Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Verteilte Systeme
Prof. Dr. P. Tran-Gia

Performance Optimization of Wireless Infrastructure and Mesh Networks

Jan Rastin Pries

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 01/10

Würzburger Beiträge zur Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Verteilte Systeme
Am Hubland
D-97074 Würzburg

Tel.: +49-931-31-86630
Fax.: +49-931-31-86632
email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage vom Autor.
Gesetzt in LaTeX Computer Modern 9pt.

ISSN 1432-8801

Performance Optimization of Wireless Infrastructure and Mesh Networks

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius–Maximilians–Universität Würzburg

vorgelegt von

Jan Rastin Pries

aus

Itzehoe

Würzburg 2010

Eingereicht am: 21.12.2009
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr.-Ing. P. Tran-Gia
2. Gutachter: Prof. Y. Koucheryavy, PhD
Tag der mündlichen Prüfung: 12.02.2010

Danksagung

Die vorliegende Arbeit ist das Ergebnis fünfjähriger Forschung im Bereich der drahtlosen Kommunikationsnetze. Auch wenn ich über die Technik der drahtlosen Kommunikation geschrieben habe, ist für mich persönlich das Wichtigste immer noch die persönliche, zwischenmenschliche Kommunikation. Diese Arbeit wäre nicht ohne die tatkräftige Unterstützung und die zahlreichen Diskussionen mit denjenigen Personen möglich gewesen, denen ich im Folgenden danken möchte.

Mein herzlichster Dank geht an meinen Doktorvater und Betreuer dieser Arbeit, Prof. Dr.-Ing. Phuoc Tran-Gia. Seiner unermüdlichen Unterstützung meiner Forschung und den vielen Diskussionen habe ich diese Arbeit zu verdanken. Dabei kam die Menschlichkeit von Prof. Tran-Gia nie zu kurz, was auch der Grund für das sehr gute und kollegiale Klima am Lehrstuhl ist. Dieses Klima wurde durch zahlreiche gemeinsame Dienstreisen ins In-und Ausland und die Eigenverantwortung bei Klein- und Großprojekten weiter gefördert und führten zu einer sehr hohen Motivation. Dabei stellte Prof. Tran-Gia häufig den Kontakt zwischen hochrangigen Wissenschaftlern und mir her, wodurch ich nicht nur meine eigene Arbeit besser einschätzen konnte, sondern wodurch sich auch gemeinsame Veröffentlichungen ergeben und Freundschaften entwickelt haben.

This leads me to the second reviewer of my thesis, Prof. Yevgeni Koucheryavy, whom I got to know and to appreciate during one of my business trips and whom I like to express my deep gratitude for the suggestions and comments on my thesis. Danken möchte ich auch Prof. Dr. Dietmar Seipel und Prof. Dr. Reiner Kolla, die mir als Prüfer bei meiner Disputation auf den Zahn gefühlt haben.

Meine Arbeit am Lehrstuhl wurde aber vor allem von Frau Gisela Förster erleichtert, die durch ihre intensive Unterstützung bei der Verwaltungsarbeit, vor allem beim Projekt G-Lab, mir viel Aufwand erspart hat. Weiterhin möchte ich mich bei Dr. Gerald Kunzmann, Dr. Nico Bayer, Dr. Matthias Siebert, Dr. Alexey Vinel, Dr. Tuo Liu und Prof. Dr. Nguyen Huu Thanh für die gute Zusammenarbeit in den Forschungsprojekten bedanken. Zusätzlich gab es immer wieder mit vielen weiteren Kollegen aus den Projekten G-Lab, Euro-NF, France Telecom, EADS, Datev und Bertelsmann Arvato rege Diskussionen, die mir ein erfolgreiches wissenschaftliches Arbeiten ermöglicht haben.

Mein besonderer Dank geht an Dr. Dirk Staehle, der mir mit fortwährender Geduld immer neue Lösungsansätze aufgezeigt hat und mir während meiner Promotionszeit nicht nur fachlich, sondern auch privat stets beiseite stand. Neben Dr. Dirk Staehle möchte ich mich natürlich bei meinen anderen Kollegen, allen voran Robert Henjes, bedanken. Nur durch das Miteinander aller am Lehrstuhl kann erfolgreich gearbeitet werden. Deshalb möchte ich an dieser Stelle den folgenden Kollegen und ehemaligen Kollegen meinen aufrichtigen Dank aussprechen: Dr. Andreas Binzenhöfer, Michael Duelli, Matthias Hartmann, Dr. Klaus Heck, Robert Henjes, Matthias Hirth, David Hock, Dr. Tobias Hoßfeld, Michael Jarschel, Alexander Klein, Dominik Klein, Dr. Stefan Köhler, Frank Lehrieder, Dr. Kenji Leibnitz, Dr. Andreas Mäder, Dr. Rüdiger Martin, Dr. Michael Menth, Dr. Jens Milbrandt, Simon Oechsner, Prof. Dr. Oliver Rose, Daniel Schlosser, Barbara Staehle, Dr. Dirk Staehle, Prof. Dr. Kurt Tutschku, Dr. Norbert Vicari, Florian Wamser und Thomas Zinner. Obwohl Florian Zeiger und Markus Sauer an einem anderen Lehrstuhl, bzw. am Telematikzentrum arbeiten, zähle ich sie zum Lehrstuhl III hinzu und möchte mich auch bei Ihnen für die unzähligen Abende und gemeinsamen Urlaube bedanken.

Besonders wichtig für meine Arbeit war auch die Hilfe meiner Diplomanden nötig gewesen. Diese sollen natürlich nicht ungenannt bleiben: Alexander Klein, Matthias Wiesen, Daniel Marsico, Thorsten Gutbrod, Stefan Menth, David Hock, Marieta Stoykova, Korhan Canbolat, Florian Wamser, Viktor Wendel, Matthias Hirth und Desheng Fu. Es ist schön zu sehen, dass fünf der ehemaligen Diplomanden mittlerweile promovieren. Zusätzlich möchte ich mich bei alle Studenten bedanken, die mich bei der Projektarbeit und der Betreuung von Vorlesungen unterstützt haben.

Zum Schluss möchte ich mich ganz besonders bei meiner Familie bedanken, meiner Mutter Berte Pries, meiner Oma Margarete Pries, meinem Vater Rolf Pries und seiner Frau Ursula, sowie meinem Bruder Gerrit Pries und seiner Frau Yvonne. Ohne deren finanzieller und moralischer Unterstützung während des Studiums und der Promotionszeit wäre diese Arbeit nicht zustande gekommen. Meinen tiefsten Dank möchte ich meiner Partnerin Anne Loosen aussprechen, die mir trotz aller eigener Aufgaben und trotz vieler einsamer Abende während meiner unzähligen Dienstreisen und Abende am Lehrstuhl stets beiseite gestanden hat.

Contents

1 Introduction **1**

1.1 Scientific Contribution . . . 3

1.2 Outline of Thesis . . . 6

2 Performance Evaluation and Optimization of Infrastructure Networks **9**

2.1 Background: The WLAN Protocol . . . 10

2.1.1 Network Architectures . . . 11

2.1.2 Physical Layer . . . 12

2.1.3 Medium Access Control . . . 13

2.2 Unfairness in Infrastructure Networks . . . 22

2.2.1 Related Work . . . 22

2.2.2 Introduction to the Unfair Channel Access Phenomenon . 27

2.2.3 Unfairness of the DCF . . . 30

2.2.4 Unfairness of the EDCA . . . 38

2.3 Dynamic Contention Window Adaptation . . . 42

2.3.1 Related Work . . . 43

2.3.2 DCWA Algorithm . . . 44

2.3.3 DCWA Parameter Evaluation . . . 49

2.3.4 Performance of the DCWA Algorithm . . . 59

2.4 Impact of Best Effort Frame Bursting . . . 63

2.4.1 Related Work . . . 63

2.4.2 Throughput Improvement Through Frame Bursting in a Saturated Environment 65

2.4.3 Impact of Frame Bursting in a Realistic Scenario 67

2.5 Lessons Learned 73

3 Quality of Experience Control in WLAN Mesh Networks 75

3.1 Wireless Mesh Networks 76

3.1.1 WMN Architecture 77

3.1.2 Routing in Wireless Mesh Networks 78

3.1.3 Challenges of QoS Support in WMNs 81

3.1.4 Related Work 82

3.2 Traffic Observation and Control in WMNs 86

3.2.1 Traffic Observation 86

3.2.2 Wireless Mesh Traffic Control 92

3.3 Performance Measurements 95

3.3.1 In-Band Traffic Disturbance 96

3.3.2 Out-Band Traffic Disturbance 99

3.4 Simulative Performance Evaluation 102

3.4.1 Validation of the Testbed Approach 104

3.4.2 Throughput Optimization 106

3.4.3 Performance Optimization of the Traffic Controller 107

3.5 Lessons Learned 113

4 Planning and Optimization of Wireless Mesh Networks 115

4.1 Network Planning and Optimization Issues 116

4.1.1 Wireless Network Planning 116

4.1.2 Global Optimization Techniques 118

4.1.3 Related Work 122

4.2 WMN Planning Using Genetic Algorithms 125

4.2.1 Problem Formulation 125

4.2.2 Fairness and Capacity in Wireless Mesh Networks 125

4.2.3 Optimization Using Genetic Algorithms 129
4.2.4 Mutation 137
4.3 Performance Evaluation 138
4.3.1 Simulation Settings 138
4.3.2 Influence of Fitness Function 140
4.3.3 Elite Set Size 142
4.3.4 Population Evolution 143
4.3.5 Effectiveness of Crossover 145
4.3.6 Effectiveness of Mutation 146
4.4 Optimization of the WMN Planning Approach 148
4.4.1 Influence of the Crossover on the GA Progress 148
4.4.2 Influence of the Mutation Operator Depending on the GA Progress 150
4.4.3 Local Optimization and Population Size 152
4.5 Lessons Learned 154

5 Conclusions **157**

Bibliography and References **163**

1 Introduction

"The last thing one knows when writing a book is what to put first."
Blaise Pascal (1623-1662): Pensées (1670).

Over the last years, several wireless technologies evolved that are designed to offer Internet access anywhere and anytime. This access is facilitated over wireless technologies like Wireless Local Area Networks (WLANs) based on the IEEE 802.11 standard, Worldwide Interoperability for Microwave Access (WiMAX), and Long Term Evolution (LTE). An increasing number of laptops, personal digital assistants, and mobile phones are equipped with these technologies enabling broadband wireless Internet access with data rates of 54 Mbps and beyond.

The common access technology is a Wireless Local Area Network. In WLANs, an Access Point (AP) connects the end user and the Internet, providing so-called infrastructure access. It is designed to transport best effort traffic to places where wired connections cannot be deployed or are too expensive. Although WLAN provides a cost-efficient alternative to traditional wired access, wireless resources are limited and the standard does not provide Quality of Service (QoS) guarantees for the end user. An efficient resource management is needed that does not only ensure QoS for real-time services but also enhances the overall throughput while still sharing the resources fairly among best effort users. In LTE or WiMAX networks, a central control unit is responsible for fair sharing of the resources among end users. In contrast, the general WLAN access is based on contention; the Access Point competes for medium access in a similar way as the end user.

In addition to the WLAN infrastructure mode, a new access structure has been lately introduced called Wireless Mesh Networks (WMNs). As opposed to traditional wireless networks, data flows in a WMN traverse several hops to connect to the Internet. This enables extensive coverage areas where no wired Internet connection is required and extends wireless local area networks to wireless metropolitan area networks since complete cities can be covered with only a few gateways to the Internet. In such a WMN, the data is sent directly from neighbor node to neighbor node and does not have to be relayed over the Access Point. For data between nodes that are not directly reachable neighbors, the packets are sent on a multi-hop route. All nodes provide relaying capabilities to forward traffic through the network towards the destination. WMNs consist normally of static devices and focus on robustness and reliability. Compared to WLAN infrastructure networks, the number of Access Points with cost-intensive Internet access connection can be reduced.

WMNs should have the so-called self-x-properties, i.e., they are self-organizing, self-configuring, and self-healing and are thus gaining an increasingly important role in next generation wireless networks. However, since WMNs usually do not have a central control unit, it is even more difficult to guarantee QoS. QoS can only be guaranteed if either a path is exclusively reserved or distributed bandwidth control management is performed in each node of the WMN. This requires that the QoS metrics are measured and evaluated at each node. A small change of the QoS metrics like packet loss, delay, and jitter has a significant impact on the Quality of Experience (QoE), a subjective measure from the user perspective of the overall value of the provided service or application.

In addition to the application of a distributed traffic observation and controlling scheme at runtime, the performance of the WMN can already be optimized during the planning process. The goal is to find a near-optimal routing and channel allocation to enhance the network throughput while fairly sharing the resources. This becomes very complex in large wireless mesh networks where each node can be equipped with several network interfaces and a trade-off has to be found between the number of hops to the Internet and the maximum supported data rate between two nodes.

1.1 Scientific Contribution

This monograph covers three different aspects of performance optimization in broadband wireless networks. Firstly, we address WLAN infrastructure networks and consider QoS, fairness, and throughput aspects. Secondly, we address wireless mesh networks and introduce a distributed bandwidth control approach to support QoS, and we finally focus on planning and optimization of WMNs.

Figure 1.1 gives an overview of the contribution of this work. The individual research studies carried out during the course of this work are classified according to their primarily used methodology on the x-axis and their main focus or primarily investigated technology on the y-axis. The methodologies can be classified into mathematical analysis, simulations, measurements, and design and optimization of new mechanisms and network structures. The respective focus of the research studies cover WiMAX, WLAN, Vertical Handover (VHO), cellular networks, and P2P systems. The markers $[x]^y$ indicate the scientific publications which provide the basis for Chapter y.

The first part covers WLAN infrastructure networks. The contention-based channel access of WLAN infrastructure networks does not ensure a fair resource sharing among the users. Stations with a larger traffic demand experience longer delays due to extensive queuing but they are favored in terms of channel access delay and number of collisions on the air interface. We design a novel analytical model to evaluate this kind of unfair channel access in different scenarios and validate the results by means of simulation. Our findings show that the network load cannot be estimated by measuring at the Access Point only. Therefore, we implement a feedback mechanism where all stations transmit their current network status to the Access Point.

The status of the network is required to guarantee QoS for real-time users because all stations including the Access Point have to content for medium access. Although real-time stations have a higher priority for medium access compared to best effort stations, they still have to compete against other real-time stations. We show that an adjustment of the channel access parameters based on the feedback

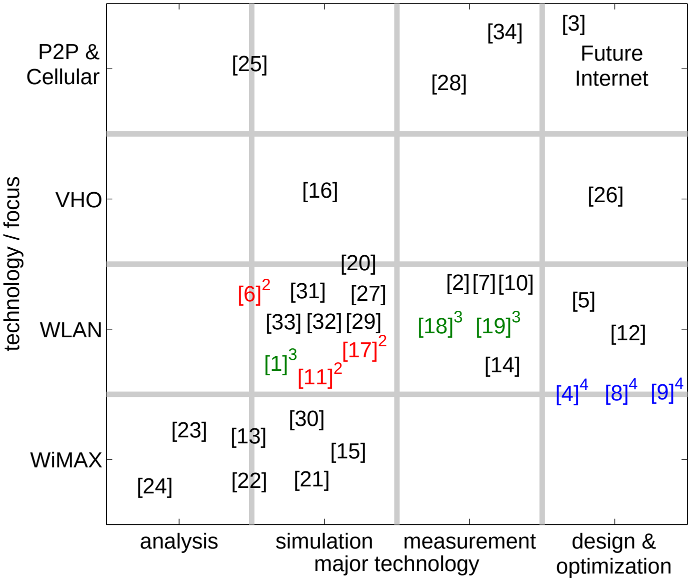


Figure 1.1: *Contribution of this work illustrated as a classification of the research studies conducted by the author. The notation* $[x]^y$ *indicates that the scientific publication* $[x]$ *is discussed in Chapter* y *of this monograph.*

of all stations helps to increase the WLAN capacity for real-time traffic. Based on the average number of retransmissions of each station, the values of the channel access parameters are increased or decreased if a threshold is exceeded. In order to keep the prioritization between real-time and best effort users on the same level, the channel access parameters of all service classes are adjusted equally. This might result in throughput degradation of best effort users. We show that transmission bursts help them to improve their throughput without violating the QoS requirements of voice users.

A similar approach can be applied in wireless mesh networks which is the second aspect of this work. Due to the fact that a WMN might have several gateways to the Internet, the parameter adjustments cannot be initiated by one single node like the Access Point. We therefore develop a distributed approach where each node in the network measures the QoS parameters and maps them to the subjective QoE. The node reacts to quality problems by dynamically adjusting the bandwidth of best effort flows in the own node, or if the cause of the quality degradation is outside the node, by informing neighboring nodes about the quality problems. The bandwidth control mechanism is first evaluated in a WMN testbed and then optimized by means of simulation. This is the second aspect of this work.

The performance of the bandwidth control mechanism can be increased a priori by carefully planning the wireless mesh network which is the third aspect of this work. Our goal is to increase the throughput of the complete WMN while sharing the resources fairly among the nodes. The planning of WMNs is in contrast to traditional wireless networks much more complex. On the one hand, a WMN consists of a multi-hop structure where not only interference on neighboring paths but also self-interference occurs. On the other hand, each node in the network can be equipped with multiple interfaces operating on different channels. The interference problems are covered by using the concept of collision domains. For the routing and channel allocation, an optimization method is required which is fast enough to optimize even large WMNs. We decided to use Genetic Algorithms (GAs) which are based on the idea of natural evolution by simulating the biological cross of genes. Although GAs are generally not able to find the best solution, they provide near-optimal results in relatively small computation time. We adjust the genetic algorithms for routing and channel allocation in WMNs and optimize the GA operators to minimize the time for the evaluation. Fairness is achieved by applying a max-min fair share algorithm and the throughput is increased by tuning the genetic parameters.

1.2 Outline of Thesis

The organization of this monograph is shown in Figure 1.2. Each chapter contains a section that shows the background and related work of the covered aspects and summarizes the lessons learned. The three columns cover from left to right (1) the problems and challenges, (2) the algorithms and mechanisms to cope with them, and (3) the impact of the applied mechanisms on the performance of the wireless network. The arrows between the sections show their relation and the background and findings which are used in later sections. The section numbers of the building blocks are given in parentheses.

The rest of this thesis is organized as follows. All topics covered in Chapter 2 are on infrastructure WLANs. First, we evaluate the WLAN unfair channel access phenomenon in scenarios with and without service differentiation for an increasing number of users and for different traffic types. The detection of the unfair channel access provides the basis for the Dynamic Contention Window Adaptation (DCWA) algorithm. This algorithm achieves resource efficiency by choosing an appropriate contention window with respect to the current channel contention. Thus, the DCWA optimizes the resources available to real-time flows and guarantees a certain QoS level. However, as a result, QoS guarantees can be provided for real-time traffic, but low priority best effort flows are prone to starvation. Frame bursting is a way to prevent best effort flow starvation without disturbing real-time voice traffic. We take a look at the influence of such a frame bursting scheme and we show that frame bursting effectively mitigates best effort flow starvation with regard to voice traffic QoS requirements. Using extended transmission bursts, best effort flows considerably benefit by an increased channel utilization through reduced protocol overhead and by the use of free resources not needed for voice traffic.

Chapter 3 focuses on wireless mesh networks. We develop a QoE control algorithm where the quality of service parameters are measured at each mesh node. If the quality of real-time traffic expressed through the Mean Opinion Score (MOS) cannot be guaranteed, the maximum available bandwidth for low priority best effort traffic is reduced. In two different testbed scenarios, we show that the mechanism successfully protects real-time flows from interfering with best effort flows. At the end of the chapter, we improve the approach by means of simulation adjusting the bandwidth limitation dynamically and including the mechanisms from Chapter 2.

Chapter 4 investigates the planning and optimization of wireless mesh networks. First, different optimization techniques are introduced and the related work, where these techniques are applied, is shown. We decided to use genetic optimization as they are able to solve this planning and optimization approach because of their simplicity and ability to optimize even large WMN scenarios. Therefore, the WMNs are encoded using a list structure. On this list structure, the two different genetic operators, crossover and mutation, are applied which we especially design for the routing and channel allocation of WMNs. We investigate the impact of every step of the genetic algorithm's workflow on the resulting network solution. Finally, we introduce the concept of local optimization which significantly improves the performance of the WMN with minimal computational overhead. In Chapter 5, we summarize the main findings and draw conclusions.

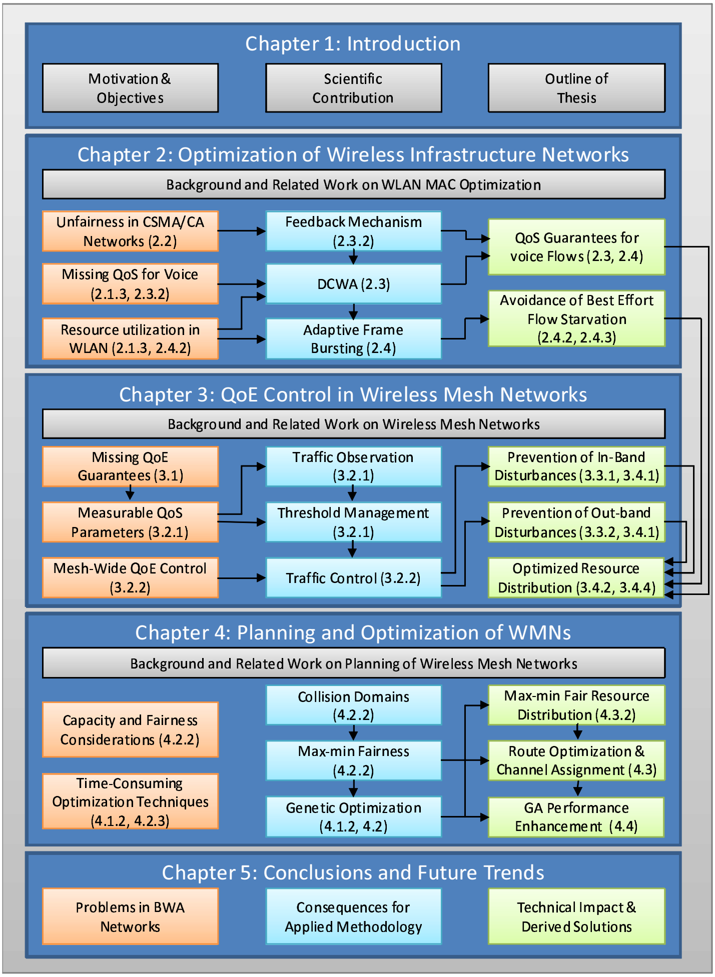

Chapter 1: Introduction
Motivation & Objectives
Scientific Contribution
Outline of Thesis
Chapter 2: Optimization of Wireless Infrastructure Networks
Background and Related Work on WLAN MAC Optimization
Unfairness in CSMA/CA Networks (2.2)
Missing QoS for Voice (2.1.3, 2.3.2)
Resource utilization in WLAN (2.1.3, 2.4.2)
Feedback Mechanism (2.3.2)
DCWA (2.3)
Adaptive Frame Bursting (2.4)
QoS Guarantees for voice Flows (2.3, 2.4)
Avoidance of Best Effort Flow Starvation (2.4.2, 2.4.3)
Chapter 3: QoE Control in Wireless Mesh Networks
Background and Related Work on Wireless Mesh Networks
Missing QoE Guarantees (3.1)
Measurable QoS Parameters (3.2.1)
Mesh-Wide QoE Control (3.2.2)
Traffic Observation (3.2.1)
Threshold Management (3.2.1)
Traffic Control (3.2.2)
Prevention of In-Band Disturbances (3.3.1, 3.4.1)
Prevention of Out-band Disturbances (3.3.2, 3.4.1)
Optimized Resource Distribution (3.4.2, 3.4.4)
Chapter 4: Planning and Optimization of WMNs
Background and Related Work on Planning of Wireless Mesh Networks
Capacity and Fairness Considerations (4.2.2)
Time-Consuming Optimization Techniques (4.1.2, 4.2.3)
Collision Domains (4.2.2)
Max-min Fairness (4.2.2)
Genetic Optimization (4.1.2, 4.2)
Max-min Fair Resource Distribution (4.3.2)
Route Optimization & Channel Assignment (4.3)
GA Performance Enhancement (4.4)
Chapter 5: Conclusions and Future Trends
Problems in BWA Networks
Consequences for Applied Methodology
Technical Impact & Derived Solutions

Figure 1.2: *Organization and contribution of this monograph according to problems and challenges, algorithms and methodologies, and impact of the applied mechanisms on the system performance.*

2 Performance Evaluation and Optimization of Infrastructure Networks

"The wireless telegraph is not difficult to understand. The ordinary telegraph is like a very long cat. You pull the tail in New York, and it meows in Los Angeles. The wireless is the same, only without the cat."
Albert Einstein (1879-1955)

Wireless infrastructure networks consist basically of a number of stations connected to the Internet via an Access Point (AP) traversing just one single hop. Such networks provide a convenient access for users at home and in public facilities, and emerge more and more in everyday life. The most common equipment is built upon the Institute of Electrical and Electronics Engineers (IEEE) 802.11 [35] standard, which is widely referred to as Wireless Local Area Network (WLAN). The first WLAN standard was released in 1997 [36] and it gradually improved its performance and evolved into a very flexible and well understood technology [35]. Today, this wireless technology is a standard equipment in laptops and other portable and mobile devices.

In the following, we give an insight into the WLAN IEEE 802.11 standard and identify problems and unfairness aspects of the WLAN channel access. Afterwards, we show how the channel access can be optimized in order to enhance the throughput while still keeping Quality of Service (QoS) requirements.

2.1 Background: The WLAN Protocol

The IEEE 802.11 standard defines the Medium Access Control (MAC) and the PHYsical (PHY) layer of the ISO/OSI protocol stack. Since its first release in 1997 [36] it has been continuously enhanced. Table 2.1 provides an overview of the complete IEEE 802.11 standard family.

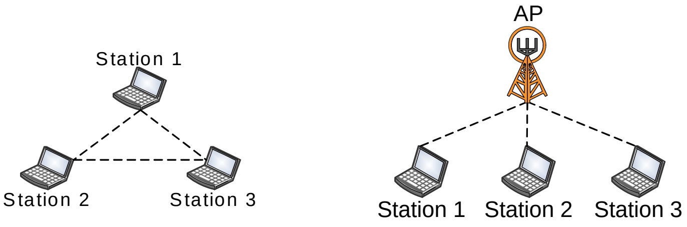
Table 2.1: *The IEEE 802.11 standard family.*

Standard	Short description
802.11	1 Mbps & 2 Mbps @ 2.4 GHz (1997)
802.11a	54 Mbps @ 5 GHz (1999)
802.11b	5.5 Mbps & 11 Mbps @ 2.4 GHz (1999)
802.11c	WLAN bridging (included in IEEE 802.11d) (2001)
802.11d	International roaming extensions (2001)
802.11e	Service differentiation (QoS support) (2005)
802.11f	Inter AP Protocol (IAPP) (2003, withdrawn 2006)
802.11g	Throughputs up to 54 Mbps @ 2.4 GHz (2003)
802.11h	Spectrum management for IEEE 802.11a (2004)
802.11i	Security enhancements (2004)
802.11j	Extensions for Japan (2004)
802.11-2007	New release including IEEE 802.11a,b,d,e,g,h,i,j (2007)
802.11k	Radio resource management enhancements (2008)
802.11n	Throughput enhancements using MIMO (2009)
802.11p	Wireless Access for Vehicular Environments (WAVE)
802.11r	Fast roaming (2008)
802.11s	Wireless mesh networking
802.11t	Wireless Performance Prediction (WPP) (canceled)
802.11u	Interworking with non IEEE 802 networks
802.11v	Wireless network management
802.11w	Protected management frames (2009)
802.11y	Operation @ 3650-3700 MHz for the US (2008)
802.11z	Extensions to Direct Link Setup (DLS)
802.11aa	Robust streaming of audio video transport streams
802.11ac	Very high throughput < 6 GHz
802.11ad	Extremely high throughput @ 60 GHz
802.11ae	QoS Management
802.11af	TV Whitespace

All amendments of the initial IEEE 802.11 standard are backwards compatible and a lot of them are still developed in task groups. In the remainder of this work, we focus on the IEEE 802.11-2007 [35] standard as it includes the main access mechanisms and the QoS enhancement.

2.1.1 Network Architectures

A WLAN can be set up using different topologies. The standard defines two modes of operation, the ad-hoc mode and the infrastructure mode. Using the ad-hoc mode, WLAN stations are allowed to directly communicate with each other, see Figure 2.1(a). The stations together form an Independent Basic Service Set (IBSS). In an infrastructure network, all communication traverses the WLAN Access Point. Thus, the AP is on the one hand responsible for relaying the traffic within the WLAN and on the other hand for forwarding traffic to the Internet. If for example Station 1 from Figure 2.1(b) wants to communicate with Station 2 from the same network, it first has to transmit the packet to the AP who then forwards the packet to its destination. Although this mode requires more wireless capacity, it might allow a communication between two stations which are not within the same coverage area. One AP together with several stations form an infrastructure Basic Service Set (BSS). Different APs connected over a wired connection form an Extended Service Set (ESS).



(a) Ad-hoc network.

(b) Infrastructure network.

Figure 2.1: *Different WLAN topologies.*

As the ad-hoc mode is not playing a major role in the context of future broadband wireless communication, we focus on the infrastructure mode in this chapter. In the subsequent chapter, we introduce another WLAN topology, the mesh mode, which is a combination of the two modes described above.

2.1.2 Physical Layer

As the main focus of this work lies on the MAC layer, the PHY layer is only shortly described here. The initial IEEE 802.11 standard [36] defines three different PHY layers in the 2.4 GHz frequency band, Frequency-Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), and Infrared. However, only the DSSS has been widely implemented. In this, the signal is modulated using either Differential Binary Phase Shift Keying (DBPSK) for data transmissions with 1 Mbps or Differential Quadrature Phase Shift Keying (DQPSK) for a data rate of 2 Mbps. These two modulation techniques are extended in the IEEE 802.11b standard [37] with Complimentary Code Keying (CCK) and Packet Binary Convolution Coding (PBCC) to achieve data rates of 5.5 Mbps and 11 Mbps, respectively.

The DSSS physical layer consists of 14 channels in the 2.4 GHz band with a channel width of 22 MHz. As the spacing between the channels is only 5 MHz and not all channels can be used in each country, only 3 non overlapping channels exist, e.g., channel 1, 6, and 11.

Concurrently to the IEEE 802.11b standard, IEEE 802.11a [38] was specified for the communication in the 5 GHz frequency band. Although it uses the same MAC layer and frame format, another PHY layer is specified, namely Orthogonal Frequency-Division Multiplexing (OFDM). This multi-carrier modulation method uses closely-spaced sub-carriers to carry the data which results in data rates of up to 54 Mbps. The modulation and coding schemes for the IEEE 802.11a standard are shown in Table 2.2.

The operation in the relatively unused 5 GHz band gives this standard an advantage over the IEEE 802.11b standard operating in the crowded 2.4 GHz band. However, the higher frequency reduces the overall range for transmissions.

Table 2.2: *Modulation and coding schemes.*

Modulation	Coding rate	Data bits per OFDM symbol	Data rate in Mbps using a 20 MHz channel
BPSK	1/2	24	6
BPSK	3/4	36	9
QPSK	1/2	48	12
QPSK	3/4	72	18
16-QAM	1/2	96	24
16-QAM	3/4	144	36
64-QAM	2/3	192	48
64-QAM	3/4	216	54

Four years after the release of the IEEE 802.11a and b standards, the IEEE 802.11g [39] was specified using a combination of both standards, the operation in the 2.4 GHz band like the IEEE 802.11b standard with the OFDM physical layer of the IEEE 802.11a standard. The IEEE 802.11g hardware is fully backwards compatible with the IEEE 802.11b hardware and was thus rapidly adopted by consumers. In the following, we introduce the basic medium access control protocol and its QoS enhancement first introduced in the IEEE 802.11e [40] standard.

2.1.3 Medium Access Control

The MAC protocol of IEEE 802.11 is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), whose collision avoidance is realized by a truncated binary exponential backoff. The latest release of the standard, IEEE 802.11-2007 [35] contains the following four different access mechanisms:

1. The Distributed Coordination Function (DCF)

2. The Enhanced Distributed Channel Access (EDCA)

3. The Point Coordination Function (PCF)
4. The Hybrid coordination function Controlled Channel Access (HCCA)

The DCF is the basic medium access function and mandatory for all WLAN equipment. It was enhanced to support different priorities by the EDCA in the IEEE 802.11e standard [40]. The latter two access mechanisms are based on a polling scheme, whereas the PCF supports a simple polling scheme and the HCCA supports a prioritized polling scheme. However, only the DCF and the EDCA are mandatory mechanisms and widely implemented by WLAN vendors. Therefore, only the first two access mechanisms are considered in this work.

Distributed Coordination Function (DCF)

The DCF is the primary access mode for sharing the wireless medium using the CSMA/CA protocol. Stations which want to transmit a packet compete with each other for medium access and all stations have equal rights. Since WLAN stations are not able to detect a collision on the wireless medium, an acknowledgment scheme is used for that purpose. The Acknowledgment (ACK) frame is transmitted by the receiving stations after a time equal to a Short Interframe Space (SIFS). If no acknowledgment is received by the sending station, it retransmits the packet. The packet is retransmitted until either an acknowledgment is received correctly or the retry limit is reached. The retry limit depends on the packet size. For small packets, the IEEE 802.11 standard introduces the short retry limit which is specified as 7, while for large packets, the long retry limit is set to 4.

A station wanting to transmit a packet first invokes a carrier sensing mechanism to determine if the medium is idle or busy. If the medium is busy, the station defers until the medium becomes idle for at least a period of time equal to a Distributed Interframe Space (DIFS) when the last frame was transmitted successfully, or for a period of time equal to an Extended Interframe Space (EIFS) if the prior transmission was not received correctly.

In order to minimize the collision probability, a random backoff period is chosen after the carrier sensing interval, unless the backoff timer already contains a non zero value from a prior deferral. The backoff timer, the so-called Contention Window (CW), is defined by the number of slots which are chosen uniformly distributed from the interval $[0, CW]$. The slot duration and the value of the CW depend on the underlying physical characteristics. Initially, the CW is set to the arbitration Contention Window minimum (aCWmin). Whenever a packet is not received correctly, the CW value is increased to $CW' = CW^2 - 1$ and the retry counter is incremented by one. The CW is enlarged every time an unsuccessful transmission takes place until the arbitration Contention Window maximum (aCWmax) is reached. Once it reaches aCWmax, the CW remains the same until the maximum number of retransmissions, the retry limit, is reached. The intention behind increasing the contention window after each unsuccessful transmission is to reduce the collision probability for the retransmission of the packet because the backoff is chosen from a larger interval. After each successful transmission, the CW is set to aCWmin again. An example for the medium access is shown in Figure 2.2.

On one condition, the backoff procedure can be skipped. When the medium is detected idle for a period of time $\geq$ DIFS, the frame can be transmitted immediately. This immediate transmission has a significant impact on the overall system, which we will show later in this chapter.

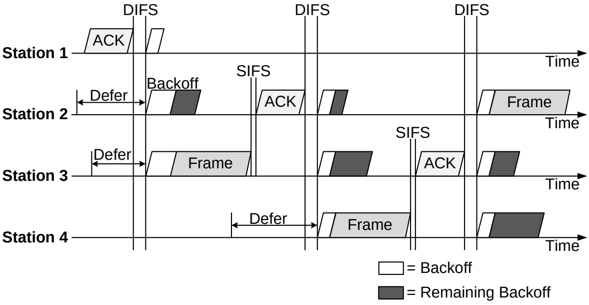


Figure 2.2: *Medium access example for DCF stations.*

Interframe Spaces

Before accessing the wireless medium, a station always has to perform carrier sensing. The duration of the sensing is called interframe space. In order to provide priority access for special frames like ACK frames, the IEEE 802.11 standard defines different interframe spaces. The values of the interframe spaces are determined according to the used physical layer. The relation between the different interframe spaces is shown in Figure 2.3.

- **Short Interframe Space (SIFS):** The SIFS is the shortest interframe space and provides the highest priority access which is used for WLAN control frames, like ACK frames and RTS/CTS frame transmissions, which are explained later in this section.
- **Point (coordination function) Interframe Space (PIFS):** The PIFS is used by stations to initiate the contention-free period of the PCF.
- **Distributed Interframe Space (DIFS):** The DIFS is the minimum idle time for stations which want to transmit management or data frames during the contention-based access period.
- **Arbitration Interframe Space (AIFS):** The length of the AIFS depends on the traffic priority class and is used by QoS stations for the enhanced distributed channel access.
- **Extended Interframe Space (EIFS):** The EIFS is the longest interframe space period and is only used when an error has occurred on the wireless channel.

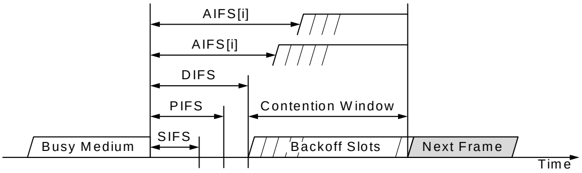


Figure 2.3: *Interframe spacing relationship.*

Request to Send/Clear to Send (RTS/CTS)

Two types of carrier sensing mechanisms are provided by CSMA/CA. First, a station accessing the channel uses a physical sensing to determine the state of the channel. This is called Clear Channel Assessment (CCA) and is provided by the physical layer. As soon as the signal on the wireless channel exceeds a fixed power threshold, the medium is marked as busy, otherwise the medium is determined to be idle. The second mechanism is the Network Allocation Vector (NAV). The NAV is a timer that is transmitted with most WLAN packets. It is used to indicate the time a transmission and its subsequent packets, e.g. ACK frames, last. Each station reads the NAV from the WLAN header and is not allowed to transmit a packet until the timer reaches zero.

However, in a hidden station scenario, these two carrier sensing mechanisms do not operate correctly. Two stations are hidden from each other if they both can communicate with the same AP or station but do not receive frames from the opposite station, see Figure 2.4. If these stations transmit a packet simultaneously, the packets will collide as they are not able to detect the transmission of the other station. Both stations wait for an ACK frame until a timeout expires and retransmit the packet afterwards. This might lead to a drastic reduction of the available bandwidth.

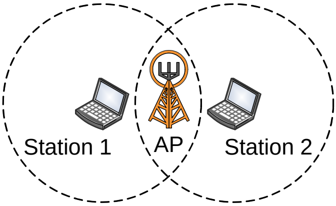


Figure 2.4: *Hidden stations scenario.*

To reduce the probability of a packet collision, the RTS/CTS mechanism is introduced in the IEEE 802.11 standard. The standard defines two small control frames, the Request To Send (RTS) and the Clear To Send (CTS) frame. RTS/CTS frames are used for data packets larger than a certain threshold, the RTS threshold. Before transmitting a large data frame, the RTS frame is transmitted, cf. Figure 2.5. The receiving station replies with a CTS frame after an SIFS. This CTS frame can be detected by the hidden station. Both control frames include the NAV which is set as long as the complete frame exchange will take. Thus, the hidden station has to wait for the NAV to expire before transmitting the next packet. Using these small control frames, the probability of a frame collision can be clearly reduced resulting in an increased performance in a hidden station scenario. On the downside, the control frames cause more protocol overhead. Thus, it has to be examined whether to use RTS/CTS or not.

