WD '08. 1st IFIP*, pages 1–5. IEEE,

November 2008. ISBN 978-1-4244-2828-1. doi: 10.1109/WD.2008.4812900. (Cited on pages 23 and 76.)

[51] Yan-jing Sun, Xue Liu, and Bei-bei Zhang. An available transmission time routing metric for wireless ad-hoc sensor networks. 2011:1–6. ISSN 1550-1329, 1550-1477. doi: 10.1155/2011/815987. URL `http://www.hindawi.com/journals/ijdsn/2011/815987/`. (Cited on pages 23 and 66.)

[52] S. S Ahmeda and E. A Esseid. Review of routing metrics and protocols for wireless mesh network. In *2010 Second Pacific-Asia Conference on Circuits,Communications and System (PACCS)*, volume 1, pages 27–30. IEEE, August 2010. ISBN 978-1-4244-7969-6. doi: 10.1109/PACCS.2010.5626819. (Cited on page 24.)

[53] A. Kumar, P. Kumar, and M.P. Singh. A survey on wireless LAN internal roaming protocols. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–6, July 2013. doi: 10.1109/ICCCNT.2013.6726785. (Cited on page 25.)

[54] IEEE standard for information technology– telecommunications and information exchange between systemsLocal and metropolitan area networks– specific requirements–Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) Specifications–Amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz. *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)*, pages 1–425, December 2013. doi: 10.1109/IEEESTD.2013.6687187. (Cited on pages 26 and 28.)

[55] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages 1–1076, June 2007. doi: 10.1109/IEEESTD.2007.373646. (Cited on pages 26 and 28.)

[56] Laurence T. Yang, Agustinus Borgy Waluyo, Jianhua, Ling Tan, and Bala Srinivasan. Mobile intelligence. In *Mobile Intelligence*. John Wiley & Sons, Inc., 2010. ISBN 9780470579398. URL `http://onlinelibrary.wiley.com/doi/10.1002/9780470579398.fmatter/summary`. (Cited on page 26.)

[57] Wikipedia. Carrier sense multiple access with collision avoidance, 2013. URL `http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_avoidance`. Online; accessed 05-April-2014. (Cited on page 27.)

[58] Christian Köbel, Walter Baluja García, and Joachim Habermann. A survey on multi-channel based, digital media- driven 802.11 wireless mesh networks. pages 169–175, June 2011. ISBN 978-1-61208-140-3. URL `http://www.thinkmind.org/index.php?view=article&articleid=icwmc_2011_8_20_20067`. (Cited on pages 26, 27, 30, and 211.)

[59] Yu Cheng, Hongkun Li, Peng-Jun Wan, and Xinbing Wang. Wireless mesh network capacity achievable over the CSMA/CA MAC. *IEEE Transactions on Vehicular Technology*, 61(7):3151–3165, 2012. ISSN 0018-9545. doi: 10.1109/TVT.2012.2204411. (Cited on page 27.)

[60] F. Birlik, O. Gurbuz, and O. Ercetin. IPTVhome networking via 802.11 wireless mesh networks: an implementation experience. *IEEE Transactions on Consumer Electronics*, 55(3):1192 –1199, August 2009. ISSN 0098-3063. doi: 10.1109/TCE.2009.5277975. (Cited on page 27.)

[61] IEEE standard for information technology–Telecommunications and information exchange between systems local and metropolitan area networks–Specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 2: MAC enhancements for robust audio video streaming. *IEEE Std 802.11aa-2012 (Amendment to IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012)*, pages 1–162, May 2012. doi: 10.1109/IEEESTD.2012.6204193. (Cited on page 28.)

[62] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. IEEE 802.11 rate adaptation: a practical approach. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, page 126–134. ACM, 2004. URL `http://dl.acm.org/citation.cfm?id=1023687`. (Cited on pages 28 and 205.)

[63] Chenzhe Zhang, K. Kowalik, and M. Davis. An experimental study of the impact of using multi-radio in WLAN mesh networks. In *5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09*, pages 1–4, September 2009. doi: 10.1109/WICOM.2009.5303103. (Cited on page 29.)

[64] R. Lasowski, C. Scheuermann, F. Gschwandtner, and C. Linnhoff-Popien. Evaluation of adjacent channel interference in single radio vehicular ad-hoc networks. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 267 –271, January 2011. doi: 10.1109/CCNC.2011.5766470. (Cited on page 29.)

[65] Paul Fuxjager, Danilo Valerio, and Fabio Ricciato. The myth of non-overlapping channels: interference measurements in ieee 802.11. In *Wireless on Demand Network Systems and Services, 2007. WONS'07. Fourth Annual Conference on*, page 1–8. IEEE, 2007. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4142700`. (Cited on page 29.)

[66] Sebastian Robitzsch, John Fitzpatrick, Seán Murphy, and Liam Murphy. Behind-the-scenes of IEEE 802.11a based multi-radio mesh networks: a measurement driven evaluation of inter-channel interference. ISSN 1942-2652. URL `http://e-archivo.uc3m.es/handle/10016/13697`. (Cited on page 29.)

[67] V. Angelakis, S. Papadakis, V.A. Siris, and A. Traganitis. Adjacent channel interference in 802.11a is harmful: Testbed validation of a simple quantification model. 49(3):160–166. ISSN 0163-6804. doi: 10.1109/MCOM.2011.5723815. (Cited on page 29.)

[68] Joshua Robinson, Konstantina Papagiannaki, Christophe Diot, Xingang Guo, and Lakshman Krishnamurthy. Experimenting with a multi-radio mesh networking testbed. In *in (WiNMee Workshop*, 2005. (Cited on page 29.)

[69] Chen-Mou Cheng, Pai-Hsiang Hsiao, H. T. Kung, and D. Vlah. WSN07-1: adjacent channel interference in dual-radio 802.11a nodes and its impact on multihop networking. In *IEEE Global Telecommunications Conference, 2006. GLOBECOM '06*, pages 1–6, November 2006. doi: 10.1109/GLOCOM.2006.500. (Cited on page 29.)

[70] D. Grossman. New Terminology and Clarifications for Diffserv. RFC 3260 (Informational), April 2002. URL `http://www.ietf.org/rfc/rfc3260.txt`. (Cited on page 30.)

[71] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001. URL `http://www.ietf.org/rfc/rfc3031.txt`. Updated by RFCs 6178, 6790. (Cited on pages 31, 66, and 69.)

[72] L. Andersson, I. Minei, and B. Thomas. LDP Specification. RFC 5036 (Draft Standard), October 2007. URL `http://www.ietf.org/rfc/rfc5036.txt`. Updated by RFCs 6720, 6790. (Cited on page 32.)

[73] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209 (Proposed Standard), December 2001. URL `http://www.ietf.org/rfc/rfc3209.txt`. Updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711, 6780, 6790. (Cited on page 32.)

[74] L. Romdhani and C. Bonnet. Cross-layer QoS routing framework for wireless mesh networks. In *The Fourth International Conference on Wireless and Mobile Communications, 2008. ICWMC '08*, pages 382–388, 2008. doi: 10.1109/ICWMC.2008.70. (Cited on page 32.)

[75] A. Acharya, Sachin Ganu, and A. Misra. DCMA: a label switching MAC for efficient packet forwarding in multihop wireless networks. *IEEE Journal*

on Selected Areas in Communications, 24(11):1995–2004, November 2006. ISSN 0733-8716. doi: 10.1109/JSAC.2006.881636. (Cited on page 32.)

[76] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, 1994. ISSN 1063-6692. doi: 10.1109/90.298432. (Cited on page 33.)

[77] R. Bless, K. Nichols, and K. Wehrle. A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services. RFC 3662 (Informational), December 2003. URL `http://www.ietf.org/rfc/rfc3662.txt`. (Cited on page 33.)

[78] PengHou Ho, D.W. Holtby, KimChuan Lim, KaeHsiang Kwong, D. Chieng, A. Ting, and SuFong Chien. End-to-end throughput and delay analysis of WiFi multi-hop network with deterministic offered load. In *IET International Conference on Wireless Communications and Applications (ICWCA 2012)*, pages 1–6, October 2012. doi: 10.1049/cp.2012.2083. (Cited on pages 33 and 108.)

[79] Deutsche Telekom AG. Mehr power durch intelligente bündelung, January 2014. URL `http://www.telekom.com/innovation/netz-der-zukunft/212446`. (Cited on page 33.)

[80] A. Delphinanto, T. Koonen, and F. den Hartog. End-to-end available bandwidth probing in heterogeneous IP home networks. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 431 –435, January 2011. doi: 10.1109/CCNC.2011.5766506. (Cited on page 33.)

[81] Pablo Rodriguez, Rajiv Chakravorty, Julian Chesterfield, Ian Pratt, and Suman Banerjee. Mar: A commuter router infrastructure for the mobile internet. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, page 217–230. ACM, 2004. URL `http://dl.acm.org/citation.cfm?id=990091`. (Cited on page 33.)

[82] C. Hornig. A Standard for the Transmission of IP Datagrams over Ethernet Networks. RFC 894 (INTERNET STANDARD), April 1984. URL `http://www.ietf.org/rfc/rfc894.txt`. (Cited on page 34.)

[83] N. Bayer, D. Hock, A. Roos, M. Siebert, B. Xu, V. Rakocevic, and J. Habermann. VoIP performance in "MeshBed" - a wireless mesh networks testbed. In *IEEE Vehicular Technology Conference, 2008. VTC Spring 2008*, pages 2218–2222, May 2008. doi: 10.1109/VETECS.2008.494. (Cited on page 35.)

[84] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato. On load distribution over multipath networks. *IEEE Communications Surveys Tutorials*, 14(3):662 – 680, 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.082511.00013. (Cited on page 35.)

[85] X.H. Wang, M. Iqbal, and X. Zhou. Design and implementation of a dual-radio wireless mesh network testbed for healthcare. In *International Conference on Information Technology and Applications in Biomedicine, 2008. ITAB 2008*, pages 300–304, . doi: 10.1109/ITAB.2008.4570647. (Cited on pages 35 and 96.)

[86] G.E.M. Zhioua and N. Tabbane. A load and QoS aware scheduling algorithm for multi-channel multi-radio wireless mesh networks. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2043–2047, 2012. doi: 10.1109/WCNC.2012.6214126. (Cited on page 35.)

[87] A. Banchs, N. Bayer, D. Chieng, A. De La Oliva, B. Gloss, M. Kretschme, S. Murphy, M. Natkaniec, and F. Zdarsky. CARMEN: delivering carrier grade services over wireless mesh networks. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008*, pages 1–6, September 2008. doi: 10.1109/PIMRC.2008.4699961. (Cited on pages 36 and 110.)

[88] R. Tahar, A. Belghith, and R. Braham. A generic mobile node architecture for multi-interface heterogenous wireless link layer. In *Network of the Future (NOF), 2012 Third International Conference on the*, pages 1–5, 2012. doi: 10.1109/NOF.2012.6463999. (Cited on page 36.)

[89] V.A. Siris, E.Z. Tragos, and N.E. Petroulakis. Experiences with a metropolitan multiradio wireless mesh network: design, performance, and application. *IEEE Communications Magazine*, 50(7):128–136, July 2012. ISSN 0163-6804. doi: 10.1109/MCOM.2012.6231289. (Cited on pages 36 and 97.)

[90] Yan He Bin Huang. Investigating deployment strategies for multi-radio multi-channel residential wireless mesh networks. pages 147–153. doi: 10.1109/WiMob.2009.34. (Cited on pages 36 and 97.)

[91] S. Robitzsch, S. Murphy, and P. Szczechowiak. Architecture of a self configuration framework for 802.11-based multi-radio mesh networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–8, 2013. doi: 10.1109/WoWMoM.2013.6583476. (Cited on page 36.)

[92] Dongsheng Wang, Pin Lv, Yingwen Chen, and Ming Xu. POCAM: partially overlapped channel assignment in multi-radio multi-channel wireless mesh network. In *2011 11th International Symposium on Communications and Information Technologies (ISCIT)*, pages 188–193, October 2011. doi: 10.1109/ISCIT.2011.6089727. (Cited on page 36.)

[93] Yi Qu, Chung-Horng Lung, and A. Srinivasan. Network layer negotiation-based channel assignment in multi-channel wireless networks. In *IEEE Global Telecommunications Conference, 2007. GLOBECOM '07*, pages 759–763, November 2007. doi: 10.1109/GLOCOM.2007.147. (Cited on page 37.)

[94] H. A Mogaibel and M. Othman. Review of routing protocols and it's metrics for wireless mesh networks. In *International Association of Computer Science and Information Technology - Spring Conference, 2009. IACSITSC '09*, pages 62–70. IEEE, April 2009. ISBN 978-0-7695-3653-8. doi: 10.1109/IACSIT-SC.2009.51. (Cited on page 37.)

[95] Yenisleidy Fernández Romero. Propuesta de algoritmo de distribución dinámica de canales basado en prioridades para redes malladas inalámbricas multi-interfaces y multi-canales. Master thesis, Instituto Superior Politécnico

José Antonio Echeverría, Facultad de Ingeniería Eléctrica, Departamento de Telecomunicaciones y Electrónica, 2013. (Cited on pages 37 and 58.)

[96] A. Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 2223 – 2234 vol. 3, March 2005. doi: 10.1109/INFCOM.2005.1498497. (Cited on page 37.)

[97] Heywoong Kim, Qijun Gu, Meng Yu, Wangyu Zang, and Peng Liu. A simulation framework for performance analysis of multi-interface and multi-channel wireless networks in INET/OMNET++. In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10, page 101:1–101:8, Orlando, Florida, 2010. ACM. ISBN 978-1-4503-0069-8. doi: 10.1145/1878537.1878643. ACM ID: 1878643. (Cited on pages 38 and 192.)

[98] Ho Ting Cheng. *Radio Resource Management for Wireless Mesh Networks Supporting Heterogeneous Traffic*. Thesis for the degree of doctor of philosophy, University of Waterloo, 2009. (Cited on page 38.)

[99] Christian Köbel, Walter Baluja García, Joachim Habermann, Nico Bayer, and Dimitry Sivchenko. Dynamic channel bundling in 802.11a based media-transport mesh networks. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 979–980, 2011. doi: 10.1109/CCNC.2011.5766659. (Cited on page 38.)

[100] Christian Köbel, Walter Baluja García, and Joachim Habermann. An ethernet bonding inspired architecture for commercial multi-interface ad-hoc networks. In *2012 IEEE 1st Global Conference on Consumer Electronics (GCCE)*, pages 339–340, 2012. doi: 10.1109/GCCE.2012.6379620. (Cited on page 38.)

[101] Christian Köbel, Walter Baluja García, and Joachim Habermann. Sistema de balance de carga para redes malladas inalámbricas multi-interfaces. *Ingeniería Electrónica, Automática y Comunicaciones*, 33(3):85–98, December 2012. ISSN

1815-5928. doi: 10.1234/rielac.v33i3.140. URL `http://rielac.cujae.edu.cu/index.php/rieac/article/view/140`. (Cited on page 38.)

[102] Sung-Ho Kim and Young-Bae Ko. Wireless bonding for maximizing throughput in multi-radio mesh networks. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07*, pages 570–576. IEEE, March 2007. ISBN 0-7695-2788-4. doi: 10.1109/PERCOMW.2007.126. (Cited on page 38.)

[103] Yun Hu. Parallel-transmission: A new usage of multi-radio diversity in wireless mesh network. *Int'l J. of Communications, Network and System Sciences*, 02 (01):51–57, 2009. ISSN 1913-3715, 1913-3723. doi: 10.4236/ijcns.2009.21006. URL `http://www.scirp.org/journal/PaperInformation.aspx?paperID=182`. (Cited on page 39.)

[104] Hong-Ning Dai, Kam-Wing Ng, R.C.-W. Wong, and Min-You Wu+. On the capacity of multi-channel wireless networks using directional antennas. In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, pages 628 –636, April 2008. doi: 10.1109/INFOCOM.2008.112. (Cited on page 46.)

[105] A. Zimmermann, V. Bohme, and A. Hannemann. IP address assignment in wireless mesh networks. In *33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008*, pages 44–51, October 2008. doi: 10.1109/LCN.2008.4664150. (Cited on page 51.)

[106] L. J. García Villalba, A. L. Sandoval Orozco, J. García Matesanz, and T.-H. Kim. E-D2HCP: enhanced distributed dynamic host configuration protocol. *Computing*, pages 1–15, February 2013. ISSN 0010-485X, 1436-5057. doi: 10.1007/s00607-013-0307-3. URL `http://link.springer.com/article/10.1007/s00607-013-0307-3`. (Cited on page 51.)

[107] E. Aryafar, O. Gurewitz, and E.W. Knightly. Distance-1 constrained channel assignment in single radio wireless mesh networks. In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, pages –, April 2008. doi: 10.1109/INFOCOM.2008.127. (Cited on page 57.)

[108] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (INTERNET STANDARD), October 1989. URL `http://www.ietf.org/rfc/rfc1122.txt`. Updated by RFCs 1349, 4379, 5884, 6093, 6298, 6633, 6864. (Cited on page 60.)

[109] Andrew Tanenbaum and David Wetherall. *Computer Networks*. Pearson, 5th (revised, german) edition, 2012. ISBN 978-3-86894-137-1. (Cited on pages 66 and 67.)

[110] Jeffrey Mogul and Steve Deering. Path mtu discovery. RFC 1191, RFC Editor, November 1990. URL `http://www.rfc-editor.org/rfc/rfc1191.txt`. `http://www.rfc-editor.org/rfc/rfc1191.txt`. (Cited on page 67.)

[111] Archi Delphinanto, Ton Koonen, Shuang Zhang, and Frank T. H. den Hartog. Path capacity estimation in heterogeneous, best-effort, small-scale ip networks. In *LCN'10*, pages 1076–1083, 2010. (Cited on page 67.)

[112] Jack McCann, Stephen E. Deering, and Jeffrey Mogul. Path mtu discovery for ip version 6. RFC 1981, RFC Editor, August 1996. URL `http://www.rfc-editor.org/rfc/rfc1981.txt`. `http://www.rfc-editor.org/rfc/rfc1981.txt`. (Cited on page 67.)

[113] A. Ariza-Quintana, Eduardo Casilari, and A. Triviño Cabrera. An architecture for the implementation of mesh networks in OMNeT++. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, page 72. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). URL `http://dl.acm.org/citation.cfm?id=1537706`. (Cited on pages 75, 79, 132, and 190.)

[114] J. Postel. Internet Protocol. RFC 791 (INTERNET STANDARD), September 1981. URL `http://www.ietf.org/rfc/rfc791.txt`. Updated by RFCs 1349, 2474, 6864. (Cited on pages 77 and 101.)

[115] Richard Chirgwin. Multipath tcp: Siri's new toy isn't a game-changer, October 2013. URL `http://www.theregister.co.uk/2013/10/01/multipath_tcp_siris_toy_isnt_a_gamechanger/`. (Cited on page 81.)

[116] A. Kostopoulos, H. Warma, T. Leva, B. Heinrich, A. Ford, and L. Eggert. Towards multipath TCP adoption: Challenges and opportunities. In *2010 6th EURO-NF Conference on Next Generation Internet (NGI)*, pages 1–8. doi: 10.1109/NGI.2010.5534465. (Cited on page 81.)

[117] Ashwin Rao, Arnaud Legout, Yeon-sup Lim, Don Towsley, Chadi Barakat, and Walid Dabbous. Network characteristics of video streaming traffic. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, page 25:1–25:12, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1041-3. doi: 10.1145/2079296.2079321. URL `http://doi.acm.org/10.1145/2079296.2079321`. (Cited on page 89.)

[118] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC 2309 (Informational), April 1998. URL `http://www.ietf.org/rfc/rfc2309.txt`. (Cited on page 89.)

[119] Hao Zhu and Guohong Cao. On improving service differentiation under bursty data traffic in wireless networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 871–881 vol.2, March 2004. doi: 10.1109/INFCOM.2004.1356975. (Cited on page 90.)

[120] K. Knightson, N. Morita, and T. Towle. NGN architecture: generic principles, functional architecture, and implementation. *IEEE Communications Magazine*, 43(10):49–56, October 2005. ISSN 0163-6804. doi: 10.1109/MCOM.2005.1522124. (Cited on page 91.)

[121] B. Pinheiro, V. Nascimento, W. Moreira, and A. Abelem. Abaré: A deployment and management framework for wireless mesh network. In *IEEE Latin-*

American Conference on Communications, 2009. LATINCOM '09, pages 1–6. doi: 10.1109/LATINCOM.2009.5305120. (Cited on page 96.)

[122] Ying-Chen Chen, Chi-Cheng Chuang, Ray-I. Chang, Jia-Shian Lin, and Te-Chih Wang. Integrated wireless access point architecture for wireless sensor networks. In *11th International Conference on Advanced Communication Technology, 2009. ICACT 2009*, volume 01, pages 713–718. (Cited on page 96.)

[123] Krishna Ramachandran, Irfan Sheriff, Elizabeth M. Belding, and Kevin C. Almeroth. A multi-radio 802.11 mesh network architecture. 13(1):132–146. ISSN 1383-469X. doi: 10.1007/s11036-008-0026-8. URL `http://dx.doi.org/10.1007/s11036-008-0026-8`. (Cited on page 96.)

[124] P. Gardner-Stephen, R. Challans, J. Lakeman, A. Bettison, D. Gardner-Stephen, and M. Lloyd. The serval mesh: A platform for resilient communications in disaster amp; crisis. In *2013 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 162–166. doi: 10.1109/GHTC.2013.6713674. (Cited on page 96.)

[125] Aruba Networks. Multiservices mesh networks for the great outdoors, 2015. URL `http://www.arubanetworks.com/products/networking/outdoor-mesh/`. (Cited on page 96.)

[126] The Serval Project. Serval mesh extender, 2015. URL `http://developer.servalproject.org/dokuwiki/doku.php?id=content:meshextender:main_page`. (Cited on page 97.)

[127] OpenWrt Wireless Freedom. Table of hardware, 2015. URL `http://wiki.openwrt.org/toh/start`. (Cited on page 97.)

[128] HSMM-Pi. A set of tools to easily configure the raspberry pi to function as a high-speed multimedia (hsmm) wireless node, 2015. URL `https://github.com/urlgrey/hsmm-pi`. (Cited on page 97.)

[129] N. Salvador, V. Filipe, C. Rabadao, and A. Pereira. Management model for wireless broadband networks. In *3rd International Conference on Systems and*

Networks Communications, 2008. ICSNC '08, pages 38–43. doi: 10.1109/ICSNC.2008.69. (Cited on page 97.)

[130] INET Developer Community. Inet package description of simple module udpbasicburst, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.udpapp.UDPBasicBurst.html. (Cited on page 106.)

[131] INET Developer Community. Inet package description of simple module udpsink, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.udpapp.UDPSink.html. (Cited on page 106.)

[132] INET Developer Community. Inet package description of simple module tcpgenericsrvapp, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.tcpapp.TCPGenericSrvApp.html. (Cited on page 113.)

[133] INET Developer Community. Inet package description of simple module tcpbasicclientapp, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.tcpapp.TCPBasicClientApp.html. (Cited on page 113.)

[134] Minseok Oh. An experimental comparison of packet delivery schemes in a linear mesh topology. In *Information Science and Applications (ICISA), 2012 International Conference on*, pages 1–4. IEEE. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6220934. (Cited on page 131.)

[135] Michael Bredel, Zdravko Bozakov, and Yuming Jiang. Analyzing router performance using network calculus with external measurements. In *Quality of Service (IWQoS), 2010 18th International Workshop on*, pages 1–9. IEEE. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5542749. (Cited on page 132.)

[136] Josias Lima, Thiago Rodrigues, Rodrigo Melo, Gregorio Correia, Judith Kelner, and Eduardo Feitosa. On the tuning of wireless heterogeneous routing. In *2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio*

Communications (PIMRC), pages 3391–3396. doi: 10.1109/PIMRC.2013.6666734. (Cited on page 133.)

[137] J. Chroboczek. The Babel Routing Protocol. RFC 6126 (Experimental), April 2011. URL `http://www.ietf.org/rfc/rfc6126.txt`. (Cited on page 178.)

[138] D. Murray, M. Dixon, and T. Koziniec. An experimental comparison of routing protocols in multi hop ad hoc networks. In *Telecommunication Networks and Applications Conference (ATNAC), 2010 Australasian*, pages 159–164, October 2010. doi: 10.1109/ATNAC.2010.5680190. (Cited on page 178.)

[139] E.Z. Tragos, S. Zeadally, A.G. Fragkiadakis, and V.A. Siris. Spectrum assignment in cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 15(3):1108–1135, 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.121112.00047. (Cited on page 181.)

[140] Pingshi He and Ziping Xu. Channel assignment and routing in multi-channel, multi-interface wireless mesh networks. In *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, volume 3, pages V3–69–V3–74. IEEE, April 2010. ISBN 978-1-4244-6347-3. doi: 10.1109/ICCET.2010.5485774. (Cited on page 182.)

[141] KwaLinux Trainingen. A real visualization of your mesh network topology, July 2010. URL `http://www.kwalinux.nl/what-a-mesh/1129/`. (Cited on page 183.)

[142] Andreas Tønnesen. Impementing and extending the optimized link state routing protocol. Master thesis, UniK University Graduate Center, University of Oslo, 2004. (Cited on page 184.)

[143] T. Clausen, C. Dearlove, J. Dean, and C. Adjih. Generalized mobile ad hoc network (manet) packet/message format. RFC 5444, RFC Editor, February 2009. URL `http://www.rfc-editor.org/rfc/rfc5444.txt`. (Cited on page 184.)

[144] J. Karlsson, P. Hurtig, A. Brunstrom, A. Kassler, and G. Di Stasi. Impact of multi-path routing on TCP performance. In *World of Wireless, Mobile and Mul-*

timedia Networks (WoWMoM), 2012 IEEE International Symposium on a, pages 1–3, June 2012. doi: 10.1109/WoWMoM.2012.6263723. (Cited on page 186.)

[145] M.X. Gong, Shiwen Mao, and S.F. Midkiff. Load- and interference-aware channel assignment for dual-radio mesh backhauls. In *IEEE Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008*, pages 1–6, November 2008. doi: 10.1109/GLOCOM.2008.ECP.967. (Cited on page 186.)

[146] J. Karlsson, P. Hurtig, A. Brunstrom, A. Kassler, and G. Di Stasi. The interaction between TCP reordering mechanisms and multi-path forwarding in wireless mesh networks. In *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 276–283, October 2012. doi: 10.1109/WiMOB.2012.6379088. (Cited on page 186.)

[147] I. Kozintsev and J. M. Veigh. Improving last-hop multicast streaming video over 802.11. *Proceedings of BROADNETS, San Jose, California, USA*, 38, 2004. URL `http://pdf.aminer.org/000/279/378/an_error_control_scheme_for_multicast_video_streaming_on_the.pdf`. (Cited on page 190.)

[148] J. Postel. Transmission Control Protocol. RFC 793 (INTERNET STANDARD), September 1981. URL `http://www.ietf.org/rfc/rfc793.txt`. Updated by RFCs 1122, 3168, 6093, 6528. (Cited on pages 191 and 203.)

[149] J. Postel. User Datagram Protocol. RFC 768 (INTERNET STANDARD), August 1980. URL `http://www.ietf.org/rfc/rfc768.txt`. (Cited on page 191.)

[150] INET Developer Community. Inet package description of simple module channelcontrol, . URL `http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.world.radio.ChannelControl.html`. (Cited on page 201.)

[151] INET Developer Community. Inet package description of simple module ieee80211radio, . URL `http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.linklayer.ieee80211.radio.Ieee80211Radio.html`. (Cited on page 201.)