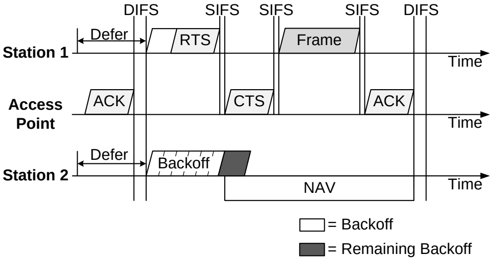


Figure 2.5: *Frame transmission using RTS/CTS.*

Enhanced Distributed Channel Access (EDCA)

In 2005, the DCF was enhanced by the EDCA. In contrast to DCF, EDCA is built on different priorities. Eight different user priorities from the IEEE 802.1D standard [41] are mapped to four Access Categories (ACs) as shown in Table 2.3 and Figure 2.6. The mapping is done by evaluating either the Differentiated Services Code Point (DSCP) field, the former type of service field from the IPv4 header or the flow label in an IPv6 header. The ACs are sorted from AC_BK to AC_VO with AC_VO having the highest priority for medium access. The service differentiation of these ACs is achieved by varying the amount of time a station senses the channel to be idle, the Arbitration Interframe Space (AIFS) as well as the contention window parameters to be used. In addition, the EDCA introduces the Transmission Opportunity Limit (TXOP Limit) which is explained in the next section.

Table 2.3: *User Priority (UP) to AC mapping.*

Priority	User Priority	802.1D Designation	AC	Designation (Informative)
Lowest	1	Background (BK)	AC_BK	Background
	2	-	AC_BK	Background
	0	Best Effort (BE)	AC_BE	Best Effort
	3	Excellent Effort (EE)	AC_BE	Best Effort
	4	Controlled Load (CL)	AC_VI	Video
	5	Video (VI)	AC_VI	Video
	6	Voice (VO)	AC_VO	Voice
Highest	7	Network Control (NC)	AC_VO	Voice

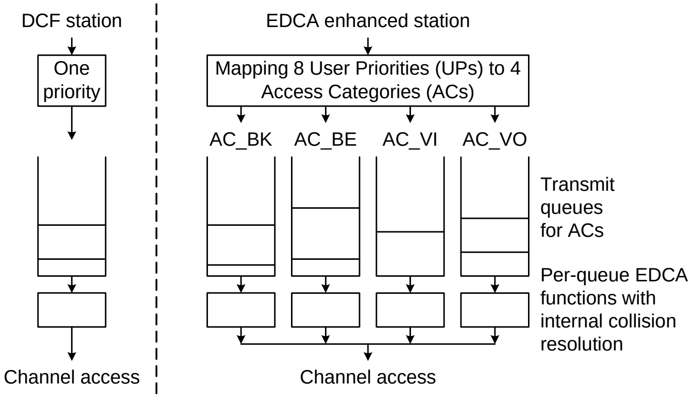

DCF station
One priority
Channel access
EDCA enhanced station
Mapping 8 User Priorities (UPs) to 4 Access Categories (ACs)
AC_BK
AC_BE
AC_VI
AC_VO
Transmit queues for ACs
Per-queue EDCA functions with internal collision resolution
Channel access

Figure 2.6: *Comparison between DCF and EDCA.*

The AIFS replaces the DIFS from the DCF. The length of the AIFS depends on the underlying physical characteristics, the so-called arbitration Slot Time (aSlotTime) and the arbitration SIFS Time (aSIFSTime) as well as on the prioritization level. It is calculated as

$$\textit{AIFS[AC]} = \textit{AIFSN[AC]} \cdot \textit{aSlotTime} + \textit{aSIFSTime}, \tag{2.1}$$

with *AIFSN[AC]* as the number of slots. Using the Extended Rate PHY (ERP) layer at 2.4 GHz, the former IEEE 802.11g standard, aSlotTime is $9\mu s$ and aSIFSTime is $10\mu s$. As lower priorities use a larger AIFS, a certain prioritization can be achieved.

The backoff procedure further supports the prioritization. For the DCF mode in an IEEE 802.11g [39] network, the default values for the contention windows are aCWmin=15 and aCWmax=1023. EDCA uses these values to define different priorities for each AC. The standard settings for every IEEE 802.11 network can be seen in Table 2.4. If we take the standard settings of aCWmin and aCWmax from the IEEE 802.11g network, we will receive the parameter set for EDCA as shown in Table 2.5. Here, the highest priority class is assigned an aCWmin of 3 and an aCWmax of 7 while the lowest priority class is assigned the values 15 and 1023. This will lead to different mean contention window sizes. Clearly, a station with a lower mean contention window will get access to the medium much more often.

Table 2.4: *Access categories and their settings.*

AC	CWmin	CWmax	AIFSN	TXOP Limit	
				DSSS	OFDM
BK	aCWmin	aCWmax	7	0	0
BE	aCWmin	aCWmax	3	0	0
VI	$\frac{aCWmin+1}{2} - 1$	aCWmin	2	6.016 ms	3.008 ms
VO	$\frac{aCWmin+1}{4} - 1$	$\frac{aCWmin+1}{2} - 1$	2	3.264 ms	1.504 ms

Table 2.5: *Default EDCA parameter set when using IEEE 802.11g.*

AC	CWmin	CWmax	AIFS	TXOP Limit
BK	15	1023	72 μs	0
BE	15	1023	37 μs	0
VI	7	15	28 μs	3.008 ms
VO	3	7	28 μs	1.504 ms

Frame Bursting Using the TXOP Limit

Besides the prioritization scheme, the TXOP Limit is also introduced with EDCA. The TXOP Limit describes the time a station is allowed to continuously transmit frames after it gained access to the medium. It is expressed in multiples of 32 μs like shown in Table 2.4. The TXOP Limit duration values are advertised by the Access Point in beacon frames. The beacon frames are sent out periodically by the Access Point and contain information about the WLAN cell and the currently used channel access parameters. A TXOP Limit field with a value of 0 indicates that a single MAC Service Data Unit (MSDU) may be transmitted at any rate for each Transmission Opportunity (TXOP).

The transmission of a frame burst is shown in Figure 2.7. The data packets and ACKs are only separated by SIFSes. It is obvious that the use of a transmission burst optimizes the link utilization because the backoff scheme does not have to be performed for every packet. However, the downside of this scheme might be longer delays and a higher collision probability during the contention phase.

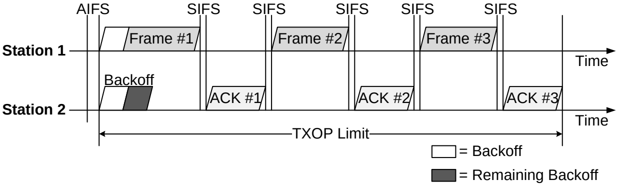


Figure 2.7: *One transmission burst.*

Before taking a look at the influence of different channel access parameters on the performance of the WLAN, we focus on fairness aspects in IEEE 802.11 infrastructure networks.

2.2 Unfairness in Infrastructure Networks

Although there are several possibilities to define fairness in IEEE 802.11 networks, we consider link layer and transport layer fairness and characterize fairness in two different ways, long-term fairness and short-term fairness. An IEEE 802.11 network is long-term fair if the successful access to the channel observed on a long term converges to 1/N for N competing and saturated stations. The network is short-term fair if the number of successful network accesses is fairly distributed over a short time period of a few milliseconds. Thus, short-term fairness implies long-term fairness, but not vice versa. In IEEE 802.11 networks, long-term fairness is sufficient for delay-insensitive packet flows whereas short-term fairness has to be achieved for real-time flows with delay and jitter requirements. A good and widely used measure for the fairness is the Jain Fairness Index (JFI) [42–44]. It is defined as

$$f(x_1, x_2, x_3, ..., x_n) = \frac{\left(\sum\limits_{i=1}^{n} x_i\right)^2}{n \cdot \sum\limits_{i=1}^{n} x_i^2}, \quad 0 \le f() \le 1, \tag{2.2}$$

where x_i is the normalized throughput of station i and n is the number of flows in the WLAN. A JFI of one indicates absolute fairness and a JFI of 0 absolute unfair resource distribution. The JFI can be used to show both, long-term and short-term fairness, by adjusting the window size over which the fairness is calculated. In the following, we present related work considering general unfairness in IEEE 802.11 networks and TCP unfairness in WLANs.

2.2.1 Related Work

A large number of papers have been published showing any kind of unfairness in IEEE 802.11 networks [45–69]. The first part of this section covers general unfairness papers and the second part focuses on unfairness of TCP over WLAN.

General Unfairness

Nandagopal et al. [45] propose a proportional fair contention resolution algorithm to achieve MAC layer fairness in wireless packet networks. The fairness of the algorithm is compared to the traditional 802.11 binary exponential backoff, a Multiplicative Increase Linear Decrease (MILD) with backoff copy mechanism [47], and a combining persistence and backoff approach [46]. The results show that the decentralized local contention resolution algorithm achieves almost ideal fairness.

Koksal et al. [48] were the first who analyzed the short-term fairness in WLAN CSMA/CA networks. Two approaches are proposed for evaluating the fairness: one based on the sliding window method with the Jain fairness index or the Kullback-Leibler distance [70], and the other one that uses renewal reward theory on Markov chain modeling. It is shown that the exponential backoff algorithm reduces the collision probabilities but at the expense of short-term unfairness. Similar publications also assuming this kind of backoff are published by Barrett et al. [49], Vaidya et al. [50], and Kwon et al. [51].

Berger-Sabbatel et al. [52] claim however that the papers above base their research on the first WaveLAN cards using plain CSMA/CA. It is described that the DCF of the IEEE 802.11 standard just uses a backoff procedure when a collision on the channel occurs. However, a look in the IEEE 802.11-2007 standard [35] disproves this statement. In Section 9.2 it is said: "...and a random backoff time following a busy medium condition". Similar to Koksal et al. [48], the JFI is used to analyze the fairness in a saturated ad-hoc network without any hidden nodes. The authors show by an analytical model, simulations, and measurement that the DCF is short-term fair.

In [53–56] the authors observe a significant unfairness between downlink and uplink flows when DCF or EDCA are used in a WLAN with an Access Point. It is claimed that the DCF allows equal utilization of the medium and thus, if the downlink has much more offered load than the uplink, the downlink becomes the bottleneck. Grilo et al. [56] define three traffic models, a voice model, a video model, and an HTTP traffic model to show that as soon as the utilization increases, the Access Point becomes the bottleneck both with the DCF and the EDCA. To solve the problem, the Access Point should use a polling based access mechanism. As an alternative solution, Kim et al. [53] and Fukuda et al. [54] propose a mechanism where the Access Point uses a shorter interframe space duration compared to the stations before accessing the shared medium. All stations use a DIFS before accessing the medium and the Access Point uses a PIFS interval, where SIFS < PIFS < DIFS, compare Section 2.1. The solution provided by Casetti et al. [55] does not only propose to use smaller interframe spaces at the Access Point but also smaller contention windows for downlink flows. However, their mechanism lacks flexibility as it uses fixed parameters.

Abeysekera et al. [57, 58] show that the general DCF is long-term unfair. Therefore, a mechanism is proposed where the minimum contention windows of the AP are adapted according to the number of flows. In contrast, the contention windows of the stations remain constant. The results show that a perfect fairness with a JFI of 1 is achieved between uplink and downlink flows. However, the results were performed using similar traffic streams in up- and downlink direction. It would be interesting if the fairness can still be achieved with different traffic streams. The work is extended in [59] to provide fairness in IEEE 802.11e networks. The authors claim that even in a network with service differentiation, fairness can be achieved between downlink and uplink best effort flows. However, only unfairness between uplink and downlink traffic flows are considered without taking a look at the unfairness between different uplink traffic flows.

Fang et al. [60] try to achieve fair medium access in IEEE 802.11 ad-hoc networks with hidden nodes. Thereby, each station measures its throughput and the throughput of other stations by evaluating the RTS, CTS, and data frames. A station doubles its contention window if its obtained throughput is larger than its fair share to ensure that other stations have a better chance to get medium access. The presented mechanism can easily be adapted to achieve a max-min fair throughput share. Another paper working on fairness in wireless ad-hoc networks is written by Malli et al. [61]. Similar to Fang et al. [60], the contention windows are adapted to improve the throughput and to achieve fairness among different flows. Their mechanism is not only applicable to wireless networks without service differentiation, but can also be applied for EDCA-based networks. Thereby, fairness is achieved between flows of the same priority.

Unfairness of TCP over WLAN

The TCP unfairness between uplink and downlink connections in WLANs is presented in [62–64]. It is shown for different traffic models that the downlink flows tend to starve. Park et al. [62] claim that the starvation is caused by both the TCP-induced and the MAC-induced unfairness. Pilosof et al. [63] propose to solve the problem by increasing the buffer size at the Access Point to avoid packet loss due to buffer overflow. Similar to this paper, Thottan et al. [64] identify the equal access probabilities of the Access Point and the stations as the reason for the TCP unfairness. However, it is shown that an increased buffer size does not solve this problem and propose an adaptive EDCA parameter set.

Another paper about TCP unfairness is presented by Blefari-Melazzi et al. [65]. It is claimed that downstream TCP connections suffer because of arising congestion and corresponding packet losses happening in the downlink buffer at the Access Point. Furthermore, for upstream TCP connections, the Access Point has to transmit the TCP acknowledgments which are delayed and lost, because the Access Point cannot access the medium with a priority higher than other stations. Leith et al. [66] look at the TCP fairness for upstream flows too. It is shown that the TCP acknowledgment is delayed using the standard DCF access mechanism. However, a scheme is proposed for how to prioritize the Access Point by using a different parameter set for the medium access compared to the IEEE 802.11e recommendations. The proposed mechanisms are tested in an experimental scenario and the results can be found in [67].

Further TCP unfairness observations are made by Jian and Chen [68]. Using ns-2 simulations is is shown that fairness between nodes depends on the distance and the difference between carrier sensing and transmission range. A Proportional Increase Synchronized Multiplicative Decrease (PISD) mechanism is proposed to ensure not only fairness but also weighted fairness in CSMA/CA networks.

Keceli et al. [69] claim that over a sufficient long interval, MAC layer fairness is achieved in IEEE 802.11 networks. However, this MAC layer fairness does not translate into achieving per-flow transport layer fairness as the AP can only transmit as many packets as each station per time interval while having more connections. In contrast to Kim et al. [53] and Jeong et al. [71], the IEEE 802.11 backoffs are not changed which is according to their opinion not practicable with today's wireless hardware. Fairness is achieved between downlink and uplink TCP flows by sending the TCP uplink ACKs only as often as the TCP data of downlink connections is sent. The ns-2 simulations show that a perfect fairness with a JFI of 1 is achieved.

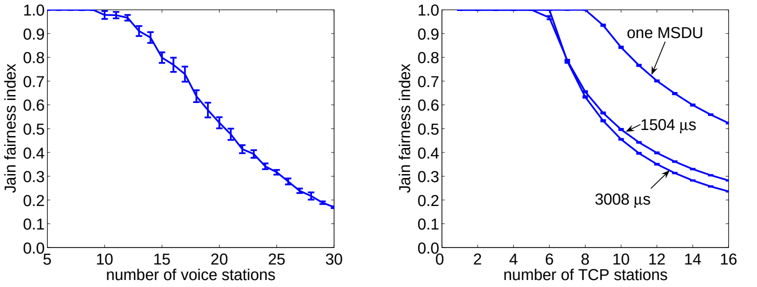
The paper from Wu et al. [72] first compares unfairness between short and long living TCP flows in several scenarios. Afterwards, the Selective Packet Marking with ACK Filtering (SPM-AF) and the Least Attained Service (LAS) mechanisms are introduced. SPM-AF differentiates TCP data packets and TCP ACK packets giving data packets a higher priority to enter the bottleneck queue [73]. LAS gives higher priority to flows that have received less service relative to other flows [74]. Simulation results show that the JFI can be improved by 20-40 % using either SPM-AF or LAS compared to the conventional drop tail queue mechanism while also reducing the variability of transfer times for small files.

All papers focus on the discrepancy of delays and buffer overflow probabilities experienced by the Access Point and the stations. A related issue that has not yet been investigated is another kind of unfairness resulting from different collision probabilities. Interestingly, the latter unfairness favors the Access Point which is contrary to the former. In the next subsections, we take a closer look on the unfairness in terms of collision probabilities.

2.2.2 Introduction to the Unfair Channel Access Phenomenon

Similar to the related work, we first take a look at the JFI. Therefore, a simulation is configured using the OPNET Modeler [75] simulation environment with the IEEE 802.11g WLAN model. Voice stations using the ITU-T G.711 [76] voice codec with a packet size of 640 bits and an inter-arrival time of 10 ms are communicating with an Access Point. We decided to use this voice codec because it is widely implemented in VoIP WLAN phones. Other codecs like the ITU-T G.729 [77] would lead to similar results. The corresponding Jain fairness indices for the voice scenario are shown in Figure 2.8(a). Although the JFI is defined to evaluate the fairness in terms of throughput, we use the JFI for evaluating the fairness in terms of collision probabilities. The JFI is computed over an interval of 100 seconds, i.e., the complete duration of each simulation run, and the simulations are repeated 50 times. Thus, long-term fairness is considered. The figure shows that for up to 9 voice stations, the IEEE 802.11 network is absolute fair in terms of collision probabilities. If the number of stations is further increased, the WLAN becomes more and more unfair and when 30 voice stations are active in the system, the JFI decreases down to 0.18. Concluding, the more the WLAN is loaded, the more unfair is the resource distribution.

In order to evaluate if the resource distribution is also unfair when using TCP downloads, a scenario is set up with 1 to 16 stations downloading 1500 Bytes TCP packets from an Access Point. The downloads are configured to saturate the WLAN. Access Point and stations are set with three different TXOP Limits, one MSDU, 1504 μs, and 3008 μs. The resulting Jain fairness indices of the collision probabilities are shown in Figure 2.8(b). With a TXOP Limit of one data frame, the system is fair up to 8 TCP downloading stations and decreases only down to 0.5 for a 16 stations scenario. However, when increasing the TXOP Limit, the resource distribution already becomes unfair when more than 5 stations are downloading. The reason is that with an increased TXOP Limit, the number of contention slots per time interval decrease. Thus, more stations compete for



(a) JFI for collision probabilities in a scenario with bi-directional UDP flows.

(b) JFI for collision probabilities in a TCP downlink scenario.

Figure 2.8: *Unfairness between stations in terms of collision probability with a 0.95 confidence level.*

channel access during each contention phase, resulting in an increased collision probability and an unfair resource distribution.

We now further analyze the reasons for this unfairness. To this end, we set up a scenario with 25 voice stations and evaluate the collision probabilities of all stations and the Access Point over time. Figure 2.9 depicts the average collision probability of the scenario during the steady-state phase. The packet collision probabilities are calculated using an interval of one second and a moving average over 20 values. The figure shows the steady-state phase after a transient phase of 30 seconds.

The collision probabilities measured at the AP are just below 10 % and the lowest in the network. The collision probabilities of the stations range from around 10 % up to over 24 %. The reason for the different collision probabilities of the stations lies in the phase pattern. An example for a phase pattern is shown in Figure 2.10. As the voice packets follow a deterministic arrival process with an inter-arrival time of 10 ms, the collision probabilities of each station depend on the start time of the voice conversation. In the figure, four stations receive their voice packets from the upper layer almost at the same time and thus

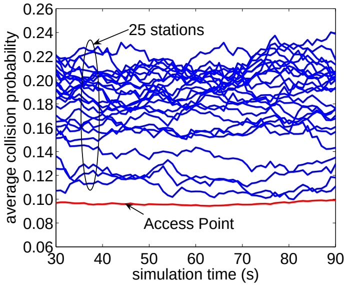


Figure 2.9: *Illustrating the unfairness between Access Point and stations.*

compete against the other three stations for medium access. This clearly results in higher collision probabilities for these four stations compared to other stations competing only against one station or against no other station at all.

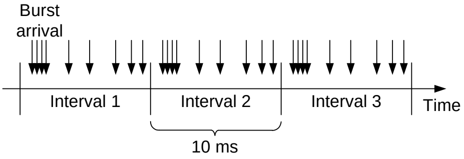


Figure 2.10: *Phase pattern illustration of voice traffic.*

The difference in collision probabilities of the AP and the stations can be traced back to the unfair channel access. A random station competes against 24 stations and the Access Point for channel access when all phases are random. On the other hand, the Access Point competes against 25 stations. It seems that every network entity has to compete against 25 others. However, when considering the number of packet transmissions, the AP competes against 25 transmissions (one from each station) and each station has to compete against 49 packet transmissions (24 from the other stations and 25 from the AP). In other words, the probability of a frame collision upon a channel access of a station is significantly higher compared to the collision probability of the AP. This explains the different collision probabilities of a single station and the Access Point seen in Figure 2.9. Nevertheless, this unfairness explanation just holds when the stations are not saturated. In the following, we evaluate this unfairness in more detail, first for the DCF and then for the EDCA.

2.2.3 Unfairness of the DCF

The simple simulation scenario above has shown the unfairness between voice stations and Access Point in terms of collision probability. In this section, we try to explain this unfair channel access phenomenon by an analytical model for the voice traffic scenario. The results are compared with a simple MATLAB simulation and a detailed OPNET simulation. The MATLAB simulation includes the CSMA/CA mechanism without regarding extensions of the DCF or influences from other layers. In contrast, the OPNET simulation includes the complete DCF with all its extensions and simulates all layers of the ISO/OSI protocol stack.

Unfair Channel Access Using Voice Traffic

To explain the unfair channel access, a simple analytical model is used. Firstly, the access probabilities of AP and stations are calculated without considering packet retransmissions due to collisions. These access probabilities are then used to calculate the collision probabilities. The resulting number of retransmissions from the collision probabilities are needed to recalculate the access probabilities. Thus, a repeated substitution of collision probabilities and access probabilities is applied to get an approximation of the collision probabilities.

Let us now define the algorithm in more detail. We consider a scenario with N stations and one Access Point. Stations and Access Point are communicating symmetrically. Let $A = 10\,\mathrm{ms}$ be the frame period of the voice application. Further, let R be the number of slots between two packet arrivals. According to the IEEE 802.11g standard, the length of a single slot is $9\,\mu s$. The slots can either be used for packet transmissions, interframe spaces, or contention.

Assume that all stations and the AP are able to transmit their packets within the interval A. This means that every station transmits one packet during this interval and the AP transmits N packets. So, during interval A, $2 \cdot N$ packets are transmitted. X slots are needed to transmit one packet, including ACK, SIFS, DIFS, and the packet transmission itself. This means that during interval A, the remaining $R - 2 \cdot N \cdot X$ slots are available for contention. The parameters are illustrated in Figure 2.11.

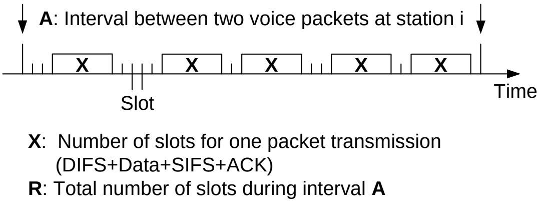

A: Interval between two voice packets at station i
X
X
X
X
X
Slot
Time
X: Number of slots for one packet transmission
(DIFS+Data+SIFS+ACK)
R: Total number of slots during interval A

Figure 2.11: *Parameter illustration of the analytical model.*

Now, the access probability and collision probability can be calculated using a repeated substitution. The iteration starts by calculating the access probabilities assuming that no collision occurs on the channel. This results in the probability

$$p_s = \frac{1}{R - (2 \cdot N - 1) \cdot X} \tag{2.3}$$

that a station accesses a given slot and the probability

$$p_{AP} = \frac{N}{R - (2 \cdot N - 1) \cdot X} \tag{2.4}$$

for the Access Point to access the medium. The numerator shows the number of packets that have to be transmitted and the denominator describes the number of available slots. One transmission is subtracted because the station or the Access Point whose access probability is calculated has not yet transmitted its packet. Having defined the initial access probabilities of the iteration process, the independent collision probabilities can be calculated as

$$q_s = 1 - (1 - p_{AP}) \cdot (1 - p_s)^{N-1} \text{ and} \tag{2.5}$$

$$q_{AP} = 1 - (1 - p_s)^N, \tag{2.6}$$

where q_s is the collision probability of a station and q_{AP} is the collision probability of the Access Point. As the Access Point competes against all stations, the collision probability is calculated using the access probability of the stations. A station on the other hand competes against all other stations and against the Access Point. Therefore, we have to take both, the access probability of the stations excluding ourselves and the access probability of the Access Point into account.

Using the collision probabilities, the access probabilities can be redefined, but before, the mean number of collisions have to be estimated. The number of retransmissions needed for a successful packet reception is calculated using the geometric distribution. Thereby, the mean number of required retransmissions is

$$X_s = E(Geom(q_s)) = \frac{q_s}{1 - q_s} \tag{2.7}$$

for the stations and

$$X_{AP} = E(Geom(q_{AP})) = \frac{q_{AP}}{1 - q_{AP}} \tag{2.8}$$

for the Access Point. The retransmission of N packets results in an N-fold geometric distribution or in

$$Y_s = E(NegBin(q_s, N)) = \frac{N \cdot q_s}{1 - q_s} \tag{2.9}$$

for all stations and in

$$Y_{AP} = E(NegBin(q_{AP})) = \frac{N \cdot q_{AP}}{1 - q_{AP}} \tag{2.10}$$

for the Access Point. Assuming that two or more packets collide, the mean number of collisions K can be defined as

$$K \cong \left\lceil \frac{\frac{N \cdot q_s}{1-q_s} + \frac{N \cdot q_{AP}}{1-q_{AP}}}{2} \right\rceil , \tag{2.11}$$

where the denominator is the minimum number of colliding packets of all stations and the Access Point. From this approximation of the mean number of collisions, the remaining number of slots available for contention are recalculated with $R - (2 \cdot N - 1 + K) \cdot X$ and the new probability that a station and respectively the Access Point accesses a slot is determined as

$$p_s = \frac{\frac{q_s}{1-q_s} + 1}{R - (2 \cdot N - 1 + K) \cdot X} \text{ and} \tag{2.12}$$

$$p_{AP} = \frac{N \cdot \frac{q_{AP}}{1-q_{AP}} + N}{R - (2 \cdot N - 1 + K) \cdot X}. \tag{2.13}$$

Finally, we can iterate between q and p, using Equation (2.3) and Equation (2.4) as the initial access probabilities.

In order to validate the results from the analytical model, we performed simulations using MATLAB and OPNET. The parameters for the simulation and the analytical model are shown in Table 2.6.

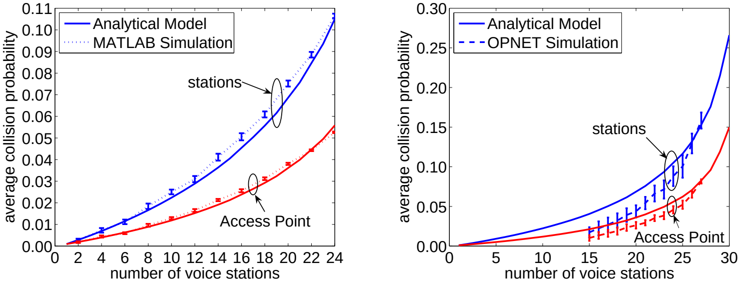
The results from the analytical model and the MATLAB simulation are illustrated in Figure 2.12(a). The 95 % confidence intervals result from 20 simulation runs with different phase patterns. The x-axis shows the number of voice stations and the y-axis illustrates the collision probabilities averaged over all stations. Two observations can be made from this experiment. Firstly, it reveals that the analytical model and the simulation fit well. The second observation is that both the analytical model and the simulation reveal the unfairness between Access Point and stations. For 24 stations, the collision probability of the Access Point is around 5.5 % and for the stations around 10.5 %. A further increase of the number

Table 2.6: *Simulation parameters.*

Parameter	Value
Voice frame duration	10 ms
WLAN standard	IEEE 802.11g
Data rate	54 Mbps
Control data rate	24 Mbps
Slot length	9 μs
DIFS time	28 μs
SIFS time	10 μs
CWmin	15
CWmax	1023
Packet length	960 bits+header
ACK length	112 bits+header
Signal extension	6 μs
AP buffer size	4,096,000 bits

of voice stations would lead to false results of the MATLAB simulation, because as it is programmed close to the analytical model, the assumption that all packets can be transmitted within an interval would not hold anymore.

For the OPNET simulation, the number of voice stations can further be increased up to 27. In a scenario with more than 27 stations, the voice connections cannot be established because of a high packet loss. In Figure 2.12(b), the OPNET simulation results are compared to the results from the analytical model. The figure reveals that the collision probability of the analytical model is higher than that of the simulation, especially when the network is not at its capacity limits. This effect results from immediate transmissions. A station can immediately transmit a packet when it is idle for at least DIFS and then receives a packet from the upper layer. In heavily loaded networks, the number of immediate transmissions decrease. This is the reason why the collision probabilities of the analytical model and simulation match well under high load. However, the figure also shows the unfairness between the Access Point and the stations. For 27 stations, the collision probability of the Access Point is 8.23 % and for the stations 15.68 %.



(a) Comparison between MATLAB simulation and analytical results.

(b) Comparison of OPNET simulation results with the complete DCF and analytical results.

Figure 2.12: *Unfairness between AP and stations.*

Unfair Channel Access for TCP Flows

All results, the OPNET simulation, the MATLAB simulation, and the analytical model show the unfairness in WLAN for bi-directional voice traffic. In this subsection, it is evaluated whether the unfairness between stations and Access Point also occurs for TCP traffic. Therefore, saturated downstream TCP traffic is considered. This means that the Access Point is continuously transmitting TCP packets and the stations acknowledge only every second TCP downlink packet. The packet size for the downlink packets is set to 1500 Bytes. With all headers, the MAC acknowledgment frame, and the interframe spaces, 37 slots are required for transmitting one TCP packet. TCP acknowledgments require 13 slots for transmission. We use TCP Reno which means that fast retransmit and fast recovery are applied. This helps to sooner recover from packet loss but does not influence the unfairness considerations. Further parameters for the TCP simulations are shown in Table 2.7.

The simulations were performed using both OPNET Modeler and MATLAB. Thereby, similar to the voice scenarios, the OPNET simulation accounts for the

Table 2.7: *Parameters for the TCP simulations.*

Parameter	Value
Application	saturated TCP
Packet size	1500 Bytes
TCP receive buffer	65,535 Bytes
Fast Retransmit	enabled (TCP Reno)
MTU	WLAN (2304 Bytes)
WLAN AP buffer	1024 kbits
WLAN station buffer	102.4 kbits
CWmin	15
CWmax	1023

complete protocol stack with a detailed TCP model and the DCF extensions and the MATLAB simulation only considers CSMA/CA and a simple TCP emulation. The TCP emulation is a saturated TCP traffic flow where every second TCP packet on the downstream is acknowledged with one TCP acknowledgment on the upstream.

An analytical model for explaining the unfairness phenomenon in a TCP traffic scenario is rather complex. The analytical voice traffic model cannot be used directly, because the packets do not arrive in fixed intervals and especially the TCP acknowledgments from the stations depend on the transmitted packets on the downlink. Therefore, only an approximation is made using an iteration process similar to the voice model. Assuming that the Access Point is saturated and the backoff is calculated between 0 and the contention window in every backoff interval, the access probabilities can be calculated using

$$p_s = \frac{1}{2 \cdot N \cdot CW_{min} + 1} \text{ and} \tag{2.14}$$

$$p_{AP} = \frac{1}{CW_{min} + 1}. \tag{2.15}$$

The access probabilities of the Access Point result from the fact that it tries to transmit a packet in every contention phase. In contrast, a station only tries to access the medium in every second frame. N is again the number of stations in the system. From this starting point of the iteration process, the collision probabilities are calculated similar to the analytical voice model with

$$q_s = 1 - (1 - p_{AP}) \cdot (1 - p_s)^{N-1} \text{ and} \tag{2.16}$$

$$q_{AP} = 1 - (1 - p_s)^N. \tag{2.17}$$

Now, the access probabilities for the stations can be redefined as

$$p_s = \frac{\frac{q_s}{1-q_s} + 1}{\sigma \cdot 2 \cdot N \cdot CW_{min} + 1} \tag{2.18}$$

and the probabilities of the Access Point as

$$p_{AP} = \frac{\frac{q_{AP}}{1-q_{AP}} + 1}{\sigma \cdot CW_{min} + 1}. \tag{2.19}$$

The factor σ depends on the average number of packets which are transmitted before the Access Point or the stations get a transmission opportunity. As it is not possible to exactly estimate this factor, it is fitted to the curves of the simulation results and set to $\sigma = \frac{2}{3}$.

The collision probabilities from the simulations and the analytical model are shown in Figure 2.13. On the x-axis, the number of TCP stations is increased from 1 up to 16 and the y-axis shows the average collision probability. The figure reveals that the simulations and the analytical model match quite well. Furthermore, the figure shows that the collision probability of the Access Point is not influenced by the number of stations. In contrast, the collision probability of the stations increase with an increasing number of stations until a constant level of around 14.4 % is reached. If we compare the collision probabilities of the bi-directional voice scenario and this TCP scenario, the unfairness between Access Point and stations becomes even more obvious. The collision probabilities of the stations is 2.6 times higher than the collision probabilities of the Access Point.

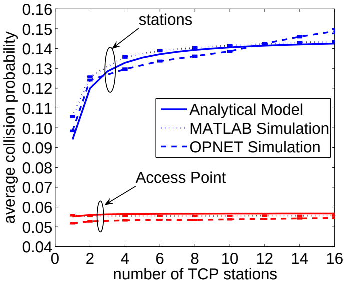


Figure 2.13: *Unfairness between AP and stations using saturated TCP traffic on the downlink.*

2.2.4 Unfairness of the EDCA

With the introduction of the IEEE 802.11e standard and the TXOP Limit, the unfairness between stations operating at different loads changed. The TXOP Limit defines the time a station is allowed to transmit packets in a row after it gained access to the medium. The packets are only separated by the acknowledgment frame and a short interframe space. For our scenario, this means that the Access Point can transmit more than one packet, up to all N packets for the N stations, after gaining access. Comparing the results from the previous section, the access probability and collision probability of the Access Point decrease. This in turn leads to the effect that more stations can be supported because the wireless medium is better utilized. However, the unfairness between stations and Access Point increases.

Influence of the TXOP Limit on UDP Voice Traffic

This time, the unfairness is shown by means of OPNET simulations only, as it is rather complex to create a simple model to show the influence of the TXOP Limit on the fairness. The parameters for the simulations are set to the values specified in Table 2.6 and the TXOP Limit for the voice queue is set to 1504 μs. With these settings, a maximum number of 32 voice stations can be supported.

The results in Figure 2.14 reveal on the one hand that the collision probability in both directions decreases compared to the results from Figure 2.12(b). On the other hand, the unfairness between Access Point and stations increases. While the average collision probability of the Access Point increases only slightly with an increasing number of stations, the collision probabilities of the stations increase from around 2 % for 20 stations up to 23 % for 32 stations.

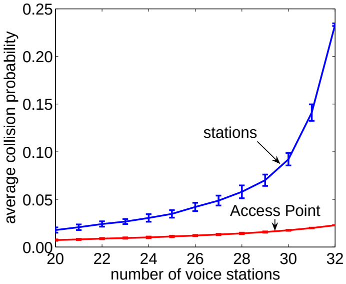
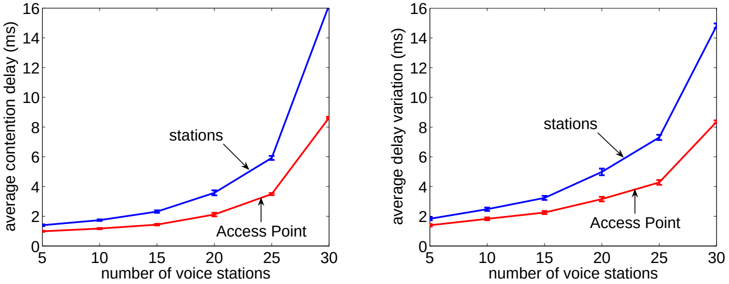


Figure 2.14: *Unfairness between AP and stations with a TXOP Limit of 1504 μs.*

Unfairness in Terms of Contention Delay and Delay Variation

In order to show that not only the collision probabilities differ between the Access Point and the stations, we take a look at the unfairness in terms of contention delay. The contention delay starts when the packet is at position zero of the queue and ends when the acknowledgment frame is successfully received. The contention delays are simulated with the same settings as in Subsection 2.2.4. Figure 2.15(a) depicts the average voice contention delay. To compare the contention delay of the stations and the Access Point, we do not consider the contention delay of individual packets but the contention delay of individual transmission opportunities. Doing this, bursting effects from Figure 2.14 are excluded and thus, solely the medium access time is considered. The prioritized Access Point exhibits contention delays that are up to 7 ms lower than the corresponding contention delay of the stations.



(a) Contention delay.

(b) Delay variation.

Figure 2.15: *Unfairness between AP and stations with a TXOP Limit of 1504 μs.*

There are two reasons for the lower delay at the Access Point. The first reason is that only up to 13 packets fit into one transmission burst and if 30 stations are active in the system, the Access Point has to transmit at least 3 bursts. With the transmission of these 3 bursts, the average collision probability of the stations is larger than the collision probability of the Access Point, see Equation (2.3) and Equation (2.4). The second reason is that the transmission of a packet from the station is delayed for at least the TXOP Limit if the Access Point has gained access prior to the station.

Now, we want to see if the unfairness phenomenon is also visible in the delay variation. The IETF [78] defines the delay variation of a pair of packets within a stream of packets as the difference between the one-way delay of these packets. Assume $P1$ and $P2$ to be two consecutive MAC packets, and the time stamps at their source and destination stations are $S1$, $S2$ and $D1$, $D2$, respectively. Then, the delay variation is calculated as

$$delay\,variation = |(D2 - D1) - (S2 - S1)|. \tag{2.20}$$

Figure 2.15(b) exhibits the increase of voice delay variation for both the stations and the Access Point when increasing the number of voice stations. We notice the difference between the delay variation of the stations and the Access Point similar to the contention delay. The delay variation at the Access Point benefits from the frame bursting feature because queuing at the Access Point rises the more voice stations are associated to it. Queuing at voice stations is not that pronounced. Consequently, bursts of voice frames on the downlink reduce contention delay and therefore, reduce the variability of the packet delay.

Influence of the TXOP Limit on TCP Traffic

Finally, the influence of the TXOP Limit is evaluated for TCP flows. The TCP traffic model from Subsection 2.2.3 is used for the simulations. Figure 2.16 exhibits the average collision probabilities for three different settings of the TXOP Limit, one MSDU, 1504 μs, and 3008 μs. With a TXOP Limit of 1504 μs, up to 4 TCP packets can be transmitted in one burst after the Access Point gained access to the wireless medium and up to 8 TCP packets can be transmitted in a burst using a TXOP Limit of 3008 μs. The Access Point recognizes a collision right after the first packet of a transmission burst is transmitted and stops the transmission of the following burst packets.

The figure reveals that an increasing TXOP Limit decreases the collision probability for both the Access Point and the stations because the access probability of the Access Point decreases. However, the unfairness between the stations and the Access Point remains the same. Therefore, we can conclude that transmission bursts do not resolve the unfairness phenomenon neither for voice UDP flows nor for TCP flows.