[152] INET Developer Community. Inet package description of compound module ieee80211nic, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.linklayer.ieee80211.Ieee80211Nic.html. (Cited on page 201.)

[153] INET Developer Community. Inet package description of simple module ieee80211mac, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.linklayer.ieee80211.mac.Ieee80211Mac.html. (Cited on page 205.)

[154] INET Developer Community. Inet package description of simple module ieee80211mgmtadhoc, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.linklayer.ieee80211.mgmt.Ieee80211MgmtAdhoc.html. (Cited on page 205.)

[155] INET Developer Community. Inet package description of simple module tcp, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.transport.tcp.TCP.html. (Cited on page 208.)

[156] INET Developer Community. Inet package description of simple module tcpsessionapp, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.tcpapp.TCPSessionApp.html. (Cited on page 208.)

[157] INET Developer Community. Inet package description of simple module tcpsinkapp, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.tcpapp.TCPSinkApp.html. (Cited on page 208.)

[158] INET Developer Community. Inet package description of simple module udpvideostreamsvr, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.udpapp.UDPVideoStreamSvr.html. (Cited on page 209.)

[159] INET Developer Community. Inet package description of simple module udpvideostreamcli, . URL http://inet.omnetpp.org/doc/INET/neddoc/

index.html?p=inet.applications.udpapp.UDPVideoStreamCli.html. (Cited on page 209.)

[160] INET Developer Community. Inet package description of simple module pingapp, . URL http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.applications.pingapp.PingApp.html. (Cited on page 209.)

ACRONYMS

AARF	Adaptive Automatic Rate Fallback
ACI	Adjacent Channel Interference
ACK	Acknowledgments
AF	Assured Forwarding
AODV	Ad-hoc On-demand Distance Vector
AODV-ST	AODV Spanning Tree
AP	Access Point
ARF	Automatic Rate Fallback
ARP	Address Resolution Protocol
B.A.T.M.A.N.	Better Approach To Mobile Adhoc Networking
BGP	Border Gateway Protocol
BMT	Bundle Management Table
CA	Channel Assignment
CBR	Constant Bit Rate
CC	Control Channel
CCTV	Closed Circuit Television
CFT	Class Flow Table
CPU	Central Processing Unit
CSC	Channel Switching Cost

CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSS	Cascading Style Sheets
CT	Cross Traffic
CTS	Clear to Send
DCDbP	Dynamic Channel Distribution based on Priorities
DCF	Distributed Control Function
DCMA	Data-driven Cut-through Medium Access
DHCP	Dynamic Host Configuration Protocol
DLL	Data Link Layer
DNS	Domain Name System
DPI	Deep Packet Inspection
DRS	Dynamic Rate Shifting
DS	DiffServ
DSCP	DiffServ Code Point
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
DST	Destination
DWCP	Dynamic Wireless Mesh Network Configuration Protocol
E-D2HCP	Extended Distributed DHCP
ECN	Explicit Congestion Notification
EDCA	Enhanced Distributed Coordination Access
EF	Expedited Forwarding

EGP	Exterior Gateway Protocol
ESSID	Extended Service Set ID
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FB	Fallback
FDMA	Frequency-Division Multiple Access
FEC	Forwarding Equivalence Class
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GW	Gateway
HDD	Hard Disk Drive
HCF	Hybrid Control Function
HNA	Host and Network Association
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HWMP	Hybrid Wireless Mesh Protocol
ICI	Inter-Carrier Interference
ICMP	Internet Control Message Protocol
IF	Interface
IP	Internet Protocol
IPsec	IP Security
IRU	Interference-aware Resource Usage

ISP	Internet Service Provider
ITC	Internal Traffic Class
LAN	Local Area Network
LB	Load Balancing
LDP	Label Distribution Protocol
LLC	Logical Link Control
LMHPC	Label-based Multi-Hop Packet Commutation
LQSR	Link Quality Source Routing
LSP	Label Switched Path
LSR	Label Switched Routers
LTE	Long Term Evolution
M-FIB	Mesh Forwarding Information Base
M-LDM	Mesh Label Distribution Message
M-LDP	Mesh Label Distribution Protocol
M-LFIB	Mesh Label Forwarding Instance Base
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MAR	Mobile Access Router
MHRRM	Multi-Hop Radio Resource Management
MIC	Metric of Interference and Channel Switching
MID	Multiple Interface Declaration
MIMC	Multi-Interface Multi-Channel

MIMO	Multiple Input Multiple Output
MPLS	Multi-Protocol Label Switching
MPR	Multi-Point Relaying
MR	Mesh Router
MR-LQSR	Multi-Radio LQSR
MSS	Maximum Segment Size
MTCP	Multipath Transmission Control Protocol
MTU	Maximum Transfer Unit
NAT	Network Address Translation
NCA	Negotiation-based Channel Assignment
NGO	Non-Governmental Organization
NHF	Next-Hop Field
NIC	Network Interface Card
OFDM	Orthogonal Frequency-Division Multiplexing
OLSR	Optimized Link State Routing
OGM	Originator Messages
OS	Operating System
OSI	Open Systems Interconnection
PCF	Point Coordination Function
PER	Packet Error Rate
PFIFO	Packet First In - First Out
PHB	Per Hop Behaviors

PHY	Physical
PLR	Packet Loss Rate
PMTUD	Path MTU Discovery
PPPoE	Point-to-Point Protocol over Ethernet
PS	Packet Scheduling
QAM	Quadrature Amplitude Modulation
QoE	Quality-of-Experience
QoS	Quality-of-Service
QSF	Queue Selector Field
RAM	Random-Access Memory
RCA	Receiver-based Channel Assignment
RF	Radio Frequency
RR	Round Robin
RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
RREP	Route Reply
RREQ	Route Request
RIFS	Reduced Inter-Frame Spacing
RSVP	Resource Reservation Protocol
RSVP-TE	RSVP-Traffic Engineering
RT	Routing Table
RTS	Request to Send

RTT	Round-Trip-Time
RX	Receive(r)
SBT	Simple Bundle Table
SIMC	Single-Interface Multi-Channel
SIP	Session Initiation Protocol
SIR	Signal-to-Interference Ratio
SISC	Single-Interface Single-Channel
SNIR	Signal-to-Noise Plus Interference Ratio
SNR	Signal-to-Noise Ratio
SPQ	Strict Priority Queuing
SRC	Source
SYN	Synchronize
TA	Traffic Analysis
TC	Topology Control
TCP	Transmission Control Protocol
TEL	Traffic Engineering Labeling
TOS	Type of Service
TTL	Time to Live
TX	Transmit(ter)
UE	User Equipment
UDP	User Datagram Protocol
URU	Universal Radio Unit

VANET	Vehicular Ad-Hoc Network
VE	Virtual Environment
VI	Virtual Interface
VoIP	Voice-over-IP
VPN	Virtual Private Network
WAT	Wireless Access Technology
WCETT	Weighted Cumulative ETT
WFQ	Weighted Fair Queuing
WFS	Weighted Fair Scheduling
WDS	Wireless Distribution System
WiBACK	Wireless Back-Haul
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WNIC	Wireless Network Interface Card
XML	Extensible Markup Language

A

APPENDIX

A.1 FURTHER RELATED WORK

In the following, additional routing protocols are described, which are not directly applicable for mesh backbones.

- AODV [42]
  - The *classic* reactive, Distance Vector protocol
  - No proactive maintenance of routes, GWs not known from the start
  - Tackles the count-to-infinity effect, by introducing sequence numbers
  - Typically used for ad-hoc networks with a high grade of mobility, constantly changing paths and few nodes [40], thus less suitable for static backbones
- Babel[1] [137]
  - Based on Destination-Sequenced Distance Vector (DSDV) [138]
  - Babel is recommended for sparsely populated, by trend unreliable WMNs
  - Designed to offer loop-free routes
  - Tends to prefer previously used, reliable routes instead of new ones, to prevent next hop oscillation between equally good routes
- Link Quality Source Routing (LQSR) protocol [12]
  - Hybrid approach: Each LQSR node builds a complete view of the topology (table-driven) but receives necessary information from packets which traverse on-demand-generated routes. A route to a GW might not be available from the start though

---

[1] Babel — a loop-avoiding distance-vector routing protocol, `http://www.pps.univ-paris-diderot.fr/~jch/software/babel/`

  - Applies source routing (Dynamic Source Routing (DSR)-based), is therefore loop-free
  - Multi-Radio LQSR (MR-LQSR) is an extension with multi-radio support

A.2 CONCEPTUAL AMENDMENTS AND DISCUSSIONS

A.2.1 *The Mesh Routing Protocol of Choice*

The OLSR protocol offers advantages, which were relevant for the system design:

- It is proactive:
  - Direct availability of an optimized path routing decision; thus OLSR is one of the better choices for QoS-related traffic
  - A node restart or topology change might imply (re)discovery of newly available, or altered routes. The proactive method is not as time consuming as with on-demand protocols
- It was designed for large, infra-structure-like mesh networks, with static node positions, heterogeneous node density and long hop distances. This suits backbones with traffic behavior described in Chapter 1, Section 1.2.5
- Its low protocol complexity plainly provides a next hop independent of the traffic type and chosen channel. Thus, OLSR is a suitable basis for further mesh enhancements which are not necessarily routing protocol-typic, like QoS, traffic awareness and custom IF management. OLSR will not interfere here
- Network-wide identification of gateways, to proactively prepare the protection of GW flows
- Built-in handling of multiple IFs, though it is simplistic. OLSR can be "multi-homed"[2] on layer 3, with MID messages
- HELLO messages can be used for piggybacking custom signaling
- Well established and popular in research community

A.2.2 *Common Control Channel*

In a SISC node, signaling and data packets are processed in the same layer 2 buffer, where both types compete in a best-effort manner. Hence, probability of a data

---

[2] Using multiple interfaces with OLSR, `http://www.olsr.org/docs/report_html/node33.html`

packet drop due to a timeout increases, which may negatively affect network performance. Deploying a (common) CC is a known concept embedded in CA protocols [139]. A CC anticipates to separate both types spectrum-wise, either in a hybrid or absolute fashion. With a CC, signaling packets are solely scheduled via a single channel, which is tuned to a fixed or negotiated channel. A designated CC can be either exclusively used for signaling and control packets, or in a hybrid fashion, where it also bears other traffic. Mesh administrators must make sure that every node has access to the CC, especially when vital information about neighborhood connectivity is provided. Tragos et al. [139] discuss whether having a CC is effective, or if it demands more resources and efforts than it's benefits may compensate. Nevertheless, most CA protocols discussed in their study apply a CC.

A CC is typically used for the following traffic:

- Broadcast signaling from OLSR, ARP, Domain Name System (DNS), DHCP, and other protocols
- Unicast signaling from TCP (synchronization and flow control), ICMP, Session Initiation Protocol (SIP), RSVP, LDP, and other protocols
- CA protocol specific exchange

A CC can be used for one or more signaling types at once. If metrics with active link probing are used, CC cannot be used solely for probing, as all links with all neighbors need to be probed.

The advantage of a fixed CC is that a new node can directly access information of assigned channels in his coverage area, and tune its interfaces to the according channels. Thus, deploying a CC facilitates initial mesh connectivity. Time-consuming scanning of all channels to sense the presence of MIMC nodes is avoided.

A.2.3 *Additional CA Protocol Requirements*

CHANNEL ASSIGNMENT:

CA proposes a set of channels per neighbor, in a way that an optimal bidirectional communication to this hop is enabled.

RADIO POOL MANAGEMENT:

CA keeps track of *all free* and currently used radios.

CHANNEL STATE AND QUALITY AWARENESS:

The evaluation of a channel's condition is done proactively (independent of traffic) and reactively (when traffic is running on this channel). Proactive distribution shall estimate the simple channel state, based on the general availability of the radio (hardware in an idle state) and previous usage (how many neighbors are addressed via this channel, i. e., channel exclusiveness). Reactive distribution shall additionally monitor activities of assigned channels and consider the channel load.

DYNAMIC EXECUTION:

CA is not static, channels can be switched during network operation. If CA shall adapt to interference, changing traffic, and other link quality fluctuations, CA algorithm complexity and signaling overhead increases significantly [26].

VARIABLE ALLOCATION STABILITY:

Generally, every radio shall have the possibility to switch a channel reactively or proactively, in case it's quality decreases below an acceptable level. A weighting system to rate the impact of a switch is desirable. A CA protocol may optionally include such policies to calculate a channel-specific switching cost (CSC). This becomes especially important in mesh networks which transport vertical traffic. Changing channels of (active) radios of GW nodes shall be more expensive. Naturally, links close to a GW carry the majority of traffic, therefore a frequent switch might interrupt too many active connections. To protect GW links from disconnection due to a channel switch, their switching characteristic must be more "inertial". He and Xu [140] describe a hybrid CA protocol whose switching strategy foresees that some radios are fixed and some are switchable, based on their topological distance to the GW.

LINK-BASED:

Channels are primarily assigned for links to neighbors, irrespective of whether the link is currently in use or not. Thus, the total hop distance of flows and per-packet assignment are not considered in the CA process; only the general load level of a channel.

INTERFACE TO OS:

OLSR selects a main IP among IPs of equipped radios. With regard to previous requirements, CA must adopt this mapping. In the topology, only the main IP is listed. Topology information is processed to assign more resources to next hops leading to a GW. All envisioned OS interfaces (especially to OLSR) are depicted in Fig. 2.2.

A.2.4 *GW Presence in a Standard OLSR Toplogy*

In the example shown in Figure A.1 [141], node 192.168.32.41 represents a GW. Numbers next to edges represent link costs (ETX in this case; 1.0 means that the link offers perfect conditions).

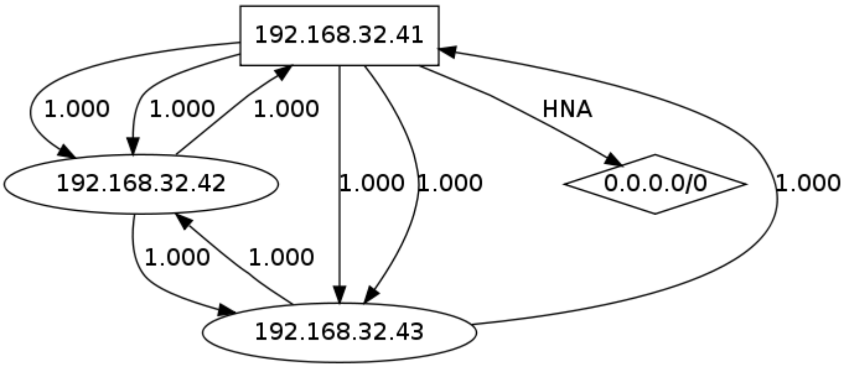


Figure A.1: Example of GW presence in topology [141]

A.2.5 *Alternative Transport Method for M-LDM Messages*

The main OLSR routing engine (not potential metric plugins) already deploys proactive signaling, to distribute its three custom messages HELLO, TC and HNA [35]. OLSR messages are gathered in a generic OLSR packet format, which is depicted in Figure A.2. Multiple instances of the MESSAGE field can be stored within one

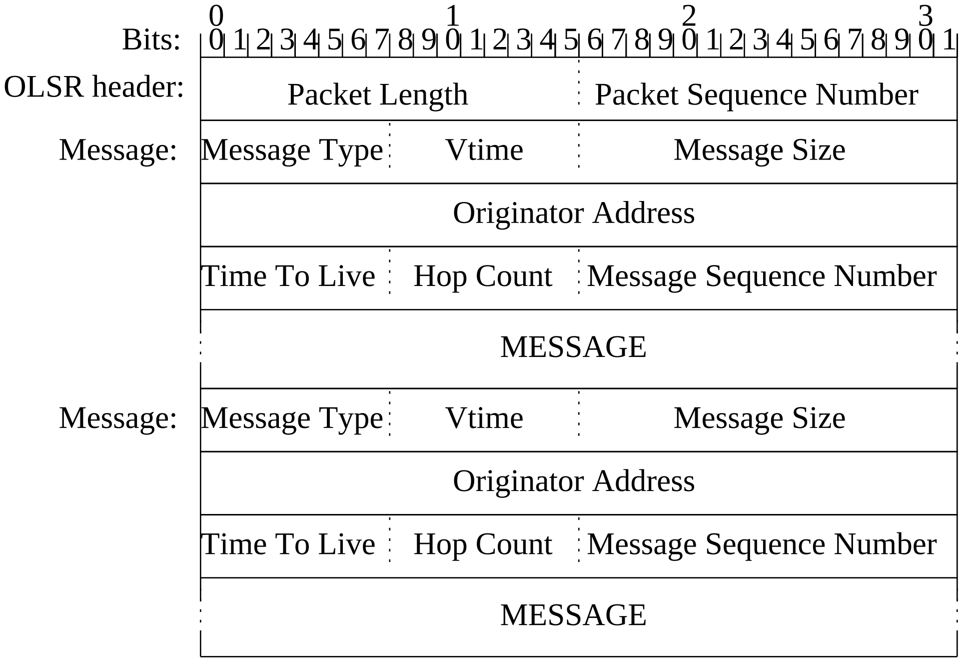


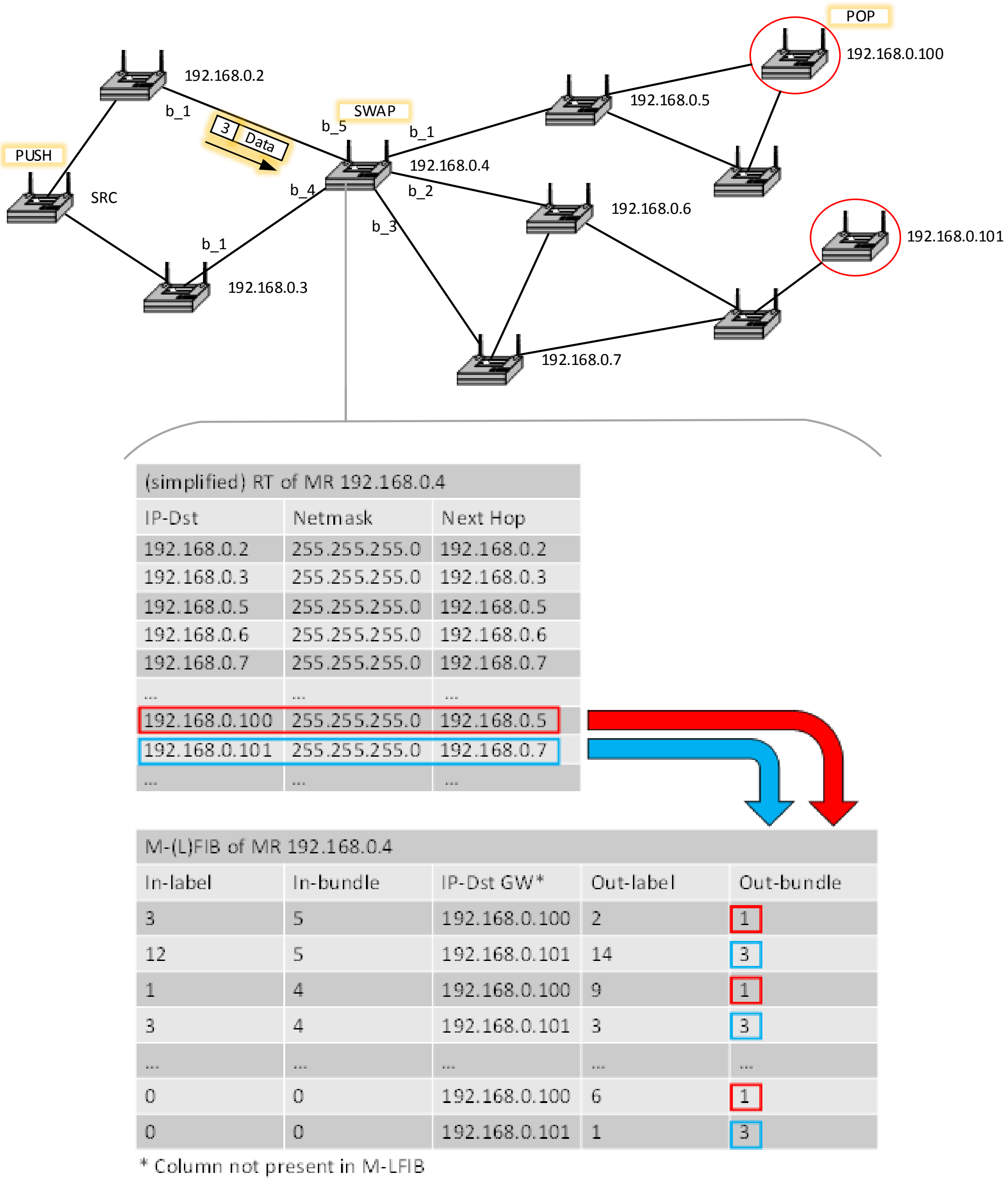
Figure A.2: The generic OLSR packet [142]

OLSR message [143]. It would be feasible to embed combinations of IPv4 addresses and labels in these multiple instances and thus define a new OLSR message type to be distributed within OLSR packets. The TTL field of the custom message is set to 1. A further adaption is not within the scope of this thesis, since storing M-LDM messages in the payload of packet-pair probing messages is preferred. Note that message types 0-127 are reserved by OLSR [142]. Type index 128-255 can be used for custom message types. Generally, the emission interval of HELLO messages must be kept short.

The disadvantage of this approach is that OLSR packets are broadcasted. M-LDP foresees a selective unicast of M-LDMs. As a compromise, all out-labels must be announced in the 1-hop neighborhood in each broadcast message. This increases flooding and size of OLSR signaling packets.

A.2.6 *Examples for Labeled Packet Communication*

In Fig. A.3, different label operations are applied. A source (left side) sends packets towards the GW at 192.168.0.100 on an upload path. Label-based forwarding is exemplarily applied for the yellow-highlighted packet between node 192.168.0.2 and node 192.168.0.4. The latter node's routing table and M-(L)FIB are closely inspected in Fig. A.3, along with context-relevant information of the node's environment (e. g., bundle indices b). The packet's label will be swapped from 3 to 2 at 192.168.0.4, as it is swiftly forwarded. M-FIB tables are not complete here, for the sake of simplicity. Also, in-labels received by neighbors 192.168.0.5 – 7 are not listed in the sample M-FIB.



(simplified) RT of MR 192.168.0.4		
IP-Dst	Netmask	Next Hop
192.168.0.2	255.255.255.0	192.168.0.2
192.168.0.3	255.255.255.0	192.168.0.3
192.168.0.5	255.255.255.0	192.168.0.5
192.168.0.6	255.255.255.0	192.168.0.6
192.168.0.7	255.255.255.0	192.168.0.7
...	...	...
192.168.0.100	255.255.255.0	192.168.0.5
192.168.0.101	255.255.255.0	192.168.0.7
...	...	...

M-(L)FIB of MR 192.168.0.4				
In-label	In-bundle	IP-Dst GW*	Out-label	Out-bundle
3	5	192.168.0.100	2	1
12	5	192.168.0.101	14	3
1	4	192.168.0.100	9	1
3	4	192.168.0.101	3	3
...	...	...	...	...
0	0	192.168.0.100	6	1
0	0	192.168.0.101	1	3

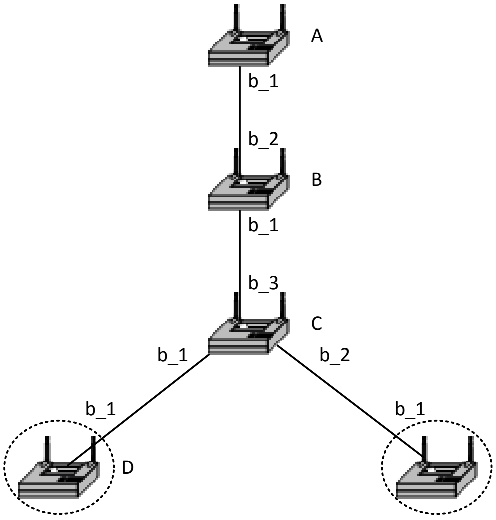
* Column not present in M-LFIB

Figure A.3: Label operations in an example

The second example in Fig. A.4 shows a WMN where M-FIB tables are already filled. Note that in this example, full IP DST addresses were replaced with characters, to enable a better overview (also, out-labels are generated in an ascending order). There are two GWs in this example; nodes D and E. Their presence has an impact on all M-FIB tables in the mesh. Both up- and download paths to them are listed in the tables.

A.2.7 *Extensions to Include Receiver Bundle Feedback and Packet Reordering*

Packet reordering is substantial in full multi-path mesh networks [144], [145], because long delay discrepancies between intra-flow packets might occur, due to multiple hops. For the presented system, LB is performed until the next hop, therefore the effect of dispersed delay is less critical. Also, labeling, queues and especially scheduler processes reveal that *the system is purely forwarding oriented*. A receiver feedback within a bundle is therefore currently not envisioned. Nevertheless, packet reordering in shared bundles between neighbors should be investigated and eventually regarded for optimization purposes in future system versions. This becomes especially important when link qualities between bundled radios differ widely. In order to avoid additional reordering measures with the current system, it is recommended to combine hop-to-hop load balancing with the use of TCP versions with built-in reordering robustness [146]. Otherwise, end-to-end TCP performance might significantly drop, when reordering is not considered in bundles along a route [145].



M-FIB of A

In-b.	In-l.	IP DST	o-b.	o-l.
0	0	D	1	1
0	0	E	1	2
1	1	A	0	1
1	2	A	0	2

M-FIB of B

In-b.	In-l.	IP DST	o-b.	o-l.
2	1	D	1	1
2	2	E	1	2
0	0	D	1	3
0	0	E	1	4
1	1	A	2	1
1	2	B	0	1
1	3	A	2	2
1	4	B	0	2

M-FIB of C

In-b.	In-l.	IP DST	o-b.	o-l.
3	1	D	1	1
3	2	E	2	1
3	3	D	1	2
3	4	E	2	2
0	0	D	1	3
0	0	E	2	3
1	1	A	3	1
1	2	B	3	2
1	3	C	0	1
1	4	E	2	4
2	1	A	3	3
2	2	B	3	4
2	3	C	0	2
2	4	D	1	4

M-FIB of D

In-b.	In-l.	IP DST	o-b.	o-l.
1	1	D	0	1
1	2	D	0	2
1	3	D	0	3
0	0	A	1	1
0	0	B	1	2
0	0	C	1	3
0	0	E	1	4
1	4	D	0	4

M-FIB of E

In-b.	In-l.	IP DST	o-b.	o-l.
1	1	E	0	1
1	2	E	0	2
1	3	E	0	3
1	4	E	0	4
0	0	A	1	1
0	0	B	1	2
0	0	C	1	3
0	0	D	1	4

Figure A.4: Example of filled M-FIB tables

A.2.8 *Determination of the TX Probability with WFS*

To determine a radio TX probability, its metric is processed in the following way. To illustrate the process, steps from the implementation are used.

1. `receiveSignal()` taps link state provided by OLSR (C++ segment):

   ```
   NodeEntry* myNodeEntry = new NodeEntry(srcObj->currentNode,srcObj->nbNode,
       srcObj->localAdd,srcObj->nbAdd,srcObj->etx,srcObj->delay,1,1,1);
   ```

   The first time a radio comes up, a new entry is created. Both available metrics ETX and ETT (`delay`) are extracted at the same time. In the implementation, WFS mode requires ETT, as it is the more accurate, QoS-related metric.

2. `WFScalculation()` stores the reciprocal of ETT in the BMT 2.9 (see following (C++ segment). This weight of a radio r is listed in the BMT column v_j.