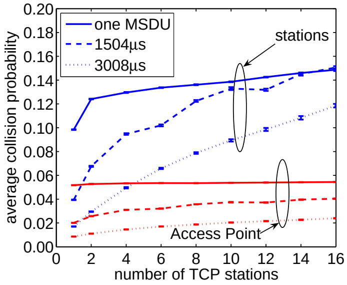


Figure 2.16: *Impact of the TXOP Limit on the collision probabilities of TCP flows.*

Concluding this section, we have to point out that this unfairness has to be taken into account when performing load or admission control in WLAN. Measuring the load in terms of collision probability only at the Access Point does not reflect the overall situation of the WLAN cell. Thus, to estimate the load in a WLAN cell and to perform admission control, the situation of each station is required. Therefore, we present a feedback mechanism in the next section which is used to estimate the current load and to reveal QoS problems of real-time flows. Using these measurements, we propose a parameter adaptation mechanism.

2.3 Dynamic Contention Window Adaptation

The QoS extensions published in the IEEE 802.11e [40] amendment in late 2005 enable service differentiation but do not guarantee a specific QoS level for real-time users. One reason for this is the lack of a load control for WLANs. Furthermore, resource efficiency has severely decreased through the service differentiation extension due to the use of small and static Channel Access Parameters (CAPs). As a result, time-varying loads cause heavily varying contention levels leading to an inefficient channel usage. In the worst case, traffic performance is degraded and QoS requirements cannot be met.

In this section, we propose a measurement-based parameter adaptation scheme which succeeds in achieving a much better resource efficiency as compared to the standard. At the same time, service differentiation is maintained and even QoS guarantees can be given to a certain extent. Resource efficiency is achieved through a dynamic CAP adaptation process according to the current channel contention level at runtime. Updates of the CAPs resulting from adaptations are broadcast via beacon frames. The mechanism is called Dynamic Contention Window Adaptation (DCWA). It achieves resource efficiency by choosing an appropriate contention window with respect to the current channel contention. Thus, voice flows significantly improve, become more robust, and are still protected.

2.3.1 Related Work

Contention window adaptation techniques in the literature can be mainly divided into Multiplicative Increase / Multiplicative Decrease (MIMD) and Additive Increase / Additive Decrease (AIAD) schemes. The authors of [79–82] use MIMD. The multiplication factor is either defined by a function of the priority and the collision rate, or simply by using a fixed value. The authors of [83–85] use AIAD and determine additive changes of the contention window through the collision rate, the priority, the distance between the minimum contention window and the maximum contention window, or simply use fix values. The complex MIMD and AIAD methods estimate the ratio between the collision rate and the contention window size.

Further, there are differences of how to change the contention windows based on measurements. A common method is a threshold-based approach [80–83, 86]. An alternative to this is to define the contention window directly as a function of parameters such as the collision rate, the number of stations, and the priority [79, 84, 85]. The problem with the latter methods is that the parameter correlation for the contention window adaptation is not clear. Kim et al. [53] prioritize the AP over the stations by shortening the interframe space from DIFS to PIFS just for the AP. Due to the lack of flexibility of this approach, Abeysekera et al. [57–

59] propose a mechanism which adapts the minimum contention window of the Access Point. The window is adapted according to the number of downlink flows, so that the system does not require a feedback from the stations. In the previous section, we have seen however that measuring at the AP only might lead to false results.

Cali et al. [87, 88] analytically derive the optimal contention window for a given channel contention level. However, the approach requires an accurate estimation of the active number of stations and the distribution of the frame length. Toledo et al. [89] also adjust the contention window parameters by estimating the number of competing stations in the network. A simple online algorithm is proposed to estimate the number of competing stations based on the sequential Monte Carlo method. A similar approach is proposed by Ge et al. [90]. The number of competing stations in the system is estimated and the contention windows are adapted in such a way that a pre-specified targeted throughput ratio among different service classes is achieved.

Gopalakrishnan et al. [91] increase the performance of a WLAN by using a concept similar to the TXOP Limit. The Access Point aggregates multiple packets into one large MSDU. The WLAN MAC layer then divides this large MSDU into smaller fragments. Once the Access Point acquires the right to transmit, all fragments are transmitted in a burst.

The next section introduces our approach, the dynamic contention window adaptation algorithm.

2.3.2 DCWA Algorithm

The DCWA dynamically adapts the contention window bounds to support a maximum number of flows with a minimum access delay. The parameter used for the contention window adaptation is the maximum number of retransmissions per packet per beacon interval R_i^{max}. Simulations showed that the number of retransmissions is suitable to reflect the current contention level in the cell. We first explain how the Access Point collects the number of retransmission per station per beacon interval $R_{i,s}$ in practice. Afterwards, we show the adaptation of the control parameters with the help of R_i^{max} and illustrate the mechanism.

Reporting the Average Number of Retransmissions to the AP

In WLAN, the Access Point distributes the channel access parameters CWmin and CWmax with beacon frames. Therefore, it is most convenient to measure the number of retransmissions there as well. If we can assume that each station including the AP experiences a similar collision rate and delay, then we could rely on measurements at the AP. However, as we have seen in the previous section, the AP can behave very differently compared to its associated stations. Due to queuing and contention effects that are differently pronounced at an individual station and the AP, the collision probabilities and delays can vary significantly.

The collision probability perceived at the AP and at a station can differ up to a factor of three, cf. Section 2.2. This means that a contention window which is efficient regarding the number of retransmissions of the AP may entail a high number of retransmissions at its associated stations, leading to a poor overall channel utilization. We can cope with this unfair channel access using an explicit feedback mechanism from the stations. The feedback contains measurements of their individual retransmission probabilities and is transmitted in the WLAN header block. Based on that feedback, the AP is able to make reasonable contention window adaptation decisions and broadcasts the new parameters in beacon frames. The mechanism is illustrated in Figure 2.17.

The feedback table at the Access Point contains feedback from each station $s \in S$, where S is the set of all stations in the cell. A feedback value $R_{i,s}$ refers to the smoothed average number of retransmissions throughout beacon interval i at station s. Then, R_i^{max} in beacon interval i is defined by

$$R_i^{max} = \underset{s \in S}{max}(R_{i,s}). \qquad (2.21)$$

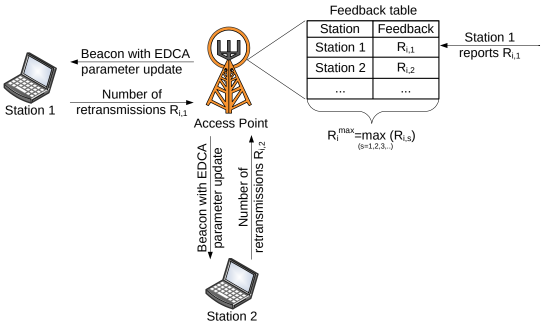

Feedback table
Station | Feedback
Station 1 | R_{i,1}
Station 2 | R_{i,2}
... | ...
Station 1 reports R_{i,1}
Beacon with EDCA parameter update
Number of retransmissions R_{i,1}
Station 1
Access Point
R_i^max=max (R_{i,s}) (s=1,2,3,..)
Beacon with EDCA parameter update
Number of retransmissions R_{i,2}
Station 2

Figure 2.17: *Feedback mechanism supporting EDCA parameter adaptation.*

The value R_i^{max} reflects the average number of retransmissions per packet of the worst station within the network. The collision probability of the AP is not taken into account as it is always lower than R_i^{max}. The DCWA algorithm at the AP determines an appropriate contention window based on R_i^{max}. Updates of the EDCA parameter set, including AIFS, CWmin, CWmax, and TXOP Limit, are then distributed via the beacon frame throughout the network. These updates include both the voice and the best effort access category parameters.

The DCWA algorithm controls the contention parameters CWmin and CWmax to keep R_i^{max} within a target range over time, independently of the current network load. The stability and efficiency of the mechanism is determined by the following four parameters:

- θ_{hi}: high control threshold
- θ_{lo}: low control threshold
- τ: inter-adaptation time, minimum time between two consecutive adaptations
- *M*: memory of the Time-Exponentially Weighted Moving Average (TEWMA) and smoothing factor for R_i^{max} [92]

θ_{hi} and θ_{lo} determine the thresholds for the contention window adaptation. Whenever R_i^{max} exceeds the higher threshold, the contention windows of all access categories are increased. The contention window parameters are increased as long as $CWmin\,[AC\_VO]$ does not exceed $maxCWmin\,[AC\_VO]$. In case R_i^{max} drops below the lower control threshold θ_{lo}, CWmin and CWmax of both access categories are decreased. They can be decreased until $CWmin\,[AC\_VO]$ reaches $minCWmin\,[AC\_VO]$. After a contention window adaptation, the algorithm waits for the duration of τ before changing the contention window again. The rationale behind this is to wait until effects of the contention window change have an impact on R_i^{max}.

In order to prevent R_i^{max} from oscillation, R_i^{max} is not only the maximum number of retransmission of the last beacon interval, but the smoothed average number of maximum retransmissions. The smoothing factor *M* determines the decay of the measured values and thus the agility of R_i^{max}. A small M means that only lately reported collision probabilities are considered, which means that R_i^{max} quickly reflects the recent contention status. With a large value of *M*, it will take longer until fundamental changes of the contention status are indicated by R_i^{max}. The exact definition of M is given in Subsection 2.3.3.

Algorithm 1 describes the DCWA in detail. Before broadcasting a beacon frame, the AP performs the DCWA procedure. The maximum value R_i^{max} of the reported feedback in the hash table serves as an input variable for the DCWA. Initially, the AP distributes the default IEEE 802.11e parameters.

The contention windows are updated - increased or decreased - depending on the current value of R_i^{max}. The contention windows are always updated for all service classes in order to maintain the prioritization between high and low priority traffic flows. An update can be repeated after the inter-adaptation time τ has elapsed.

An illustration of the DCWA algorithm and its parameters is shown in Figure 2.18(a). The x-axis shows the simulation time in seconds. The left y-axis marks the maximum number of retransmissions per packet and the second y-axis illustrates the currently used CWmin. The figure points out that the DCWA keeps

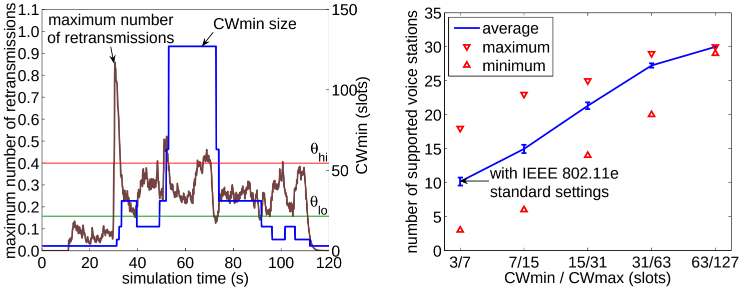
Algorithm 1 DCWA Algorithm.

```
$R_i^{max}$: maximum number of retransmissions per packet in beacon interval i
$\theta_{hi}$: high control threshold triggering CW increase
$\theta_{lo}$: low control threshold triggering CW decrease
$maxCWmin\,[AC\_VO]$: maximum $CWmin\,[AC\_VO]$
$minCWmin\,[AC\_VO]$: minimum $CWmin\,[AC\_VO]$
$last\ CW\ update\ time$: time of the last contention window update
$\tau$: inter-adaptation time, minimum time to elapse before the next update

if $(current\ time - last\ CW\ update\ time) > \tau$ then
    if $R_i^{max} > \theta_{hi}$ and $CWmin\,[AC\_VO] < maxCWmin\,[AC\ HP]$ then
        $CWmin\,[AC\_VO] = 2 \cdot CWmin\,[AC\_VO] + 1$
        $CWmax\,[AC\_VO] = 2 \cdot CWmax\,[AC\_VO] + 1$
        $CWmin\,[AC\_BE] = 2 \cdot CWmin\,[AC\_BE] + 1$
        $CWmax\,[AC\_BE] = 2 \cdot CWmax\,[AC\_BE]) + 1$
    else if $R_i^{max} < \theta_{lo}$ and $CWmin\,[AC\_VO] > minCWmin\,[AC\_VO]$ then
        $CWmin\,[AC\_VO] = (CWmin\,[AC\_VO] - 1)/2$
        $CWmax\,[AC\_VO] = (CWmax\,[AC\_VO] - 1)/2$
        $CWmin\,[AC\_BE] = (CWmin\,[AC\_BE] - 1)/2$
        $CWmax\,[AC\_BE] = (CWmax\,[AC\_BE] - 1)/2$
    else
        channel contention $R_i^{max}$ is within target range $\theta_{lo} \leq R_i^{max} \leq \theta_{hi}$
    end if
end if
```

R_i^{max} within the target range between θ_{hi} and θ_{lo}. Larger contention windows lead to smaller collision probabilities, but to longer access delays. Smaller contention windows lead to a high R_i^{max} and to a waste of available resources.

To show the performance gain of the DCWA compared to the standard settings, we implemented the DCWA algorithm into the OPNET simulation environment. Voice stations using the ITU-T G.711 [76] voice codec are configured with the IEEE 802.11g standard and a data rate of 54 Mbps. Figure 2.18(b) shows the increased capacity when using DCWA. The y-axis marks the number of supported voice stations whose QoS requirements can be met. Starting with a con-



(a) Influence of DCWA on CWmin and R_i^{max}.

(b) Increased capacity for voice traffic using DCWA.

Figure 2.18: *Illustration of DCWA and its impact on the number of supported voice stations.*

tention window of 3/7, the DCWA algorithm increases the contention window when more stations enter the system, until $maxCWmin\,[AC\_VO] = 63/127$ is reached. A further increase of the contention windows does not result in more supported voice stations because of large contention delays.

In addition to the increased performance when using the DCWA algorithm, the figure also reveals that the number of supported voice stations is not fixed for one setting. This was however shown in [93–101]. The reason why the number of supported stations cannot be exactly defined is the phase pattern as already indicated in Section 2.2. When several voice stations start their transmission at the same time within one voice frame, the collision probabilities due to same backoffs increase which lead to a lower number of supported voice stations.

2.3.3 DCWA Parameter Evaluation

The DCWA algorithm can be configured by four parameters θ_{up}, θ_{lo}, τ, and M. In the following, we identify good values for these parameters in practice. In order to evaluate the impact of a single parameter on the system performance, the concerned parameter is varied in a value range while keeping the other three parameters fixed.

The simulations for the parameter studies are based on the IEEE 802.11g standard with a saturated traffic model. The intention of the use of a saturated traffic model is to get an idea of the general system behavior. A station configured with a saturated traffic model always has a packet to transmit meaning that the transmission queue is never empty. In the following evaluations, the packet size is chosen according to the ITU-T G.711 voice packet size. The transmissions are started uniformly distributed within an interval of 100 ms. All stations have the same priority and use the highest priority access category, AC_VO, for their transmissions. The duration of an individual simulation run is 100 s whereas the first 20 s are considered as the transient phase. The simulations are repeated five times and a 90 % confidence interval is shown in the figures.

Influence of θ_{hi}

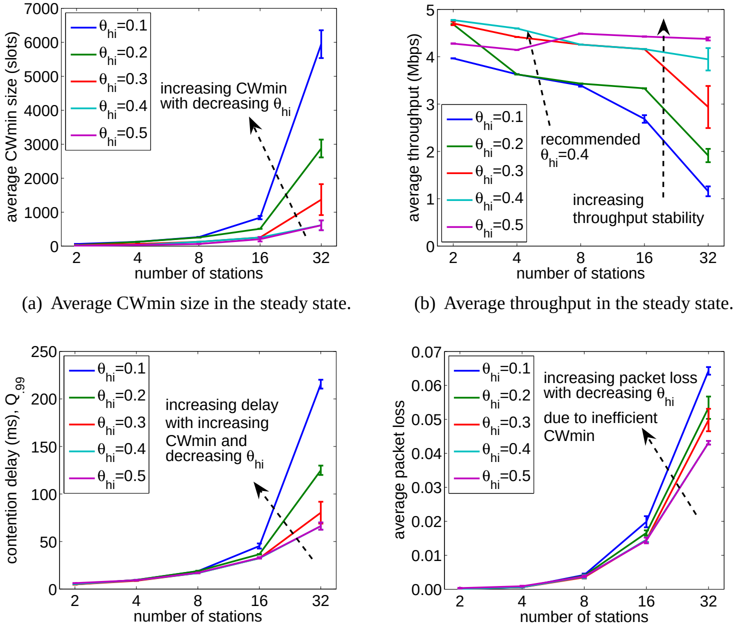
The first parameter to be evaluated is θ_{hi}, which is the high threshold for the contention window adaptation. It represents a limit for the maximum tolerable average number of retransmissions per packet over all stations in a WLAN cell. There is a trade off between the collision probability and the performance which can be tuned by the choice of a suitable contention window. If the contention window is chosen too small, many stations compete for the same transmission slot. Choosing the contention window too large wastes slots that are not used for any transmission. Hence, to achieve an optimal system performance, a certain level of retransmissions per packet is required. In order to study the influence of the threshold, θ_{hi} is varied from 0.1 to 0.5. The other DCWA parameters are fixed and set to $\theta_{lo} = 0.05$, $\tau = 1\,s$, and $M = 1$. Other parameter combinations were simulated as well, but they showed the same behavior and therefore, only selected parameter combinations are presented.

For the parameter evaluation, all stations start with the initial contention windows of $CWmin[AC\_VO] = 3$ and $CWmax[AC\_VO] = 7$ and no bursting is used, meaning that the TXOP Limit of the IEEE 802.11e standard is set to one packet. We distinguish between the transient phase and the steady state and evaluate appropriate measures in both phases. During the transient phase, new stations start their voice calls and the DCWA adapts the contention window until the number of stations reaches a fixed level. The duration of this phase is influenced by all parameters. The steady state considers the time when all stations have started their voice call and there are only a few more contention window adaptations until the end of the simulation.

Let us first take a look at the contention window as its size is controlled by the DCWA algorithm. The development of the average CWmin size during the steady state phase is shown in Figure 2.19(a). The increasing minimum contention window with an increasing number of stations reflects the higher contention when many stations compete concurrently for medium access. Besides this increasing CWmin based on the number of stations, we can observe that the smaller θ_{hi}, the higher the minimum contention window is increased. The reason is that for lower θ_{hi}, the threshold is reached sooner and the contention window is enlarged.

The DCWA control mechanism is in a stable state when the system reaches a steady state and the contention window parameters are not changed by the DCWA algorithm anymore. The stability depends on the choice of an appropriate parameter set of all four DCWA parameters. If the control range between θ_{lo} and θ_{hi} is chosen too narrow, the DCWA algorithm will never reach a stable state. In this case, the contention window suffers from oscillations. A wider control range ensures a stable contention window size where the DCWA algorithm only performs a few adaptations.

An inappropriate change of the contention windows due to a false setting of θ_{hi} leads to a performance loss in terms of throughput as shown in Figure 2.19(b). The figure depicts the impact of θ_{hi} on the average throughput during steady state. Three observations can be made from the figure. Firstly, an increasing number of stations lead to a decrease of the average throughput which is caused by higher contention. Secondly, the larger θ_{hi} is set, the more stable is the throughput when more stations are active in the WLAN cell. Finally, the average throughput



(a) Average CWmin size in the steady state.

(b) Average throughput in the steady state.

(c) Contention delay in the steady state.

(d) Average packet loss in the transient phase.

Figure 2.19: *Impact of the DCWA parameter θ_{hi} on the WLAN performance.*

increases up to a value of $\theta_{hi} = 0.4$. A θ_{hi} of more than 0.4 results in a lower average throughput when few stations are active in the system.

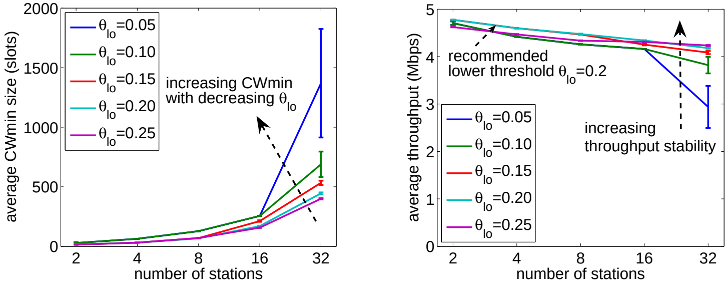
The influence of θ_{hi} on the contention delay is depicted in Figure 2.19(c). The 99 % quantile is plotted as it is an appropriate way to generate meaningful metrics. Long contention delays would just be considered as packet loss as the information becomes obsolete. The contention delay increases the larger the contention windows are set by the DCWA and the smaller θ_{hi} is chosen. The reason is obvious because large minimum contention windows lead to large average backoffs resulting in high contention delays.

Finally, the impact of the DCWA on the packet loss during the transient phase is shown in Figure 2.19(d). During this transient phase, packets are dropped due to high contention. An average packet loss of up to 6.5 % can be observed and even a slightly higher packet loss for $\theta_{hi} = 0.1$ which occurs due to CWmin oscillations. During steady state, the packet loss rate is extremely low and in an order of magnitude from 0.01 % to 0.1 % on average. This is an indicator for the robustness of the DCWA and can be observed in all results of the simulation series. Since the system is not experiencing an overall throughput gain for values above 0.4, and having the objective to maintain a system that is still sensitive to traffic changes, a high control threshold of $\theta_{hi} = [0.3, 0.4]$ is recommended.

Influence of θ_{lo}

After having found a suitable setting for θ_{hi}, the performance influence of θ_{lo} is evaluated. θ_{hi} is now fixed to 0.3 and θ_{lo} is varied between 0.05 and 0.25. This lower control threshold θ_{lo} represents the minimum empirical collision probability that is tolerable and is responsible for decreasing the contention windows as soon as the measured average collision probability drops underneath θ_{lo}. The lower it is set, the lesser the DCWA responds to traffic fluctuations. This behavior can be observed in Figure 2.20(a). Small values of θ_{lo} result in higher average contention window sizes, while the highest value $\theta_{lo} = 0.25$ effects the smallest average contention window size, but it effects as well the highest number of contention window adaptations throughout the simulation. The reason for the contention window oscillation is the small control range between θ_{lo} and θ_{hi}. Both, a too small θ_{lo} and a too small control range, lead to performance degradation in terms of average throughput as shown in Figure 2.20(b). The best performance is achieved when setting θ_{lo} to 0.15 or to 0.2. Using these values, the control range between θ_{lo} and θ_{hi} is large enough to prevent the system from oscillating.

The 99 % quantile of the contention delays is not plotted as it is always below 100 ms and only slightly affected by θ_{lo}. Furthermore, we want to point out that the average packet loss during the steady state phase is below 10^{-4} and partly even below 10^{-5} for all settings of θ_{lo}.



(a) Average CWmin size in the steady state.

(b) Average throughput in the steady state.

Figure 2.20: *Impact of the DCWA parameter θ_{lo} on the WLAN performance.*

We conclude that the width of the control range should be at least 0.1 and thus recommend to set the DCWA parameters to $\theta_{lo} = [0.15, 0.2]$ for a $\theta_{hi} = [0.3, 0.4]$.

Influence of M

Besides the two thresholds θ_{hi} and θ_{lo}, the memory M has a significant impact on the performance of the DCWA algorithm. Although the channel contention status changes very frequently in WLAN, we do not want to adapt the contention windows based on short-term fluctuations. Instead, the DCWA algorithm should react to real transient phases that are due to changed load conditions. In order to smoothly react on these changes, the Time-Exponentially Weighted Moving Average (TEWMA) introduced by Menth et al. [92] is used. The TEWMA calculates the time-dependent means of a series of values $X_i, i = 0, ..., n$ and is defined by

$$E[X](t) = \frac{S[X](t)}{N(t)}, \tag{2.22}$$

with $S[X](t_0) = X_0$ and $N(t_0) = 1$. The sums $S[X](t)$ and $N(t)$ are updated by

$$S[X](t_i) = S[X](t_{i-1}) \cdot e^{-\gamma \cdot (t_i - t_{i-1})} + X_i \text{ and} \tag{2.23}$$
$$N(t_i) = N(t_{i-1}) \cdot e^{-\gamma \cdot (t_i - t_{i-1})} + 1, \tag{2.24}$$

whenever a new value X_i updates the measurement. Here, γ is the devaluation factor. The memory of the TEWMA is

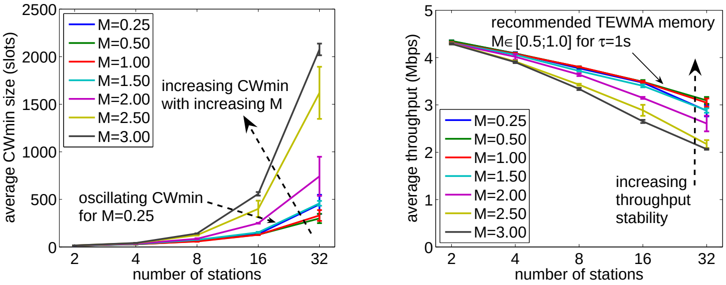
$$M = \int_0^\infty e^{-\gamma \cdot t} dt = \frac{1}{\gamma}. \tag{2.25}$$

An exact half-life period $T_H = \frac{\ln(2)}{\gamma}$ of a value can be derived by $\frac{1}{2} = e^{-\gamma \cdot T_H}$. For evaluating the impact of the memory M on the performance of the DCWA algorithm, the parameter settings listed in Table 2.8 are used.

Table 2.8: *Parameter settings for the evaluation of the memory M.*

DCWA parameter	Value range
θ_{hi}	0.3
θ_{lo}	0.2
M	0.25, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0
T_H	0.17, 0.35, 0.69, 1.04, 1.39, 1.73, 2.08
τ	1.0 s

Setting M to large values assigns greater importance to old values while current measured values are not reflected that soon. Consequently, the DCWA algorithm reacts slowly to a changing channel contention status which is shown in Figure 2.21(a). The figure displays the average CWmin size for different number of stations in the system and for different memories of the TEWMA during the transient phase. A large memory results in very large minimum contention windows due to too many successive contention window enlargements. However, when setting the memory too small, e.g. $M = 0.25$, the contention windows exhibit an oscillating behavior resulting in an increased average minimum contention window.

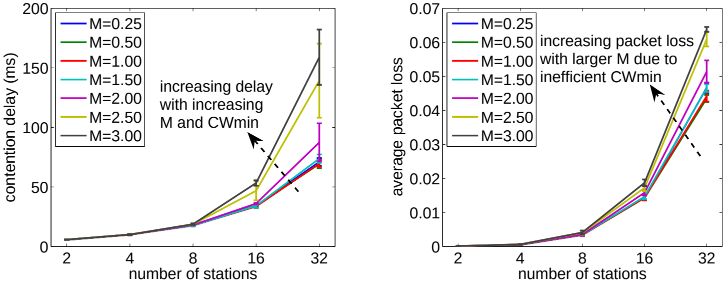


(a) Average CWmin in the transient phase.

(b) Average throughput in transient phase.

Figure 2.21: *Impact of the memory M on CWmin and average throughput.*

The impact of the TEWMA memory on the throughput in the transient phase is shown in Figure 2.21(b). A large memory leads to very large average contention windows and thus to a very long waiting time before accessing the channel which decreases the average throughput. The best performance in terms of throughput is achieved with a memory of 0.5 or 1.0. Beside the throughput decrease, the large contention windows cause long contention delays as shown in Figure 2.22(a) and higher average packet loss as shown in Figure 2.22(b). Similar to the highest throughputs, a memory of 0.5 or 1.0 achieves the smallest delays and lowest average packet loss. However, we have to outline that larger memories slightly outperform small memories during the steady state phase.



(a) Contention delay in the transient phase.

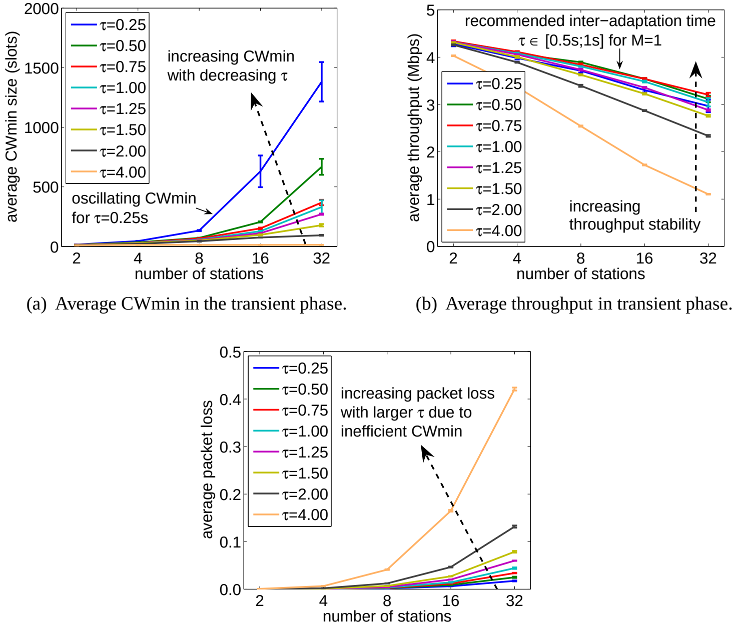
(b) Average packet loss in the transient phase.

Figure 2.22: *Impact of the DCWA parameter M on the contention delay and packet loss.*

Concluding the results of the impact of the memory, we have to point out that the performance decrease using a higher sensitivity is only marginal during the steady state phase, but the benefit of a more sensitive DCWA response during the transient phase is large. Thus, we recommend a TEWMA memory of $M \in [0.5, 1.0]$.

Influence of τ

Finally, we want to evaluate the influence of the inter-adaptation time. As we have seen, the memory of the TEWMA and the inter-adaptation time are strongly correlated. Thus, we have to find a proper adjustment of the two parameters in order to achieve a good performance with the DCWA algorithm. As already mentioned, τ defines the minimum waiting time between two successive contention window adaptations. For evaluating its impact, θ_{hi} is set to 0.3, θ_{lo} to 0.2, and the memory M to 1.0. The inter-adaptation time τ is varied between 0.25 and 4.0 seconds. Figures 2.23(a)-(c) show the impact on the contention window minimum, the average throughput, as well as the average packet loss during the transient phase.



(a) Average CWmin in the transient phase.

(b) Average throughput in transient phase.

(c) Average packet loss in the transient phase.

Figure 2.23: *Impact of the DCWA parameter τ on the WLAN performance.*

The figures reveal that the shorter the parameter τ is chosen, the larger are the average minimum contention windows. In contrast to the memory M, this does not result in performance degradation in terms of throughput. On the contrary, the smaller τ is chosen, the higher the average throughput is, except for $\tau = 0.25$. For such a small inter-adaptation time, the system adapts too fast, resulting in a form of oscillation. Thus, an inter-adaptation time of $\tau \in [0.5s, 1.0s]$ for a memory of $M = 1.0$ is recommended. With such a small inter-adaptation time, the contention windows are increased as much as needed to reduce the number of collisions significantly, resulting in a high average throughput.

Recommended DCWA Parameter Settings

The results underline that the memory M and the inter-adaptation time τ should be set to similar values to achieve a well working contention window adaptation. We recommend to set $M \in [0.5, 1.0]$ and $\tau \in [0.5s, 1.0s]$. As evaluated, the control range between θ_{lo} and θ_{hi} should be at least 0.1. Thus, we recommend to set θ_{lo} to 0.15 or 0.2 and θ_{hi} to 0.3 or 0.4. All recommended values for the parameters of the DCWA algorithm are summarized in Table 2.9.

Table 2.9: *Recommendation for the DCWA parameter configuration.*

DCWA parameter	Value range
θ_{hi}	$[0.3, 0.4]$
θ_{lo}	$[0.15, 0.2]$
M	$[0.5, 1.0]$
τ	$[0.5\,s, 1.0\,s]$

2.3.4 Performance of the DCWA Algorithm

After having derived a near-optimal parameter setting for the DCWA algorithm, we evaluate the performance of the algorithm in the presence of two different service classes.

Simulation Settings

For the evaluation, the DCWA is configured with θ_{hi}=0.4, θ_{lo}=0.2, M=1, and $\tau(s)$=1 s. The simulation was performed using the OPNET Modeler and the duration of a single simulation run is set to 100 s. The first 20 s are considered as the transient-phase. All of the following performance figures are generated on the basis of five replications by calculating the 95 % confidence interval. We use the same saturated traffic model for both service classes to get an idea about the general system behavior. The traffic model generates a bit stream with a packet size of 736 bit which is transmitted using UDP resulting in a MAC layer packet length of 1257 bit. The arrival rate of the packets is configured to saturate the MAC buffer at a station at any time. The number of stations in the system is set to 16 for the evaluation of the throughput and to 32 for the performance analysis in terms of contention delay. Again, the stations use the IEEE 802.11g standard with a data rate of 54 Mbps.

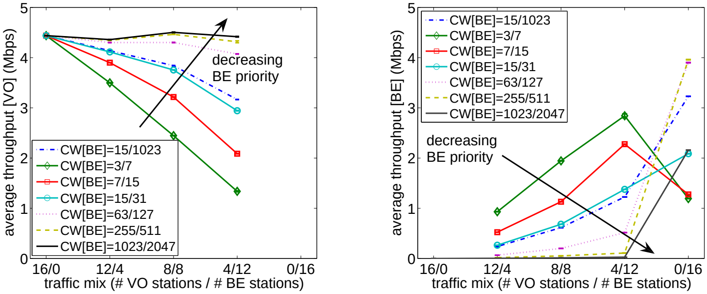
In order to study the effect of different priorities between high priority and low priority traffic flows, we define the prioritization level as the ratio between CW[AC_BE] and CW[AC_VO] as shown in Table 2.10. The prioritization level is calculated over the initial contention window settings. A contention window of 3/7 means that CWmin is set to 3 and CWmax is set to 7. As the IEEE 802.11 standard proposes a contention window of 15/1023, we also include this in our simulation settings.

Table 2.10: *Initial CW prioritization for AC_VO and AC_BE traffic using DCWA.*

CW[AC_VO]	CW[AC_BE]	Prioritization level≈ $\frac{CW[AC\_BE]}{CW[AC\_VO]}$
3/7	3/7	1
3/7	7/15	2
3/7	15/31	3
3/7	15/1023	-
3/7	63/127	5
3/7	255/511	7
3/7	1023/2047	9

Average Throughput and Contention Delay with different Prioritization Levels

The influence of prioritization levels on the throughput and contention delay is illustrated for different combinations of VO and BE traffic stations in Figure 2.24 and Figure 2.25. A prioritization level of 1, CW[AC_BE]=3/7, shows a bad performance in terms of throughput and contention delay for voice traffic flows.



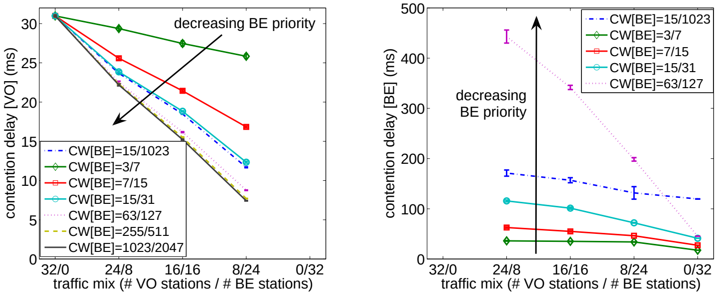
(a) Throughput of VO traffic.

(b) Throughput of BE traffic.

Figure 2.24: *VO and BE throughput of a 16 station scenario.*

The higher the prioritization level, the higher the VO throughput and the lower the BE throughput and vice versa. A rise of the prioritization level results in a linear decrease of BE throughput. In presence of 100 % BE traffic, the behavior is different. This results from the fact that the DCWA only reacts on retransmissions of VO traffic and as there is no VO traffic, the contention windows remain constant. Thus, a small contention window setting like 7/15 leads to a lot of collisions. The choice of a ’broad’ contention window of 15/1023 mitigates this problem and achieves a good throughput performance for each scenario. Its prioritization level corresponds approximately to 15/31 and additionally has the ability to better adapt to the given traffic mix.

Furthermore, the higher the prioritization level, the shorter the contention delay for VO traffic and the longer the contention delay for BE traffic. Up to a prioritization level of 5, CW[BE]=63/127, the VO delays are reduced. The absolute reduction amounts to 20 ms, inferred from Figure 2.25(a), data point 8/24. At the same time, BE delays exceed 200 ms. A prioritization beyond this level does not reduce VO delays anymore, but increases BE delays tremendously.



(a) Contention delay of VO traffic.

(b) Contention delay of BE traffic.

Figure 2.25: *VO and BE contention delay of a 32 station scenario.*

We can summarize that a larger contention window of BE traffic reduces the contention delay of VO traffic. The reason are smaller contention windows compared to BE traffic which also results in longer contention delays of BE traffic. The higher the prioritization level is chosen, the more obvious this phenomenon is. We recommend thus to set the contention windows for the BE traffic class to 63/127 or to the initial settings 15/1023 as recommended by the IEEE 802.11-2007 standard. Using this prioritization level between VO and BE traffic, up to 30 high priority voice users can be served no matter if there is BE traffic present in the cell or not. This was already shown in Figure 2.18. The DCWA increases the contention window of VO traffic to CWmin=63 and CWmax=127 while increasing the contention window of BE traffic equally, meaning that the prioritization level is preserved. Thus, up to 200 % more voice users can be supported. However, BE traffic suffers from this performance increase for VO traffic. In the next section, we try to improve the performance of the BE traffic in the presence of the DCWA algorithm by increasing the TXOP Limit.

2.4 Impact of Best Effort Frame Bursting

In the previous section, we demonstrated how the DCWA can handle both VO and BE traffic. It was shown that even with a low prioritization level, BE traffic has to accept severe resource limitations. For this reason, we explore the throughput improvement for BE traffic through frame bursting. Frame bursting, expressed by the parameter TXOP Limit, allows a station to transmit several data packets during each won Transmission Opportunity (TXOP). When setting the TXOP Limit to more than one data frame for BE traffic, more medium resources are granted to BE traffic. Before showing the influence of frame bursting on the VO traffic and the throughput improvement of BE traffic, the work related to frame bursting is reviewed.

2.4.1 Related Work

Burst adaptation mechanisms for WLAN mainly focus on burst adaptations of real-time traffic flows. Analytical models to show the impact of the TXOP Limit are presented in [102–105]. The first two papers present an analytical model showing the influence of the transmission burst size when using the DCF. The latter two papers analyze the impact of the TXOP Limit for the EDCA. However, the limit is set similar for every service class. It is claimed that the size of the TXOP Limit should be configured carefully to prevent low priority traffic from starvation.

Simulation studies of the TXOP Limit are presented in [106, 107]. Similar to the analytical papers, it is claimed that the TXOP Limit should be set proportional to the buffer size. Majkowski and Palacio [108, 109] on the other hand do not only measure the buffer size, but also take the transmission speed into account and introduce a new TXOP Limit mechanism called Enhanced TXOP (ETXOP) to optimize the system throughput. The mechanism is validated through OPNET simulations. Another OPNET simulation is performed by Liu and Zhao [110]. In this paper, the TXOP Limit is adjusted according to estimations of the incoming video frame size. The duration of a burst is then set to the time necessary to transmit the video frames pending in the buffer and additional frames expected by the estimator.

Cranley et al. [111] present another TXOP Limit study for video streaming. However, it is also claimed that the TXOP Limit is not suitable for audio streams because of the constant bit rate streams. The algorithm proposed by Huang et al. [112] is capable of dynamically adapting the TXOP Limit to a varying PHY rate, channel condition, and network load. Compared to the values proposed by the IEEE 802.11 standard, the proposed mechanism assures a more superior and stable throughput performance for voice and video ACs. Andreadis and Zambon [113, 114] claim that the amount of traffic transmitted on the downlink might sometimes also be lower than the uplink traffic which is, according to them, not considered in other papers. In order to cope with the varying traffic load, they adapt the TXOP Limit at the Access Point according to the network load which can lead to a smaller TXOP Limit at the AP than at the stations.