   ```
   BMTtableEntry* myBMTentry = new BMTtableEntry(temp->getCurrentNode(), temp
       ->getNbNode(), temp->getLocalAdd(), temp->getNbAdd(), 1/(temp->
       getDelay()),0,0,0);
   ```

3. `WFScalculation()` calculates the sum of weights in a bundle b (C++ segment):

   ```
   ett_sum += 1/(temp->getDelay());
   ```

4. `WFScalculation()` calculates the TX probability g for each radio and stores it in the corresponding field in BMT (C++ segment):

   ```
   double probability_temp = bmt_temp->getWeight()/ett_sum;
   ```

The above steps illustrate how ETT is processed in the `scheduler` module. After these steps, we obtain the probabilities of each link and we can assign the packets statistically based on their probabilities.

A.2.9 *WFS Output*

Listing A.1 (extract / output of OMNeT++ 4.1 event window) contains the WFS calculation output for a single packet. It outlines the link tuple and the related TX probability (lines 9, 10, 22 and 23) for a link between a local and a neighbor radio

Listing A.1: Examples of WFS output for a single packet

```
   # sending UNICAST msg on radio index:: 0
   ** Event #448701  T=32.000046333332  MiNetwork.mihost[0].scheduler (Scheduler,
       id=10), on ‘VideoStrmPk’ (IPDatagram, id=3955349)
   SENDING: IP SRC ADDRESS:: 10.10.0.1 IP DST ADDRESS:: 10.10.0.3
 4 Scheduling algorithm: WFS
   ETT info:(Current node, Neighbor node, Local interface, Neighbor interface,
       Link delay(ETT))
   { 10.10.0.1, 10.10.0.3, 10.10.0.2, 10.10.0.4,0.00570172},
   { 10.10.0.1, 10.10.0.3, 10.10.0.1, 10.10.0.3,0.00561647},
   ETT info:(Neighbor node, Local interface, Neighbor interface, TX probability,
       number of sent packets, periodWFQ)
 9 { 10.10.0.3, 10.10.0.2, 10.10.0.4,0.496234,2988,1},
   { 10.10.0.3, 10.10.0.1, 10.10.0.3,0.503766,2935,1}

   # 6 seconds passing...

14 # sending UNICAST msg on radio index:: 1
   ** Event #1824552  T=38.000191333332  MiNetwork.mihost[0].scheduler (Scheduler,
        id=10), on ‘VideoStrmPk’ (IPDatagram, id=4989735)
   SENDING: IP SRC ADDRESS:: 10.10.0.1 IP DST ADDRESS:: 10.10.0.3
   Scheduling algorithm: WFS
   ETT info:(Current node, Neighbor node, Local interface, Neighbor interface,
       Link delay(ETT))
19 { 10.10.0.1, 10.10.0.3, 10.10.0.2, 10.10.0.4,0.00205549},
   { 10.10.0.1, 10.10.0.3, 10.10.0.1, 10.10.0.3,0.00272974},
   ETT info:(Neighbor node, Local interface, Neighbor interface, TX probability,
       number of sent packets, periodWFQ)
   { 10.10.0.3, 10.10.0.2, 10.10.0.4,0.570451,1116,1},
   { 10.10.0.3, 10.10.0.1, 10.10.0.3,0.429549,764,1}
```

kept within a bundle. In the example, two neighbors with the IPs 10.10.0.1 and 10.10.0.3 have a bundled connection with two radios each. The second output from line 14 on shows that a shift of link state has been provoked. TX probabilities and the number of sent packets change accordingly.

A.2.10 *Distinctive Features of RR-Extended Mode and Alternative Parameters*

The extended RR mode sets itself apart from the WFS mode. Extended RR mode has been designed for nodes where no suitable layer 3 link-quality metric might be available. Unsuitable would be for example the simple hop count (whose metric value is always 1). Also, the routing protocol might be configured in a way that link state is refreshed in long periods. An imminently required switch is bound to this interval then. Thus, switching criteria in this mode shall be independent of protocol-specific timers. Another difference to WFS mode is that the switching decision is solely based on statistics of the currently used radio. Presence and state of other radios is not considered. Compared to the more sophisticated WFS mode, extended RR is more radical in its scheduling behavior; underachieving radios will be disabled completely until resurrected anew, based on a reactive decision. As a natural conclusion, it is recommended to use a parameter F provided by the PHY or MAC layer [113].

R typically defines a minimum or maximum rate, such as the frame error rate on the MAC. To provide another example, the current MAC data rate represents a simple 802.11x parameter. WirelessLAN Dynamic Rate Shifting (DRS) adjusts the modulation scheme reactively. This is typically triggered when interference occurs or when SNR drops, because the device gains in distance to a receiver. When a MAC ACK frame is not received within the ACK window and a packet has to be re transmitted, the radio may choose to shift to a more robust rate. Since WLAN data rates are predefined (see [13]), the mesh administrator can easily pick one as a threshold rate, in case this fallback criterion is used. With this behavior, different rates along a multi-hop route can be avoided, which is beneficial for a mesh network [147].

A.3 SIMULATION TOOLS

A.3.1 *Environment*

For all simulations, OMNeT++ version 4.1 (build "100611-4b63c38") is used, with INETMANET (build "4116c0c371") as the framework of choice.

OMNeT++ was installed on a virtual machine, which runs on a Proxmox Virtual Environment (VE)[3] server. The virtual machine has the following hardware specifications:

- 8 GB Random-Access Memory (RAM)
- 8-core processor (single socket)
- 128 GB Hard Disk Drive (HDD)
- Debian 6.0.8 Squeeze
- Linux Kernel 2.6.32-5.686
- 32-bit OS

Each radio in the WMN receives an individual IPv4 address within the same address range (address class is chosen automatically). This is managed by the module `NetworkConfigurator` in each .ini file.

Four `thruputMeters` are placed between transport and network layer in the custom node in Fig. 3.1. They measure bidirectional TCP [148] and UDP performance [149] (including overhead). All other modules already generate internal statistics.

Generally, two types of traffic flows are considered. There is engineered traffic, which is in the focus of investigations. This type of traffic experiences the various advantages of the layer 2.5 module, like multi-radio usage or class-sensitive queuing. And then there is Cross Traffic (CT), which is used to congest single channels or links in the mesh topology. This type is not measured. For cross-traffic only a single channel is used, which eliminates the use of PS modes.

By default there is no switch of channels during operation. All channels are assigned statically at the beginning of the simulation. This manual assignment is declared for

---

[3] Proxmox Virtual Environment, `http://www.proxmox.com/`

every single node pair in the corresponding parameters list. As a requirement for the used CA protocol, it is demanded that a worsening channel condition may trigger a switch. This effect is regarded in Chapter 3, 3.3.5.

Solely IEEE 802.11g [13] radios are used, with their regarding standard layer 1 and 2 properties. Unlike with real hardware, it is possible to specify an arbitrary amount of available channels. The PHY layer of INET[4]/ INETMANET considers all 20 MHz 802.11 channels to be orthogonal [97].

Although supported by OMNeT++, node *mobility* and solid *obstacles* (to cause shadowing) are not included in simulations. All mesh nodes have a static position and remain active during simulation period. Only link states change in the topology during simulation. Generally, obstacles and mobility are aspects which strongly influence the response of the physical layer. To avoid that PHY layer causes unexpected performance variations of flows monitored in layer 2 and above, PHY parameters are chosen in a way that a homogeneous environment is created. This allows a better and more precise analysis of core characteristics.

Using a grid structure (or parts of it) for node positioning also has performance advantages, as discussed in [27]. In their study on city-wide mesh backbones, Vural et al. state that a grid topology may achieve up to 50% higher throughput levels than with a random node placement.

A.3.2 *Interference Generators*

In Chapter 3, 3.3, different methods are used to congest a channel or radio resource. *Intra-mesh* congestion is caused by concurrent flows within the same WMN. To maintain two completely separated layer 3 topologies, which can interfere with each other on shared channels, the focus network runs with OLSR, whereas the background network uses AODV. Listing A.2 (segment from .ini file) treats this case. Each ad-hoc network runs in a different sub-network. All radios have static IPs. Example interface configuration of a node of the focus network is shown in A.3 (segment from .ini file). Example interface configuration of a node of the external background network is listed in A.4 (segment from .ini file).

---

[4] INET - An open-source communication networks simulation package, `http://inet.omnetpp.org/`

Listing A.2: Deploying OLSR and AODV simultaneously - .ini configuration

```
**.mihost[0..1].manetrouting.manetmanager.routingProtocol = "OLSR_ETX"
**.mihost[2..3].manetrouting.manetmanager.routingProtocol = "AODV"
```

Listing A.3: Deploying OLSR and AODV simultaneously - interface setup for OLSR

```
ifconfig:
# interface 0 to mihost[0]
name: wlan0
    inet_addr: 10.10.0.1
    Mask: 255.255.255.0
    MTU: 1500
    POINTTOPOINT MULTICAST
# interface 1 to mihost[0]
name: wlan1
    inet_addr: 10.10.0.2
    Mask: 255.255.255.0
    MTU: 1500
    POINTTOPOINT MULTICAST
ifconfigend.
```

Listing A.4: Deploying OLSR and AODV simultaneously - interface setup for AODV

```
ifconfig:
# interface 0 to host[0]
name: wlan0
    inet_addr: 192.0.0.1
    Mask: 255.255.255.0
    MTU: 1500
    POINTTOPOINT MULTICAST
# interface 1 to host[0]
name: wlan1
    inet_addr: 192.0.0.2
    Mask: 255.255.255.0
    MTU: 1500
    POINTTOPOINT MULTICAST
ifconfigend.
```

A.4 MAIN CONFIGURATION PARAMETERS

This Section lists the most important configuration parameters for the simulations conducted in Chapter 3.

A.4.1 *Quality-of-Service and Priority Queueing*

Table A.1 lists selected OMNeT++ parameters relevant for queue testing. The complete configuration file of the simulation can be found in the code attachments (disc). `repeat` specifies the number of runs / repetitions. The column *Node* has been introduced here, to indicate configuration of specific nodes. When fields are empty, the parameter applies to all nodes (which is still the case in table A.1).

Table A.1: Parameters for queue testing

MODULE	NODE	PARAMETER	TP.	VALUE
(general)		`sim-time-limit`	C.	130s
		`repeat`	C.	40
`manetrouting`		`routingProtocol`	S.	OLSR_ETX
		`Link_delay`	B.	true
(network)		`numMIHosts`	C.	8
		`playgroundSizeX`	C.	720m
		`playgroundSizeY`	C.	300m
		`mobilityType`	S.	NullMobility
		node distance	C.	140m
(custom)		`schedulingAlgorithm`	S.	*Variable*
		`periodWFQ`	C.	1
		`periodETT`	C.	2
(MIMC)		`numChannels`	C.	2
		`numMIRadios`	C.	2
		ch. wlan0	C.	0
		ch. wlan1	C.	1

The ETT metric is used. The settings in table A.2 show alternative metrics from the INETMANET OLSR version.

Table A.2: Routing metric configuration in OLSR

SETTING	EFFECT
`routingProtocol = OLSR`...	activates hop count metric
`routingProtocol = OLSR_ETX`, `Link_delay` disabled...	activates ETX metric
`routingProtocol = OLSR_ETX`, `Link_delay` enabled...	activates ETT metric

The parameters of the UDP "Basic Burst" and "Sink" applications are listed in table A.3.

Table A.3: Exemplary parameters for UDP basic burst and sink app

MODULE	PARAMETER	TYPE	VALUE
(host)	`numUdpApps`	Constant	1
	`udpAppType`	String	UDPBasicBurst
`udpApp`	`localPort`	Constant	9001
	`destPort`	Constant	9001
	`destAddresses`	String	*Variable*
	`limitDelay`	Constant	10s
	`messageLength`	Constant	*Variable*
	`messageFreq`	Constant	*Variable*
	`time_begin`	Constant	30s
	`time_end`	Constant	0s
	`message_freq_jitter`	Constant	0s
	`burstDuration`	Constant	1ms
(host)	`numUdpApps`	Constant	1
	`udpAppType`	String	UDPSink
`udpApp`	`localPort`	Constant	9001

A.4.2 *Vertical Traffic in a Mesh Network*

Table A.4 shows which main parameters define the grid scenario. The complete configuration file can be found in the code attachments (disc).

Table A.4: Parameters for vertical traffic simulation

MODULE	NODE	PARAMETER	TP.	VALUE
(general)		`sim-time-limit`	C.	55s
		`repeat`	C.	20
`manetrouting`		`routingProtocol`	S.	OLSR
		`Link_delay`	B.	false
`tcp`		`receiveQueueClass`	S.	TCPMsgBasedRcvQueue
		`sendQueueClass`	S.	TCPMsgBasedSendQueue
(network)		`numMIHosts`	C.	37
		`playgroundSizeX`	C.	980m
		`playgroundSizeY`	C.	980m
	36	`mobilityType`	S.	NullMobility
	0 – 35	`mobilityType`	S.	StaticGridMobility
	36	node distance	C.	99m
	0 – 35	node distance	C.	140m
(custom)		`queueModule`	S.	""
		`schedulingAlgorithm`	S.	RR
(MIMC)		`numChannels`	C.	*Variable*
		`numMIRadios`	C.	*Variable*
		ch. wlan0	C.	0
		ch. wlan1	C.	1
		ch. wlan2	C.	2
		ch. wlan3	C.	3
		ch. wlan4	C.	4
		ch. wlan5	C.	5

Table A.5 lists the settings for the `TCPBasicClientApp` and `TCPGenericSrvApp` applications in the scenario.

A.4.3 *Multi-Modal Load Balancing*

Table A.6 lists common settings for the two test scenarios. The complete configuration file of the simulation can be found in the code attachments (disc).

Table A.7 specifies the parameters for the evaluation of the WFS PS mode.

Table A.8 specifies the parameters for the evaluation of the Extended RR PS mode.