Ksentini et al. [115] differentiate between the TXOP Limit for high priority ACs and low priority ACs. The TXOP Limit for AC_VO and AC_VI is dynamically set according to the flow's data rate and flow's priority, whereas the TXOP Limit for AC_BE and AC_BK is fixed. The results show that the throughput is improved and the delay is reduced.

None of these paper analyze the performance effects of the TXOP Limit if it is just used for the low priority best effort traffic class. In this section, we focus on a burst adaptation scheme for this traffic class while additionally using the DCWA algorithm from the previous section. Our goal is to set the TXOP Limit for the best effort class as large as possible to ensure a high throughput without disrupting high priority traffic flows.

2.4.2 Throughput Improvement Through Frame Bursting in a Saturated Environment

After reviewing the related work, we first want to explore the throughput improvement for best effort frame bursting in a saturated environment, before considering a more realistic scenario.

Simulation Settings

To study the impact of the TXOP Limit[BE] extension, we configure a saturated traffic model. The intention behind this is to get an idea about the general system behavior. A saturated station, no matter which service class is used, always has a packet of size 1500 Bytes to transmit. In the following, we use two different station types, high priority voice and low priority best effort stations. The TXOP Limit[VO] is set to one MSDU for all simulation scenarios. The differentiation is realized by extending the TXOP Limit[BE] to the values listed in Table 2.11.

Table 2.11: *TXOP Limit[BE] extension.*

Duration (ms)	# MSDU per burst
0.32	1
0.64	2
1.28	4
3.20	10
6.40	20
16.00	50

The duration of a complete transmission cycle of a single MAC packet including SIFS and its ACK takes approximately 0.32 ms. A TXOP Limit[BE]=6.4 ms means that once a BE station has won a TXOP, it has the right to transmit frames for 6.4 ms which corresponds to 20 packets. Furthermore, in order to support more stations in the system, the DCWA algorithm is used which increases the contention windows to approximately CWmin=63 and CWmax=127 for voice traffic and to CWmin=255 and CWmax=16,383 for best effort traffic.

Impact of Frame Bursting on Throughput and Contention Delay

In order to study the impact of frame bursting on the throughput, we configure a scenario with 16 voice and 16 best effort stations. Figure 2.26 shows the average throughput where each curve is plotted as a function of TXOP Limit[BE]. With increasing the TXOP Limit[BE] up to 50 MSDUs per burst, the total throughput increases up to 34.2 Mbps which is a gain of 40 % relative to 24.5 Mbps for a burst size of 1. The explanation of the flattening of the total throughput curve is that performance degradation due to collisions is mitigated through long burst sizes and reduced contention. This effect becomes saturated at a certain point, as also observed by the authors in [102].

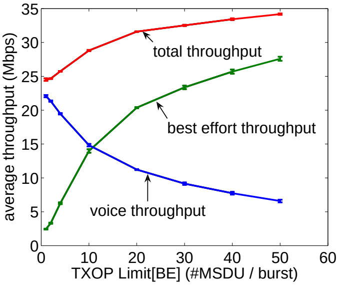


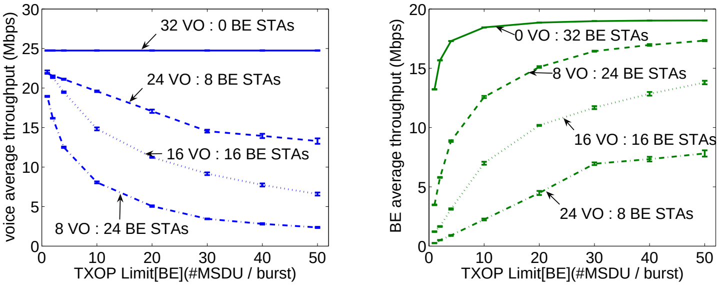
Figure 2.26: *Impact of an extended TXOP Limit[BE] on the total throughput and on the throughput distribution among voice and best effort traffic.*

Altogether, this shows that the parameter TXOP Limit[BE] is an extremely powerful means to realize throughput optimization. Figure 2.27 shows the average throughputs of voice and BE stations for the same scenario and for different traffic mixes. The decrease in voice throughput and the increase in BE throughput is visible for all traffic mix constellations.

Most interesting are the differences among the set of curves and their progression with the traffic mix. In general, the fewer voice stations and the more BE stations are in the scenario, the lower the voice throughput and the higher the

BE throughput. For TXOP Limit[BE] values of 20 and 30 the decline in voice throughput and the increase in BE throughput becomes tremendous. At that point, the contention window prioritization of voice traffic is far outweighed by TXOP Limit[BE] prioritization of BE traffic.

A main result of the throughput analysis is that applying a TXOP Limit[BE] extension increases the capacity for BE traffic. This capacity increase is realized independently of the traffic mix. A further result is that BE throughput prioritization with TXOP Limit[BE] works very effectively and can counterbalance negative throughput impacts through voice contention window prioritization. It is a useful means to distribute available resources among traffic classes.



(a) Throughput of voice stations.

(b) Throughput of best effort stations.

Figure 2.27: *Impact of an extended TXOP Limit[BE] and of the traffic mix on voice and best effort throughput, 32 station scenario.*

2.4.3 Impact of Frame Bursting in a Realistic Scenario

In this subsection, we assess the applicability of the concurrent use of the DCWA and a TXOP Limit[BE] enlargement in presence of voice and best effort traffic. We want to explore the capabilities of a joint DCWA - TXOP Limit[BE] control under more realistic conditions to draw conclusions about its benefits in practice.

Simulation Settings

For the following simulations we set up high priority voice stations and low priority best effort stations. A voice station uses the ITU-T G.711 [76] voice codec with a packet size of 640 bits and an inter-arrival time of 10 ms. This voice codec is used because it is widely implemented in VoIP WLAN phones. However, tests with other voice codecs showed a similar behavior. A best effort station downloads files from the Access Point using TCP with a packet size of 1500 Bytes. The performance figures presented here refer to scenarios with 20 best effort stations, while the number of voice stations varies from 5 to 30. In order to support a maximum number of voice stations, the DCWA algorithm is configured as recommended in Table 2.9 with θ_{hi}=0.3, θ_{lo}=0.15, M=1, and $\tau(s)$=1 s. The DCWA adapts to the contention level of the high priority voice queue and controls both the contention windows of AC_VO and AC_BE, with the latter being linearly controlled with AC_VO. Their initial values are set to CWmin[AC_VO]=3, CWmax[AC_VO]=7, CWmin[AC_BE]=15, and CWmax[AC_BE]=1023.

The duration of an individual simulation is 200 s. The stations start equally distributed within [0 s;50 s]. Thus, the first 60 s are regarded as transient phase and are not considered for the statistics. The performance figures are generated on the basis of 30 replications, applying a 95 % confidence interval. In order to study the influence of best effort frame bursting, we simulate the scenario described above with the TXOP Limit[BE] values provided in Table 2.11.

Throughput Improvement

The average throughput improvement of all best effort stations is shown in Figure 2.28. The figure illustrates the improvement for an increasing number of voice stations from 5 to 30. The larger the number of voice stations, the lower is the best effort average throughput which is caused by the strict prioritization of voice stations. For 30 voice stations, WLAN operates close to its capacity limit as already shown in the previous section. When the TXOP Limit[BE] is increased, the capacity limit is reached sooner, because the QoS for the voice stations cannot be met anymore. In detail, through the use of a large TXOP Limit[BE], the arrival rate of voice packets surpasses the sending rate of the voice access category, leading to buffer overflow and hence to dropped voice packets at the Access Point.

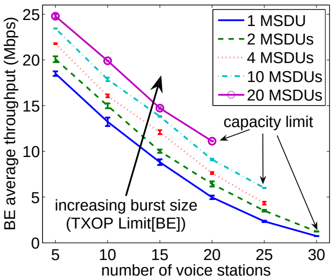


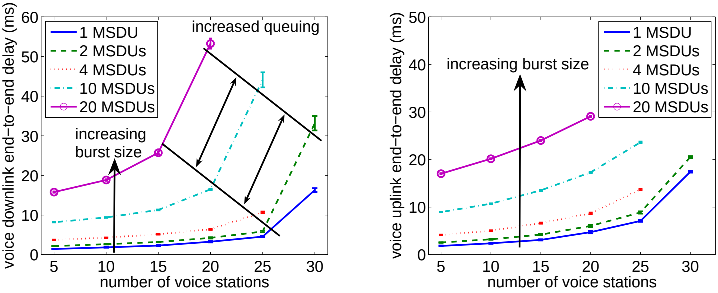
Figure 2.28: *Impact of an extended TXOP Limit[BE] on the best effort throughput.*

We recognize that best effort traffic achieves as much bandwidth as possible in dependency of the amount of prioritized voice traffic. Furthermore, it is shown that using extended frame bursts is very beneficial to increase best effort throughput performance. However, enlarging the burst size of the best effort access category must be done with care as we will see in the next paragraphs.

Consequences on Delay

Frame bursting affects voice and best effort traffic in different ways. Voice delay suffers in both directions, from the Access Point to the station and vice versa. Best effort delay on the other hand only suffers on the uplink. On the downlink, the delay is reduced. For our simulation we consider the end-to-end delay. The end-to-end delay consists of the queuing and the contention delay. Furthermore, we differentiate between downlink and uplink delays.

The end-to-end delays for voice traffic of both downlink and uplink are depicted in Figures 2.29. The more voice stations are in the system, the larger the end-to-end delay gets. This is on the one hand caused by longer medium busy times when more stations are transmitting and on the other hand by the use of the DCWA control algorithm. The DCWA increases the contention windows when more stations are present in the cell, resulting in larger backoff times. The figures furthermore show the impact of an increased burst size on the end-to-end delay. Especially the downlink suffers from an increased burst size. When the WLAN is close to its capacity limit, the queuing delay for each packet increases.

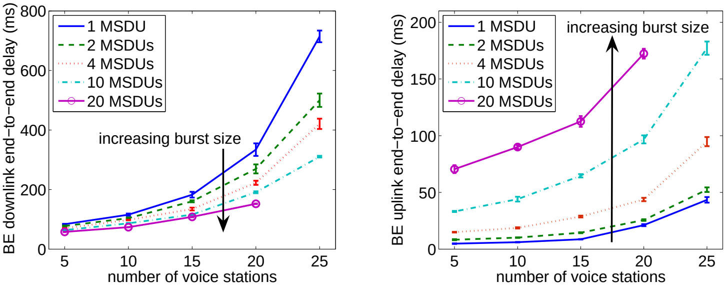


(a) Voice downlink end-to-end delay.

(b) Voice uplink end-to-end delay.

Figure 2.29: *Impact of voice traffic and TXOP Limit[BE] on voice delay.*

Figure 2.30 shows the impact on the best effort end-to-end delay. As already mentioned, best effort delay benefits and suffers from an increased buffer at the same time. It benefits because if more frames are transmitted per TXOP, the contention delay for each packet transmitted in one burst decreases. In addition to the contention delay, the queuing delay at the MAC buffer decreases. Looking at Figure 2.30(a), the end-to-end delay decreases from 370 ms down to 150 ms when the TXOP Limit[BE] is increased from 1 MSDU to 20 MSDUs in the presence of 20 voice stations. However, the uplink end-to-end delay suffers from an increased burst size as can be derived from Figure 2.30(b). In contrast to the Access Point, queuing is not relevant for the stations and the fewer amount of contention slots during a time period with an increased burst size lead to larger end-to-end delays on the uplink.



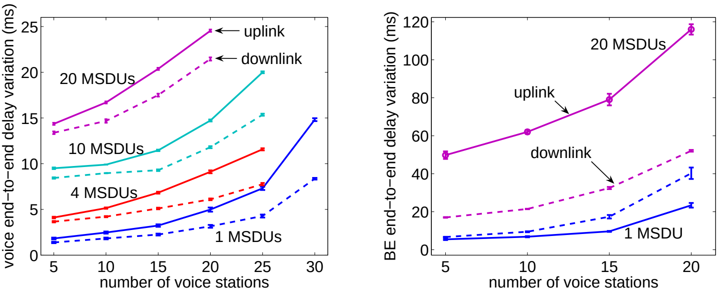
(a) BE downlink end-to-end delay.

(b) BE uplink end-to-end delay.

Figure 2.30: *Impact of voice traffic and TXOP Limit[BE] on best effort delay.*

Delay Variation Influences

Finally, we take a look at the end-to-end delay variation. Figure 2.31(a) exhibits the increase of the voice end-to-end delay variation for both the uplink and the downlink when increasing the number of voice stations and when increasing the best effort burst size. We notice the difference between uplink and downlink. This phenomenon has two reasons. Firstly, the voice downlink benefits from frame bursting. With an increasing number of voice stations, queuing at the Access Point rises, resulting in downlink frame bursts with several voice packets. Consequently, bursts of voice frames on the downlink reduce contention delay, and therefore reduce the variability of the packet delay. The second reason is the unfairness between stations and Access Point described in Section 2.2. The contention delay of the voice downlink transmissions is much smaller compared to the uplink contention delay because the Access Point competes against fewer transmissions than the stations. This results in a larger end-to-end delay variation of the uplink flows.



(a) Voice delay variation.

(b) BE delay variation.

Figure 2.31: *Impact of voice traffic and TXOP Limit[BE] on voice and best effort delay variation.*

The results of best effort end-to-end delay variation can be seen in Figure 2.31(b). It also exhibits differences among downlink and uplink. The use of frame bursting leads to significantly increased variability of the uplink delay. Larger burst durations force other stations to wait longer before they can access the medium. The impact on the downlink delay variation is twofold. On the one hand, transmission bursts increase the contention delay variation for voice traffic similarly to the variability of the voice uplink delay. On the other hand, the increase of the variability is not that pronounced for best effort traffic. The reason is that the first packet within a burst experiences a large delay variation due to longer frame bursts while the variablility of the other packets within a burst remains constant.

Summarizing the results of an increased burst size, we want to point out that an increased burst size for the best effort traffic class can prevent best effort flow starvation without harming voice QoS requirements.

2.5 Lessons Learned

The objective of this chapter was to enhance the performance in WLAN infrastructure networks. Therefore, a measurement-based adaptation algorithm for the channel access parameters is proposed. In a broad range of simulation scenarios, the performance of the algorithm is evaluated and improved. The results show that the traffic performance for both, high priority voice and low priority best effort traffic is significantly enhanced.

Before optimizing the channel access, we revealed an unfairness between stations and Access Point in terms of collision probabilities. Other works on fairness in wireless networks just consider throughput unfairness per flow but do not take into account the collision probabilities and the contention delays. Thus, they show that downlink flows experience lower throughputs. The result is that the contention parameters are tuned based on measurements at the Access Points. However, measuring the network load in terms of collision probabilities at the Access Point does not reflect the overall situation in WLAN. In order to cope with this unfairness, a measurement-based feedback mechanism is designed which enables a very accurate assessment of the channel contention level. Using the measured channel contention level, we developed the Dynamic Contention Window Adaptation (DCWA) algorithm to keep the channel contention at an efficient level independent of the current network load.

An efficient DCWA parameter configuration was derived to optimize the achievable capacity and it is shown that the algorithm effectively increases the wireless resources available for high priority traffic. Up to 200 % more voice connections can be supported compared to the IEEE 802.11e standard while still meeting the QoS requirements. A key finding is that the contention window is a very powerful means to realize service differentiation. The amount of wireless resources granted to an access category and the experienced packet delivery delay are heavily impacted depending on the degree of the contention window prioritization. The DCWA was extended to simultaneously control the contention windows of both high priority voice and low priority best effort traffic. By maintaining the prioritization level between the two service classes, the QoS requirements for voice traffic can still be met at any time and best effort traffic is granted the remaining wireless resources.

These remaining wireless resources can however be very low, depending on the number of voice stations in the WLAN and can even get starved under some conditions. Therefore, we proposed frame bursting for best effort traffic to increase its throughput while controlling the loss of prioritization for high priority voice traffic. We showed that voice traffic is still prioritized over best effort traffic whose throughput is significantly increased. Furthermore, the best effort throughput improves because of an increased WLAN resource efficiency which is due to reduced protocol overhead and due to reduced contention. Best effort frame bursting can effectively counterbalance the negative impact of contention window prioritization on best effort traffic.

Simulations of voice and best effort stations in a WLAN cell showed that increased frame bursts lead to more residual capacity for best effort traffic but also reduce the number of supportable voice stations that enjoy prioritized transmission. The limitation of the number of voice stations is caused by too large queuing delays for typical voice applications.

Lessons learned from this chapter are that it is neither sufficient to measure the channel contention level at one site nor to set the contention parameters to fixed values. The proposed measurement-based feedback algorithm increases the overall cell capacity significantly compared to fixed settings of the channel access parameters by adaptively minimizing the contention delay when possible and maximizing the throughput when needed. Furthermore, an increased TXOP Limit for best effort traffic helps to prevent best effort flow starvation.

3 Quality of Experience Control in WLAN Mesh Networks

> "If you think in terms of a year, plant a seed; if in terms of ten years,
> plant trees; if in terms of 100 years, teach the people."
> Confucius (551 B.C.-479 B.C.)

In the last chapter, we have seen how to improve the performance in a WLAN infrastructure network. In this chapter, we take a closer look at Wireless Mesh Networks (WMNs) [116]. In general, WMNs are a combination of infrastructure and ad-hoc networks. Similar to wireless ad-hoc networks, no central unit is established to distribute traffic and the data is sent directly from neighbor node to neighbor node. If data would reach nodes that are not directly reachable neighbors, the packets are sent on a multi-hop route. All nodes provide relaying capabilities to forward traffic through the network to reach the destination. However, in contrast to wireless ad-hoc networks, WMNs are normally comprised of static devices and focus on reliability, network capacity, and are mainly used as an alternative to a wired network infrastructure.

Due to the advantages of WMNs like self-organization and self-healing, several standardization groups have been set up. The first standardization group for WLANs was started in 2003 under the extension IEEE 802.11s [117]. However, the standard is not completely finished and thus, no implementations are available yet. Besides the IEEE 802.11s standard, further standardization groups for WMNs like IEEE 802.15.5 [118] and IEEE 802.16-2004 [119] underline the importance of wireless mesh networks [120].

Major research aspects in WMNs are intelligent path reservation, routing strategies, and Quality of Service (QoS) support [121]. In this chapter, we present a distributed, measurement-based approach to support real-time traffic in WLAN-based mesh networks. The aim of the proposed mechanism is to keep track of the services currently present in the network and to meet the QoS requirements. The objective measurable QoS parameters are then mapped to the user-perceived Quality of Experience (QoE), expressed through the Mean Opinion Score (MOS) [122].

In contrast to the previous chapter, the measurement-based approach is implemented on the network layer. An implementation on the WLAN MAC layer would be possible within a simulation environment, but our approach is also evaluated in a WMN testbed and this would imply an update or recreation of all WLAN drivers. The network layer is instead totally software-based and thus easy exchangeable.

The remaining of this chapter is structured as follows. We first give an introduction to wireless mesh networks by showing the WMN architecture as well as routing mechanisms for WMNs. Subsequently, the challenges of QoS provisioning in WMNs and the related work is presented. In the second part of this chapter, the mechanism is introduced, validated in a testbed implementation, and extended by means of simulation.

3.1 Wireless Mesh Networks

WMNs are normally wireless networks with fixed node positions and without any energy constraints. These characteristics enable the possibility to perform cross-layer optimization to ensure a sufficient QoS level. Especially this QoS support and the resulting QoE might lead to a great success in future broadband wireless access [123, 124]. Before reviewing the work related to QoS support in WMNs, we take a look at the WMN architecture, introduce WMN routing mechanisms, and show the challenges of QoS support in WLAN-based mesh networks.

3.1.1 WMN Architecture

A WMN is normally organized in a three-tier, hierarchical structure like shown in Figure 3.1. Starting at the bottom, normal non-mesh capable wireless or wireline clients are attached to the mesh network by Mesh Access Points (MAPs). These MAPs, together with other Mesh Points (MPs), form the mesh network itself. An MP is responsible for mesh relaying, meaning that it is capable of forming an association with its neighbors and forwarding traffic on behalf of other MPs. An MP can be equipped with one or more wireless interfaces. Using several wireless interfaces, the interference in a WMN can be reduced but the complexity of the channel assignment for the interfaces increases. In this chapter, all wireless mesh points are configured with only one interface operating on one channel in the 5 GHz frequency band. At the top of the hierarchy stands a Mesh Gateway (MGW). An MGW bridges traffic between different WMNs or connects the WMN to the Internet. In general, there can be more than one MGW. This however leads to further problems, e.g., which MGW to use for routing traffic to and from the Internet. Thus, we consider only one MGW.

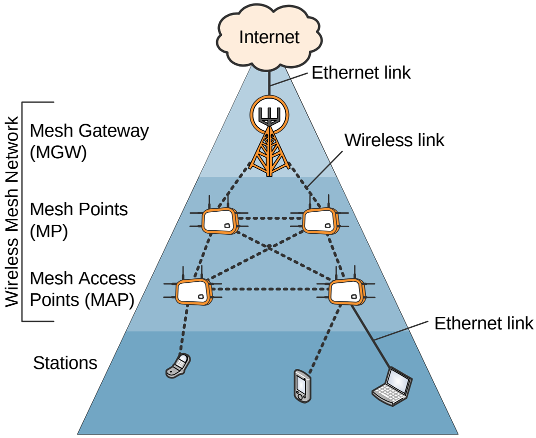


Figure 3.1: *Wireless Mesh Network Hierarchy.*

Besides problems with multiple gateways and MPs with multiple interfaces, path reservation and routing are major problems in WMNs. In the following, we first give a general overview of routing in WMNs and then introduce the Optimized Link State Routing (OLSR) in more detail as we use this routing protocol for QoE optimization later in this chapter.

3.1.2 Routing in Wireless Mesh Networks

Generally, routing protocols can be divided into two different classes, proactive routing protocols and reactive routing protocols. When using proactive routing protocols, the routers or nodes maintain a current list of destinations and their routes by distributing routing tables periodically and in case of a recognized change. During the last years, a large variety of proactive routing protocols have been introduced or extended. Two widely-used proactive protocols are Optimized Link State Routing (OLSR), standardized in the Request for Comments (RFC) 3626 [125] and Destination Sequence Distance Vector (DSDV) [126].

Reactive routing on the other hand means that the routes are found on demand by flooding the network with route request packets. Popular reactive routing protocols are Dynamic Source Routing (DSR) and Ad-hoc On-demand Distance Vector (AODV). DSR was first brought up in 1994 by Johnson [127] and is now standardized in the RFC 4728 [128]. It is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad-hoc networks. AODV is a combination of DSDV and DSR and was introduced by Perkins and Royer in 1999 [129] and standardized under the Request for Comments 3561 in 2003 [130]. The philosophy in AODV is that topology information is only transmitted by clients on demand.

Besides these two main routing mechanisms, a lot of new methods have been designed for routing. The Zone Routing Protocol (ZRP) and the Hazy Sighted Link State (HSLS) routing protocol use a combination of proactive and reactive routing called hybrid routing protocol. The IEEE 802.11s standard also introduces a hybrid routing protocol called Hybrid Wireless Mesh Protocol (HWMP), which is located on the data link layer and uses a combination of on demand path selection and proactive topology tree extensions. The modes can not be used exclusively because the proactive mode is an extension to the on demand mode. The on demand mode is used when no gateway is present or when the on demand path can provide a better path to the destination. To establish a path using the on demand mode, a mesh point broadcasts a path request frame. The target mesh point replies with a path reply message similar to AODV. For the proactive tree mode, two different mechanisms, a proactive path request and a root announcement, are described in the IEEE 802.11s draft. For the proactive path request, the gateway broadcasts a path request periodically and the attached mesh points update their tables according to the path on which the request is received. If a bidirectional path is required, the mesh points reply to this request. The second mechanism, the router announcement works similar in such a way that the gateway broadcasts root announcements. The mesh points do not reply to these announcements but take the opportunity to send path requests to the gateway if they have to create or refresh a path. These individually addressed request work similar to the on demand mode.

Performing routing on the data link layer gives the possibility to get rid of the network layer at the mesh points. However, as the standard is not yet completed and implementations on the data link layer are rather complex, we perform the routing in the WMN on the network layer, using an OLSR implementation.

Optimized Link State Routing

The aim of OLSR is to inherit the stability of link state algorithms. Due to the fact that it is a proactive routing protocol, it has the advantage of having the routes immediately available when needed. Each OLSR node transmits control messages periodically and can therefore sustain a loss of some packets from time to time. Furthermore, the protocol does not need an in-order delivery of its messages because each control message contains a sequence number. The control messages are transmitted to specific nodes in the network to exchange neighborhood information. These messages include the node's IP address, the sequence number, and a list of the distance information of the node's neighbors. After receiving this information from its neighbors, a node sets up a routing table. Whenever a node wants to transmit a data packet, it can calculate the route to every node with the shortest path algorithm. A node only updates its routing table if a change in the neighborhood list is detected, a route to any destination has expired, or a better, shorter route is detected for a destination.

Up to this point, OLSR resembles the normal Link State Routing (LSR) protocol. The difference between OLSR and LSR is that OLSR relies on Multi-Point Relays (MPRs). An MPR is a node which is selected by its direct neighbor. For the selection of an MPR, a node picks a number of one-hop neighbors as an MPR set so that all two-hop neighbors can be reached by these MPRs and the number of required MPRs is minimal. The first idea of multi-point relays is to minimize the flooding of broadcast messages in the network. An information packet should not be sent twice to the same region of the network. The second idea is that the size of the control messages is reduced. Figure 3.2 shows the benefits of the MPR algorithm compared to normal link state routing.

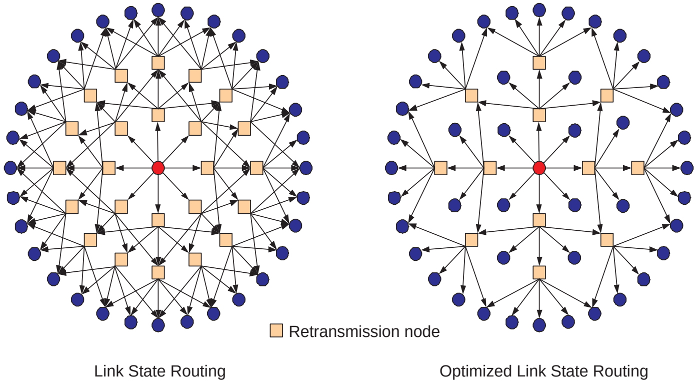


Figure 3.2: *Comparison of LSR and OLSR.*

For the measurements in our WLAN-based mesh testbed, we use the OLSRd implementation by Andreas Tønnesen [131]. He developed OLSRd in his master thesis, with the project moving on after he finished. It is widely used and well tested on community WMNs like Athens wireless network, Freifunk network, and Funkfeuer with 2000 nodes, 600 nodes, and 400 nodes respectively.

3.1.3 Challenges of QoS Support in WMNs

In wireline networks, QoS is normally supported using over-provisioning of bandwidth or other resources. This does however not work in WMNs, as WMNs suffer from limited bandwidth. Furthermore, if we want to support voice traffic flows in WMNs, we face problems such as a) packet loss due to interference on the unlicensed frequency band, b) high overhead of the protocol stack, c) queuing problems due to interfering best effort flows on the same path, and d) interference problems due to traffic on neighboring paths.

Compared to wireless infrastructure networks where no interference problems on neighboring paths and self-interference problems on the same path occur, WMNs can support a smaller amount of QoS dependent traffic. If a WMN runs on one single channel, the number of supported voice traffic flows decreases with the number of hops as shown in Figure 3.3. The figure illustrates the average number of supported G.711 voice traffic flows with activated DCWA. In a single hop environment, a wireless infrastructure network, about 30 voice stations are supported, cf. Figure 2.18(b) from the previous chapter. However, when increasing the number of hops to the mesh gateway, the number of supported voice stations decreases down to an average of 2.83 for six hops. The reason for this tremendous decrease is the self-interference produced by different packets of the same flow competing for medium access at different mesh points. If, in addition to the self-interference, interfering traffic on neighboring paths is added to the network, the number of supported voice stations decrease even more.

There are basically four different possibilities to increase the number of supported voice stations in WMNs. The simplest way is to increase the number of

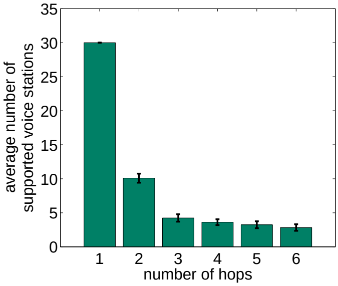


Figure 3.3: *Average number of supported voice stations, depending on the number of hops.*

interfaces at each MP. In such a multi-radio, multi-channel environment, the interference between neighboring mesh points can significantly be reduced. The second possibility is to reduce the protocol overhead. Although the header compression of the RTP, UDP, and IP headers alone has no positive effect on the number of supported voice stations as we have shown in [27], in combination with packet aggregation, more voice stations can be supported, cf. Niculescu et al. [132]. Third, traffic flows can be locally separated as much as possible. This is however not feasible as such a WMN would have to be very large and marginally loaded which is not cost efficient. Finally, the bandwidth of interfering traffic can be reduced. In this chapter, we show a mechanism to increase the QoE of voice stations by monitoring their performance and controlling the bandwidth of interfering flows on the same or adjacent paths when needed. We first take a closer look at the work related to QoS support in WMNs.

3.1.4 Related Work

The feasibility of routing metrics for WMNs has been studied in several papers. Campista et al. [133] differentiate between ad-hoc routing protocols, controlled flooding, traffic aware routing, and opportunistic routing. The evaluation of four routing metrics, Expected Transmission Count (ETX), Expected Transmission Time (ETT), Minimum Loss (ML), and Hop Count show that the Hop Count metric performs worse in terms of loss rate and roundtrip times over several hops compared to the other three metrics. Further comparisons of routing metrics for wireless mesh networks are made by Draves et al. [134, 135]. In several experiments, it is shown that the ETX metric significantly outperforms the hop count metric. However, a round trip time metric and a packet pair metric perform even worse. Draves et al. [135] propose an own routing metric which is called Multi-Radio Link-Quality Source Routing (MR-LQSR) with a new metric Weighted Cumulative Expected Transmission Time (WCETT). In comparison with other routing metrics in a multi-radio testbed, the proposed WCETT shows the best performance. Another routing protocol to support QoS in WMNs is proposed by Cordeiro et al. [136]. It is an extension of OLSR to provide QoS based on link delay measurements. The protocol is compared to the classical OLSR and another OLSR extension called OLSR-ML for minimum losses.

Besides routing metrics, path reservation schemes alone and in combination with routing protocols are evaluated in [137–142]. Also the IEEE 802.11s standard introduces an optional path reservation method that allows to access the medium with lower contention. Formerly named Mesh Deterministic Access (MDA) [143], it is now called Mesh Controlled Channel Access (MCCA) [117]. It allows to set up a periodic medium reservation. However, in order to reserve the medium properly and not to waste resources, the application layer requirements have to be known in advance. Another medium reservation scheme is proposed by Carlson et al. [137]. The scheme is called Distributed End-to-End Allocation of Time Slots for Real-Time Traffic (DARE). Thereby, time slots for real-time traffic are reserved at all intermediate mesh points as well as mesh points in the interference range. Besides MCCA and DARE, another reservation mechanism is introduced by Kone et al. [138] which is named Quality of Service Routing in Wireless Mesh Networks (QUORUM). In contrast to DARE, it includes a complete, reactive routing protocol and a resource reservation mechanism. Furthermore, an admission control scheme is implemented which is carried out in the reactive route discovery phase. Wireless Mesh Routing (WMR) proposed by Xue et al. [139] is similar to QUORUM as it uses a reactive discovery approach and a reservation and admission control scheme to support QoS for real-time traffic. Marwaha et al. [140] compare all four path reservation schemes and show their advantages and drawbacks for supporting QoS. However, none of the path reservation schemes is favored and no own reservation scheme is proposed. Jiang et al. [142] try to support QoS in a wireless mesh backbone by using a wireless Differentiated Services (DiffServ) architecture. It is shown that wireless DiffServ is a promising approach to support QoS traffic but a lot of open issues remain, like the possibility to operate in a multi-channel environment, and the influences of a routing scheme on the underlying MAC layer. The latest route reservation scheme found is introduced by Gallardo et al. [141]. In the paper, the Medium Access through Reservation (MARE) scheme is compared to MCCA and it is shown that MARE reduces the protocol overhead and provides a collision-free transmission to all MPs who have successfully reserved bandwidth.

In contrast to the QoS optimization by introducing new routing metrics or path reservation schemes, Niculescu et al. [132] optimize the performance of voice traffic in wireless mesh networks using three different mechanisms. Firstly, it is shown how multiple interfaces at the mesh nodes can reduce the interference and thus, increase the voice capacity. Secondly, a label-based routing protocol is introduced as an extension to DSDV. The routing protocol uses an adaptive path switching by considering and probing path delays. Finally, the performance is further optimized using packet aggregation in combination with header compression. All mechanisms are evaluated in an IEEE 802.11b mesh testbed with 15 nodes. Another paper optimizing the voice performance in WMNs by using packet aggregation is presented by Kim et al. [144]. Similar to Niculescu et al. [132], the number of supported G.729 voice calls is shown with and without packet aggregation. Wei et al. [145] present an interference capacity model which can be used to perform voice traffic admission control in WMNs. With the model, the number of admitted voice calls is maximized while still maintaining QoS for existing calls.

Kim et al. [146] look at the coexistence of TCP and voice traffic in a WMN. Firstly, the performance of different TCP variants in WMNs is compared and it is shown that none of the variants works well. Afterwards, different TCP controlling mechanisms like priority queues and window resizing are evaluated. With window resizing, meaning that the advertisement window is modified at the gateways, the highest TCP throughput is achieved while still protecting voice traffic. However, for this solution TCP has to be controlled before entering the WMN and the examined methods only work for fixed wireless capacities.

There are also propositions for QoE provisioning on higher layers. He et al. [147] introduce a middleware-based QoS control in 802.11 wireless networks. The idea is to implement a traffic prioritization inside the mesh nodes based on control theory. To perform this prioritization, a "middleware design with cross-layer framework" is introduced and implemented in a Linux-based testbed. Above the middleware, the applications have the possibility to define requirements for single connections. Before a service is started, the application informs the middleware that certain QoS specifications are needed for the desired flow between two end points. The middleware's task is to choose an adequate service class on a dynamic basis depending on the current performance of the service and the demanded requirements. The current quality is measured and compared to the desired one by a control loop. However, all services that need a certain QoS performance have to be announced first.

In Mogre et al. [148] and Bayer et al. [149] it is claimed that an efficient and adaptive bandwidth management is required to support different traffic classes. These bandwidth management mechanism have to adapt to the needs of the applications. Furthermore, the solution has to be kept as simple and transparent as possible. This leads us to our approach for QoE control in WLAN-based mesh networks described in detail in the following.

3.2 Traffic Observation and Control in WMNs

The general idea of our approach is to perform bandwidth control of best effort traffic flows when real-time flows are disturbed in a way that their QoS requirements can not be met anymore. The approach is implemented on the network layer to be on the one hand platform independent and on the other hand due to simplicity. A change on the WLAN MAC layer would also be possible but it means an update or recreation of all WLAN drivers and implies possible hardware changes as well. In contrast, the network layer is easily exchangeable as it is totally based on software. The communication between the network layer and the network device can be realized via driver independent interfaces.

In order to achieve maximum flexibility and to reduce signaling overhead, the approach is distributed in the WMN. Every wireless mesh point keeps track of the relayed traffic flows. This is done by differentiating between real-time services and non real-time services. Each service is thereby assigned to a priority queue on the network layer whose bandwidth can be limited when needed. QoS parameters of the real-time service are monitored continuously and an exceedance of a QoS threshold is recognized as fast as possible. The mesh point then reacts to meet the QoS requirements.

The monitoring and the appropriate controlling are performed by two tools, the Traffic Observer and the Traffic Controller. We split the functionalities since we want to be able not only to react on problems occurring within one single mesh point but also to react when the problem comes from a neighboring, interfering path. Figure 3.4 shows the general structure of the developed mechanisms. In the following, we describe the Traffic Observer and Traffic Controller in detail.

3.2.1 Traffic Observation

The traffic observation performed by the Traffic Observer is the heart of the presented approach. The Traffic Observer is responsible for monitoring all traffic flows passing through the mesh point. It then has to judge whether the performance of the real-time traffic flows is sufficient to ensure a good perceived quality

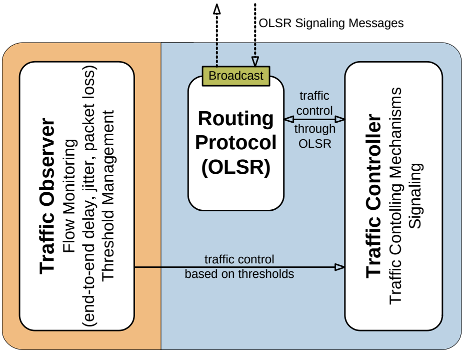


Figure 3.4: *General structure.*

for the end user. If certain QoS thresholds are exceeded, the Traffic Observer informs the Traffic Controller as soon as possible in order to promptly react to these problems. In the following, both the flow monitoring and the threshold management are described.

Flow Monitoring

Due to the fact that the Traffic Observer is placed in each mesh point, a lot of information can be observed and analyzed. The information can be separated into packet-related traffic information and non packet-related traffic information. The latter one includes CPU and memory usage of the mesh point. Although this information is also relevant, we focus on the description of the packet-related information as this is relevant for our QoE-based decision process.

Packet-related traffic information includes all information which can be observed when relaying traffic, in our case IP packets, due to the fact that we operate on the network layer. As the approach has to be very flexible, meaning that it has to work in all different kinds of scenarios and topologies, and since the storage capacity on a mesh point is very limited, a proper choice of the monitored information is required. Furthermore, information from neighboring nodes can not be included. The reason is that we neither want to produce a lot of signaling overhead by transmitting the traffic-related information in separate packets, nor is it possible to add the information to normal data packets. OLSRv2 might solve this problem since it provides a more flexible signaling framework. However, the standard is not yet finished and thus, it is impossible to evaluate it in a WMN testbed. In addition, the complex signaling framework produces more overhead due to periodic signaling.

All information about the currently active services is obtained by the observation of packets passing through the own node. Three different types of information can be gathered for a certain packet stream. First of all, there is the explicit time-independent information readable out of the packets content, for instance source or destination address or protocol type. Next, there is the implicit time-dependent information which is obtainable at the moment of the packet monitoring, e.g., the absolute packet arrival time or relative arrival time after the last packet of the same service. Finally, there is statistical information that is based on a series of packets rather than on a single one. This information provides a long-term analysis of the monitored services, e.g. packet loss over the last n packets or the standard deviation of the packet inter-arrival time. The measurement of the widely-used one-way delay metric is rather difficult. Current approaches work with delay budgets, compare Staehle et al. [14] and thus, only the end points of a WMN can react on delay problems. Thus, the one-way delay is only included in the simulative approach.

The following information can be gathered by the Traffic Observer: For all real-time services using the RTP protocol, source, destination, and next hop IP address of a packet are obtained as explicit information. This can be done either out of the packet header or in case of the next hop address out of the routing table by knowing the destination address. The payload type of the RTP service and its unique Synchronization Source (SSRC) identifier are also explicitly readable from the packet header. The combination of SSRC and next hop address is used to assign a unique ID to each service. Packets with the same SSRC and next hop obtain the same ID and are observed together.

Besides the addresses and next hop information, the following implicit and explicit information is obtained from every packet p_i:

ϕ_i: unique identification number of p_i,
t_i: absolute arrival time of p_i,
$\Delta t_i = t_i - t_{i-1}$: relative arrival time of p_i, and
l_i: total length of p_i in Bytes.

To prevent an oscillation of the approach due to fast changing traffic conditions, the values are averaged over a window size w. Therefore, a set of values is stored and its packets $P = \{p_{last-w+1}, \ldots, p_{last}\}$ are sorted by time of packet arrival:

$\Phi = \{\phi_{last-w+1}, \ldots, \phi_{last}\}$,
$T = \{t_{last-w+1}, \ldots, t_{last}\}$,
$\Delta T = \{\Delta t_{last-w+1}, \ldots, \Delta t_{last}\}$, and
$L = \{l_{last-w+1}, \ldots, l_{last}\}$.

With these definitions, the statistical information can be obtained as follows. The mean inter-packet delay $mean_{IPD}$ is defined as

$$mean_{IPD} = mean[\Delta T] = \frac{\sum\limits_{x \in \Delta T} x}{w}. \tag{3.1}$$

The standard deviation of the inter-packet delay std_{IPD} is defined as

$$std_{IPD} = \sqrt{\frac{w}{w-1} \cdot \left(\frac{\sum\limits_{x \in \Delta T} x^2}{w} - \left(\frac{\sum\limits_{x \in \Delta T} x}{w} \right)^2 \right)}. \tag{3.2}$$

As we consider only constant bit rate real-time traffic, the std_{IPD} is equal to the jitter. Finally, the packet loss pkt_{loss} is defined as

$$pkt_{loss} = 1 - \frac{|\Phi|}{max[\Phi] - min[\Phi] + 1} = 1 - \frac{w}{\phi_{last} - \phi_{last-w+1} + 1}. \tag{3.3}$$

For non real-time traffic, the following information is retrieved. Similar to real-time traffic, the protocol type, source and destination addresses, and ports are gathered from the packet header. The combination of source and destination addresses and ports are used to assign a packet to the correct monitored service. Besides this information, the number of packets per second and bits per second are calculated as follows using the above definitions. The bandwidth bps in bits per second is defined as

$$bps = \frac{\sum\limits_{l \in L} l}{max[T] - min[T]} = \frac{\sum\limits_{l \in L} l}{t_{last} - t_{last-w+1}}. \tag{3.4}$$

The packet rate $pktps$ is defined as

$$pktps = \frac{|L|}{max[T] - min[T]} = \frac{w}{t_{last} - t_{last-w+1}}. \tag{3.5}$$

Threshold Management

The flow monitoring of the Traffic Observer is responsible for collecting all information relayed over the mesh point. However, the monitoring alone is not sufficient to provide QoE guarantees. In addition, a threshold management is needed to judge the current situation and to react on quality decreases. The flow monitoring supports the threshold management by providing all information up to the most recent packet. All gathered information is evaluated according to the requirements of the real-time flows. Thereby, we introduce three different quality levels, namely good, average, and bad quality.

As it is difficult to measure the one-way delay, only the std_{IPD} and pkt_{loss} parameters are evaluated and used for the quality assignment in the testbed. The judgment is done service dependent. Each RTP payload type can be configured with four own values describing the thresholds. It is thus possible to have different thresholds for various real-time services. The four thresholds are the thresholds from good to average std_{IPD}, from average to bad std_{IPD}, from good to average

pkt_{loss}, and from average to bad pkt_{loss}. Although the thresholds can be adapted to the network situation, we set them to fixed values for every type of service in our experiments.

Each mesh point saves the explicit and implicit information for the last w packets internally. This information is requested not just on demand but regularly by the Traffic Observer. Using the above equations, the Traffic Observer calculates the statistical information std_{IPD} and pkt_{loss}. In our case, the statistical information is calculated when $\lfloor \frac{w}{10} \rfloor$ new packets have arrived. If for example w is set to 100, the std_{IPD} and pkt_{loss} values are updated after every 10th packet. Afterwards, these values are compared to the thresholds shown in Table 3.1. In the testbed, we react only when the quality drops from average to bad quality, e.g., when a packet loss of 1.5 % is measured, an alert is sent to the Traffic Controller. To avoid an alert flooding during the reaction process of the Traffic Controller, the minimum interval between two alerts is set to 1 second.

Table 3.1: *QoE thresholds.*

Quality level	MOS	pkt_{loss}	Threshold pkt_{loss}	std_{IPD}	Threshold std_{IPD}
Good	3.8-5.0	$<$ 0.3 %		$<$ 1.7 ms	
Average	3-3.8	0.3-1.7 %	0.1 %	1.7-7.2 ms	1.5 ms
Bad	1-3	$>$1.7 %	1.5 %	$>$7.2 ms	7.0 ms

The four thresholds are chosen smaller than the quality bounds in order to react before the quality decreases to an unacceptable QoE and not afterwards. In all measurements, the std_{IPD} did not exceed 5 ms. Therefore, the parameter is neglected in the following. Instead, we just use the QoS parameter pkt_{loss} to calculate the Mean Opinion Score (MOS). The MOS value gives a numerical indication of the perceived quality, the QoE, of voice data after being transmitted and eventually compressed using different voice codecs. According to Hoßfeld et al. [150, 151] there is a clear exponential relationship between packet loss ratio and MOS for the ITU-T G.711 voice codec [76]. As we are using this codec for the measurements, the MOS can be calculated using the following equation from

Hoßfeld et al.:

$$MOS = 2.861 \cdot e^{-29.816 \cdot pkt_{loss}} + 1.134. \tag{3.6}$$

However, we want to point out that this equation is based on real measurements with some additional performance degradation. This degradation stems from the used microphones and loudspeakers. In an ideal case, which we will see later in the section with the simulative performance control, the thresholds are different. However, using the thresholds of the ideal case would result in MOS values always below 3 in a real WMN testbed before the Traffic Controller reacts to the performance degradation.

3.2.2 Wireless Mesh Traffic Control

The second tool for the QoE bandwidth control mechanism is the Traffic Controller. As soon as the Traffic Observer detects any quality problems of real-time traffic flows, it sends alerts to the Traffic Controller. The Traffic Controller is now in charge of signaling the quality problems to other mesh points in the WMN and to react on the disturbing influences to increase the quality.

Traffic controlling mechanism

Quality degradation can occur for several reasons like packet loss, jitter, and long one-way delays. Similar to the DCWA algorithm on the WLAN MAC layer, packet loss and jitter can be reduced using prioritization with the Differentiated Services Code Point (DSCP) field of the IP header. However, in contrast to the DCWA approach for WLAN infrastructure networks, a prioritization alone would not help in a WMN because of interference problems when relaying traffic over several hops.

Assuming that the quality degradations are originating from disturbances within a WMN, the QoE can be increased by a flexible reduction of the interfering traffic's packet amount. By decreasing the allowed bandwidth of non real-time traffic as soon as an alert is received from the Traffic Observer, and increasing the bandwidth of non real-time traffic flows when no alert is received for a period of time, the QoE of voice traffic can be increased without a non real-time flow starvation.

This is realized by using different priority queues like shown in Figure 3.5. The prioritization is again performed using the DSCP field from the IP header and the traffic flows are assigned to four different queues. Three priority queues are executed on a Packet First In First Out (PFIFO) basis. Packet here means that the number of packets in the queue is limited. The scheduler handles the different queues using a strict priority. In addition to the PFIFO queues, another queue is used for best effort traffic. This queue is used for bandwidth control. As soon as a quality degradation is signalized by the Traffic Observer, the Traffic Controller limits the bandwidth of the Token Bucket Filter (TBF) queue. TBF is thereby a pure traffic shaper which allows to throttle the bandwidth to a configured rate. Within the WMN testbed, the queuing is realized by the Linux traffic controller (tc) tool.

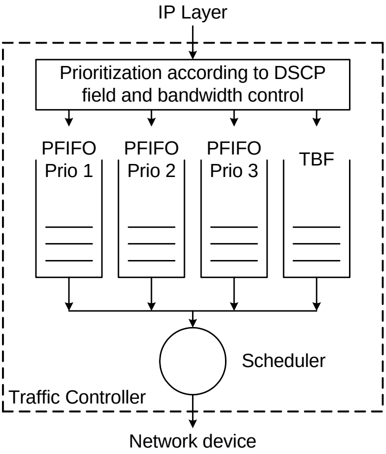
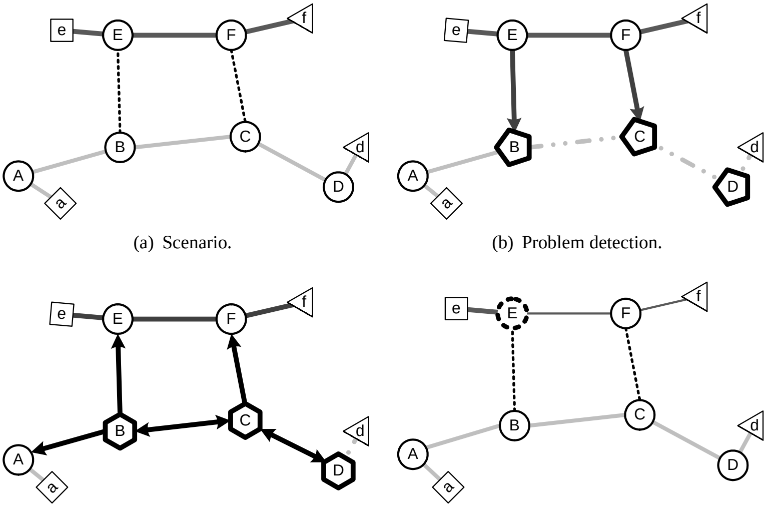


Figure 3.5: *Traffic Controller prioritization and bandwidth control.*

Steps of Controlling

The different steps of the Traffic Controller are explained with an example network shown in Figure 3.6. The WMN consists of six mesh points, marked as a circle. A constant bit rate real-time connection is established between end user *a* and end user *d* via the mesh points *A-B-C-D*. In addition, an interfering best effort traffic flow is set up between *e* and *f* via *E-F*, see Figure 3.6(a).



(a) Scenario.

(b) Problem detection.

(c) Neighbor broadcast.

(d) Problem location and reaction.

Figure 3.6: *Steps of controlling.*

Although both traffic flows do not use the same path, interference is produced by mesh points *E* and *F* symbolized with the dashed lines between *E* and *B* and *F* and *C*. This leads to a quality degradation of the real-time traffic flow as illustrated with the dash-dotted line in Figure 3.6(b). The quality problem is recognized by the Traffic Observers in mesh points *B*, *C*, and *D*. The Traffic Observers then send an alert to their Traffic Controllers.

Before broadcasting the problems using OLSR broadcast messages, the Traffic Controllers try to find out if the quality decrease is caused at their own queues. This might happen when the queues are overloaded and thus, the mesh points check all non real-time applications in the own mesh point first. If this is the case, the bandwidth of the best effort traffic flows is limited by the TBF queue. In our example however, the performance problems are caused by best effort traffic on a neighboring path. The next step of the Traffic Controller is now to send signaling messages to all one-hop neighbors, shown in Figure 3.6(c). The reason is that the one-hop neighbors of the disturbed mesh points might have caused the quality problems. The one-hop neighbor mesh points then check their own queues similar to the disturbed mesh points. In the example, mesh point *E* and mesh point *F* identify the quality problems within their own queues and limit the bandwidth of the best effort flow. Figure 3.6(d) shows the situation after the reaction of the mechanism. The thinner line between mesh point *E* and *F* symbolizes the bandwidth limitation. If the reduced bandwidth is not sufficient to ensure a high QoE for the real-time service, the bandwidth is again decreased after an inter-adaptation time of one second.

If, however, no threshold is exceeded for a period of time, the Traffic Controller stepwise increases the bandwidth of the non real-time flows. The performance of this Additive Increase Multiplicative Decrease (AIMD) mechanism is only evaluated in the simulative approach in the next section. Within the testbed, the bandwidth limitation is set to a fixed value.

In the following section, we first describe the testbed setup and then evaluate the performance of the bandwidth control approach in a scenario with interfering best effort traffic on the same path and on a neighboring path.

3.3 Performance Measurements

For the performance measurement, a WLAN mesh network is set up at T-Systems in Darmstadt. An example of the testbed is shown in Figure 3.7. The mesh points are embedded AMD Geode SC1100 systems with 266 MHz CPUs and 64 MB

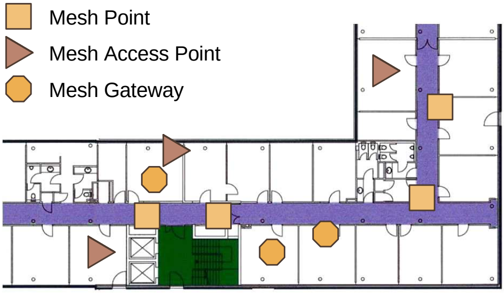


Figure 3.7: *The building layout of the WLAN-based mesh testbed.*

RAM. The mesh gateways are normal PCs with a 3 GHz Intel Pentium 4 processor and 1 GB of RAM. Although the mesh points are equipped with two Atheros WLAN cards, only one card is used for the wireless mesh backbone. The interconnection between the mesh points is set up on the 5 GHz frequency band using the IEEE 802.11a standard. As already mentioned, OLSR routing between the mesh points is realized by the OLSRd implementation of Andreas Tønnesen [131].

The Traffic Observer is implemented as a kernel module. It is runnable independently of OLSRd and can be compiled and used on any Linux machine with the correct kernel version. The Traffic Controller is set up as a plugin to the OLSRd plugin interface. It includes a signaling unit making use of the OLSRd broadcast messages and thus allows communication between different Traffic Controllers. Located on one single node, Traffic Observer and Traffic Controller are contacting each other via Linux netlink sockets.

3.3.1 In-Band Traffic Disturbance

The first scenario to evaluate the performance of the QoE approach is shown in Figure 3.8. In this scenario, the interfering best effort traffic is transmitted using the same wireless link between mesh point *A* and mesh point *B*. The scenario is called in-band scenario and the disturbance should be directly recognized by mesh point *B*.

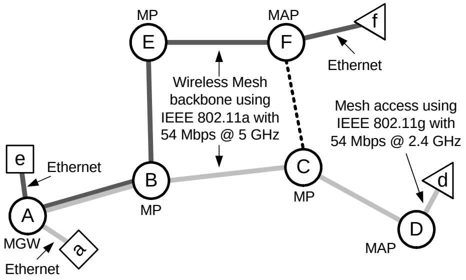
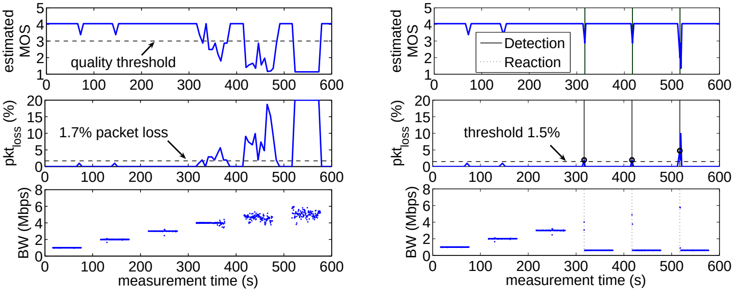

MP
E
MAP
F
f
Ethernet
Wireless Mesh backbone using IEEE 802.11a with 54 Mbps @ 5 GHz
Mesh access using IEEE 802.11g with 54 Mbps @ 2.4 GHz
e
Ethernet
B
MP
C
MP
d
A
MGW
a
Ethernet
D
MAP

Figure 3.8: *In-band interfering traffic.*

In total, two traffic flows are set up in the scenario, one real-time flow between *a* and *d* and one best effort traffic flow between *e* and *f*. The real-time flow is a voice connection similar to the ITU-T G.711 voice codec with an inter-arrival time of 20 ms and a packet size of 200 Bytes. The bandwidth of the interfering best effort flow is stepwise increased from 1 Mbps to 6 Mbps. As soon as quality problems of the voice traffic flow are detected by the Traffic Observer, the Traffic Controller limits the bandwidth of the best effort flow to 1 Mbps.

To show the effect of the QoE approach, the tests are conducted twice, once with activated bandwidth control mechanism and once without controlling the bandwidth. The results are shown in Figure 3.9. Figure 3.9(a) depicts the results with deactivated Traffic Controller and Figure 3.9(b) illustrates the results when the Traffic Controller is activated. The x-axis shows the measurement time in seconds and the three y-axes present from bottom to top the bandwidth of the best effort traffic disturbance at mesh point *F*, the packet loss of the real-time traffic, and the resulting MOS at mesh point *D*, respectively. The std_{IPD} has also been computed but the measurements have shown that it was always below



(a) In-band scenario without Traffic Controller.

(b) In-band scenario with Traffic Controller.

Figure 3.9: *Comparison of in-band scenario with and without Traffic Controller.*

5 ms, even for the highest interfering bandwidth and thus, the std_{IPD} threshold from average to bad quality is not exceeded.

What can be observed from Figure 3.9 is that the packet loss has a large influence on the perceived quality of the end user. Looking at Figure 3.9(a), we can see that as soon as the bandwidth of the best effort flow is increased to 4 Mbps, the packet loss quality threshold of 1.7 % is exceeded, resulting in a mean opinion score below 3 which is not acceptable for the end user. When further increasing the bandwidth to 6 Mbps, the packet loss increases to more than 20 percent which leads to a MOS value of 1.

Looking now at Figure 3.9(b) with activated Traffic Controller, we can observe that the MOS is kept on a high level above 4 and only shortly drops when the Traffic Controller reacts on the quality problems. As soon as the threshold of 1.5 % pkt_{loss} is exceeded, marked with a circle, the Traffic Observer of mesh point *A* sends an alert to the Traffic Controller who then limits the bandwidth of the best effort flow to 1 Mbps. The vertical lines in the curves show the time when the problem is detected and the time of the reaction of the Traffic Controller.

Although the time difference between detection and reaction of the mechanism is not visible, the quality decrease between the detection and the reaction seems to be very large. However, we have to mention that this is a worst case scenario as the best effort flow is a UDP traffic stream meaning that the bandwidth is immediately increased to 6 Mbps. If a TCP best effort stream is used instead, the bandwidth would slowly increase, the quality decrease would be detected sooner, and the MOS would not drop to 1.5.

The impact of the bandwidth control mechanism is most obvious when the bandwidth of the best effort flow is equal or larger than 5 Mbps. Such an interfering bandwidth leads to 6 percent packet loss of the real-time connection and to a MOS value always below 3. With activated mechanism, the MOS immediately increase to 4 again.

3.3.2 Out-Band Traffic Disturbance

After having seen the performance of the bandwidth control approach in the in-band scenario, we now evaluate if the mechanism also works when the interfering best effort traffic is transmitted on a neighboring path. The scenario is shown in Figure 3.10. It looks similar to the first scenario except that the best effort traffic flow is now just using the wireless mesh link *E-F*. The path of the voice traffic flow remains the same. We call this scenario the out-band scenario. If the best effort flow now interferes the real-time flow in such a way that the packet loss drops below the threshold, the reaction should be as shown in the example scenario in Section 3.2.2.

In contrast to the previous scenario, the limitation of the Traffic Controller is set to 5 Mbps instead of 1 Mbps. The reason is that the degradation results from interferences on the air interface and not from overloaded queues. These interferences have a lower influence on the voice traffic flow. This also means that the bandwidth of the interfering best effort traffic can be further increased. We start with an interfering bandwidth of 5 Mbps and increase the bandwidth in steps of 5 Mbps to a maximal interfering bandwidth of 25 Mbps.

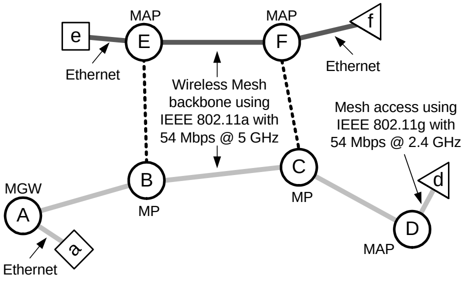
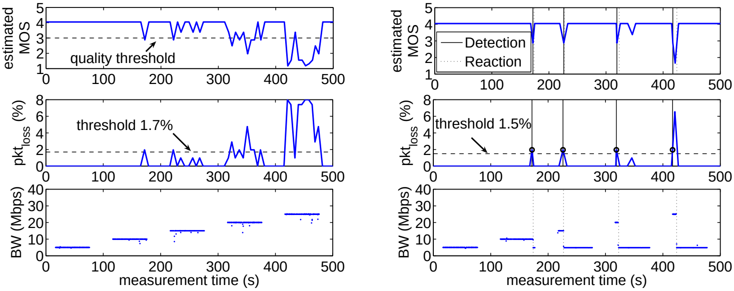


Figure 3.10: *Out-band interfering traffic.*

The tests are again performed with and without activated control algorithm. Figure 3.11 shows the results of both measurements. Similar to the in-band scenario, the std_{IPD} has no influence on the quality perceived by the voice user, and is thus not plotted. The same three subplots are generated and Figure 3.11(a) shows the results without Traffic Controller. Already at an interfering bandwidth of 10 Mbps, the quality drops below the quality threshold meaning that a packet loss of over 1.7 % is observed over a short period of time. If the interfering bandwidth is increased to 20 Mbps or more, the packet loss of the voice traffic flow is always above 1.7 % and finally, a best effort bandwidth of 25 Mbps results in a quality which is unacceptable for the voice user.

If the bandwidth control mechanism is activated, the results look different, shown in Figure 3.11(b). As soon as the 1.5 % pkt_{loss} threshold of the Traffic Observer is exceeded, the Traffic Observers in mesh point *B*, *C*, and *D* send an alert to the Traffic Controllers. Due to the fact that no best effort traffic flows are relayed by these mesh points, an OLSR broadcast message is then transmitted to mesh point *A*, *E*, and *F*. Mesh point *E* recognizes that it causes the performance problems and reduces the bandwidth of the best effort flow to 5 Mbps. Immediately after that, the MOS of the real-time traffic increases again to an acceptable value.



(a) Out-band scenario without Traffic Controller.

(b) Improvements by the Traffic Controller in the out-band scenario.

Figure 3.11: *Comparison of out-band scenario with and without Traffic Controller.*

In this scenario, the time between the recognition of the quality problems and the bandwidth limiting lasts longer, because the quality problems have to be signaled to the neighbor mesh points. In general, the reaction time depends on the parameter w and the time for signaling the problems. When setting w to 100, quality problems are identified after w packets have been transmitted. With a packet inter-arrival time of 20 ms, this means that it takes 2 seconds. If this reaction time of about 2 seconds is too long for the real-time traffic, w can be halved or even quartered to ensure a sufficient QoE level.

For an interfering bandwidth of 20 Mbps or more, compare Figure 3.11(b), the reaction time increases sometimes to up to 7 seconds. The reason here is not the setting of w, but the hardware of the mesh points. We have seen that the CPU usage and the memory of the mesh points are operating on their limits. However, newly available WMN hardware is equipped with a larger CPU and with up to 1 GByte of RAM. Thus, the long reaction delays would disappear. Furthermore, those high bandwidths are not expected to occur very often in WMNs.

Besides the reaction time, it is also interesting to take a look at the overhead produced by signaling the performance problems. This overhead depends on the WMN scenario and the number of neighbors of a mesh point. In our measurement scenario, a mesh point receives between 400 Bytes, about 3 to 4 packets, and 2000 Bytes, 15 to 20 packets, of OLSRd messages per second. The Traffic Observer itself sends a maximum of one alert per second in order to avoid alert flooding and one alert fits into a single OLSRd signaling packet. This results in a complete signaling overhead of less than 16 kbps which is negligible even in highly loaded networks.

In the following section, we optimize the bandwidth control algorithm so that the mechanism does not only limit the bandwidth of the interferer but also increases the bandwidth when no quality problems are observed over a period of time. Additionally, a combination of the bandwidth control mechanism, the DCWA, and extended frame bursting is simulated. The optimization is however performed by means of simulation, because further measurement runs in the WMN testbed would be too time consuming.

3.4 Simulative Performance Evaluation

For the validation of the measurement results, we implemented the Traffic Observer and the Traffic Controller in the OPNET Modeler simulation environment. The only difference between the simulation approach and the measurements is the calculation of the mean opinion score. In the testbed, we used the equation by Hoßfeld et al. [151] which is based on measurements of the ITU-T G.711 voice codec. As no hardware influences have to be considered in the simulation approach, we decided to use the E-Model from the ITU-T G.107 [152] recommendation to calculate the MOS.

The E-Model is expressed by the R-Factor which has a range between 0 and 100. However, values below 50 are generally unacceptable and typical voice connections do not get above 94 giving a typical range between 50 and 94. The basic model is

$$R\text{-}Factor = R_0 - I_s - I_d - I_e + A. \tag{3.7}$$

The R-Factor is calculated by estimating the signal to noise ratio of a connection (R_0) and subtracting the network impairments I_s, I_d, and I_e. I_s represents signal impairments occurring simultaneously with the voice signal transmission, including loudness, quantization distortion, and non-optimum sidetone level. I_d represents the mouth to ear delay and I_e is the equipment impairment factor associated with losses. According to Cole and Rosenbluth [153] and the ITU-T G.133 [154] recommendation, I_e can be calculated for an ITU-T G.711 voice conversation as

$$I_e = 30 \cdot log(1 + 15 \cdot pkt_{loss}). \tag{3.8}$$

Afterwards, the advantage factor A is added. The advantage factor A is used to represent the convenience to the user of being able to make a voice call, i.e. a cell phone is convenient to use therefore people are more forgiving on quality.

The mapping between MOS and R-Factor can be done using the following expression

$$MOS = 1 + 0.035 \cdot R + 7 \cdot 10^{-6} \cdot R \cdot (R - 60) \cdot (100 - R). \tag{3.9}$$

It has to be mentioned that with this mapping, a maximum MOS value of 4.5 can be achieved. This also shifts the quality thresholds. An illustration of the mapping is shown in Figure 3.12 and the R-Factor quality ratings are shown in Table 3.2.

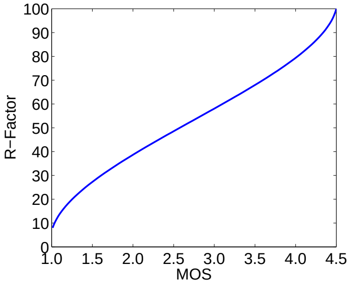


Figure 3.12: *R-Factor vs. MOS score.*

Table 3.2: R-Factor quality ratings.

User opinion	R-Factor
Maximum for G.711	93
Very satisfied	90-100
Satisfied	80-90
Some users satisfied	70-80
Many users dissatisfied	60-70
Most users dissatisfied	50-60
Not recommended	0-50

Using these equation, the MOS can be calculated for different packet losses and delays. The resulting values for the ITU-T G.711 voice codec are shown in Figure 3.13. The exponential relation between packet loss and MOS is similar to Hoßfeld et al. [151]. However, the quality thresholds are different. A mean opinion score of 3.6 is still achieved with a packet loss of 7 % and a delay of up to 177 ms. Thus, the thresholds for the bandwidth control mechanism are adapted later in this section, but first we compare the testbed results of the in- and out-band scenario with simulation results using the same thresholds.

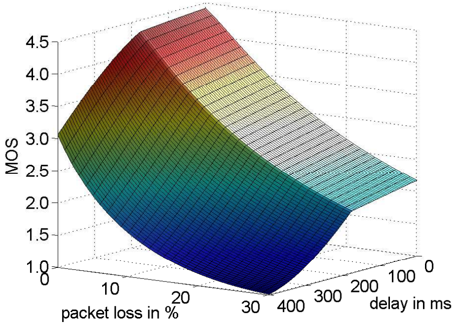
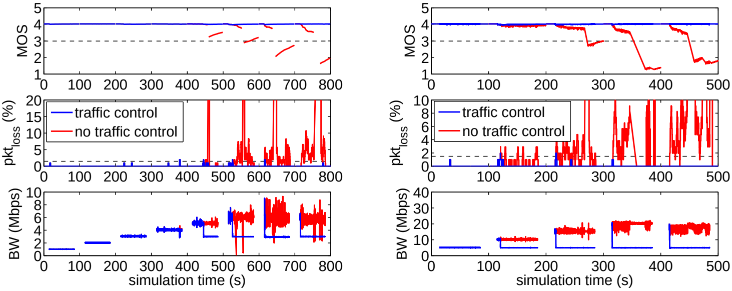


Figure 3.13: *MOS vs. packet loss rate and delay for ITU-T G.711.*

3.4.1 Validation of the Testbed Approach

In order to validate the testbed results, the simulation is configured using the same scenario setup. Only the bandwidth limitation is increased from 1 Mbps to 3 Mbps because of larger WLAN MAC layer queues. The results of the in-band scenario are shown in Figure 3.14(a). In contrast to the previous section, we placed the results with and without the Traffic Controller in one figure.

As soon as the interfering bandwidth is increased to 5 Mbps, the MOS drops below 3 and the Traffic Controller decreases the allowed bandwidth of the best effort traffic flow down to 3 Mbps. Comparing this result with the measurements, the Traffic Controller reacts later, not at 4 Mbps but at 5 Mbps. The reason for



(a) Simulated in-band scenario.

(b) Simulated out-band scenario.

Figure 3.14: *Simulation results of in-band and out-band scenario.*

this might be larger queues within the OPNET simulation. Without controlling the interfering traffic, the packet loss due to queue blocking increases to over 20 % similar to the testbed measurements.

Within the simulation, we are able to measure the one-way delay as well. However, the delay was always below 45 ms in both scenarios, with and without activated Traffic Controller, which does not necessitate a reaction of the Traffic Controller. Furthermore, the std_{IPD} was always below 7 ms so that the Traffic Controller is just activated due to a high pkt_{loss} and not due to a high std_{IPD} or delay.

Similar to the in-band traffic scenario, we also simulated the out-band scenario. The results are shown in Figure 3.14(b). Comparing them to the measurement results, the Traffic Controller similarly regulates the interfering traffic after it increases to 10 Mbps. However, the packet loss responsible for the traffic control, increases more drastically compared to the measurements. The peak packet loss within the simulation is 70 %.

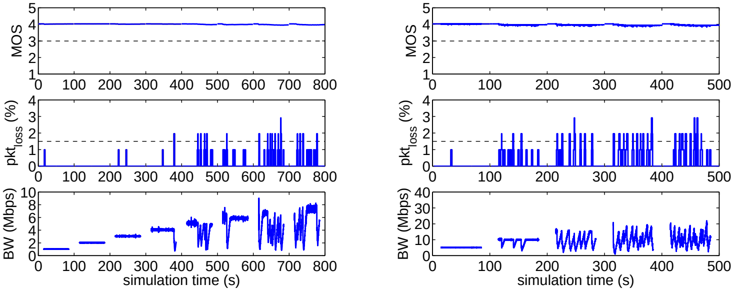
Looking at the MOS figure, we can see that the QoE does not drop below the threshold of 3 for an interfering bandwidth of 10 Mbps. The first significant quality degradation is shown for an interfering bandwidth of 20 Mbps. With this interfering bandwidth on a neighboring path, the packet loss is almost always above the quality threshold of 1.7 %.

In this scenario, the variance of the one-way delay is with up to 60 ms higher compared to the in-band scenario. In the in-band scenario, the real-time traffic is disturbed due to extensive queuing, whereby it is disturbed on the air interface in the out-band scenario. Packet transmissions have to be deferred due to a busy air interface and a lot of collisions cause packet retransmissions. However, the delay is still low enough and also the std_{IPD} threshold is not exceeded at any time.

3.4.2 Throughput Optimization

The simulation results of the in- and out-band scenario are very similar to the measurements in the testbed. The only downside of the approach is that in case of a quality problem, the bandwidth of the best effort flow is limited to a fixed amount. This leads to a waste of resources and thus we decided to implement an adaptive controlling mechanism using an Additive Increase Multiplicative Decrease (AIMD) approach. As soon as one of the thresholds is exceeded, the bandwidth of the interferer is halved. This decrease is done at most every second. Whenever no threshold is exceeded for one second, the bandwidth limitation is stepwise increased by 100 kbps every second until the maximum allowed bandwidth is reached. These values where obtained from extensive parameter studies.

The results using the adaptive mechanism for the in-band and the out-band scenario are shown in Figure 3.15. Both bandwidth figures show the saw tooth behavior similar to TCP. The average bandwidth of the interfering traffic is increased up to 100 % compared to the static scenario with only a small influence on the MOS of the real-time traffic. However, the MOS stays always above 3.8. These results illustrate that a good QoE level can be achieved without wasting too much resources.



(a) In-band scenario.

(b) Out-band scenario.

Figure 3.15: *In-band and out-band scenario with throughput optimization.*

3.4.3 Performance Optimization of the Traffic Controller

Up to now, we have evaluated the bandwidth control algorithm only in two WMN scenarios. With the simulative approach we are however able to simulate a larger scenario and can also take a look at the performance of the control algorithm when used in an IEEE 802.11e network. Therefore, we first have to adapt the QoE thresholds. The quality thresholds of the E-Model are shown in Table 3.3. Due to the fact that we are able to determine the MOS with not only one value but in combination of the packet loss and the one-way delay, we set the thresholds pkt_{loss} to 3 %, one-way delay 250 ms, and std_{IPD} to 30 ms. Thus, we already react when the quality drops from good to average, according to one quality threshold.

Table 3.3: QoE thresholds.

Quality level	MOS	pkt_{loss}	One-way delay	std_{IPD}
Good	4.0-4.5	$\leq$3 %	$\leq$250 ms	$\leq$30 ms
Average	3.6-4.0	$\leq$7 %	$\leq$320 ms	$\leq$50 ms
Bad	1.0-3.6	$>$7 %	$>$320 ms	$>$50 ms

The first scenario which we want to evaluate is shown in Figure 3.16. The scenario consists of one gateway node and nine mesh points which are placed in a triangle pattern. A real-time station and a non real-time station are connected to each of the lowermost mesh points, the MAPs, using an Ethernet connection. The stations communicate with nodes behind the mesh gateway router. The mesh points use the IEEE 802.11a standard at 5 GHz and have a communication and interference range as indicated in Figure 3.16 by dotted lines, which means that the mesh point in the middle can communicate with the six surrounding mesh points. All mesh points operate on one single channel and use RTS/CTS for packets larger than 256 Bytes.

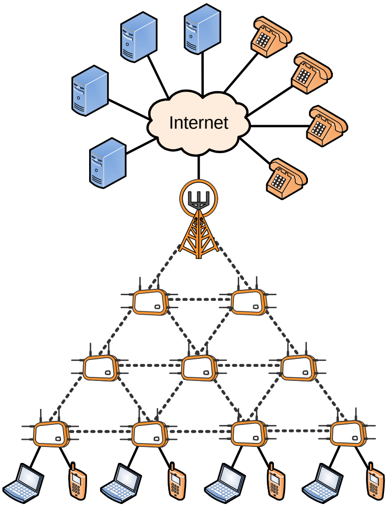


Figure 3.16: *Simulation scenario.*

The non real-time nodes are configured with FTP traffic in both downlink and uplink direction and the traffic amount per node is increased from 0.5 Mbps to 3.0 Mbps. The real-time traffic is again produced using the ITU-T G.711 voice codec. The AIMD bandwidth control mechanism is configured with a decrease factor of 2, an inter-decrease time of one second, an increase volume of 100 kbps, and an inter-increase time of one second.

Traffic Control in a Scenario Without Service Differentiation

For the first evaluation of the flexible bandwidth control mechanism, the mesh network from Figure 3.16 is set up without any service differentiation. This means that the real-time traffic is not prioritized on the WLAN MAC layer. In Figure 3.17 the influence of the flexible bandwidth control mechanism ("flex mode") on the MOS and the mean throughput of the non real-time traffic is shown. The values are averaged over 10 simulation runs which each lasting 160 seconds. The first 60 seconds are considered as the transient phase and are not included in the results.

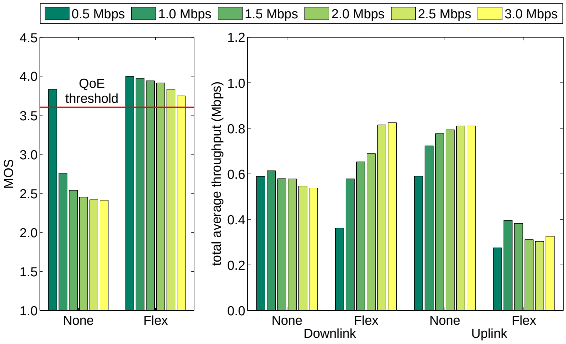


Figure 3.17: *Comparison of no reaction and flex mode.*

When no bandwidth control mechanism is used, a sufficient MOS value is only achieved with 0.5 Mbps FTP traffic per non real-time station. As soon as the traffic is increased to 1 Mbps or more, the MOS drops below 3.0 or even below 2.5 which is not acceptable for any user. With activated bandwidth control mechanism, the MOS is always above 3.7, no matter how much bandwidth is requested by the non real-time users. This shows that the presented mechanism can guarantee a high QoE for the end user even without service differentiation on the WLAN MAC layer. Furthermore, it can be observed that the total average throughput increases with increasing requested bandwidth of the non real-time users. The reason is that the wireless links are not completely utilized. A further increase beyond 3.0 Mbps would result in more collisions on the air interface and thus in lower average throughput.

The downside of the flexible bandwidth mechanism is the total average throughput of the non real-time stations shown on the right of Figure 3.17. The throughput is about 60 % lower compared to not using any bandwidth control mechanism.

The main reason for the bandwidth limitation in the scenario is the packet loss measured at the Traffic Observers. Without any control mechanism, a packet loss of up to 20 % can be observed for real-time traffic. The one-way delay increases up to 250 ms and is reduced to about 60 ms with the control mechanism. Similar to the one-way delay, the std_{IPD} is reduced from 50 ms to 20 ms with the control mechanism.

Summarizing the results of the scenario without MAC layer service differentiation, it was observed that the Traffic Controller successfully protects real-time traffic flows by controlling the bandwidth of non real-time flows. In the next subsection, we evaluate the need for a control mechanism in a WLAN mesh network with service differentiation. Service differentiation means that we use the EDCA channel access mechanism with different priorities instead of the standard DCF channel access, cf. Section 2.1.3. We evaluate if the service differentiation on the MAC layer alone can already guarantee a high QoE for real-time flows.

Traffic Control in a Scenario with Service Differentiation

The WLAN MAC layer service differentiation should ensure a certain quality for real-time traffic flows by assigning different channel access parameters to real-time and non real-time traffic flows. Figure 3.18 shows the comparison of no bandwidth control mechanism and the proposed flexible mechanism in a scenario with service differentiation.

Compared to the previous results, the MOS does not decrease down to 2.5 when no control mechanism is used. However, a MOS value of 3.1 also means that many real-time users are dissatisfied with the quality. Similar to the previous scenario, the quality decrease is mainly caused by packet loss. The flexible bandwidth control mechanism again ensures a high MOS value. Comparing the average throughput of the non real-time user with the previous scenario reveals that the average throughput is lower which is caused by service differentiation on the WLAN MAC layer. However, the throughput distribution between uplink and downlink flows is fairer no matter whether the control mechanism is used or not.