Table A.5: Exemplary parameters for TCP basic client and generic server app

MODULE	PARAMETER	TYPE	VALUE
(host)	`numTcpApps`	Constant	*Variable*
	`tcpAppType`	String	TCPBasicClientApp
tcpApp	`address`	String	“”
	`connectAddress`	String	*Variable*
	`connectPort`	Constant	21
	`idleInterval`	Constant	1s
	`numRequestsPerSession`	Constant	3
	`port`	Constant	2000
	`reconnectInterval`	Constant	100ms
	`replyLength`	Constant	5MB
	`requestLength`	Constant	$(5-20)$B
	`startTime`	Constant	$\approx 30{,}5$s
	`thinkTime`	Constant	2s
(host)	`numTcpApps`	Constant	1
	`tcpAppType`	String	TCPGenericSrvApp
tcpApp	`address`	String	“”
	`port`	Constant	21
	`replyDelay`	Constant	0s

Table A.6: Common parameters for multi-modal load balancing simulation

MODULE	NODE	PARAMETER	TP.	VALUE
(general)		`sim-time-limit`	C.	130s
		`repeat`	C.	40
manetrouting		`routingProtocol`	S.	OLSR_ETX
		`Link_delay`	B.	true
tcp		`advertisedWindow`	C.	65535
		`mss`	C.	1024
		`tcpAlgorithmClass`	S.	TCPReno

Table A.7: Parameters for evaluation of WFS PS mode

MODULE	NODE	PARAMETER	TP.	VALUE
(network)		`numMIHosts`	C.	9
		`playgroundSizeX`	C.	580m
		`playgroundSizeY`	C.	300m
		`mobilityType`	S.	NullMobility
	0 – 4	node distance	C.	140m
	5 – 8	node distance	C.	40m
(custom)	0 – 4	`schedulingAlgorithm`	S.	*Variable*
	5 – 8	`schedulingAlgorithm`	S.	none
		`queueModule`	S.	“”
		`periodWFQ`	C.	1
		`periodETT`	C.	2
(MIMC)		`numChannels`	C.	3
		`numMIRadios`	C.	3
	0 – 4	ch. wlan0	C.	0
	0 – 4	ch. wlan1	C.	1
	0 – 4	ch. wlan2	C.	2
	5 – 6	ch. wlan0	C.	0
	5 – 6	ch. wlan1	C.	0
	5 – 6	ch. wlan2	C.	0
	7 – 8	ch. wlan0	C.	1
	7 – 8	ch. wlan1	C.	1
	7 – 8	ch. wlan2	C.	1

Table A.8: Parameters for evaluation of Extended RR PS mode

MODULE	NODE	PARAMETER	TP.	VALUE
(network)		`numMIHosts`	C.	9
		`playgroundSizeX`	C.	1140m
		`playgroundSizeY`	C.	300m
		`mobilityType`	S.	NullMobility
		node distance	C.	140m
(custom)		`schedulingAlgorithm`	S.	fallback
		`queueModule`	S.	“”
		`periodWFQ`	C.	99
		`periodETT`	C.	99
		`numFBRadios`	C.	*Variable*
		`collisionRatio`	C.	*Variable*
(MIMC)		`numChannels`	C.	*Variable*
		`numMIRadios`	C.	*Variable*
		ch. wlan0	C.	0
		ch. wlan1	C.	1
		ch. wlan2	C.	2
		ch. wlan3	C.	3

A.5 ADDITIONAL MEASUREMENS

A.5.1 *Single-Interface Radio Performance*

The goal of this preceding simulation series is to create simple reference measurements as a basis for other, more complex setups. A single radio-to-radio link and its capacity is continuously found as an element of multi-hop routes in the following. A WLAN radio including layer 1 and 2 is the smallest manageable unit in the proposed system, therefore only common standard settings are applied in these layers. The impact of these settings is investigated here, with an emphasis on typical PHY layer parameters. For the upper layer 2.5 module, the frame error rate, dependent on the transmit power of a WLAN interface, is the most interesting product of PHY layer.

A.5.1.1 *Configuration*

A simple SISC setup is constructed, comprised of two nodes in Fig. A.5. They are placed within reach and in line-of-sight. Each node is equipped with one 802.11g radio. The circles in Fig. A.5 represent a node's reception range, which is mainly determined by the receiver `sensitivity` value. The handling of frames received within the coverage area is described in the following. Frames received outside of these boundaries are received, but only recognized as disturbing noise. The latter

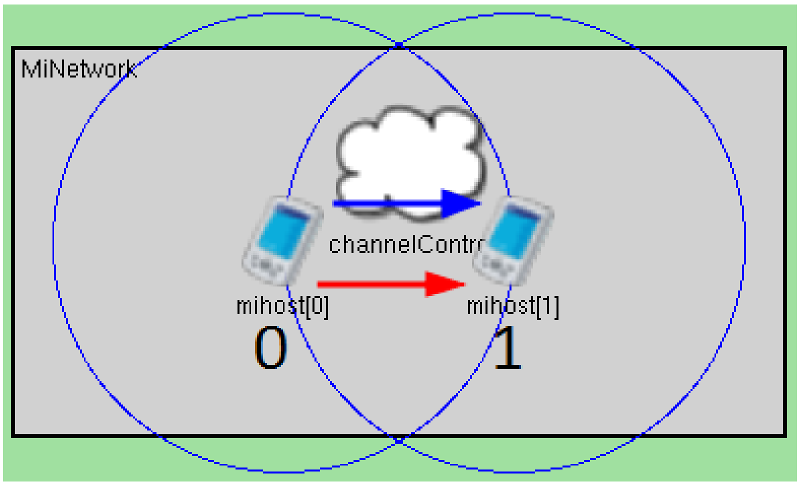


Figure A.5: Scenario for testing PHY layer response condition is only given, if a frame is still received within the receiver's `thermalNoise` boundary.

Between 802.11g radios, an uncongested 20 MHz WLAN channel is used. The choice of parameters is based on the physical conditions of a typical outdoor WMN with static node positions. Nodes use omni-directional antennas. A selection of relevant 802.11 radio parameters is shown in table A.9. PHY parameters are provided by INETMANET modules `ChannelControl` [150] and `Radio` [151]. `ChannelControl` influences the channel itself, whereas `Radio` is interface-specific. Both form part of the used 802.11g Network Interface Card (NIC) compound module. Its complete set of layer 1 and 2 default parameters is listed in [152], including a *description* of each parameter.

The complete configuration file of the simulation can be found in the code attachments (disc). The deployed TCP and UDP packet generators / sinks shall be explained in Chapter 3.

In INETMANET / INET, physical layer is a highly abstracted component. PER values are estimated based on the `berTableFile` (file "per_table_80211g_Trivellato.dat", INETMANET resource). The table contains packet error probabilities, previously obtained by a dedicated OFDM physical layer simulator. Chosen probability depends on the used coding and modulation scheme, Signal-to-Noise Plus Interference Ratio (SNIR) and frame size for an incoming packet. The table lookup is applied only for packets received within a node's reception range.

VARIABLES

Since mobility and obstacles are not considered, two channel models have been tested; free space / path loss reception and two-ray-ground model.

`mihost[0]` establishes a TCP connection to `mihost[1]` (lower arrow) and transmits a fixed amount of data. Afterwards, two consecutive UDP streams (upper arrow) are transferred in the same direction. The first one has a set TX rate of 7 Mbit/s with 256 bytes datagram size, whereas the second has a TX rate of 21 Mbit/s with 1 kB datagram size.

The transmitter power of `mihost[0]` shall vary from 0.5mW to 20mW.

Table A.9: Parameters for SI Radio Performance simulation

MODULE	PARAMETER	TYPE	VALUE
(general)	`sim-time-limit`	Constant	300s
	`warmup-period`	Constant	30s
`ChannelControl`	`pMax`	Constant	20mW
	`sat`	Constant	−110dBm
	`numChannels`	Constant	1
	`propagationModel`	String	*Variable*
`Radio`	`channelModel`	String	AWGN
	`channelNumber`	Constant	0
	`carrierFrequency`	Constant	2.4GHz
	`bitrate`	Constant	54Mbps
	`transmitterPower`	Constant	*Variable*
	`thermalNoise`	Constant	−110dBm
	`sensitivity`	Constant	−90dBm
	`pathlossAlpha`	Constant	2.8
	`TransmissionAntennaGainIndB`	Constant	0dB
	`ReceiveAntennaGainIndB`	Constant	0dB
	`SystemLossFactor`	Constant	0dB
	`TransmiterAntennaHigh`	Constant	1m
	`ReceiverAntennaHigh`	Constant	1m
	`snirThreshold`	Constant	4dB
	`berTableFile`	File	
	`phyOpMode`	String	g
(network)	`numMIHosts`	Constant	2
	`playgroundSizeX`	Constant	400m
	`playgroundSizeY`	Constant	200m
	node distance	Constant	60m
`manetrouting`	`routingProtocol`	String	OLSR

A.5.1.2 *Results*

The MAC loss rate depending on the received signal strength is depicted in Fig. A.6. Fig. A.7 shows the different throughput results for TCP and UDP. TX Radio (blue legend entry) refers to the radio of `mihost[0]`, whereas RX Radio (red entry) refers to the measured loss rate of the radio from `mihost[1]`.

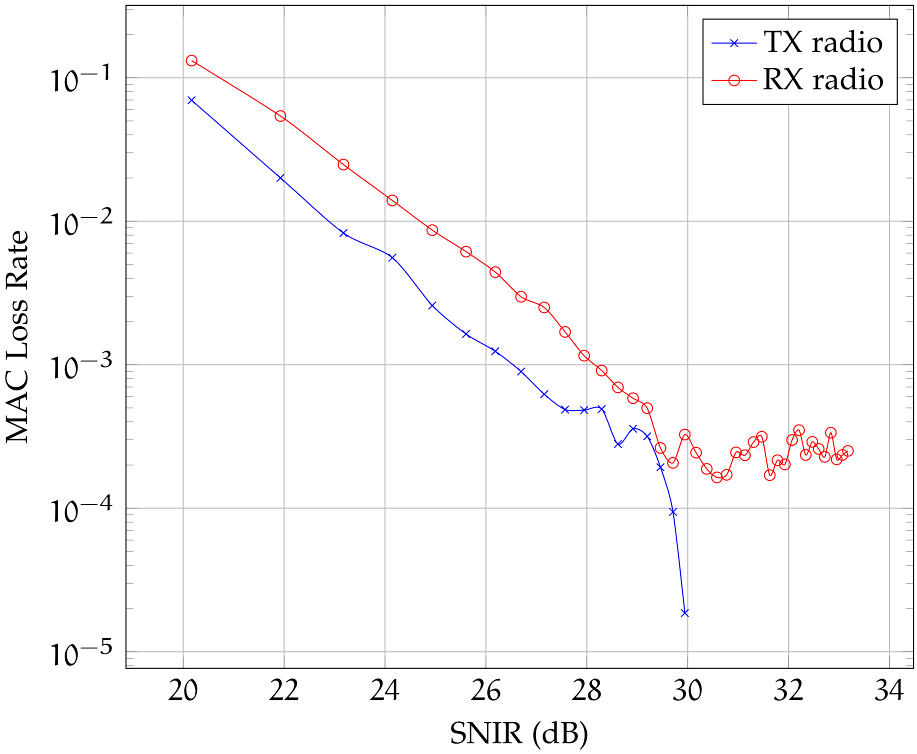
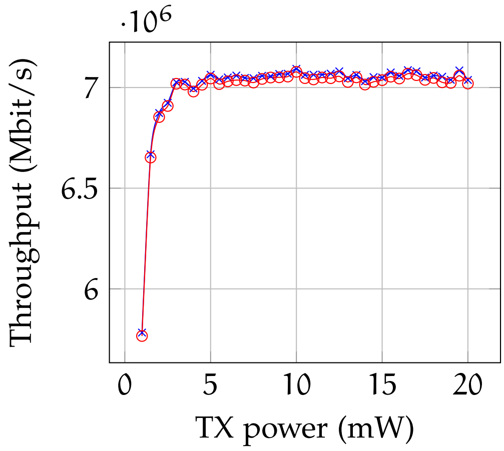


Figure A.6: Loss rate comparison with single-radio connection

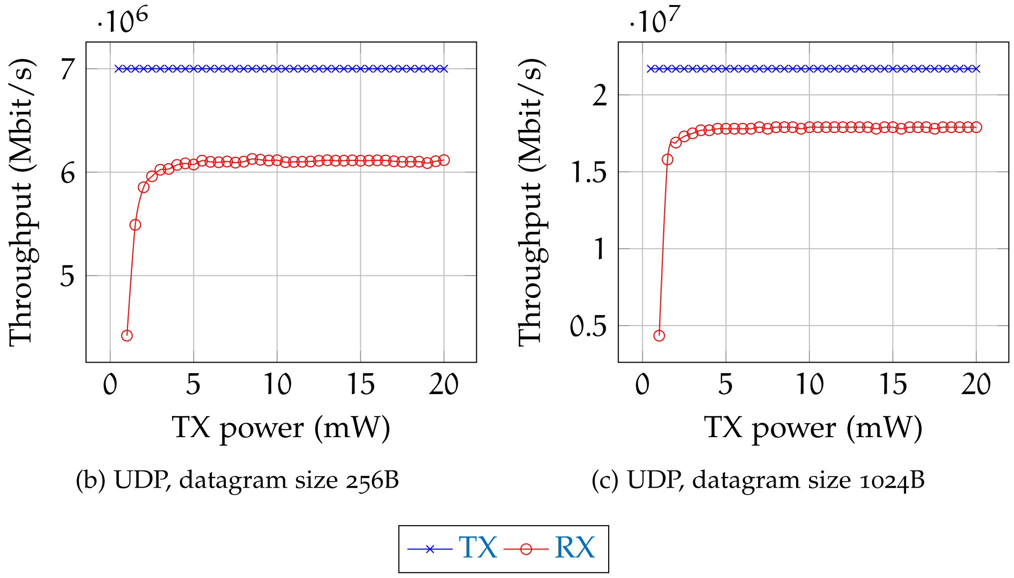
A.5.1.3 *Evaluation*

Mainly the chosen sensitivity value (-90dBm), path loss coefficient (2.8) and minimum TX power (0.5 mW) are responsible for the resulting connectivity range of 60 m. It was observed in this simulation series that the close proximity between nodes leads to the instance that free space and two-ray-ground model show similar results. Thus, the measured curves are independent of the channel model. For the other measurements, free space model will be used, with a TX power of 20 mW and a fixed node distance of 140m (with exceptions found in tables A.4 and A.7).