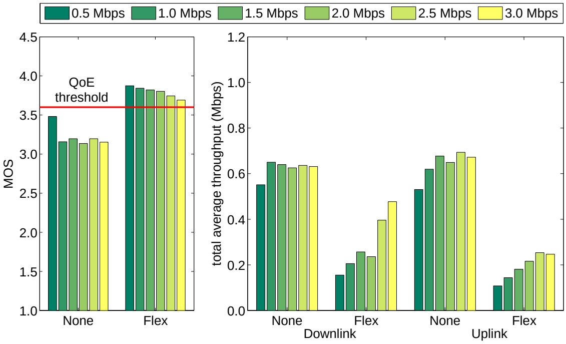


Figure 3.18: *Comparison of no reaction and flex mode using EDCA.*

Traffic Control in Combination with DCWA and Frame Bursting

Although the average throughput of the non real-time traffic was increased compared to a fixed bandwidth limitation, it is still low. In order to further improve the bandwidth, we extended the WLAN MAC layer with the DCWA and an extended frame bursting from the previous chapter. We start with the same scenario in order to compare the results. The TXOP Limit is set to 1280 μs for the non real-time flows. The results are shown in Figure 3.19.

Taking first a look on the MOS curves for different interfering bandwidths and comparing them to the results of Figure 3.18, we can see that the quality of the real-time stations is even higher. For an interfering bandwidth of 3.0 Mbps, the MOS of the real-time users was 3.7 without using frame bursting and the DCWA algorithm. When using them, the quality increases to 3.85. However, not only the MOS increases, but also the throughput of the non real-time stations. The right figure, illustrating the total average throughput for the different FTP downlink and uplink bandwidths, shows that the throughput is increased between 52 % and 75 % compared to a setting without frame bursting and DCWA. Thus, the scarce wireless resources can be efficiently utilized using an adaptive bandwidth limitation on the network layer in combination with the DCWA and an extended frame bursting on the WLAN MAC layer.

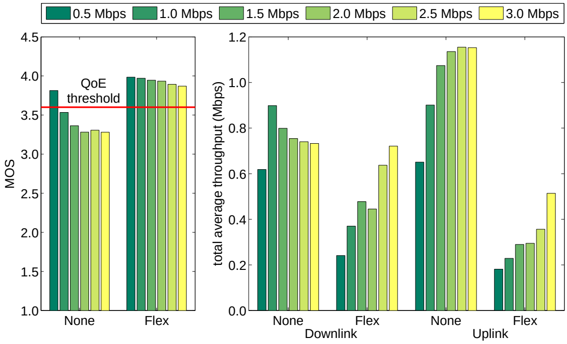


Figure 3.19: *Optimization with DCWA and extended frame bursting of 1280 μs.*

3.5 Lessons Learned

Future wireless networks are expected to be organized in a three-tier mesh architecture, with a wireless backbone to forward the traffic between the access networks and the Internet. In this chapter, we proposed a monitoring and controlling mechanism to guarantee a certain QoE level in a wireless mesh network.

The approach is based on two main entities, a Traffic Observer and a Traffic Controller. The Traffic Observer continuously measures the network situation. Whenever a problem is detected in the wireless mesh network, for example a high rate best effort flow blocking a real-time application, the Traffic Controller forces this low priority flow to reduce the bandwidth.

Two different scenarios have been investigated in a WMN testbed, one with interfering best effort traffic on the same path and one scenario with interfering traffic within the coverage area of the mesh points. The results show that without the Traffic Controller, the subjective quality, expressed by the mean opinion score, decreases drastically when only a small interfering bandwidth is set up. However, when the Traffic Controller is activated, the MOS only drops for one to three seconds below four. Comparing the two scenarios, it can be said that the real-time application is more influenced by a best effort flow on the same path than on a crossing path. This is due to queuing effects on the MAC layer.

In the next part, we implemented the approach in the OPNET Modeler simulation environment and compared the results of the same scenarios with the measurement results. The results differed only slightly. Only the amount of non real-time traffic until the Traffic Controller reacts, increases. However, the limitation to a fixed bandwidth leads to a waste of resources. Therefore, we have extended the controlling mechanism using an AIMD approach. As soon as one of the thresholds is exceeded, the bandwidth of the non real-time traffic flows is halved. Whenever no quality problems are measured by the Traffic Controller for a period of time, the bandwidth limitation is stepwise increased. The results show that the bandwidth can be increased up to 100 % compared to the static scenario with only a small influence on the MOS of the real-time traffic.

Afterwards, we investigated a larger WMN scenario with nine mesh points, one mesh gateway, four real-time, and four non real-time users. The performance of the adaptive bandwidth controlling algorithm was shown in a scenario with and without service differentiation. The results in the scenario without WLAN service differentiation illustrate that the mechanism successfully protects real-time traffic flows from non real-time flows. Although it was expected that the WLAN service differentiation alone would guarantee a high QoE for real-time services, the results reveal that the MOS increases but is still not acceptable for most users. However, also in the scenario with WLAN service differentiation, the presented approach was able to provide QoE guarantees with a MOS value above 3.6. Even though the bandwidth of the non real-time traffic is limited, the flows are not prone to starvation.

In order to further increase the bandwidth of the non real-time traffic, the DCWA algorithm as well as an extended frame bursting from the previous chapter was implemented. Using these mechanisms, an increase of the total throughput between 52 % and 75 % is achieved without interfering the real-time traffic flows. The quality of the real-time traffic even increases compared to the scenario without DCWA and frame bursting.

Lessons learned from this chapter are that it is not possible to support real-time traffic in a wireless mesh network without any controlling mechanism for non real-time traffic. Even when using the IEEE 802.11e standard with service differentiation, no QoE guarantees can be given to the end users. The proposed bandwidth control mechanism successfully keeps the MOS values on a high level above 3.6. In combination with the DCWA algorithm and an extended frame bursting from the last chapter, the total average throughput of non real-time traffic can be successfully increased without interfering real-time traffic.

4 Planning and Optimization of Wireless Mesh Networks

"A common mistake people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools."
Douglas Adams (1952-2001): The Hitchhiker's Guide to the Galaxy (1979).

In the previous chapter, we have optimized the QoE of real-time traffic in wireless mesh networks by dynamically limiting the bandwidth of interfering best effort flows. In this chapter, we focus on the planning and optimization of WMNs. As the performance of the bandwidth control algorithm also depends on the initial setup of the WMN, we try to optimize the channel allocation as well as the routing.

Compared to traditional wireless networks, WMN planning and optimization is more challenging since several wireless hops are needed to connect a node to the Internet. As already mentioned in the previous chapter, interference aspects have to be taken into account and also the estimation of a link capacity is a challenge. A suitable tool for the planning and optimization of such WMNs are Genetic Algorithms (GAs) due to their ability to solve complex problems in relatively small computation time. GAs belong to the group of global optimization algorithms. In contrast to exact optimization techniques for this problem, genetic algorithms scale well so that we are able to optimize even large WMNs. Thereby, we want to increase the throughput of the complete WMN while sharing the resources fairly among the nodes. This is achieved by applying a max-min fair share algorithm presented in [155] and by tuning the genetic parameters. A solution is max-min fair if no rate can be increased without decreasing another rate to a smaller value [156]. A max-min fair share algorithm is used instead of proportional fairness because the main goal is not to optimize the maximum overall throughput on the cost of fairness but to ensure a fair resource distribution between the users.

The rest of this chapter is structured as follows. We first give an introduction to wireless network planning, comparing traditional cellular network planning with wireless mesh network planning. This is followed by a short overview of global optimization techniques which are applied in the related work section for optimizing WMNs. Afterwards, we describe our genetic algorithms for routing and channel assignment in detail and evaluate the performance of different genetic operators. Finally, we optimize the WMN planning by using a local optimization approach.

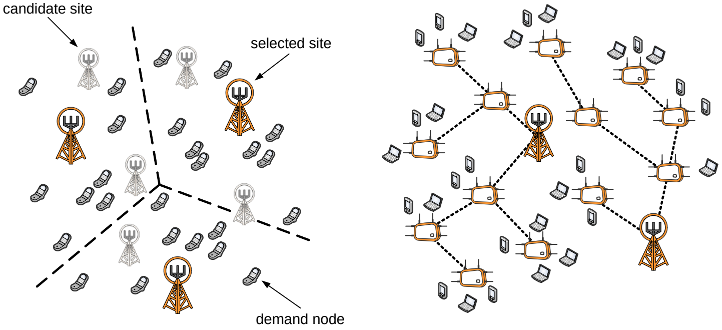
4.1 Network Planning and Optimization Issues

Network planning and optimization can be done using several techniques. On the one hand, signal quality measurements can be performed which is very time-consuming and necessitates the access to all areas in which the network should be supported. On the other hand, a demand node concept can be used. This mechanism is often applied to cellular network planning. Furthermore, network planning can be done using an optimization mechanism. Meanwhile, a huge number of optimization techniques have been proposed and we decided to use genetic optimization due to its simplicity and the ability to plan even large networks.

4.1.1 Wireless Network Planning

The planning of wireless mesh networks can be applied to a variety of wireless networks, like WiMAX, WLAN, and sensor networks. Although the network technology changes, the planning challenges remain similar. In contrast to traditional cellular network planning, the planning and optimization of WMNs is much more complex. A widely used concept for cellular network planning is the demand node concept introduced by Tutschku [157] and illustrated in Figure 4.1(a).

The algorithm first looks for the demands of cellular services. Therefore, different demographic areas are taken into account. For example, more phone calls occur in urban areas than in rural areas. According to these demographic regions, a different number of demand nodes are set up like shown in Figure 4.1(a). In addition to the demand nodes, candidate sites for base stations are inserted into the optimization algorithm. As each base station is able to support a fixed amount of users in cellular systems of the second generation, candidate sites are selected for base station placement in such a way that all demand nodes can be served with a certain probability.



(a) Cellular network planning.

(b) Wireless mesh network planning.

Figure 4.1: *Comparison of traditional cellular network planning and wireless mesh network planning.*

In contrast, the planning of WMNs is much more complex. Not only the covered area or number of end users has to be considered, but also the capacity and the interference of the relaying links. The capacity of a link does not only depend on the distance between two mesh points, but also on the interference which in turn depends on the used channels. Looking again at Figure 4.1(a), we can see that the channel assignment has to be performed in such a way that neighboring base stations do not use the same channel. In WMNs, such as shown in Figure 4.1(b), each mesh point can be equipped with multiple interfaces which can be assigned one channel each.

In addition to the more complex channel assignment in WMNs compared to traditional cellular networks, also the routing has to be considered. In the previous chapter, we have used the minimum hop count metric from the OLSR protocol. This routing metric might however lead to a solution with a rather poor performance in a fixed wireless mesh network where each mesh point is equipped with multiple interfaces. As we consider such a network in this chapter, we use a fixed routing and optimize it in such a way that the minimal flow throughput is maximized.

4.1.2 Global Optimization Techniques

Due to the complexity of routing and channel assignment in WMNs, global optimization techniques are applied. By the time of this monograph, over 90 different optimization techniques have been proposed, ranging from ant colony optimization to tabu search. We only describe four of them which are used for the planning and optimization of wireless mesh networks, namely tabu search, branch and bound, simulated annealing, and genetic algorithms.

Tabu Search

Tabu search is an extension of the local search technique for solving optimization problems. The algorithm was introduced by Glover in 1986 [158]. It enhances the local search method by using a memory structure. To avoid cycles of the possible solutions found by the algorithm, the solutions are marked as “tabu”. All solutions on the tabu list can not be used for the next iteration step.

The tabu search algorithm starts by using either a random solution or by creating a solution with a heuristic. From this initial solution x, the algorithm iteratively moves to a solution x’ in the neighborhood of x. From all possible solutions in the neighborhood of x, the best one is selected as the new solution if it is not on the tabu list. Afterwards, the tabu list is extended and the algorithm proceeds until a stopping criterion is reached.

Branch and Bound

The branch and bound method generally finds the optimal solutions with the disadvantage of being slow. In general, it is a search and comparison of different possibilities based upon partition, sampling, and subsequent upper bounding procedures. The first step, the branch, is used to split a problem into two or more subproblems. The iteration of the branch step creates a search tree. To avoid the calculation of all subtrees, the algorithm uses the bound step. It searches for the first valid solution whose value is the upper bound. All following calculations are canceled if their costs exceed the upper bound. If a new, cheaper solution is found, the upper bound will be set to the value of this new solution. Thus, the branch step increases the search space while the bound step limits it. The algorithm proceeds until either all subtrees have been evaluated or a threshold is met.

Simulated Annealing

The goal of simulated annealing is to find a good solution rather than to find the optimal solution like branch and bound. The name of the algorithm comes from metallurgy. Metal is heated up and then cooled down very slowly. The slow cooling allows to form larger crystals, which corresponds to finding something nearer to a global minimum-energy configuration.

When applying simulated annealing for the channel allocation in a WMN, the algorithm starts assigning channels randomly. If a small change in the channel assignment improves the throughput, i.e. lowers the cost or energy, the new solution is accepted and if it does not improve the solution it might be accepted based on a random function. If the change only slightly worsens the solution, it has a better chance to get accepted in contrast to a solution which heavily decreases the performance. Worse solutions are accepted with a probability given by the Boltzmann factor

$$e^{-\frac{E}{k_B \cdot T}} > R(0,1), \tag{4.1}$$

where E is the energy, k_B is the Boltzmann constant, T is the temperature, and $R(0,1)$ is a random number in the interval [0,1]. This part is the physical process of annealing. For a given temperature, all solutions are evaluated and then, the temperature is decremented and the entire process repeated until a stable state is achieved or the temperature reaches zero. This means that worse solutions are accepted with a higher probability when the temperature is high. As the algorithm progresses and the temperature decreases, the acceptance criterion gets more and more stringent.

Genetic Algorithms

Genetic algorithms are similar to simulated annealing and are also not applied to find the optimal solution but rather good ones. In contrast to the branch and bound method, they are much faster and therefore applicable for the planning and optimization of large wireless mesh networks.

GAs are based on the idea of natural evolution and are used to solve optimization problems by simulating the biological cross of genes. A randomly created population of individuals represents the set of candidate solutions for a specific problem. The genetic algorithm applies a so-called fitness function to each individual to evaluate its quality and to decide whether to keep it in the new population. However, the selection without any other operation will lead to local optima. Therefore, two operators, crossover and mutation, are used to create new individuals. These new individuals are called progenies. Figure 4.2 shows the influence of crossover and mutation on the fitness landscape of two traits. As mutation is just a swapping of one or two bits, it leads to only small changes in the fitness landscape. The crossover operator instead can lead to complete new solutions as indicated in the figure with the creation of the progeny. Thus, the crossover operator can protect the genetic algorithm from running into local optima, while a mutation is just a small step around a previous solution. Both operators together are used to find a near-optimal solution.

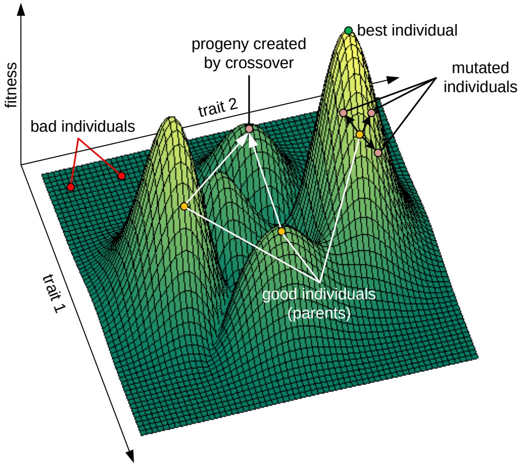


Figure 4.2: *Influences of crossover and mutation in the fitness landscape.*

Simulated annealing, genetic algorithms, and tabu search are well suited for the planning of wireless mesh networks. Applying the branch and bound algorithm would be too time consuming, especially when considering large WMNs. In the next section, we take a closer look at the work related to WMN planning and optimization where one of the described optimization methods is applied.

4.1.3 Related Work

Wireless mesh networks have attracted the interest of various researchers and Internet providers. Hence, a number of papers have been published on the problem of planning WMNs and estimating their performance. We divide the related work into three parts. The first part shows general WMN planning approaches. In the second part, the work related to channel assignment and routing is presented. Finally, we present papers working with genetic algorithms for planning radio networks.

Wireless Mesh Network Planning Using Optimization Techniques

Sen and Raman [159] introduce a variety of design considerations and a solution approach which breaks down the WMN planning problem into four tractable parts. These sub-problems are inter-dependent and are solved by heuristics in a definite, significant order. The evaluations of the presented algorithms show that they are able to generate long-distance WLAN deployments of up to 31 nodes in practical settings.

Other related works [160–162] deal with creating a wireless mesh network model, planning its parameters, and evaluating the solutions via linear programming. He et al. [160] propose mechanisms for optimizing the placement of integration points between the wireless and wired network. The developed algorithms provide best coverage by making informed placement decisions based on neighborhood layouts, user demands, and wireless link characteristics. Amaldi et al. [161] propose other planning and optimization models based on linear programming. The aim is to minimize the network installation costs by providing full coverage for wireless mesh clients. Thereby, traffic routing, interference, rate adaptation, and channel assignment are taken into account. Another cost minimizing, topology planning approach is presented by So and Liang [162]. An optimization framework is proposed which combines a heuristic with Bender's decomposition to calculate the minimum deployment and maintenance cost of a given heterogeneous wireless mesh network. Furthermore, an analytical model is presented to investigate whether a particular relay station placement and channel assignment can satisfy the user demands and interference constraints.

Routing and Channel Assignment for Wireless Mesh Networks

One of the first contributions on channel assignment is presented by Raniwala et al. [163,164]. The channels are assigned according to the expected load evaluated for shortest path and randomized multi-path routing. It is shown that by using only two network interface cards per mesh point, the throughput increases up to eight times. In contrast to Raniwala et al. [163,164], Chen et al. [165] do not only consider the expected load for the channel assignment, but also consider the link capacities. Based on the link metrics, called expected-load and expected-capacity, the channel assignment is optimized using simulated annealing.

Further papers based on the paper presented by Raniwala et al. [163] are published by Ramachandran et al. [166] and Subramanian et al. [167]. Both papers take the interference between links into account. The first paper solves the channel assignment using a straightforward approach while the second one uses a tabu search algorithm. Another paper on channel assignment and routing is presented by Alicherry et al. [168]. A linear programming based routing algorithm is shown which satisfies all necessary constraints for the joint channel assignment, routing, and interference free link scheduling problem. Using the algorithm, the throughput is fairly optimized. The fairness constraint means that for each node the demands are routed in proportion to the aggregate traffic load.

Raniwala and Chiueh [164] and Chen et al. [165] only consider non overlapping, orthogonal channels. Mohsenian Rad and Wong [169, 170] instead also consider partially overlapping channels and propose a congestion-aware channel assignment algorithm. It is shown that the proposed algorithm increases the aggregate throughput by 9.8 % to 11.4 % and reduces the round trip time by 28.7 % to 35.5 % compared to the approach of Raniwala and Chiueh [164].

Genetic Algorithms for Radio Network Planning

Genetic algorithms have been used for radio network planning for years [171–174]. Calégari et al. [171] apply a genetic algorithm for UMTS base station placement in order to obtain a maximum coverage. It is claimed that the performance of the GA strongly depends on the fitness function. Another paper on UMTS optimization with genetic algorithms was published by Ghosh et al. [172] in 2005. Genetic algorithms are used to minimize the costs and to maximize the link availability of a UMTS network with optical wireless links to the radio network controllers.

Besides Gosh et al. [172], Badia et al. [173] use genetic algorithms for a joint routing and link scheduling for WMNs. The packet delivery ratio is optimized depending on the frame length. It is shown that genetic algorithms solve the studied problems reasonably well, and also scale, whereas exact optimization techniques are unable to find solutions for larger topologies. The performance of the GA is shown for a single-rate, single-channel, single-radio WMN.

Vanhatupa et al. [174, 175] apply a genetic algorithm for the WMN channel assignment. Capacity, AP fairness, and coverage metrics are used with equal significance to optimize the network. The routing is fixed, using either shortest path routing or expected transmission times. An enormous capacity increase is achieved with the channel assignment optimization. Compared to manual tuning, the algorithm is able to create a network plan with 133 % capacity, 98 % coverage, and 93 % costs, while the algorithm needs 15 minutes for the optimization whereas the manual network planning takes hours.

In contrast to the related work, we focus not only on a single-radio or single-rate WMN, but evaluate the performance of a multi-channel, multi-radio, multi-rate WMN using both channel and route assignment. Our genetic algorithm optimizes the throughput while still maintaining a max-min fair throughput allocation between the nodes. In the next section, the complexity of a fair resource allocation in WMNs is described before introducing genetic algorithms and its modifications for the planning and optimization of WMNs.

4.2 WMN Planning Using Genetic Algorithms

The objective of this chapter is to support the WMN planning process by optimizing the performance of a WMN. With the help of genetic algorithms, near-optimal solutions can be achieved in relatively small computation time. In this section, we show the parameters which we have to consider and to evaluate in order to achieve a near-optimal WMN solution, meaning that the throughput in the WMN is fairly shared among the mesh points.

4.2.1 Problem Formulation

We assume that each mesh point is connected to only one gateway with a fixed routing and we can thus define the mesh network as a directed graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of mesh points $n_1, ..., n_V$ and $\mathcal{E} = \mathcal{L}$ is a set of links connecting the mesh points. A subset $GW \subseteq \mathcal{V}$ contains the gateways which are connected to the Internet. Each mesh point $n_i \in \mathcal{V} \setminus GW$ has a fixed route and gateway to the Internet. The route is denoted as $\mathcal{R}_i$ and consists of a set of links, $\mathcal{R}_i \subseteq \mathcal{L}$. Thus, the mesh points connected to one gateway can be considered as a tree and the complete WMN as a forest.

As we do not have a fully meshed network, a link (i, j) between mesh point i and mesh point j only exists, if a communication between these mesh points is possible within the mesh network. Let $dr_{i,j}$ be the data rate of the link (i, j). The goal is now to optimize the paths from each mesh point $n_i \in \mathcal{V} \setminus GW$ to the gateway so that the throughput in the WMN is fairly shared among the mesh points.

4.2.2 Fairness and Capacity in Wireless Mesh Networks

To achieve a fair resource distribution among the mesh points, we use a max-min fair share approach introduced by Bertsekas and Gallager [156]. A solution is max-min fair if no rate can be increased without decreasing another rate to a smaller value. Max-min fairness is achieved by using an algorithm of progressive filling. First, all data rates are set to zero. Then, the data rates of all flows are equally increased until one flow is constrained by the capacity set. This is the bottleneck flow and all other flows have to be faster than this one. Afterwards, the data rates of the remaining flows are increased equally until the next bottleneck is found. This procedure is repeated until all flows are assigned a data rate.

Before assigning the data rates to the flows, the capacity of the network has to be estimated. Therefore, we first have to estimate the link capacities. The capacity of a single link is determined by the pathloss and the Signal to Noise Ratio (SNR). For the pathloss calculation, we use a modified COST 231 Hata [176] pathloss model for carrier frequencies between 2 GHz and 6 GHz. The model is proposed by the IEEE 802.16 working group as the WiMAX urban macrocell model, but is also valid for WLAN mesh networks and is defined as

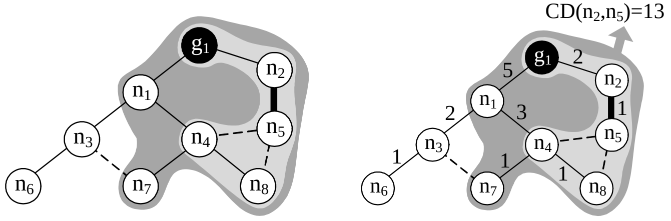
$$PL = 35.2 + 35 \cdot log_{10}(d(n_i, n_j)) + 26 \cdot log_{10}\left(\frac{f}{2}\right). \tag{4.2}$$

Here, f denotes the operating frequency and d denotes the euclidean distance between mesh points n_i and n_j. The pathloss model is used to calculate the SNR which is required to determine the maximum achievable throughput. The SNR is calculated as

$$\gamma_{n_i,n_j} = T_x - PL(n_i, n_j, f) - (N_0 + 10 \cdot log_{10}(W)), \tag{4.3}$$

where T_x is the transmit power, N_0 is the thermal noise spectral density (-174 dBm/Hz), and W is the system bandwidth. Now, the Modulation and Coding Scheme (MCS) mcs is selected with an SNR requirement γ^*_{mcs} that is smaller or equal to the link's SNR γ_{n_i,n_j}. The MCS is chosen in such a way that the frame error rate lies below 1 %. If the SNR requirement for the most robust MCS cannot be met, the two mesh points n_i and n_j are not within communication and interfering range.

Having computed the maximum data rate of each link according to the pathloss, we now have to calculate the capacity of each link taking interference from neighboring mesh points into account. Therefore, we use the concept of Collision Domains (CDs) introduced by Jun and Sichitiu [177]. The collision domain $\mathcal{D}_{i,j}$ of a link (i,j) corresponds to the set of all links (s,t) which can not be used in parallel to link (i,j) because the interference from a transmission on link (s,t) alone is strong enough to disturb a parallel transmission on link (i,j). Figure 4.3(a) shows the collision domain of link (n_2, n_5). The one-hop collision domain illustrated in light-gray denotes the area for a WLAN-based mesh network without using RTS/CTS. The dark gray area shows the two-hop area where no station can transmit a packet when using RTS/CTS.



(a) One- and two-hop collision domain of link (n_2, n_5).

(b) Link load calculation depending on the carried number of flows.

Figure 4.3: *Collision domain and its link loads.*

The nominal load of such a collision domain is the number of transmissions taking place in the collision domain. A transmission $tr_{k,i,j}$ corresponds to the hop from mesh point n_i to mesh point n_j taken by the flow towards mesh point k, i.e. $(i,j) \in \mathcal{R}_k$. The number of transmissions $\lambda_{i,j}$ on link (i,j) corresponds to the number of end-to-end flows crossing it:

$$\lambda_{i,j} = \Big|\{k|(i,j) \in R_k\}\Big|. \tag{4.4}$$

Figure 4.3(b) shows the load per link for the same example network as before. Each mesh point on the way to the gateway produces traffic resulting in a traffic load of 5 on the link (n_1, g_1) and a load of 2 on the link (n_2, g_1). Correspondingly, the number of transmissions in collision domain $\mathcal{D}_{i,j}$ is

$$m_{i,j} = \sum_{(s,t)\in\mathcal{D}_{i,j}} \lambda_{s,t}. \tag{4.5}$$

Thus, the collision domain of link (n_2, n_5) consists of 13 transmissions in total.

In order to fairly supply all mesh points, we share the time resources among all transmissions taking place within the collision domains of the corresponding links. Thereby, we take the rates $dr_{i,j}$ and the number of flows $\lambda_{i,j}$ into account. The throughput $t_{i,j}$ of link i, j is then defined as

$$t_{i,j} = \frac{1}{\sum\limits_{(s,t)\in\mathcal{D}_{i,j}} \frac{\lambda_{s,t}}{dr_{s,t}}}. \tag{4.6}$$

If we assume that link (n_2, n_5) supports 54 Mbps based on the pathloss and the SNR, the throughput would be 4.15 Mbps due to a collision domain size of 13. However, before setting this throughput to node n_5 we have to follow the principle of max-min fairness.

An algorithm to determine the max-min fair throughput allocation based on the definition of collision domains is given by Aoun and Boutaba [178]. The algorithm iteratively determines the bottleneck collision domain and allocates the data rates of all flows traversing this domain. If in our example in Figure 4.3 the link (n_1, g_1) would be the bottleneck, all mesh points traversing the link would be assigned to this throughput, in our case n_3, n_4, n_6, n_7, n_8. As link (n_2, n_5) and link (g_1, n_2) also belong to the collision domain of link (n_1, g_1) but do not transmit over the bottleneck link, the time resources occupied by the bottleneck link are subtracted from the two links.

In the next step of the iteration process, only the remaining collision domains are considered. This way, we calculate the throughput of each flow which is needed to evaluate the fitness of the WMN. The iteration stops when all flows are assigned. If in our example the next bottleneck collision domain is link (g_1, n_2), the remaining maximum supported rates are assigned to the last two links. Algorithm 2 clarifies the procedure of assigning the rates.

Algorithm 2 Max-min fair resource distribution based on collision domains.

```
$\mathcal{O} = \mathcal{F}$                                                    all flows are unassigned
$\mathcal{L} = \{(i,j) | n_{i,j} > 0\}$                                        all active links
$p_{i,j} = 1$, $(i,j) \in \mathcal{L}$                                         all links have full capacity

*Iteration*
    **for all** links $(i,j) \in \mathcal{L}$
        $m_{i,j} = \sum_{(s,t) \in \mathcal{D}_{i,j}} \lambda_{s,t}$           nominal load
        $t_{i,j} = \frac{1}{\sum_{(s,t) \in \mathcal{D}_{i,j}} \frac{\lambda_{s,t}}{dr_{s,t}}}$   throughput share per flow
    **end for**
    $(u,v) = \arg\min_{(i,j) \in \mathcal{L}} t_{i,j}$                          bottleneck collision domain
    $\mathcal{B} = \{k \in \mathcal{O} | \mathcal{R}_k \cap \mathcal{D}_{u,v} \neq \emptyset\}$   bottleneck flows
    $b_k = r \cdot t_{u,v}$ for all $k \in \mathcal{B}$                          set bottleneck rates
    $\mathcal{O} = \mathcal{O} \setminus \mathcal{B}$                           adapt unassigned flows
    $p_{i,j} = p_{i,j} - \sum_{k \in \mathcal{B}} |\mathcal{R}_k \cap \mathcal{D}_{i,j}| \cdot t_{u,v}$   adapt free capacity of all CDs
    $\mathcal{L} = \mathcal{L} \setminus \mathcal{D}_{u,v}$                     adapt active links
*Stop criterion:* $\mathcal{O} = \emptyset$
```

4.2.3 Optimization Using Genetic Algorithms

After describing the principle of collision domains and max-min fair throughput allocation, we now explain the workflow of a genetic algorithm in detail. Figure 4.4 shows the complete procedure of a genetic algorithm for the planning and optimization of WMNs. Firstly, a random population is created with a predefined number of individuals. The fitness of each individual is evaluated using the fitness function and the individuals are ordered according to the fitness value. The best individuals, the elite set, is kept for the new population. Afterwards, the crossover and mutation operator are used to create the remaining number of individuals for the new population. The procedure is repeated until a sufficient solution is achieved. In the following, we explain the steps of our WMN optimization approach in more detail.

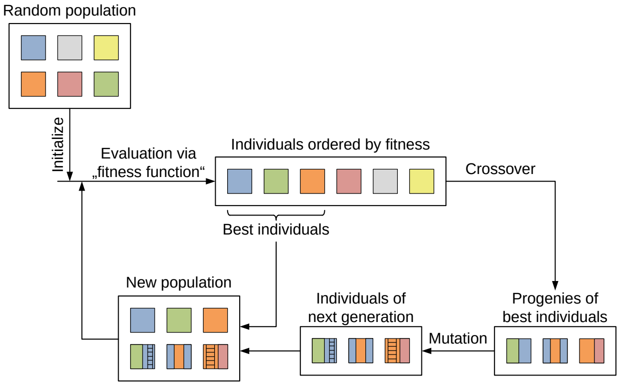
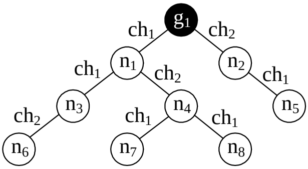

Random population
Initialize
Evaluation via „fitness function“
Individuals ordered by fitness
Crossover
Best individuals
New population
Individuals of next generation
Progenies of best individuals
Mutation

Figure 4.4: *Workflow of a genetic algorithm.*

Network Encoding

Before going through the steps of the genetic algorithm, the WMN has to be encoded. The encoding must be simple without any redundancy in order not to prolong the runtime of the genetic algorithm. As we assume that each mesh point is connected to only one gateway, the network encoding has to represent a spanning tree with the gateway as root, cf. Figure 4.5(a). This means that the graph does not contain any cycles and each mesh point has only one route towards the gateway. Such a tree structure can easily be encoded in a list, where the next hop of each mesh point, which the traffic has to take in order to reach the gateway, is stored. This list representation of the example network from Figure 4.5(a) is shown in Figure 4.5(b). Considering for example mesh point n_4, the next hop is node n_1 and the next hop of mesh point n_1 is the gateway. Thus, the complete routing of a WMN is handled with a simple list representation.



(a) Example network.

node	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
next hop	g_1	g_1	n_1	n_1	n_2	n_3	n_4	n_4
channel	ch_1	ch_2	ch_1	ch_2	ch_1	ch_2	ch_1	ch_1

(b) List representation.

Figure 4.5: *Example network and its list representation.*

Besides the routing, we also want to optimize the channel allocation. Although each mesh point can be equipped with several network interface cards, the channel of the link towards the gateway is fixed as shown in Figure 4.5(a). Thus, the channel allocation can be done in a similar way as the routing. Therefore, the list is extended with one more row, showing the channel of the next hop towards the gateway, cf. Figure 4.5(b). This simple list represents the tree structure of one gateway and each gateway in the wireless mesh network is encoded in a similar way. The list representation is later used to perform the genetic operations and to evaluate the fitness of the WMN.

Evaluation via Fitness Function

The evaluation part of the optimization is the heart of the genetic algorithm. Based on the fitness value, the GA decides which individuals should be kept in the new population. Hence, it rates the performance of the genes and allows only the best to be replicated.

The fitness of the WMN is estimated using the allocated throughputs of each flow. The fitness function $f(\mathcal{N})$ of the evaluation represents the user satisfaction and the fairness of the resource allocation. Some fitness functions might lead to a complete unfair resource distribution in the WMN. Therefore, we evaluate the performance of several different fitness functions in Section 4.2.3. Several combinations of the functions $min(\mathcal{R}_\mathcal{N})$, $median(\mathcal{R}_\mathcal{N})$, $mean(\mathcal{R}_\mathcal{N})$, and $var(\mathcal{R}_\mathcal{N})$ are used, which are applied on all links of a network solution $\mathcal{N}$. The function $min\left(\mathcal{R}_\mathcal{N}\right)$ calculates for example the minimum throughput of all links used in routing scheme $\mathcal{R}_\mathcal{N}$. We define the following eight different fitness functions:

$$\begin{aligned}
f_1(\mathcal{N}) &= min(\mathcal{R}_\mathcal{N}) = \text{minimum throughput}(\mathcal{R}_\mathcal{N}) \\
f_2(\mathcal{N}) &= median(\mathcal{R}_\mathcal{N}) = \text{median throughput}(\mathcal{R}_\mathcal{N}) \\
f_3(\mathcal{N}) &= mean(\mathcal{R}_\mathcal{N}) = \text{mean throughput}(\mathcal{R}_\mathcal{N}) \\
f_4(\mathcal{N}) &= min(\mathcal{R}_\mathcal{N}) + \frac{median(\mathcal{R}_\mathcal{N})}{s} \\
f_5(\mathcal{N}) &= mean(\mathcal{R}_\mathcal{N}) - var(\mathcal{R}_\mathcal{N}) \\
f_6(\mathcal{N}) &= min(\mathcal{R}_\mathcal{N}) + \frac{median(\mathcal{R}_\mathcal{N})}{s} + \frac{mean(\mathcal{R}_\mathcal{N})}{|\mathcal{L}|} \\
f_7(\mathcal{N}) &= \sum_{i=0}^{\left|\tilde{T}\right|-1} (\left|\tilde{T}\right| - i) \cdot \tilde{T}(i) \\
f_8(\mathcal{N}) &= \sum_{i=0}^{\left|\tilde{T}\right|-1} c^{\left|\tilde{T}\right|-i} \cdot \tilde{T}(i).
\end{aligned}$$

The last two functions weight the link throughputs with a factor depending on the corresponding throughput value. Therewith, we aim to achieve a kind of max-min fairness not only with the throughput allocation made by the evaluating algorithm but also with the fitness value from a reasonable fitness function. For this purpose, an ascendingly sorted list $\tilde{T}$ of the throughputs of all routing links in the solution $\mathcal{N}$ is used. Each throughput value from $\tilde{T}$ is weighted with a factor depending on its place in the list, giving more weight to lower positions.

This results in a fitness value with which mainly smaller link throughputs are optimized at the expense of higher ones. The parameter c of function $f_8(\mathcal{N})$ is a constant which we set to 1.5 and s is set to 8 for the experiments in Section 4.2.3.

Selection Principle

After the evaluation of a population, we select a set of solutions which have the highest fitness of all and keep them in the new generation. This set is called the elite set. In Section 4.3.3 we vary the size of the elite set in order to see the influence on the solution. As the number of individuals of a population is fixed for all generation steps, the remaining number of individuals are created by crossing and mutating the genes.

The selection of the individuals for applying the genetic operators is thereby based on the fitness and furthermore depends on the number of needed new individuals. Let w be the number of needed new individuals and $s(x)$ be the selection probability for individual x. Then, the number of progenies generated based on individual x are

$$g(x) = \|w \cdot s(x)\| \, . \tag{4.7}$$

The selection probability $s(x)$ depends on the relation between the fitness of solution x and the sum of all fitness values from the complete population which means that new individuals are more likely to be created from individuals with a better fitness. This results in

$$s(x) = \frac{f(x)}{\sum\limits_{j=1}^{n} f(j)} . \tag{4.8}$$

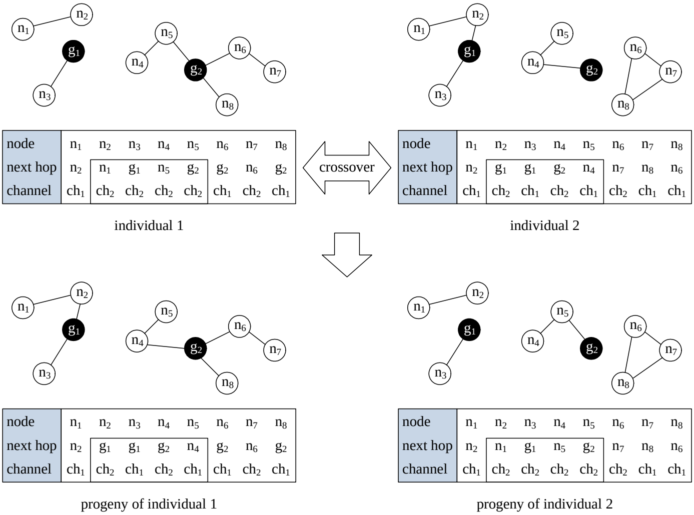
Crossover Types

The crossover operator as well as the mutation operator are now applied to the selected number of individuals. For the cross of genes, we use the standard 2-Point Crossover [179] and two other variants which we especially created for the planning of WMNs, the Cell and the Subtree Crossover.

2-Point Crossover

The 2-Point Crossover is a widely used extension of the 1-Point Crossover. While the 1-Point Crossover changes the list representations of two individuals until a certain point or from a certain point on, the 2-Point Crossover exchanges subsets which are randomly chosen sublists of the individuals representation, the genotype. Thus, a start and an end point, denoting the range of the sublist, are chosen each time the 2-Point Crossover is applied.

An example of the crossover is shown in Figure 4.6. The sublists of two individuals should be crossed, namely the routing and channel allocation of mesh points n_2 to n_5. The resulting progenies of the individuals show one characteristic of this reproduction approach. It created solutions which contain mesh points with no connection to any gateway. This happens due to the unregulated and absolutely arbitrary selection of the gene subset which is meant to be exchanged.



node	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
next hop	n_2	n_1	g_1	n_5	g_2	g_2	n_6	g_2
channel	ch_1	ch_2	ch_2	ch_2	ch_2	ch_1	ch_2	ch_1

individual 1

node	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
next hop	n_2	g_1	g_1	g_2	n_4	n_7	n_8	n_6
channel	ch_1	ch_2	ch_1	ch_2	ch_1	ch_2	ch_1	ch_1

individual 2

node	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
next hop	n_2	g_1	g_1	g_2	n_4	g_2	n_6	g_2
channel	ch_1	ch_2	ch_1	ch_2	ch_1	ch_1	ch_2	ch_1

progeny of individual 1

node	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
next hop	n_2	n_1	g_1	n_5	g_2	n_7	n_8	n_6
channel	ch_1	ch_2	ch_2	ch_2	ch_2	ch_2	ch_1	ch_1

progeny of individual 2

Figure 4.6: *2-Point Crossover between two individuals.*

Looking at the progeny of individual 2, mesh points n_1, n_2, n_6, n_7, n_8 have no connection to any gateway and thus, the crossover results in an unreasonable solution. On the other hand, the 2-Point Crossover has created a reasonable progeny of individual 1.

Since the 2-Point Crossover may lead to unconnected solutions, we have to be careful when evaluating the fitness of the resulting solutions. Thus, we adapt the fitness function to

$$\tilde{f}(\mathcal{N}) = f(\mathcal{N}) - diss(\mathcal{V}). \tag{4.9}$$

which includes now the $diss(\mathcal{V})$ term denoting the number of nodes with no connection to any gateway. Hence, the throughput contained in $f(\mathcal{N})$ presents the positive costs of the network while $diss(\mathcal{V})$ stands for the penalty costs.

Cell Crossover

In contrast to the 2-Point Crossover, the Cell Crossover does not exchange sublists but complete cells. The crossover operator randomly chooses a gateway and exchanges the entire cell meaning that the routing information as well as the channel allocation is exchanged.

Figure 4.7 shows an example for the crossover of two solutions. Black nodes denote the network gateways and the light gray areas mark the chosen cell which is exchanged. In the resulting progenies, the mesh points that have changed their connection are marked dark gray. We can see that not only link connections from mesh points are crossed, but some mesh points are now also connected to other gateways. Mesh points $n_{10}, n_{12}, n_{17}, n_{18}$ are connected to gateway g_2 in the progeny of individual 2 while they were attached to gateway g_1 before. The reason is that the number of mesh points belonging to one cell differ between the individuals. Therefore, we also have to attach unconnected nodes after the Cell Crossover which can be seen in the progeny of individual 1. In addition to the exchange of routes, the assigned channels are exchanged which is not shown in the figure for the sake of readability.

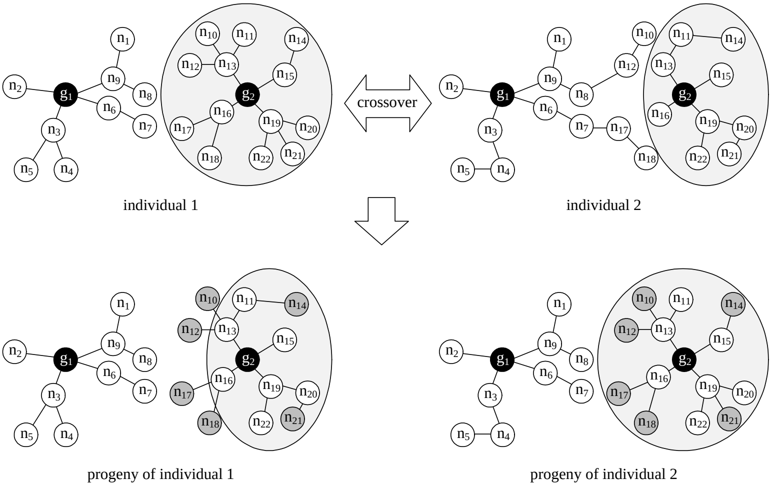


Figure 4.7: *Cell Crossover between two individuals.*

Subtree Crossover

The last crossover type is the Subtree Crossover. In contrast to the Cell Crossover, not a complete gateway tree is exchanged but only a subtree. Therefore, the Subtree Crossover chooses mesh points randomly and crosses the entire subtree with the mesh point as root. Similar to the Cell Crossover, the channel allocation is exchanged together with the routing information.

The Subtree Crossover of two subtrees is shown in Figure 4.8. The chosen mesh points are n_3 and n_{13}. The crossover of subtree n_3 only causes a small change in the tree structure in contrast to the subtree crossover of n_{13}. Here, some nodes of the subtree are connected to different gateways in the two individuals. After the crossover, mesh points n_{10} and n_{12} belong to gateway g_2 in the progeny of individual 2. This reduces the number of long branches of gateway g_1 but there is still potential for further optimization.

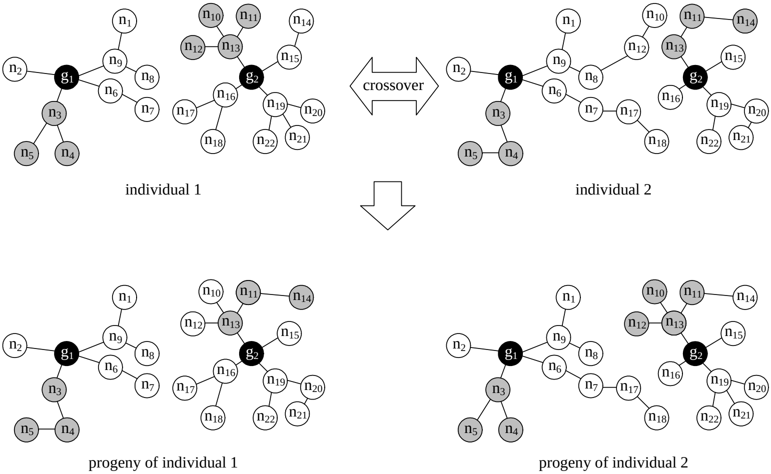


Figure 4.8: *Subtree Crossover between two individuals.*

4.2.4 Mutation

While the different crossover variants help to avoid running into local optima, the mutation operator increases the performance of WMNs with slightly modifications of the routing structure and channel allocation. For the optimization of WMNs, the number of mutations are chosen based on the scenario size and the mutation of the routing and channel allocation are applied independently from each other.

For the routing scheme, the mutation operator substitutes some randomly chosen positions of the routing code with new information taken from a set of potential neighbors which would not cause the creation of cycles and would not harm the tree structure of the solution. An example for the mutation of the routing scheme from three nodes is shown in Figure 4.9. Here, the links towards the gateway of the three gray nodes are mutated. For the channel allocation, the mu-

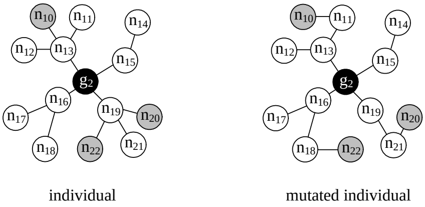


Figure 4.9: *Routing mutation of three mesh points.*

tation operator randomly chooses a channel from a list of possible channels and substitutes randomly chosen links from the WMN.

According to the workflow diagram shown in Figure 4.4, the mutation operator is applied after the crossover on the progenies of the crossover. The mutated individuals together with the elite set form then the new population and close the circle of the genetic algorithm.

4.3 Performance Evaluation

After introducing genetic algorithms in detail and showing our modifications and extensions for wireless mesh networks, we now want to evaluate the performance of the genetic algorithm. The influence of every part of the genetic algorithm's workflow is thereby evaluated separately. First, we take a look at the influence of the fitness function on the resulting solution. Afterwards, the size of the elite set is investigated followed by the population evolution for the three different crossover types. Finally, we show the influence of the two genetic operators crossover and mutation on the resulting network solution.

4.3.1 Simulation Settings

For the creation of the results presented in this section, we use the two scenarios introduced in Table 4.1. Although we evaluated a large number of different scenarios, we highlight only the two most different ones here. The first one consists of 2 gateways and 71 mesh points distributed over an area of 2 km x 1.2 km. Thereby, the minimal distance between mesh points is 60 m and between the two gateways it is 700 m. For the sake of readability, we call this scenario G2MP71. The second scenario contains a smaller number of mesh points and a larger number of gateways. We choose this clearly different topology in order to show the influence of the crossover operators depending on the number of mesh points. The 38 mesh points and 6 gateways of the second scenario are allocated in an area of 1.5 km x 1 km. The minimal distance between users is 60 m and between gateways 450 m. We call this scenario G6MP38.

Table 4.1: *Simulation Scenarios.*

Parameter	Scenario S1	Scenario S2
Topology	G2MP71	G6MP38
Population size	150	
Elite set size	50	
Number of generations	400	
Crossover type	Subtree Crossover Cell Crossover 2-Point Crossover	
Number of crossed subtrees	rand(0,7)	rand(0,5)
Number of mutations	rand(0,20)	rand(0,10)
Fitness function	$f_1(\mathcal{N})$	

The differences in the settings of the two configurations depend on the used topology of the corresponding scenario. Due to the larger number of mesh points contained in G2MP71, we configure Scenario S1 with more mutations and more exchanged subtrees than Scenario S2. Thereby, we keep the relation between crossover and mutation at a fixed level suitable for the investigation of the genetic operators.

Besides the parameters of the genetic algorithm, the general parameter settings are shown in Table 4.2. These parameters only affect the characteristics of the network connections. The parameters carrier frequency, channel bandwidth, and available channels decide to some extent the performance of the mesh point connections in a network solution but they do not have an impact on the effectiveness of the genetic algorithm. Therefore, we do not consider their impact on the resulting solutions.

Table 4.2: *General Parameter Settings.*

Parameter	Value
Carrier frequency	3500 MHz
Channel bandwidth	20 MHz
Maximum throughput	67.2 Mbps
Available channels	3500 MHz, 3510 MHz
Antenna power	25 dBm
Pathloss model	WiMAX urban macrocell model

4.3.2 Influence of Fitness Function

As the fitness function is the heart of the genetic algorithm, we first take a look on the influence of different fitness functions on the resulting solution. Therefore, eight different fitness functions, described in Section 4.2.3, are applied.

Figure 4.10 shows the throughputs of the mesh points of the best individual after 400 generations of Scenario S1. For the sake of readability, the curves of the eight different fitness functions are shown in two separate subfigures. The x-axis shows the normalized flow IDs, meaning the 71 mesh points sorted by throughput, and the y-axis lists the throughput in Mbps of the flows.

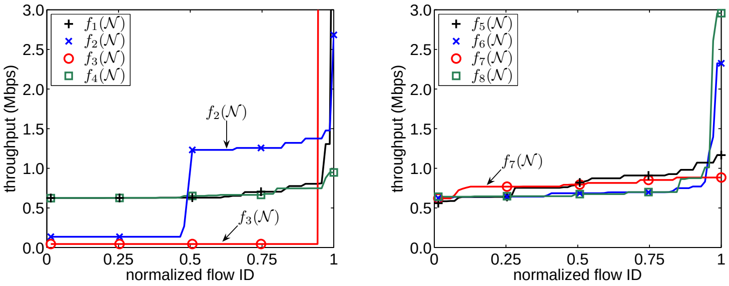
A curve completely parallel to the x-axis would mean a perfect fairness between all flows and a curve whose minimum throughput is above $f_1(\mathcal{N})$ would mean that the solution is max-min fair. This allows to see that the unfairest resource distributions are achieved with the fitness functions $f_2(\mathcal{N})$ and $f_3(\mathcal{N})$.

Optimizing only the median with $f_2(\mathcal{N})$, we do not pay attention to the rest of the throughput allocation. This is why the left part of the $f_2(\mathcal{N})$ curve stays very low. The distribution of $f_2(\mathcal{N})$ also shows that some mesh points have a very high throughput compared to others. This happens accidentally because the fitness function does not control their behavior as it focuses just on the throughput of the median.

Fitness function $f_3(\mathcal{N})$, optimizing only the mean throughput, also results in a very unfair solution. Here, the number of hops towards the gateway are minimized in order to get some nodes with very high throughput which boost the mean value. In this scenario, four mesh points have a throughput of over 24 Mbps while the throughput of all other flows is about 0.05 Mbps.

All other fitness functions result in a max-min fair resource distribution with a maximized minimal throughput. In the resulting solutions of $f_1(\mathcal{N})$, $f_6(\mathcal{N})$, and $f_8(\mathcal{N})$, some flows have a very high throughput but not at the costs of other flows.

The fairest solution is achieved with fitness function $f_7(\mathcal{N})$ where all flows have a similar throughput of about 0.7 Mbps. The fitness function weights the throughputs of the mesh points. Thereby, smaller throughputs have a stronger influence on the fitness than higher throughputs. This is achieved by multiplying the throughputs with the inverse of the ascendingly sorted flow ID.



(a) $f_1(\mathcal{N})$ to $f_4(\mathcal{N})$.

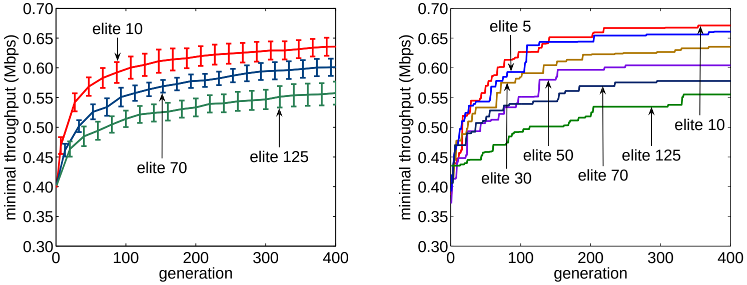
(b) $f_5(\mathcal{N})$ to $f_8(\mathcal{N})$.

Figure 4.10: *Throughput allocation of the best individual.*

4.3.3 Elite Set Size

In this section, we examine the impact of the elite set size on the progress of the evolution using Scenario S1 and applying the Subtree Crossover only. Figure 4.11(a) illustrates the minimal throughput of three different elite set sizes averaged over 15 different initial populations. This time, the x-axis shows the generation number while the y-axis lists the minimal throughputs.

From the figure it can be observed that the best performance is achieved with a small elite set size. On the one hand, a large elite set includes a number of bad individuals which are kept in the next generation and decrease the minimal throughput. On the other hand, with an elite set size of 125, only 25 new progenies are generated. With this small number of new unexplored genes, the progress of the genetic algorithm slows down which can be seen on the left side of the figure. Similar solutions compared to an elite set size of 10 might be achieved after several more generations. This means that the larger the elite set size is, the slower is the progress of the genetic algorithm. To prove this statement, we performed the optimization of the same scenario for more different elite set sizes. The results are shown in Figure 4.11(b).



(a) Three elite set sizes averaged over 15 seeds.

(b) Performance of six elite set sizes.

Figure 4.11: *Comparison of different elite set sizes.*

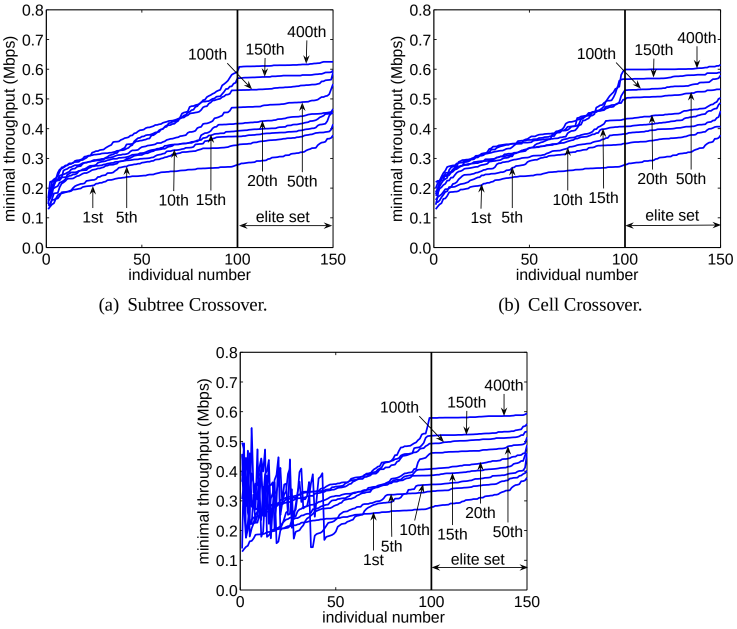
The figure reveals almost the same behavior as the previous one. Smaller elite sets cause faster evolution and lead to better solutions. However, a too small elite set size is also bad as the figure shows for an elite set size of 5. With a too small elite set, there might be a discrepancy between the fitness of the elite set and the fitness of new progenies. Thus, the elite set size should be chosen in dependence of the population size.

4.3.4 Population Evolution

Examining the evolution of the population is an important consideration needed to demonstrate the effectiveness of the genetic algorithm. Observing the evolution of the population with every generation step helps to decide when to terminate the algorithm. When the fitness is not increasing after an additional number of generations, the genetic algorithm can be stopped because either a near-optimal solution is found or the genetic algorithm is stuck in a local optimum. As the crossover operator helps to get out of a local optimum, we take a look at the population evolution for all three introduced crossover types.

The results shown in Figure 4.12 are generated with Scenario S1 from Table 4.1. The x-axis shows the individuals sorted by fitness and the y-axis displays the minimal throughput of each individual. The different curves illustrate the generation progress during the genetic optimization. The elite set size is chosen to be one third of the complete population size.

In order to compare all three crossovers, we did not plot the fitness but the minimal throughput on the y-axis. As the penalty costs are included in the fitness function of the 2-Point Crossover, cf. Section 4.2.3, the fitness values would be much lower for the 2-Point Crossover. Hence, we consider only the minimal throughputs which only represent the positive costs. This is also the reason for the strongly varying curves on the left side of Figure 4.12(c). The individuals have a large minimal throughput but there are a lot of unconnected nodes which result in a lot of penalty costs and thus in lower fitness.



(a) Subtree Crossover.

(b) Cell Crossover.

(c) 2-Point Crossover.

Figure 4.12: *Generations progress using Subtree, Cell, and 2-Point Crossover.*

In all subfigures, we can observe that the higher the generation number is, the smaller is the fitness growth. This slowdown is caused by the similarity of individuals. After several generations, the individuals are quite similar, which means that the crossover does not generate new, unexplored genes. The only possibility to find better solutions is to apply the mutation operator only. Therefore, we introduce the concept of local optimization in Section 4.4.

Evaluating the population evolution in other scenarios has shown that it highly depends on the topology structure but a good solution is always found after 400 generations. We tested the performance of Scenario S1 also after 1000 and

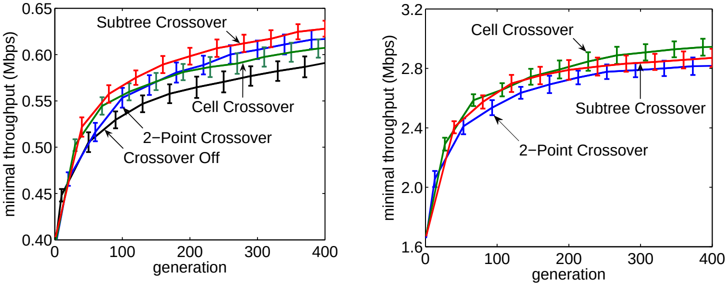
1500 generations, but the performance increase was negligible compared to the throughput after 400 generations.

A comparison of the three crossover types shows that the highest minimal throughput after 400 generations is achieved with the Subtree Crossover, followed by the results of the Cell Crossover. The network solution with the worst performance is achieved when applying the 2-Point Crossover. In the next section, we want to see if this is an exception or if the Subtree Crossover always leads to the best solutions.

4.3.5 Effectiveness of Crossover

In order to show the effectiveness of the crossover type, we compare the performance of the three crossover operators depending on the number of mesh points and gateways in the network. Furthermore, we want to find out if there is an interaction between the efficiency of the crossover types depending on the topology.

The results for both scenarios from Table 4.1 are presented in Figure 4.13. Figure 4.13(a) shows the evolution of the best individual during 400 generations with different crossover types and for not using the crossover operator at all for Scenario S1. It illustrates the average results of 20 seeds while applying a 95% confidence interval.



(a) Scenario S1. (b) Scenario S2.

Figure 4.13: *Effectiveness of the crossover operator.*

This scenario includes a high number of mesh points which are distributed in the coverage areas of only two gateways. This results in deep tree structures with long ways over multiple hops towards the corresponding gateway. Such network structures seem to be crucial for the effectiveness of the crossover types. We can observe that the Subtree Crossover leads to a better solution than the other two crossover types. The better performance of the subtree approach is the result of the exchange of small connectivity components which causes reasonable gene variations without disturbing the tree structure. The other two crossover types show a lower performance whereby the unregulated 2-Point Crossover even outperforms the intelligent Cell Crossover approach. This results from the small number of gateways which causes the cross of only one cell per new progeny and quickly leads to similar individuals.

The results from Scenario S2 are shown in Figure 4.13(b). In contrast to the previous scenario, the higher number of available gateways leads to a better efficiency of the Cell Crossover. Moreover, the small number of nodes belonging to one gateway allows a larger variety of individuals. This is due to the fact that small changes in the routing structure cause higher changes in the network performance than in Scenario S1. However, the Cell and Subtree Crossover which exchange only connectivity components have a better performance than the 2-Point Crossover.

The comparison of the crossover types shows that the crossover operator should be selected based on the considered topology to achieve the best solutions. In the next section, we take a look at the influence of the mutation operator on the evolution of the population.

4.3.6 Effectiveness of Mutation

The mutation operator causes small changes in the fitness landscape and normally does not help to get out of local optima. However, in the last section we have seen that applying only the mutation operator almost increases the performance of the wireless mesh network to the same level as compared to a scenario where both, crossover and mutation are applied. To investigate the influence of the mutation operator, Scenario S1 is considered. Both mutation operations, the routing and the channel mutations, are applied on the progenies of the crossed individuals. The number of routing and channel mutations on each individual are chosen randomly in the interval [0,20]. Figure 4.14 shows the minimal throughputs during the progress of the genetic algorithm for all three crossover types.

Surprisingly, the performance of the genetic algorithm without mutation is generally low and the genetic algorithm runs into a local optimum after a few generations. In contrast, when activating the mutation operator, the fitness of the solution grows even after 400 generations and there is still potential for further evolution. This shows how crucial the mutation operator is for the evolution of the genetic algorithm. Without using the mutation operator, similar individuals are created by the three different crossovers. The best performance is here seen for the Subtree Crossover as the Subtree Crossover has the largest possibilities to create new genes. The mutation operator instead ensures the creation of new unexplored genes with slight changes in the routing scheme and channel allocation which fosters the evolution.

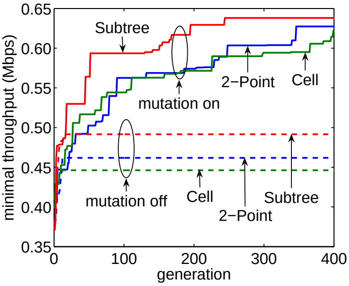


Figure 4.14: *Mutation ON/OFF in combination with three crossover types tested on Scenario S1.*

4.4 Optimization of the WMN Planning Approach

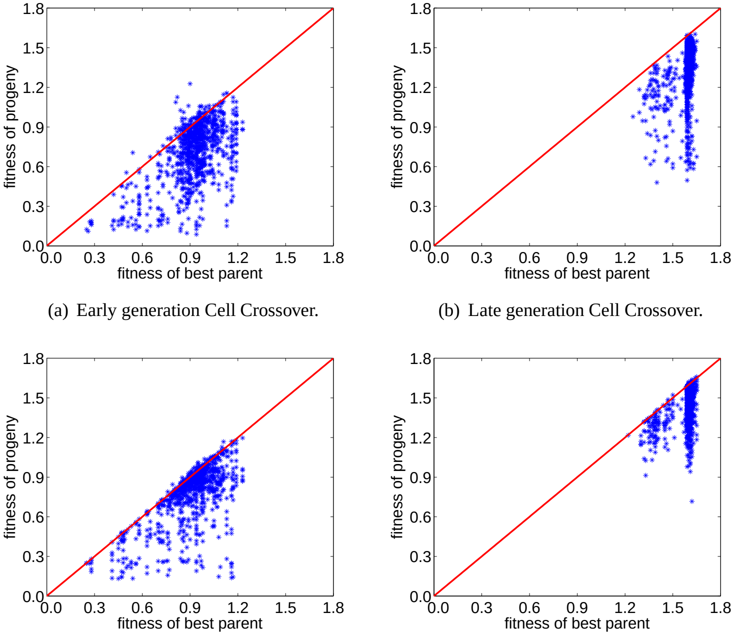
In the last section, we have seen the influence of the genetic operators on the performance of the resulting wireless mesh network. In this section, we take a look at the influence of the genetic operators in dependence of the GA progress and introduce a local optimization technique to quickly improve the performance of the wireless mesh network.

4.4.1 Influence of the Crossover on the GA Progress

As crossover operations are very time consuming, we want to see if the crossover types lead to better network solutions during all generations. Therefore, we compare the fitness of the best parent with the fitness of the resulting progeny for early generations as well as for late generations. The genetic optimization runs for 500 generations and the results in Figure 4.15 show the fitness of the Cell Crossover and Subtree Crossover of 2000 samples.

Looking at Figure 4.15(a), we can see that about 10 % to 20 % of all crossover operations lead to better progenies. Although this amount seems to be very low, we have to take a look at the exact improvements. One early Cell Crossover increases the fitness from 0.9 to 1.2. This might be a step out of two local optima in the fitness landscape. However, performing a Cell Crossover in the late stages of the genetic algorithm always leads to worse progenies. The reason is simple as a Cell Crossover of two near-optimal solutions are likely to create unreasonable progenies.

When applying the Subtree Crossover, the results are a little bit different as shown in Figure 4.15(c) and Figure 4.15(d). Although the percentage of better progenies is similar to the Cell Crossover, the improvements are lower. The reason is that the Subtree Crossover performs only small variations by exchanging subtrees, whereas the Cell Crossover changes two complete cells. However, these small changes also have a bad influence when performing them at the end of the generation process and only one or two progenies are better than their parents.



(a) Early generation Cell Crossover.

(b) Late generation Cell Crossover.

(c) Early generation Subtree Crossover.

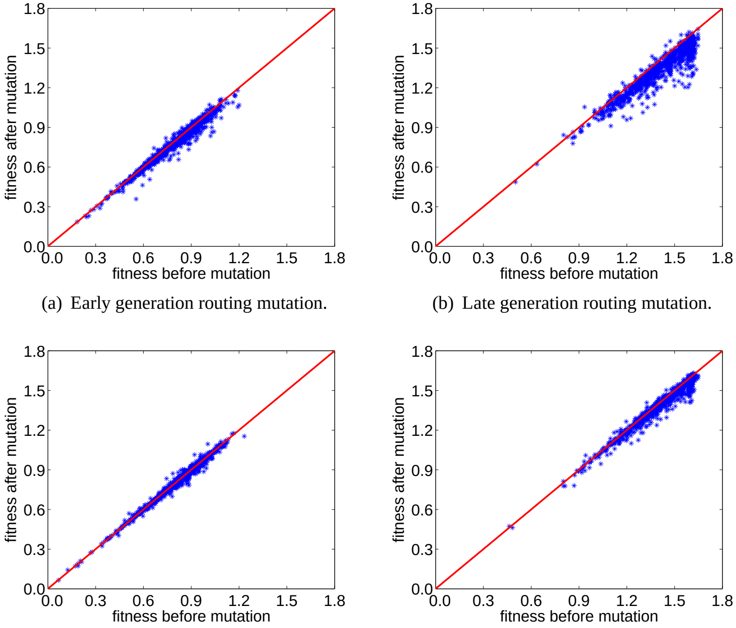
(d) Late generation Subtree Crossover.

Figure 4.15: *Influence of the crossover operator on the fitness of the resulting progenies.*

Thus, the amount of crossovers can be reduced with increasing number of generations. Before doing this, we take a look at the influence of the mutation operator in dependence of the number of generations.

4.4.2 Influence of the Mutation Operator Depending on the GA Progress

The mutation operator conducts only small modifications of the individuals and it is thus expected that the fitness only slightly changes after the mutation is performed. Furthermore, we want to evaluate if, in contrast to the crossover, the mutation also leads to better results when applied on late generation steps. The results for the two mutation operators, routing and channel allocation, are shown in Figure 4.16. The plots are generated based on 2000 samples taken at the beginning and at the end of a 500 generation run.



(a) Early generation routing mutation.

(b) Late generation routing mutation.

(c) Early generation channel mutation.

(d) Late generation channel mutation.

Figure 4.16: *Influence of the mutation operator on the GA progress.*

As expected, the change in the fitness value is only small after the mutation is applied. However, the number of improved individuals is larger for both mutation operators compared to the crossover operations. The channel mutation even yields better results in 50 % of all mutations. Although the performance of both mutation operators decreases with an increasing number of generations, still better individuals are achieved in 5 % to 10 % of all mutations, cf. Figure 4.16(b) and Figure 4.16(d).

Thus, the mutation operator should be applied during the complete generation process. However, when reducing the number of crossover operations with an increasing number of generations, the number of performed mutations are also decreased. In order to keep the number of mutations, the following mechanism is applied. Firstly, the elite set size is increased with each generation which means that increasingly more individuals are kept for the following population. This reduces the number of crossover and mutation operations. Secondly, in order to apply the mutation operator during the complete generation process, both mutation operations are performed with each individual of the elite set. If the fitness after the mutation is higher than before the mutation, the new individual is taken for the next population instead of the old one. If the fitness is worse, the new individual is discarded.

In Figure 4.17 we compare the fitness values of the ten best individuals in a scenario with an enlargement of the elite set size with increasing generation number and without an enlargement. The values are averaged over 10 simulation runs with 500 generations. Except for the worst of all 10 individuals, the enlargement of the elite set has a positive influence on the fitness. On average, the fitness is increased by 8 %.

Summarizing, a reduction of the number of crossover operations achieved by a stepwise enlargement of the elite set size has a positive effect on the fitness value. In addition, the runtime of the genetic algorithm is reduced due to the smaller number of complex calculations of the crossover operations.

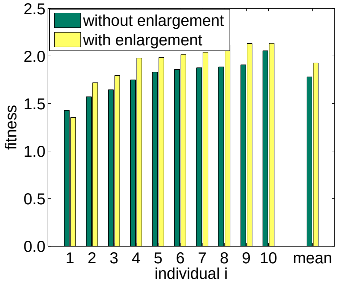


Figure 4.17: *Influence of the enlargement of the elite set.*

4.4.3 Local Optimization and Population Size

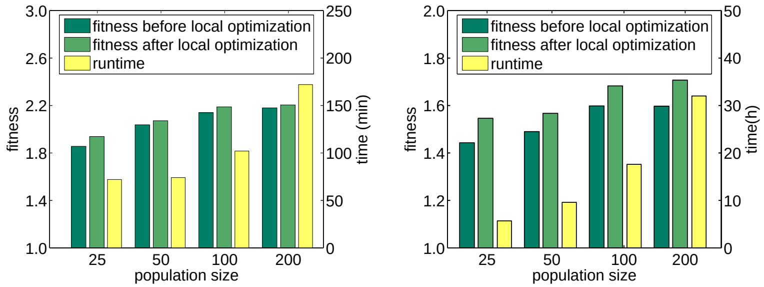
As we have seen in the previous figures, applying the crossover operator on late generations almost always results in worse individuals. However, the mutation operator might improve the individuals because it only slightly changes the individuals. To take advantage of this, we introduce the concept of local optimization. After the normal genetic algorithm finishes, we take the five best individuals of the last generation, copy them three times, and perform several mutations with them. Similar to the previous improvement, the resulting individual is only kept if its fitness value is higher compared to the fitness value before the mutation, else it is discarded. This can be repeated more than a thousand times because the computation time for mutating 15 individuals is negligible.

In order to investigate the effect of the local optimization, we take a look at the influence of the population size. The larger the population size, the more new individuals are created per generation resulting in a larger number of good individuals. This means that a large population size has the potential to get to the optimal solution but requires more computation time. In order to find a good population size, we need to look at the fitness of the best individual for a variety of population sizes and compare it to the runtime.

To see the influence of the population size as well as the local optimization, a genetic algorithm run with 500 generations is performed with an additional local optimization of 2500 generations. We investigate the influence on two different scenarios, with different average numbers of mesh points per gateway, and increase the population size from 25 to 200. The results are shown in Figure 4.18.

The fitness values are averaged values of the best individual over ten runs of the genetic algorithm. The runtime shows the minimal total runtime. In Figure 4.18(a) the local optimization only slightly increases the fitness of the best individual. However, in a scenario with a larger number of mesh points, the local optimization increases the fitness between 5 % and 7 % depending on the population size, cf. Figure 4.18(b). The reason is that such a scenario offers more possiblities to assign the routes and channels which are evaluated in the local optimization process.

Taking a look at the population size, we want to point out that the performance increase is only visible up to a population size of 100. When increasing the population size to 200, a run takes twice as long as a run with a population size of 100, while the fitness increases only by 1.5 % at most. Thus, a population size of



(a) Scenario with an average of 18.6 MPs per mesh gateway.

(b) Scenario with an average of 23.6 MPs per mesh gateway.

Figure 4.18: *Relationship between population size, fitness, and runtime.*

100 is a good compromise between the runtime of the genetic algorithm and the fitness of the resulting individuals.

Summarizing, we want to point out that a local optimization of the best individuals is a good means to get to better solutions without significantly prolonging the runtime of the genetic algorithm. A similar result might be achieved after an additional 500 or 1000 generations but this would take much more time. Also the performance increase by enhancing the population size is negligible and almost doubles the runtime.

4.5 Lessons Learned

The complex multi-hop structure of wireless mesh networks induces the need to investigate a large number of network configurations in order to optimize the throughput and to fairly distribute it between the users. In this chapter, we investigated the usability of genetic algorithms for the planning and optimization of wireless mesh networks. The goal of the optimization was to find a near-optimal routing and channel assignment to achieve a max-min fair throughput allocation for the users attached to the wireless mesh points.

The performance of the genetic algorithm depends on the applied fitness function. The fitness function is used to evaluate the resulting network solution. We investigated eight different fitness functions optimizing for example the minimum, mean, and maximum throughput. The results show that the fitness function should be chosen with care because some functions lead to an unfair share of resources. Using a fitness value built on weighted throughputs of all network flows results in the best solutions. In addition to choosing a good fitness function, we illustrated that it is also important to choose the elite set size according to the population size. A small population with a large elite set size often results in a local optimum. The elite set size also has an impact on the required number of generations to get to a good solution. We showed that with an elite set size of one third of the population size, a near-optimal solution is achieved after 400 generations.

Besides the fitness function and the size of the elite set, the genetic operators crossover and mutation have to be carefully applied. We adapted the operators to the requirements of wireless mesh networks and introduced two new crossover variants called Cell and the Subtree Crossover. The evaluation of the influence of these operators revealed that the WMN-specific Cell and Subtree Crossover lead to better solutions compared to the well-known 2-Point Crossover. However, they have to be applied according to the network topology. The Subtree Crossover shows the best performance in scenarios with a large number of mesh points per gateway whereas the Cell Crossover leads to the best solutions in scenarios with a small number of mesh points per gateway.

During the progress of the genetic algorithm, the contribution of the crossover operator to find the optimal solution decreases. After several generation steps, almost no better solutions are achieved by applying the crossover operator. Here, only mutation leads to a better fitness of the solution. We have shown that a reasonable network optimization is only possible by using mutation. The influence of the mutation operator in combination with all crossover types was tested and it was proven that in all cases it strongly fosters the evolution. Even in late generation steps, the fitness of the resulting solution improved.

In order to benefit from the crossover operator to get out of local optima at the beginning of the evolution process and to still get to better solutions at the end of the genetic optimization, we introduced the concept of an elite set increase and a local optimization. With every generation of the genetic algorithm, the elite set is increased which decreases the number of crossover and mutation operations. In order to still mutate the individuals, the mutation operator is applied to the elite set and if a better solution is found, it is taken to the next generation. The local optimization is done after the normal generations procedure finishes. Thereby, several mutations are performed of the five best individuals and the resulting individuals are only kept in the new generation of the local optimization if the fitness value is higher compared to the fitness value before the mutation. Using these concepts, the performance of the WMN can be significantly increased with a minimal computational overhead.

Lessons learned from this chapter are that genetic algorithms are well-suited for the optimization and planning of wireless mesh networks. While other optimization techniques like linear programming fail to optimize large WMNs, genetic algorithms solve the complex structure of WMNs in relatively small computation time. However, the parameters of the genetic algorithm have to be carefully chosen and adapted to the applied topology.

5 Conclusions

> "Begin at the beginning and go on till you come to the end; then stop."
> Lewis Carrol (1832-1898): Alice’s Adventures in Wonderland (1864).

Future broadband wireless networks should be able to support not only best effort traffic but also real-time traffic with strict QoS constraints. In addition, their available resources are scare and limit the number of users. To facilitate QoS guarantees and increase the maximum number of concurrent users, wireless networks require careful planning and optimization.

In this monograph, we studied three aspects of performance optimization in wireless networks: resource optimization in WLAN infrastructure networks, quality of experience control in wireless mesh networks, and planning and optimization of wireless mesh networks.

The contention-based access of WLAN infrastructure networks requires a proper configuration of the channel access parameters. The IEEE 802.11 standard as well as several scientific publications propose a static configuration of these parameters. In order to support real-time traffic, the IEEE 802.11 introduces service differentiation, where each service class has its own set of channel access parameters. However, we showed that resource efficiency severely decreased through the service differentiation extension due to the use of small and static contention windows. As a result, time-varying loads cause heavily varying contention levels, leading to an inefficient channel usage. In the worst case, traffic performance is degraded and QoS requirements cannot be met.

To cope with this problem, we propose a dynamic adaptation of the contention parameters. During runtime, the algorithm adjusts the parameters based on the network load. In contrast to assumptions in the literature, we revealed that this load cannot be accurately estimated at the Access Point. There is a disparity between the load observed at each station and the load measured at the Access Point. Highly loaded and low loaded stations experience different collision probabilities and channel access delays. This unfairness has been shown in a mathematical model and validated by means of simulation. We can handle the unfairness using a feedback mechanism, where all stations transmit their contention status which enables a very accurate assessment of the channel contention level.

To keep the prioritization between different services, the channel access parameters are equally adjusted based on the contention level of the worst station. Thus, the algorithm does not only ensure QoS for real-time services but also increases their capacity. However, low priority best effort traffic suffers from starvation under some conditions. Therefore, we proposed a solution based on frame bursting for best effort traffic to increase its throughput while controlling the reduction of the loss of prioritization for real-time traffic. We showed that real-time traffic is still prioritized over best effort traffic whose throughput is significantly increased. The throughput profits from reduced protocol overhead and reduced contention. Best effort frame bursting can effectively counterbalance the negative impact of contention window prioritization on best effort traffic. Simulations of voice and best effort stations in a WLAN cell showed that increased frame bursts lead to more residual capacity for best effort traffic.

Both mechanisms, the dynamic contention window adaptation and frame bursting of best effort traffic can also be applied for wireless mesh networks. However, these two mechanisms are not sufficient to provide QoS guarantees. Quality problems in wireless mesh networks can occur due to (1) self-interference, (2) interference on neighboring paths, and (3) buffer overloads. In order to encounter all three problems, we introduced a dynamic bandwidth control scheme on the network layer, which is in contrast to our approach for WLAN infrastructure networks not located in the Access Point but distributed on all network nodes. The scheme is based on a monitoring and controlling unit. As both are located on each node of the mesh network, signaling costs and reaction time in case of quality problems can be reduced. Quality problems are solved by limiting the bandwidth of best effort traffic. We measured the performance of the approach in a wireless mesh testbed and later optimized it by means of simulation. In combination with the two mechanisms for WLAN infrastructure networks, the limited wireless resources of a WLAN-based mesh network can be efficiently utilized while the QoS requirements of real-time users can still be kept.