The throughput measurements shall serve as a reference for the achievable layer 4 performance on an uncongested link. TCP implementation in INETMANET allows for an average throughput of approx. 7 Mbit/s (including TCP headers and overhead). The later Fig. A.9a reveals that the MSS has a significant influence on TCP throughput. By default, it is set to 536 bytes [148] (which applies here). In other measurements, MSS is set to 1024 bytes, so the average TCP throughput on a single link is up to 11 Mbit/s. When a second fallback radio is used to carry solely TCP ACKs, achievable throughput per uncongested link can be raised to approx. 16



(a) TCP



(b) UDP, datagram size 256B

(c) UDP, datagram size 1024B

Figure A.7: UDP and TCP throughput with single-radio connection for PHY testing

Mbit/s. This effect is described in Chapter 3, 3.3.3.3. Maximum UDP throughput has been tested with TX rates of 7, respectively 21 Mbit/s. The result is an RX rate of approx. 6, respectively 17 Mbit/s. It is shown that the UDP payload size causes this large performance gap, since more payload is transported with less UDP header overhead and less packets compete to be sent in the MAC layer.

All other constants and file inputs *will be adopted as specified in table A.9* in subsequent simulations.

A.5.2 *MAC Parameters*

The used MAC components consist of the standard MAC layer (Ieee80211Mac) [153] for reception and transmission of frames, plus a MAC management extension [154] for peer-to-peer ad-hoc communication. Table A.10 lists all used MAC settings, which shall be valid in all other simulations. It contains the most relevant parameters from sources [153] and [154] in the INET documentation, where all parameters are further explained in detail. The implemented `autoBitrate` feature has no impact on results, as shown in Figures A.7b and A.7c. Although ARF and AARF [62] can be enabled, they have no effect. PHY modulation / bitrate is not adapted (no change in TCP / UDP throughput), even if the receiver is almost out of range. Therefore, the default setting (disabled) for `autoBitrate` is applied in the following simulations.

Table A.10: General MAC parameters

MODULE	PARAMETER	TYPE	VALUE
`mac`	`ForceBitRate`	Boolean	false
	`PHY_HEADER_LENGTH`	Constant	−1
	`autoBitrate`	Constant	0
	`basicBitrate`	Constant	2e6bps
	`bitrate`	Constant	54Mbps
	`maxQueueSize`	Constant	14
	`mtu`	Constant	1500
	`opMode`	Constant	2 (g)
	`retryLimit`	Constant	7
	`rtsThresholdBytes`	Constant	2346B
	`slotTime`	Constant	9us
`mgmt`	`frameCapacity`	Constant	10

By default, threshold frame size to use RTS/CTS virtual carrier sensing is set to a value of 2346 bytes. The hidden terminal problem is common in ad-hoc networks and single-channel congestion will be often provoked in the presented investigations. Still, in order to evaluate the effectiveness of novel features, varying this particular parameter and additionally consider its impact is not in the scope of this work.

The opmode "g" activates 802.11g mode, with a maximum raw data rate of 54 Mbit/s.

A.5.3 *Single-Interface Multi-Hop Environment*

This measurement shall provide information on the expected performance degradation caused by multi-hop transmission. In a mesh, nodes typically maintain one or more of these routes to different destinations. When looking at vertical traffic, one of the ends may be either a GW or a consumer. To better follow performance degradation in a mesh, at first such a single route is tested. To provoke mesh-typical cross traffic, disruptor nodes have been placed crosswise. The deployed scenario hosts 10 nodes and is shown in Figure A.8. Still, a single radio / channel is used. The stream from `mihost[0]` to `[5]` is in the focus of investigation. The influence of cross-traffic will harm it even further.

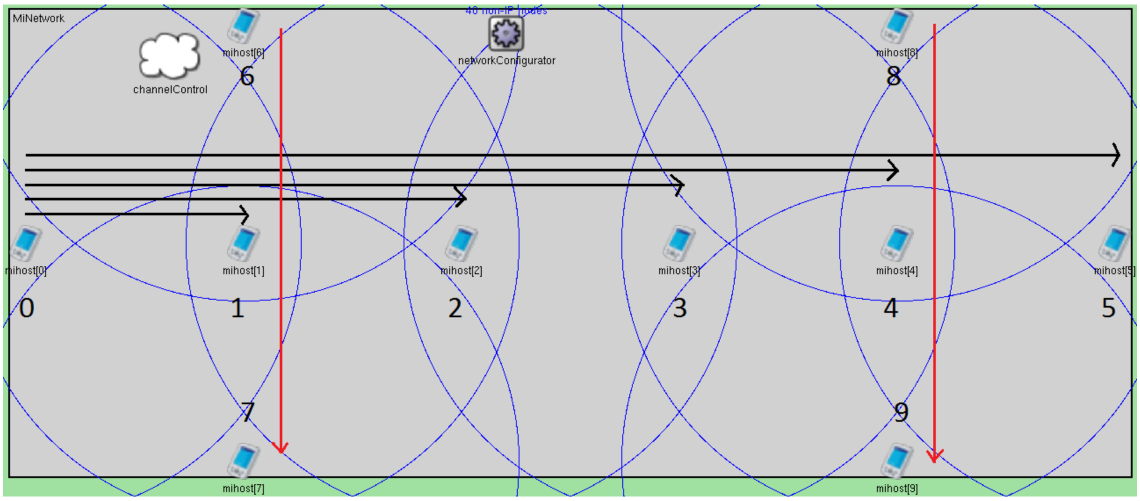


Figure A.8: Scenario for testing a single-radio mesh route

A.5.3.1 *Configuration*

New parameters, which haven't been listed so far in subsections A.5.1 and A.5.2, as well as those which have been updated, are listed in table A.11. All parameters listed are overtaken in Chapter 3. Explicit modifications, or amendments to these parameters will be separately indicated in tables in Chapter 3. OLSR is deployed, using the hop count metric. This avoids that packets on the focus route are passing one of the disturbing nodes `mihost[6-9]`, due to better link states. As in A.9, a 30s warm up period is granted, so OLSR can establish all routes. Standard MPR scheme

Table A.11: Parameters for multi-hop route simulation

MODULE	PARAMETER	TYPE	VALUE
(general)	`sim-time-limit`	Constant	100s
`manetrouting`	`routingProtocol`	String	OLSR
	`Hello_ival`	Constant	2
	`Tc_ival`	Constant	5
	`Mid_ival`	Constant	5
	`Mpr_algorithm`	Constant	1
	`routing_algorithm`	Constant	1
	`Link_quality`	Constant	2
	`Fish_eye`	Boolean	false
	`Link_delay`	Boolean	false
`tcp`	`advertisedWindow`	Constant	65535
	`mss`	Constant	1024
	`tcpAlgorithmClass`	String	TCPReno
(network)	`numMIHosts`	Constant	10
	`playgroundSizeX`	Constant	720m
	`playgroundSizeY`	Constant	300m
	`mobilityType`	String	NullMobility
	node distance	Constant	140m
(custom)	`queueModule`	String	“”
	`schedulingAlgorithm`	String	none
(MIMC)	`numMIRadios`	Constant	1

is used; TC messages are sent out every 5 seconds and HELLO messages every 2 seconds. Intervals were overtaken as default settings from [15].

General TCP settings are described in [155]. The MSS has been increased to 1024 bytes (advertised window is $64 * \mathrm{MSS}$), which improves the maximum throughput between two nodes to approx. 11 Mbit/s. There are no general settings required for UDP.

The custom section in table A.11 lists settings of the layer 2.5 module. LMHPC is disabled, so the used nodes (carrying the custom module) react like regular single-radio mesh nodes.

TCP "Session" [156] app is used as packet generator and TCP "Sink" app [157] is used as a receiver in this scenario. Main parameters are listed in table A.12. Session app opens a single TCP connection and sends the specified amount of data and closes afterwards.

Table A.12: Exemplary parameters for TCP session and sink app

MODULE	PARAMETER	TYPE	VALUE
(host)	`numTcpApps`	Constant	1
	`tcpAppType`	String	TCPSessionApp
`tcpApp`	`active`	Boolean	true
	`address`	String	""
	`connectAddress`	String	*Variable*
	`connectPort`	Constant	1000
	`port`	Constant	1000
	`sendBytes`	Constant	20MB
	`tClose`	Constant	150s
	`tOpen`	Constant	29s
	`tSend`	Constant	uniform(34s, 36s)
	`rng-0`	Constant	1
(host)	`numTcpApps`	Constant	1
	`tcpAppType`	String	TCPSinkApp
`tcpApp`	`address`	String	""
	`port`	Constant	1000
	`connectPort`	Constant	1000

To evaluate the achievable throughput and minimum delay with UDP on the focus route, UDP "Video Stream Server [158] and - Client [159]" apps are used (see table A.13). The UDP connection simulates video streaming, where the packet length (typically low) and the video size is specified. TX rate is calculated with $\frac{\mathtt{packetLen}}{\mathtt{waitInterval}}$ (3.2 Mbit/s in this case).

Table A.13: Exemplary parameters for UDP video stream-server and stream-client app

MODULE	PARAMETER	TYPE	VALUE
(host)	`numUdpApps`	Constant	1
	`udpAppType`	String	UDPVideoStreamSvr
`udpApp`	`packetLen`	Constant	512B
	`serverPort`	Constant	3088
	`videoSize`	Constant	20MB
	`waitInterval`	Constant	1302us
(host)	`numUdpApps`	Constant	1
	`udpAppType`	String	UDPVideoStreamCli
`udpApp`	`serverAddress`	String	*Variable*
	`serverPort`	Constant	3088
	`localPort`	Constant	9999
	`startTime`	Constant	$\mathrm{uniform}(34s, 36s)$
	`rng-0`	Constant	2

In parallel to the UDP stream, pings are constantly emitted by the "Ping" app [160]. ICMP payload equals UDP datagram size. Ping app is set up with parameters from table A.14.

The complete configuration file of the simulation can be found in the code attachments (disc).

VARIABLES AND EVENT TIMELINE

A single TCP connection between `mihost[0]` and each member of the focus route is measured (black arrows in Fig. A.8). `connectAddress` specified at `mihost[0]` starts with `mihost[1]` and is increased by one hop until `mihost[5]` is reached. Same con-

Table A.14: Exemplary parameters for ICMP Ping app

MODULE	PARAMETER	TYPE	VALUE
`pingApp`	`count`	Constant	20
	`destAddr`	String	*Variable*
	`hopLimit`	Constant	32
	`interval`	Constant	1s
	`packetSize`	Constant	512B
	`printPing`	Boolean	false
	`startTime`	Constant	uniform(34s, 36s)
	`rng-0`	Constant	3
	`stopTime`	Constant	0s

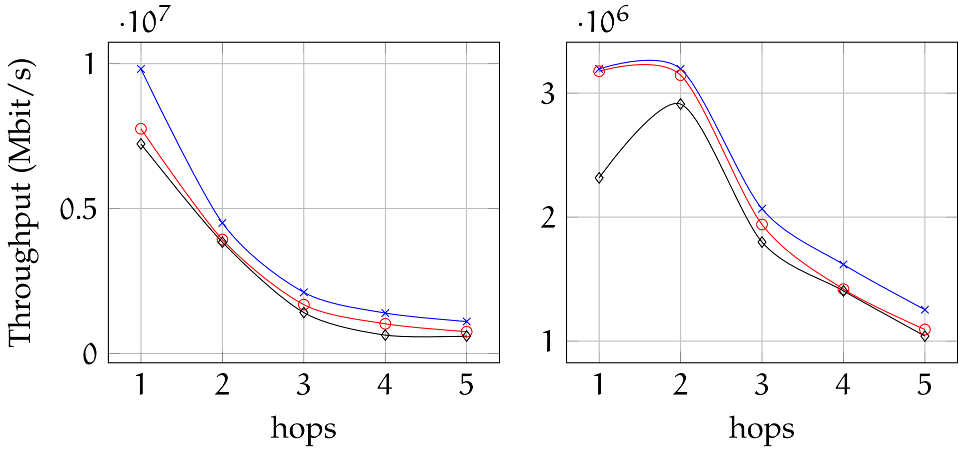
stellation is used for UDP, using the streaming app. There is either no, or one CT pair (`mihost[6]` to `mihost[7]`) or two cross-traffic pairs (`mihost[8]` to `mihost[9]`) active (TCP, red arrows). At first, at t=28s, CT starts (when active). Afterwards, around t=35s, the focus stream attempts to transfer data.

A.5.3.2 *Results*

Average throughput and RTT results are shown in Figure A.9.

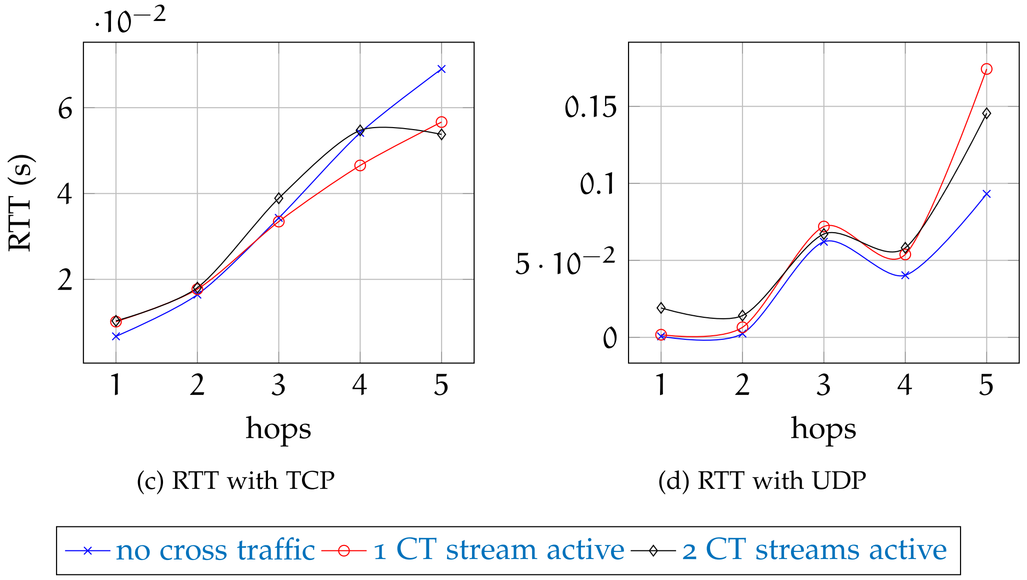
To better highlight the impact of the two cross-streams, Fig. A.10 and Fig. A.11 show the packet distribution between nodes of the TCP focus stream to `mihost[5]`, when no CT is active and when both CT streams are active. Data is obtained from a *single run* for Fig. A.10 and A.11. The blue bar (sent after retry) is the result of a previously initiated, unknown number of TX retries (might be zero retries as well), which was finally successful, and is independent from the packets which were directly send without retry.

To understand why packet TX amounts of the focus nodes are so drastically worsened when 2 CT streams are running, throughput of cross-traffic has to be taken into account. As revealed in Fig. A.12, both streams always achieve throughput levels of approx. 4 Mbit/s over a 2-hop distance.



(a) TCP throughput

(b) UDP throughput (@ TX rate 3.2 Mbit/s)



(c) RTT with TCP

(d) RTT with UDP

Figure A.9: UDP and TCP throughput with single-radio connection in a standard WMN

A.5.3.3 *Evaluation*

Looking at the throughput and delay in Fig. A.9, the increasing hop distance has the most severe impact on performance [58]. The significant impact of hop count on mesh performance plays a continuous role in Chapter 3. As shown in Fig. A.9, it affects both UDP and TCP. In the case of TCP with no CT, the available capacity is halved between first and second hop receiver. The lack of full-duplex transmission causes intra-flow interference, which is most severe on the first three hops. This issue needs to be resolved by using additional channels. With the presented system,

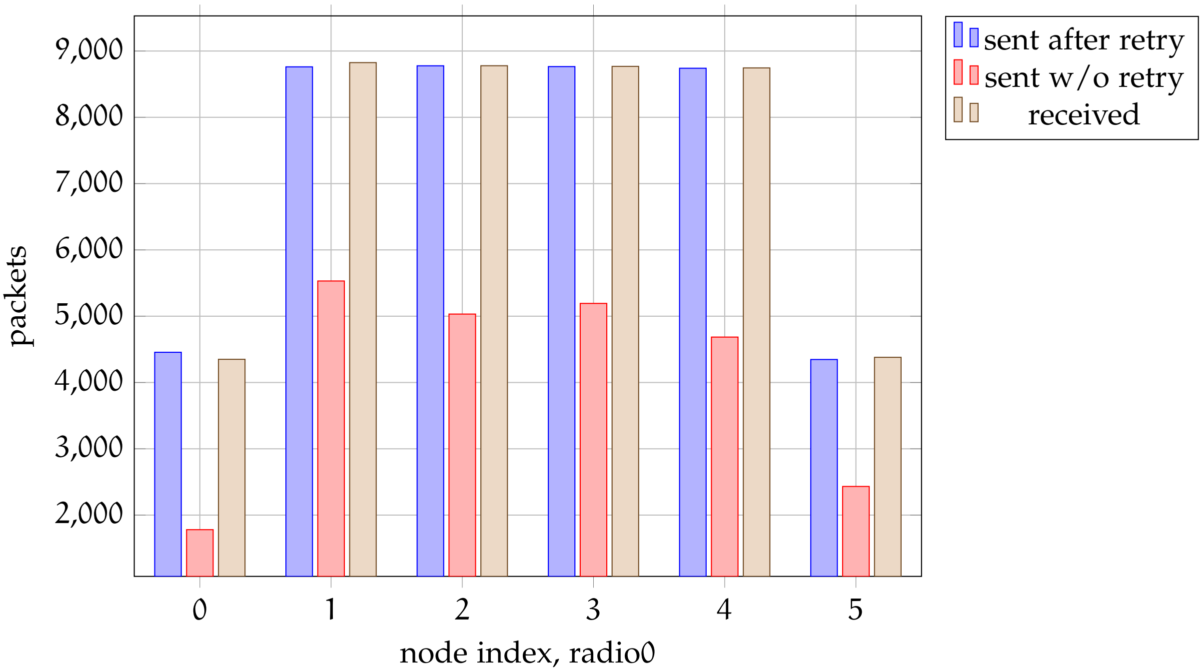


Figure A.10: Packet distribution of each hop when no cross-traffic is active

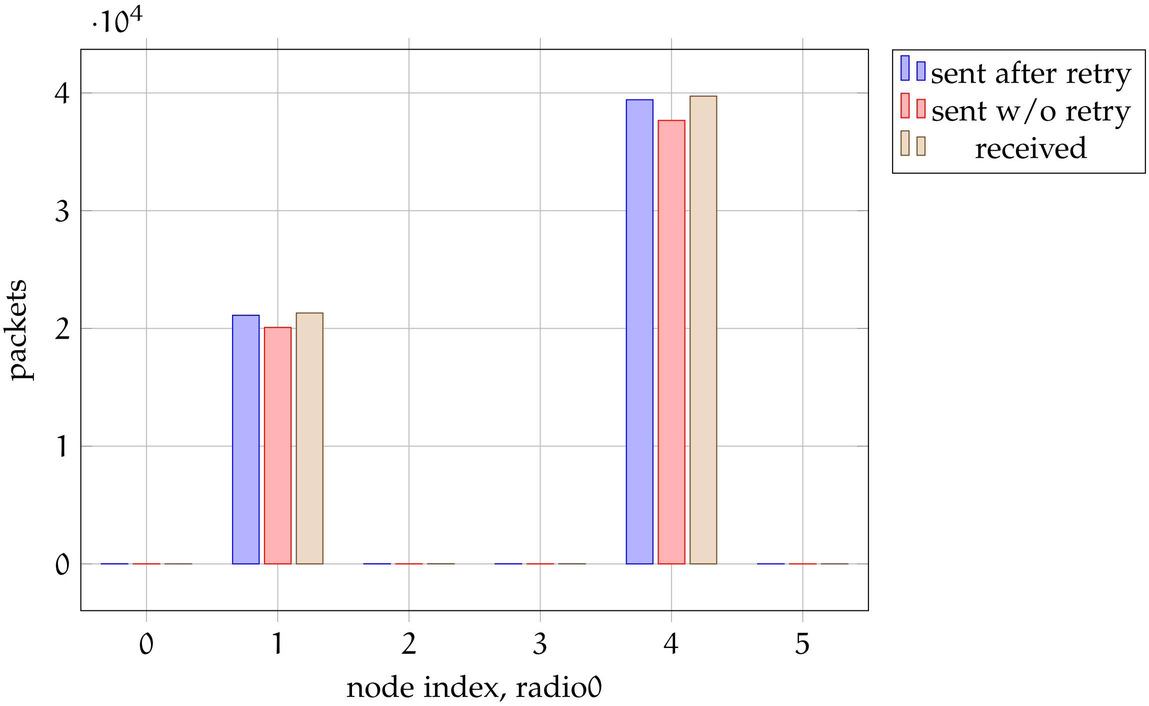


Figure A.11: Packet distribution of each hop when two cross-traffic streams are active

a node can sense unsuitability of links between neighbors and use less congested links instead, which is enabled by WFS and Extended RR PS modes.

Cross-traffic takes more influence on TCP throughput than on UDP at the first hop. UDP is most affected by 1 and 2 CT streams at the first two hops.

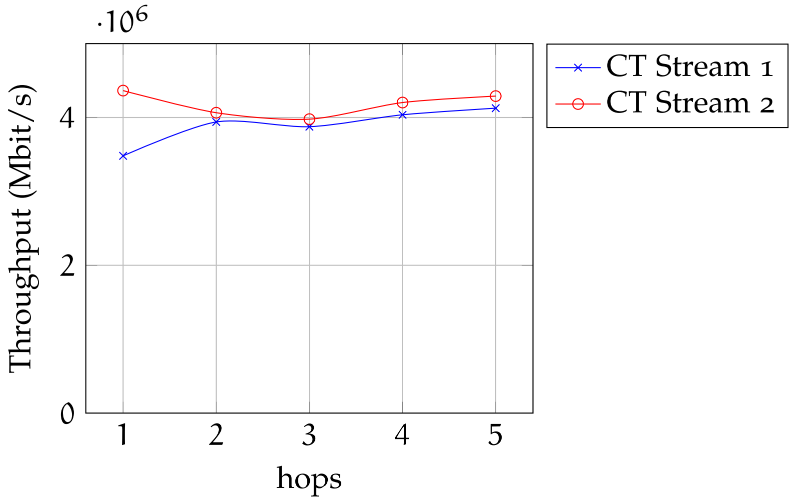


Figure A.12: TCP throughput of cross-traffic

The impact of CT is now interpreted with packet distributions in figures A.10 and A.11. Fig. A.10 depicts how TCP is still well functioning over 5 hops, since the amounts of sent and received packets at `mihost[0]` and `mihost[5]` are similar (TCP signaling - especially ACKs - is taken into account). Intermediate hops process more packets (and TX retries), due to higher levels of congestion caused by multi-hop forwarding. Fig. A.11 depicts the case when both CT streams are active. The focus stream in the chosen run offers a very poor performance in comparison to both CT streams; in fact, in Fig. A.11 no TCP connection could be established with the fifth hop in Fig. A.9a. It is decisive that both CT streams start earlier than the focus stream. It is clearly visible that `mihost[1]` and `mihost[4]` are mostly occupied with processing cross-traffic, which blocks the focus route. This does not change during the entire CT transmission. When the presented system is applied, nodes in coverage range of `mihost[1]` and `mihost[4]` shall avoid the occupied channel 0 and use others instead, so CT and focus stream can coexist.

A.5.4 *IP Processing as a Discrete Event*

The listing A.5 contains an extract of the OMNeT++ event log. A single ping ICMP packet enters the IP layer, is rerouted and leaves it again. The log reveals that the processing is discrete, because no time is spent for the rerouting.

Listing A.5: Event log for a single packet entering and leaving the IP layer

```
** Event #4960  T=30.000214466666  MiNetwork.mihost[1].wlan[0].mgmt (Ieee80211MgmtAdhoc, id=55), on ‘ping0’ (Ieee80211DataFrame, id=3404)
Frame arrived from MAC: (Ieee80211DataFrame)ping0
** Event #4961  T=30.000214466666  MiNetwork.mihost[1].scheduler (Scheduler, id=36), on ‘ping0’ (IPDatagram, id=3405)
RECEIVING: MY HOST:: mihost[1]
###### sending message to upper layer in gate: 0

# Packet is leaving scheduler module at T=30.000214466666 and is handed to IP layer:

** Event #4962  T=30.000214466666  MiNetwork.mihost[1].networkLayer.ip (IP, id=49), on ‘ping0’ (IPDatagram, id=3405)
Routing datagram ‘ping0’ with dest=10.0.4.1: output interface is wlan0, next-hop address: 10.0.3.1
** Event #4963  T=30.000214466666  MiNetwork.mihost[1].manetrouting.manetmanager (ManetManager, id=54), on ‘{}’ (ControlManetRouting, id=3414)
** Event #4964  T=30.000214466666  MiNetwork.mihost[1].networkLayer.arp (ARP, id=50), on ‘ping0’ (IPDatagram, id=3405)

# Packet is leaving IP layer and is scheduled again, at T=30.000214466666:

Packet (IPDatagram)ping0 arrived from higher layer, using next-hop address 10.0.3.1
** Event #4965  T=30.000214466666  MiNetwork.mihost[1].scheduler (Scheduler, id=36), on ‘ping0’ (IPDatagram, id=3405)
SENDING: IP SRC ADDRESS:: 10.0.1.1 IP DST ADDRESS:: 10.0.4.1
Unknown Mode!
# sending UNICAST msg on radio index:: 0
** Event #4966  T=30.000214466666  MiNetwork.mihost[1].wlan[0].mgmt (Ieee80211MgmtAdhoc, id=55), on ‘ping0’ (IPDatagram, id=3405)
Packet arrived from upper layers: (IPDatagram)ping0
** Event #4967  T=30.000214466666  MiNetwork.mihost[1].wlan[0].mac (Ieee80211gMac, id=56), on ‘ping0’ (Ieee80211DataFrame, id=3415)
->Enter handleUpperMsg...
frame (Ieee80211DataFrame)ping0 received from higher layer, receiver = 0A-AA-00-00-00-03
deferring upper message transmission in WAITSIFS state
->Leave handleUpperMsg...
```

Listing A.6: Dynamic channel switch with the scenario manager

```
<?xml version="1.0"?>
<scenario>
        <at t="30">
                <change-channel module="mihost[1].wlan[1].mgmt">
                        <newChannelNum>1</newChannelNum>
                </change-channel>
                <change-channel module="mihost[1].wlan[2].mgmt">
                        <newChannelNum>2</newChannelNum>
                </change-channel>
        </at>
        <at t="70">
                <change-channel module="mihost[1].wlan[1].mgmt">
                        <newChannelNum>3</newChannelNum>
                </change-channel>
                <change-channel module="mihost[1].wlan[2].mgmt">
                        <newChannelNum>4</newChannelNum>
                </change-channel>
        </at>
</scenario>
```

A.5.5 *Channel Map Input for Scenario Manager*

Listing A.6 allows to dynamically switch channels during run-time. The OMNeT++ `scenario.Manager` requires a Extensible Markup Language (XML) input format.

DECLARATION

I declare that this dissertation is the product of my own work, that it has not been submitted before for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

Havana, April 2015

Christian Köbel