Finally, the complex structure of wireless mesh network does not only require dynamic resource optimization during runtime, but also a careful a priori network planning and optimization. If an operator plans a wireless mesh network, several parameters like routing, number of available channels, number of interfaces per node, number of gateways, and node locations are crucial. Due to this large parameter space, standard optimization techniques like linear programming fail for large wireless mesh networks. We revealed that biology-inspired optimization techniques, namely genetic algorithms, are well-suitable for the planning of wireless mesh networks. We adapted the parameters of the genetic algorithm and introduced new genetic operators which we designed especially for the optimization of wireless mesh networks. Although genetic algorithms generally do not always find the optimal solution, we showed that with a good parameter set for the genetic algorithm, the overall throughput of the wireless mesh network can be significantly improved while still sharing the resources fairly among the users.

In the course of this monograph, we showed that an efficient resource management of wireless networks is essential to provide QoS guarantees for real-time users. Neglecting the performance of best effort users, however, would lead to dissatisfied users. A dynamic control of the access parameters can increase the utilization of the wireless resources successfully, i.e. the overall throughput of best effort users can be enhanced without harm for real-time users. However, the performance of these control mechanisms strongly depends on careful network planning. The planning problem can be solved by applying genetic algorithms which are able to optimize the complex structure of wireless mesh networks in relatively small computation time.

List of Acronyms

AC	Access Category
ACK	Acknowledgment
aCWmax	arbitration Contention Window maximum
aCWmin	arbitration Contention Window minimum
AIFS	Arbitration Interframe Space
AIMD	Additive Increase Multiplicative Decrease
AODV	Ad-hoc On-demand Distance Vector
AP	Access Point
aSIFSTime	arbitration SIFS Time
aSlotTime	arbitration Slot Time
BSS	Basic Service Set
CAP	Channel Access Parameter
CCA	Clear Channel Assessment
CCK	Complimentary Code Keying
CD	Collision Domain
CDF	Cumulative Distribution Function
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DBPSK	Differential Binary Phase Shift Keying

DCF	Distributed Coordination Function
DCWA	Dynamic Contention Window Adaptation
DIFS	Distributed Interframe Space
DSCP	Differentiated Services Code Point
DSDV	Destination Sequence Distance Vector
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
EIFS	Extended Interframe Space
ERP	Extended Rate PHY
ESS	Extended Service Set
FHSS	Frequency-Hopping Spread Spectrum
GA	Genetic Algorithm
HCCA	Hybrid coordination function Controlled Channel Access
HWMP	Hybrid Wireless Mesh Protocol
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
JFI	Jain Fairness Index
LSR	Link State Routing
MAC	Medium Access Control
MAP	Mesh Access Point
MCCA	Mesh Controlled Channel Access
MCS	Modulation and Coding Scheme
MGW	Mesh Gateway
MOS	Mean Opinion Score
MP	Mesh Point
MPR	Multi-Point Relay

MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency-Division Multiplexing
OLSR	Optimized Link State Routing
PBCC	Packet Binary Convolution Coding
PCF	Point Coordination Function
PHY	PHYsical
PIFS	Point (coordination function) Interframe Space
QoE	Quality of Experience
QoS	Quality of Service
RFC	Request for Comments
RTS	Request To Send
SIFS	Short Interframe Space
SNR	Signal to Noise Ratio
TEWMA	Time-Exponentially Weighted Moving Average
TXOP	Transmission Opportunity
TXOP Limit	Transmission Opportunity Limit
VHO	Vertical Handover
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network

Bibliography of the Author

— Journals and Book Chapters —

[1] R. Pries, D. Hock, and D. Staehle, “QoE based bandwidth management supporting real time flows in IEEE 802.11 mesh networks,” *Special Issue PIK on Mobile Ad-hoc-Networks*, January 2010.

[2] F. Wamser, R. Pries, D. Staehle, K. Heck, and P. Tran-Gia, “On traffic characteristics of a broadband wireless internet access,” *Special Issue of the Telecommunication Systems (TS) Journal*, 2010.

[3] P. Tran-Gia, T. Hoßfeld, M. Menth, and R. Pries, “Emerging issues in current future internet design,” *e&i Elektrotechnik und Informationstechnik, Special Issue "Future Internet", ISSN: 0932-383X (print), ISSN: 1613-7620 (online)*, July/August 2009.

— Conference Papers —

[4] D. Staehle, B. Staehle, and R. Pries, “Max-min fair throughput in multi-gateway multi-rate mesh networks,” in *IEEE VTC Spring 10*, (Taipei, Taiwan), May 2010.

[5] B. Staehle, D. Staehle, and R. Pries, “Effects of link rate assignment in IEEE 802.11 mesh networks,” in *16th European Wireless Conference*, (Lucca, Italy), April 2010.

[6] R. Pries, D. Staehle, S. Oechsner, M. Menth, S. Menth, and P. Tran-Gia, “On the unfair channel access phenomenon in wireless LANs,” in *21st International Teletraffic Congress (ITC 21)*, (Paris, France), September 2009.

[7] R. Pries, F. Wamser, D. Staehle, K. Heck, and P. Tran-Gia, “On traffic characteristics of a broadband wireless internet access,” in *Next Generation Internet Networks 2009 (NGI 2009)*, (Aveiro, Portugal), July 2009.

[8] R. Pries, D. Staehle, M. Stoykova, B. Staehle, and P. Tran-Gia, “A genetic approach for wireless mesh network planning and optimization,” in *First International Workshop on Planning and Optimization of Wireless Communication Networks (PlanNet2009) in conjuction with the 5th International Wireless Communications and Mobile Computing Conference (IWCMC)*, (Leipzig, Germany), June 2009.

[9] R. Pries, D. Staehle, M. Stoykova, B. Staehle, and P. Tran-Gia, “Wireless mesh network planning and optimization through genetic algorithms,” in *The Second International Conference on Advances in Mesh Networks, MESH 2009*, (Athens, Greece), June 2009.

[10] R. Pries, F. Wamser, D. Staehle, K. Heck, and P. Tran-Gia, “Traffic measurement and analysis of a broadband wireless internet access,” in *IEEE VTC Spring 09*, (Barcelona, Spain), April 2009.

[11] R. Pries, D. Staehle, S. Menth, M. Menth, and P. Tran-Gia, “Impact of best effort frame bursting in IEEE 802.11 networks,” in *IEEE VTC Spring 09*, (Barcelona, Spain), April 2009.

[12] B. Staehle, D. Staehle, and R. Pries, “Effects of link rate assignment on the max-min fair throughput of wireless mesh networks,” in *21st International Teletraffic Congress (ITC 21)*, (Paris, France), September 2009.

[13] D. Staehle, R. Pries, A. Vinel, and A. Mäder, “Performance evaluation and parametrization of the IEEE 802.16 contention-based cdma bandwidth request mechanism for the OFDMA physical layer,” in *12th ACM-IEEE International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, (Tenerife, Canary Islands, Spain), October 2009.

[14] B. Staehle, D. Staehle, R. Pries, M. Hirth, P. Dely, and A. J. Kassler, “Measuring one-way delay in wireless mesh networks - an experimental investigation,” in *The 4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks (PM2HW2N)*, (Tenerife, Canary Island, Spain), October 2009.

[15] R. Pries, D. Staehle, and D. Marsico, “IEEE 802.16 capacity enhancement using an adaptive TDD split,” in *IEEE VTC Spring 08*, (Singapore), May 2008.

[16] R. Pries, D. Staehle, P. Tran-Gia, and T. Gutbrod, “A seamless vertical handover approach,” in *Wireless Systems and Mobility in Next Generation Internet* (Springer, ed.), Springer, 2008.

[17] R. Pries, S. Menth, D. Staehle, M. Menth, and P. Tran-Gia, “Dynamic contention window adaptation (DCWA) in IEEE 802.11e wireless local area networks,” in *HUT-ICCE*, (Hoi An, Vietnam), June 2008.

[18] R. Pries, D. Hock, N. Bayer, M. Siebert, D. Staehle, V. Rakocevic, B. Xu, and P. Tran-Gia, “Dynamic bandwidth control in wireless mesh networks: A quality of experience based approach,” in *18th ITC Specialist Seminar on Quality of Experience*, (Karlskrona, Sweden), May 2008.

[19] D. Hock, N. Bayer, R. Pries, M. Siebert, D. Staehle, V. Rakocevic, and B. Xu, “QoS provisioning in WLAN mesh networks using dynamic band-

width control," in *European Wireless 2008*, (Prague, Czech Republic), June 2008.

[20] R. Pries, A. Mäder, D. Staehle, and M. Wiesen, "On the performance of mobile IP in wireless LAN environments," in *Wireless Systems and Mobility in Next Generation Internet* (Springer, ed.), Springer, 2007.

[21] R. Pries, D. Staehle, and D. Marsico, "Performance evaluation of piggyback requests in IEEE 802.16," in *IEEE VTC Fall 07*, (Baltimore, MD, USA), October 2007.

[22] D. Staehle and R. Pries, "Comparative study of the IEEE 802.16 random access mechanisms," in *Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST*, (Cardiff, UK), pp. 334–339, September 2007.

[23] A. Vinel, D. Staehle, and R. Pries, "Random multiple access in WiMAX: Problems and solutions," in *1st Workshop on WiMAX Wireless and Mobility (WEIRD) in conjunction with the WWIC 2007 Conference*, (Coimbra, Portugal), May 2007.

[24] A. Vinel, D. Staehle, R. Pries, and Q. Ni, "Performance analysis of the polling scheme in IEEE 802.16," in *Distributed Computer and Communication Networks (DCCN-2007): Theory and Applications*, (Moskow, Russia), September 2007.

[25] A. Mäder, D. Staehle, R. Pries, and M. Spahn, "Impact of HSDPA transmit power allocation schemes on the performance of UMTS networks," in *Australasian Telecommunication Networks and Applications Conference (ATNAC) 2007*, (Christchurch, New Zealand), December 2007.

[26] R. Pries, A. Mäder, and D. Staehle, "A network architecture for a policy-based handover across heterogeneous networks," in *OPNETWORK 2006*, (Washington D.C., USA), August 2006.

[27] R. Pries, A. Mäder, and D. Staehle, “Do we need header compression for VoIP in wireless LANs?,” in *12th EUNICE Open European Summer School 2006*, (Stuttgart, Germany), September 2006.

[28] R. Pries, T. Hoßfeld, and P. Tran-Gia, “On the suitability of the short message service for emergency warning systems,” in *IEEE VTC Spring 06*, (Melbourne, Australia), May 2006.

[29] R. Pries, K. Heck, P. Tran-Gia, and T. Wirth, “QoS traffic in wireless LAN overlapping cells,” in *European Wireless 2006*, (Athens, Greece), April 2006.

[30] A. Klein, R. Pries, and D. Staehle, “Performance study of the WiMAX FDD mode,” in *OPNETWORK 2006*, (Washington D.C., USA), August 2006.

[31] R. Pries and K. Heck, “Simulative study of the WLAN handover performance,” in *OPNETWORK 2005*, (Washington D.C., USA), August 2005.

[32] K. Heck, R. Pries, and T. Wirth, “Simulative study of the RTS/CTS solution to the hidden node problem in infrastructure wireless LANs,” in *5th World Wireless Congress (WWC 04)*, (San Francisco, CA, USA), May 2004.

[33] R. Pries and K. Heck, “Performance comparison of handover mechanisms in wireless LAN networks,” in *Australasian Telecommunication Networks and Applications Conference (ATNAC) 2004*, (Sydney, Australia), December 2004.

[34] T. Hoßfeld, K. Leibnitz, R. Pries, K. Tutschku, P. Tran-Gia, and K. Pawlikowski, “Information diffusion in eDonkey filesharing networks,” in *Australasian Telecommunication Networks and Applications Conference (ATNAC) 2004*, (Sydney, Australia), December 2004.

General References

[35] IEEE Std. 802.11-2007, “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” June 2007.

[36] IEEE Std. 802.11-1997, “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” 1997. ISO/IEC 8802-11:1997.

[37] IEEE Std. 802.11b-1999, “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher speed physical layer (PHY) extension in the 2.4 GHz band,” 1999. IEEE 802.11b-1999.

[38] IEEE Std. 802.11a-1999, “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed physical layer in the 5 GHz band,” 1999. IEEE 802.11a-1999.

[39] IEEE Std. 802.11g-2003, “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - amendment 4: Further higher data rate extension in the 2.4 GHz band,” 2003. IEEE 802.11g-2003.

[40] IEEE 802.11e, “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 8: Medium access control (MAC) quality of service enhancements,” November 2005. IEEE 802.11e-2005.

[41] IEEE Std. 802.1D, “IEEE standard for local and metropolitan area networks: Medium access control (MAC) bridges,” 2004. IEEE 802.1D.

[42] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, Inc., April 1991.

[43] Dah-Ming and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, 1989.

[44] R. Jain, D.-M. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared systems,” Tech. Rep. DEC-TR-301, Digital Equipment Corporation, September 1984.

[45] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, “Achieving MAC layer fairness in wireless packet networks,” in *MobiCom ’00: Proceedings of the 6th annual international conference on Mobile computing and networking*, (Boston, MA, USA), pp. 87–98, ACM Press, August 2000.

[46] T. Ozugur, M. Naghshineh, P. Kermani, C. M. Olsen, B. Rezvani, and J. A. Copeland, “Balanced media access methods for wireless networks,” in *MobiCom ’98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, (Dallas, TX, USA), pp. 21–32, 1998.

[47] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: a media access protocol for wireless LAN’s,” in *SIGCOMM ’94: Proceedings of the conference on Communications architectures, protocols and applications*, (London, United Kingdom), pp. 212–225, 1994.

[48] C. E. Koksal, H. Kassab, and H. Balakrishnan, “An analysis of short-term fairness in wireless media access protocols (poster session),” in *SIGMETRICS ’00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (Santa Clara, CA, USA), pp. 118–119, ACM, June 2000.

[49] C. L. Barrett, M. V. Marathe, D. C. Engelhart, and A. Sivasubramaniam, “Analyzing the short-term fairness of IEEE 802.11 in wireless multi-hop radio networks,” in *10th IEEE International Symposium on Modeling,*

Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002., (Fort Worth, TX, USA), pp. 137–144, October 2002.

[50] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Mobile Computing and Networking*, (Boston, MA, USA), pp. 167–178, August 2000.

[51] Y. Kwon, Y. Fang, and H. Latchman, "A novel MAC protocol with fast collision resolution for wireless LANs," in *IEEE Infocom 2003*, (San Francisco, CA, USA), pp. 853–862, March-April 2003.

[52] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau, "Fairness and its impact on delay in 802.11 networks," in *IEEE Globecom 2004*, (Dallas, TX, USA), p. 7121, December 2004.

[53] S. W. Kim, B.-S. Kim, and Y. Fang, "Downlink and uplink resource allocation in IEEE 802.11 wireless LANs," *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 320–327, January 2005.

[54] Y. Fukuda and Y. Oie, "Unfair and inefficient share of wireless LAN resource among uplink and downlink data traffic and its solution (wireless communication technologies)," *IEICE Transactions on Communications*, vol. E88-B, no. 4, pp. 1577–1585, 2005.

[55] C. Casetti and C.-F. Chiasserini, "Improving fairness and throughput for voice traffic in 802.11e EDCA," in *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004.*, (Barcelona, Spain), pp. 525–530, September 2004.

[56] A. Grilo and M. Nunes, "Performance evaluation of IEEE 802.11e," in *IEEE PIMRC'02*, (Lisboa, Portugal), September 2002.

[57] B. A. H. S. Abeysekera, T. Matsuda, and T. Takine, "Dynamic contention window control to achieve fairness between uplink and downlink

in IEEE 802.11 WLANs," in *IEEE Wireless Communications and Networking Conference (WCNC) 2007*, (Hong Kong, China), pp. 2109–2114, March 2007.

[58] B. A. H. S. Abeysekera, T. Matsuda, and T. Takine, "Dynamic contention window control mechanism to achieve fairness between uplink and downlink flows in IEEE 802.11 wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 3517–3525, September 2008.

[59] B. A. H. S. Abeysekera, T. Matsuda, and T. Takine, "Dynamic contention window control scheme in IEEE 802.11e wireless LANs," in *IEEE VTC Spring 09*, (Barcelona, Spain), April 2009.

[60] Z. Fang, B. Bensaou, and Y. Wang, "Performance evaluation of a fair backoff algorithm for IEEE 802.11 DFWMAC," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, (Lausanne, Switzerland), pp. 48–57, ACM, 2002.

[61] M. Malli, Q. Ni, T. Turletti, and C. Barakat, "Adaptive fair channel allocation for QoS enhancement in IEEE 802.11 wireless LANs," in *IEEE International Conference on Communications (ICC)*, (Paris, France), pp. 3470–3475, June 2004.

[62] E.-C. Park, D.-Y. Kim, and C.-H. Choi, "Analysis of unfairness between TCP uplink and downlink flows in Wi-Fi hot spots," in *IEEE Globecom 2006*, (San Francisco, CA, USA), November 2006.

[63] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over wireless LAN," in *IEEE Infocom 2003*, (San Francisco, CA, USA), pp. 863–872, April 2003.

[64] M. Thottan and M. C. Weigle, "Impact of 802.11e EDCA on mixed TCP-based applications," in *WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet*, (Boston, MA, USA), August 2006.

[65] N. Blefari-Melazz, A. Detti, I. Habib, A. Ordine, and S. Salsano, "TCP fairness issues in IEEE 802.11 networks: Problem analysis and solutions based on rate control," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 1346–1355, April 2007.

[66] D. Leith and P. Clifford, "Using the 802.11e EDCF to achieve TCP upload fairness over WLAN links," in *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2005.

[67] D. J. Leith, P. Clifford, D. Malone, and A. Ng, "TCP fairness in 802.11e WLANs," *IEEE Communications Letters*, vol. 9, pp. 964–966, November 2005.

[68] Y. Jian and S. Chen, "Can CSMA/CA networks be made fair?," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, (San Francisco, CA, USA), pp. 235–246, September 2008.

[69] F. Keceli, I. Inan, and E. Ayanoglu, "Achieving fair TCP access in the IEEE 802.11 infrastructure basic service set," in *IEEE International Conference on Communications ICC*, (Beijing, China), pp. 2637–2643, May 2008.

[70] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–81, 1951.

[71] J. Jeong, S. Choi, and C. kwon Kim, "Achieving weighted fairness between uplink and downlink in IEEE 802.11 DCF-based WLANs," in *Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, IEEE QShine*, (Lake Vista, FL, USA), August 2005.

[72] Q. Wu, M. Gong, and C. Williamson, "TCP fairness issues in IEEE 802.11 wireless LANs," *Computer Communications*, vol. 31, no. 10, pp. 2150–2161, 2008.

[73] C. Williamson and Q. Wu, “A case for context-aware TCP/IP,” *SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 4, pp. 11–23, 2002.

[74] E. W. Biersack, B. Schroeder, and G. Urvoy-Keller, “Scheduling in practice,” *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 4, pp. 21–28, 2007.

[75] OPNET Modeler, OPNET University Program:
http://www.opnet.com/services/university/.

[76] ITU-T Recommendation G.711, “Pulse code modulation (PCM) of voice frequencies.” International Telecommunication Union, November 1998.

[77] International Telecommunication Union, “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (cs-acelp).” Recommendation G.729, Telecommunication Standardization Sector of ITU, March 1996.

[78] C. Demichelis and P. Chimento, “IP packet delay variation metric for IP performance metrics (IPPM).” RFC 3393, `http://www.ietf.org/rfc/rfc3393.txt`, November 2002.

[79] L. Romdhani, Q. Ni, and T. Turletti, “AEDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, (New Orleans, LA, USA), March 2003.

[80] V. Scarpa, G. Convertino, A. Palmieri, and J. C. D. Martin, “Adaptive techniques to guarantee QoS in a IEEE 802.11 wireless LAN,” in *IEEE VTC Spring 04*, (Milan, Italy), pp. 3014 – 3018, May 2004.

[81] L. Zhang and S. Zeadally, “A framework for efficient resource management in IEEE 802.11 WLANs: research articles,” *Wirel. Commun. Mob. Comput.*, vol. 4, no. 8, pp. 835–848, 2004.

[82] L. Scalia, I. Tinnirello, J. W. Tantra, and C. H. Foh, “Dynamic MAC parameters configuration for performance optimization in 802.11e networks,” in *IEEE Globecom 2006*, (San Francisco, CA, USA), November 2006.

[83] A. Nafaa, A. Ksentini, and A. Mehaoua, “SCW: sliding contention window for efficient service differentiation in IEEE 802.11 networks,” in *IEEE WCNC 2005 The IEEE Wireless Communications and Networking Conference 2005*, (New Orleans, LA, USA), March 2005.

[84] S. E. Housseini and H. Alnuweiri, “Adaptive contention-window MAC algorithms for QoS-enabled wireless LANs,” in *International Conference on Wireless Networks, Communications, and Mobile Computing*, (Maui, HI, USA), June 2005.

[85] L. Gannoune, “A comparative study of dynamic adaptation algorithms for enhanced service differentiation in IEEE 802.11 wireless ad hoc networks,” in *AICT-ICIW ’06: Proceedings of the Advanced Int’l Conference on Telecommunications and Int’l Conference on Internet and Web Applications and Services*, (Guadeloupe, French Carribean), February 2006.

[86] F. Cacace, G. Iannello, M. Vellucci, and L. Vollero, “A reactive approach to QoS provisioning in IEEE 802.11e WLANs,” in *NGI 2008, 4th EURO-NGI Conference on Next Generation Internet Networks*, (Krakow, Poland), April 2008.

[87] F. Cali, M. Conti, and E. Gregori, “Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit,” *IEEE/ACM Transactions on Networking (TON)*, vol. 8, pp. 785–799, December 2000.

[88] F. Cali, M. Conti, and E. Gregori, “IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism,” *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1774–1786, September 2000.

[89] A. L. Toledo, T. Vercauteren, and X. Wang, "Adaptive optimization of IEEE 802.11 DCF based on bayesian estimation of the number of competing terminals," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1283–1296, 2006.

[90] Y. Ge, J. C. Hou, and S. Choi, "An analytic study of tuning systems parameters in IEEE 802.11e enhanced distributed channel access," *Computer Networks*, vol. 51, no. 8, pp. 1955–1980, 2007.

[91] P. Gopalakrishnan, D. Famolari, and T. Kodama, "Improving WLAN voice capacity through dynamic priority access," in *IEEE Globecom 2004*, (Dallas, TX, USA), pp. 3245– 3249, December 2004.

[92] M. Menth, J. Milbrandt, and J. Junker, "Time-exponentially weighted moving histograms (TEWMH) for self-adaptive systems," in *IEEE Globecom 2006*, (San Francisco, CA, USA), November 2006.

[93] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama, "Voice capacity of IEEE 802.11b, 802.11a and 802.11g wireless LANs," in *IEEE Globecom 2004*, (Dallas, TX, USA), November/December 2004.

[94] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama, "Voice capacity of IEEE 802.11b, 802.11a and 802.11g wireless LANs in the presence of channel errors and different user data rates," in *IEEE VTC Fall 2004*, (Los Angeles, CA, USA), pp. 4543–4547, September 2004.

[95] S. Garg and M. Kappes, "Admission control for VoIP traffic in IEEE 802.11 networks," in *Global Telecommunications Conference, Globecom 2003*, vol. 6, (San Francisco, CA, USA), pp. 3514–3518, December 2003.

[96] S. Garg and M. Kappes, "An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks," in *Wireless Communication and Networking, WCNC*, vol. 3, (New Orleans, LA, USA), pp. 1748–1753, March 2003.

[97] D. Chen, S. Garg, M. Kappes, and K. S. Trivedi, “Supporting VoIP traffic in IEEE 802.11 WLAN with enhanced medium access control (MAC) for quality of service,” in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2003)*, (Montreal, Canada), July 2003.

[98] S. Garg and M. Kappes, “Can I add a VoIP call?,” in *International Conference on Communications (ICC 2003)*, (Anchorage, AK, USA), May 2003.

[99] F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, and P. Agrawal, “Voice performance in WLAN networks - an experimental study,” in *Global Telecommunications Conference, Globecom 2003*, vol. 6, (San Francisco, CA, USA), December 2003.

[100] M. Coupechoux, V. Kumar, and L. Brignol, “Voice over IEEE 802.11b capacity,” in *Proc. of the 16th ITC Specialist Seminar*, (Antwerp, Belgium), September 2004.

[101] M. Veeraraghavan, N. Cocker, and T. Moors, “Support of voice services in IEEE 802.11 wireless LANs,” in *IEEE Infocom 2001*, vol. 2, (Anchorage, AK, USA), pp. 488–497, April 2001.

[102] Y. P. Fallah and H. M. Alnuweiri, “Modeling and performance evaluation of frame bursting in wireless LANs,” in *IWCMC ’06: Proceeding of the 2006 International Conference on Communications and Mobile Computing*, (New York, NY, USA), pp. 869–874, ACM Press, July 2006.

[103] J. Hu, G. Min, and M. E. Woodward, “Analysis and comparison of burst transmission schemes in unsaturated 802.11e WLANs,” in *IEEE Globecom 2007*, (Washington D.C., USA), pp. 5133–5137, November 2007.

[104] F. Peng, H. M. Alnuweiri, and V. C. Leung, “Analysis of burst transmission in IEEE 802.11e wireless LANs,” in *IEEE International Conference on Communications (ICC)*, (Istanbul, Turkey), pp. 535–539, June 2006.

[105] L. Romdhani and C. Bonnet, “Performance analysis and optimization of the 802.11e EDCA transmission opportunity (TXOP) mechanism,” in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB)*, no. 3, (White Plains, NY, USA), October 2007.

[106] S. Selvakennedy, “The impact of transmit buffer on EDCA with frame-bursting option for wireless networks,” in *29th Annual IEEE International Conference on Local Computer Networks (LCN’04)*, (Washington D.C., USA), pp. 696–697, November 2004.

[107] S. Selvakennedy, “The influence of MAC buffer on the contention-based access scheme with bursting option for IEEE 802.11e wireless networks,” *Journal of Engineering Science and Technology (JESTEC)*, vol. 1, pp. 119–138, December 2006.

[108] J. Majkowski and F. C. Palacio, “Dynamic TXOP configuration for QoS enhancement in IEEE 802.11e wireless LAN,” in *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, (Split/Dubrovnik, Croatia), September 2006.

[109] J. Majkowski and F. C. Palacio, “Enhancement TXOP scheme for efficiency improvement of WLAN IEEE 802.11e,” in *IEEE VTC Fall 06*, (Montreal, Canada), September 2006.

[110] H. Liu and Y. Zhao, “Adaptive EDCA algorithm using video prediction for multimedia IEEE 802.11e WLAN,” in *IEEE International Conference on Wireless and Mobile Communications (ICWMC’06) collocated with ICCGI’06*, (Bucharest, Romania), July 2006.

[111] N. Cranley, T. Debnath, and M. Davis, “An experimental investigation of parallel multimedia streams over IEEE 802.11e WLAN networks using TXOP,” in *IEEE International Conference on Communications, ICC 2007*, (Glasgow, Scotland), pp. 1740–1746, June 2007.

[112] M. L. Huang, S. Lee, and S.-C. Park, "An adaptive EDCA TXOP with rate adaptation for QoS provision," in *Proceedings of the 6th WSEAS International Conference on Instrumentation, Measurement, Circuits & Systems*, (Hangzhou, China), April 2007.

[113] A. Andreadis and R. Zambon, "QoS enhancement for multimedia traffics with dynamic TXOPLimit in IEEE 802.11e," in *Q2SWinet '07: Proceedings of the 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks*, (New York, NY, USA), pp. 16–22, ACM, October 2007.

[114] A. Andreadis and R. Zambon, "QoS enhancement with dynamic TXOP allocation in IEEE 802.11e," in *Personal, Indoor and Mobile Radio Communications (PIMRC)*, no. 18, (Athens, Greece), September 2007.

[115] A. Ksentini, A. Guéroui, and M. Naimi, "Adaptive transmission opportunity with admission control for IEEE 802.11e networks," in *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, (Montréal, Canada), pp. 234–241, ACM Press, 2005.

[116] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, pp. 23–30, September 2005.

[117] IEEE 802.11s/D3.02, "Draft amendment to standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - ESS mesh networking," May 2009. IEEE 802.11s/D3.02.

[118] IEEE Std. 802.15.5, "IEEE 802.15 standard group web site. available from: http://www.ieee802.org/15/."

[119] IEEE 802.16-2004, "IEEE standards for local and metropolitan area networks - part16: Air interface for fixed broadband wireless access systems," 2004. IEEE 802.16-2004.

[120] M. J. Lee, J. Zheng, Y.-B. Ko, and D. M. Shrestha, “Emerging standards for wireless mesh technology,” *IEEE Wireless Communications*, vol. 13, pp. 56–63, April 2006.

[121] Y. Koucheryavy, G. Giambene, D. Staehle, F. Barcelo-Arroyo, T. Braun, and V. Siris, *Traffic and QoS Management in Wireless Multimedia Networks, COST 290 Final Report*. Springer, May 2009.

[122] ITU-T, “Series P: telephone transmission quality, methods for objective and subjective assessment of quality; mean opinion score (MOS) terminology,” July 2006. ITU-T Recommendation P.800.1.

[123] I. F. Akyildiz and X. Wang, *Wireless Mesh Networks*. Advanced Texts in Communications and Networking, Wiley, John & Sons, Incorporated, 2009.

[124] S. M. Faccin, C. Wijting, J. Kneckt, and A. Damle, “Mesh WLAN networks: Concept and system design,” *IEEE Wireless Communications*, vol. 13, pp. 10–17, April 2006.

[125] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR).” RFC 3626, `http://www.ietf.org/rfc/rfc3626.txt`, October 2003.

[126] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *SIGCOMM ’94: Proceedings of the conference on Communications architectures, protocols and applications*, (London, United Kingdom), pp. 234–244, ACM Press, 1994.

[127] D. B. Johnson, “Routing in ad hoc networks of mobile hosts,” in *Workshop on Mobile Computing Systems and Applications*, (Santa Cruz, CA, USA), pp. 158–163, December 1994.

[128] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 111–116, ACM, 2007.

[129] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *2nd IEEE Workshop on Mobile Computing Systems and Applications*, (New Orleans, LA, USA), pp. 90–100, February 1999.

[130] C. Perkins and E. Belding-Royer, "Ad-hoc on-demand distance vector (AODV) routing." RFC 2460, `http://www.ietf.org/rfc/rfc2460.txt`, July 2003.

[131] A. Tønnesen, "Implementing and extending the optimized link state routing protocol," Master's thesis, UniK University Graduate Center, University of Oslo, 2004.

[132] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov, "Performance of VoIP in a IEEE 802.11 wireless mesh network," in *25th IEEE International Conference on Computer Communications (Infocom)*, (Barcelona, Spain), April 2006.

[133] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. K. Costa, O. C. M. 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*, vol. 22, pp. 6–12, January/February 2008.

[134] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *ACM SIGCOMM*, (Portland, OR, USA), September 2004.

[135] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *ACM MobiCom*, no. 10, (Philadelphia, PA, USA), pp. 114–128, September 2004.

[136] W. Cordeiro, E. Aguiar, W. Moreira, A. Abelem, and M. Stanton, “Providing quality of service for mesh networks using link delay measurements,” in *ICCCN’07*, (Honolulu, HI, USA), August 2007.

[137] E. Carlson, C. Prehofer, C. Bettstetter, H. Karl, and A. Wolisz, “A distributed end-to-end reservation protocol for IEEE 802.11-based wireless mesh networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 2018–2027, November 2006.

[138] V. Kone, S. Das, B. Y. Zhao, and H. Zheng, “QUORUM - quality of service in wireless mesh networks,” *Mobile Networks and Applications*, vol. 12, pp. 358–369, December 2007.

[139] Q. Xue and A. Ganz, “QoS routing for mesh-based wireless lans,” *International Journal of Wireless Information Networks*, vol. 9, pp. 179–190, July 2002.

[140] S. Marwaha, J. Indulska, and M. Portmann, “Challenges and recent advances in QoS provisioning in wireless mesh networks,” in *8th IEEE International Conference on Computer and Information Technology (CIT)*, (Sydney, Australia), July 2008.

[141] J. R. Gallardo, D. Makrakis, and H. T. Mouftah, “MARE: an efficient reservation-based MAC protocol for IEEE 802.11s mesh networks,” in *Second International Conference on Advances in Mesh Networks*, (Glyfada, Greece), June 2009.

[142] H. Jiang, W. Zhuang, X. Shen, A. Abdrabou, and P. Wang, “Differentiated services for wireless mesh backbone,” *IEEE Communications Magazine*, vol. 44, pp. 113–119, July 2006.

[143] G. R. Hiertz, S. Max, T. Junge, D. Denteneer, and L. Berlemann, “IEEE 802.11s – mesh deterministic access,” in *Proceedings of the 14th European*

Wireless Conference – EW 2008, (Prague, Czech Republic), IEEE, June 2008.

[144] K. Kim, S. Ganguly, R. Izmailov, and S. Hong, "On packet aggregation mechanisms for improving VoIP quality in mesh networks," in *IEEE VTC Spring 06*, vol. 2, (Melbourne, Australia), May 2006.

[145] H. Wei, K. Kim, A. Kashyap, and S. Ganguly, "On admission of VoIP calls over wireless mesh network," in *IEEE International Conference on Communications (ICC)*, vol. 5, (Istanbul, Turkey), pp. 1990–1995, June 2006.

[146] K. Kim, D. Niculescu, and S. Hong, "Coexistence of VoIP and TCP in wireless multihop networks," *IEEE Communications Magazine*, vol. 47, pp. 75–81, June 2009.

[147] W. He, H. Nguyen, and K. Nahrstedt, "Experimental validation of middleware-based QoS control in 802.11 wireless networks," in *Third International Conference on Broadband Networks (BROADNETS '06)*, (San Jose, CA, USA), October 2006.

[148] P. S. Mogre, M. Hollick, and R. Steinmetz, "QoS in wireless mesh networks: Challenges, pitfalls, and roadmap to its realization," in *17th International workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, (Urbana-Champaign, IL, USA), June 2007.

[149] N. Bayer, M. C. de Castro, P. Dely, A. Kassler, Y. Koucheryavy, P. Mitoraj, and D. Staehle, "VoIP service performance optimization in Pre-IEEE 802.11s wireless mesh networks," in *IEEE ICCSC*, (Shanghai, China), May 2008.

[150] T. Hoßfeld, P. Tran-Gia, and M. Fiedler, "Quantification of quality of experience for edge-based applications," in *20th International Teletraffic Congress (ITC20)*, (Ottawa, Canada), June 2007.

[151] T. Hoßfeld, D. Hock, K. Tutschku, P. Tran-Gia, and M. Fiedler, "Testing the IQX hypothesis for exponential interdependency between QoS and QoE of voice codecs iLBC and G.711," in *18th ITC Specialist Seminar on Quality of Experience*, (Karlskrona, Sweden), May 2008.

[152] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning." International Telecommunication Union, March 2005.

[153] R. Cole and J. Rosenbluth, "Voice over IP performance monitoring," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2, pp. 9–24, 2001.

[154] ITU-T Recommendation G.133 Appendix I, "Transmission impairments due to speech processing, appendix I: Provisional planning values for the equipment impairment factor Ie and packet-loss robustness factor Bpl." International Telecommunication Union, May 2002.

[155] D. Staehle, B. Staehle, and R. Pries, "Max-min fair throughput in multi-gateway multi-rate mesh networks," Tech. Rep. 454, University of Würzburg, January 2009.

[156] D. P. Bertsekas and R. G. Gallager, *Data networks*. Prentice-Hall, 1987.

[157] K. Tutschku, "Demand-based radio network planning of cellular communication systems," in *IEEE Infocom 1998*, no. 17, (San Francisco, CA, USA), March 1998.

[158] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986.

[159] S. Sen and B. Raman, "Long distance wireless mesh network planning: problem formulation and solution," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*, (New York, NY, USA), pp. 893–902, ACM, 2007.

[160] B. He, B. Xie, and D. P. Agrawal, “Optimizing deployment of internet gateway in wireless mesh networks,” *Computer Communications*, vol. 31, no. 7, pp. 1259–1275, 2008.

[161] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, “Optimization models and methods for planning wireless mesh networks,” *Computer Networks*, vol. 52, pp. 2159–2171, August 2008.

[162] A. So and B. Liang, “Minimum cost configuration of relay and channel infrastructure in heterogeneous wireless mesh networks,” in *Networking*, (Atlanta, GA, USA), pp. 275–286, May 2007.

[163] A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, pp. 50–65, April 2004.

[164] A. Raniwala and T. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *IEEE Infocom 2005*, vol. 3, (Miami, FL, USA), pp. 2223–2234, March 2005.

[165] Y.-Y. Chen, S.-C. Liu, and C. Chen, “Channel assignment and routing for multi-channel wireless mesh networks using simulated annealing,” in *IEEE Globecom 2006*, (San Francisco, CA, USA), November/December 2006.

[166] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, “Interference-aware channel assignment in multi-radio wireless mesh networks,” in *IEEE Infocom 2006*, (Barcelona, Spain), April 2006.

[167] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, “Minimum interference channel assignment in multiradio wireless mesh networks,” *IEEE Transactions on Mobile Computing*, vol. 7, pp. 1459–1473, December 2008.

[168] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, (Cologne, Germany), pp. 58–72, ACM, 2005.

[169] A. H. Mohsenian Rad and V. W. S. Wong, "Joint channel allocation, interface assignment and MAC design for multi-channel wireless mesh networks," in *IEEE Infocom 2007*, (Anchorage, AK, USA), pp. 1469–1477, May 2007.

[170] A. H. Mohsenian Rad and V. W. S. Wong, "Congestion-aware channel assignment for multi-channel wireless mesh networks," *Computer Networks*, vol. 53, pp. 2502–2516, September 2009.

[171] P. Calégari, F. Guidec, P. Kuonen, and D. Wagner, "Genetic approach to radio network optimization for mobile systems," in *IEEE VTC Spring 97*, (Phoenix, AZ, USA), May 1997.

[172] S. Ghosh, P. Ghosh, K. Basu, and S. K. Das, "GaMa: an evolutionary algorithmic approach for the design of mesh-based radio access networks," in *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, (Washington D.C., USA), pp. 374–381, IEEE Computer Society, November 2005.

[173] L. Badia, A. Botta, and L. Lenzini, "A genetic approach to joint routing and link scheduling for wireless mesh networks," *Elsevier Ad Hoc Networks Journal*, vol. Special issue on Bio-Inspired Computing, p. 11, April 2008.

[174] T. Vanhatupa, M. Hännikäinen, and T. D. Hämäläinen, "Performance model for IEEE 802.11s wireless mesh network deployment design," *Journal of Parallel and Distributed Computing*, vol. 68, pp. 291–305, March 2008.

[175] T. Vanhatupa, M. Hännikäinen, and T. D. Hämäläinen, "Genetic algorithm to optimize node placement and configuration for WLAN planning," in *4th International Symposium on Wireless Communication Systems, 2007. ISWCS 2007*, (Trondheim, Norway), pp. 612–616, October 2007.

[176] E. Damosso and L. M. Correia, *Digital Mobile Radio Towards Future Generation Systems, COST 231 Final Report*. European Commission, 1999.

[177] J. Jun and M. L. Sichitiu, "The nominal capacity of wireless mesh networks," *IEEE Communications Magazine*, vol. 10, pp. 8–14, October 2003.

[178] B. Aoun and R. Boutaba, "Max-min fair capacity of wireless mesh networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, (Vancouver, Canada), pp. 21–30, October 2006.

[179] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: University of Michigan Press, 1975.

ISSN 1432-8801