



Architectures and Algorithms for Future Wireless Local Area Networks

Peter Dely

Faculty of Economic Sciences, Communication and IT

Computer Science

DISSERTATION | Karlstad University Studies | 2012:53

Architectures and Algorithms for Future Wireless Local Area Networks

Peter Dely

Architectures and Algorithms for Future Wireless Area Networks

Peter Dely

DISSERTATION

Karlstad University Studies | 2012:53

ISSN 1403-8099

ISBN 978-91-7063-464-2

© The author

Distribution:

Karlstad University

Faculty of Economic Sciences, Communication and IT

Computer Science

SE-651 88 Karlstad, Sweden

+46 54 700 10 00

Print: Universitetstryckeriet, Karlstad 2012

WWW.KAU.SE

Architectures and Algorithms for Future Wireless Local Area Networks

PETER DELY

Department of Computer Science, Karlstad University

Abstract

Future Wireless Local Area Networks (WLANs) with high carrier frequencies and wide channels need a dense deployment of Access Points (APs) to provide good performance. In densely deployed WLANs associations of stations and handovers need to be managed more intelligently than today.

This dissertation studies when and how a station should perform a handover and to which AP from a theoretical and a practical perspective. We formulate and solve optimization problems that allow to compute the optimal AP for each station in normal WLANs and WLANs connected via a wireless mesh backhaul. Moreover, we propose to use software defined networks and the OpenFlow protocol to optimize station associations, handovers and traffic rates. Furthermore, we develop new mechanisms to estimate the quality of a link between a station and an AP. Those mechanisms allow optimization algorithms to make better decisions about when to initiate a handover. Since handovers in today's WLANs are slow and may disturb real-time applications such as video streaming, a faster procedure is developed in this thesis.

Evaluation results from wireless testbeds and network simulations show that our architectures and algorithms significantly increase the performance of WLANs, while they are backward compatible at the same time.

Keywords: WLAN, IEEE 802.11, network management, software defined networking, OpenFlow, optimization, handover, mobility

Acknowledgments

The thesis you are just reading would not exist without the help from many people that supported me throughout the years of research and writing.

First of all, I would like to take the opportunity to thank my advisor Andreas Kassler, who, like a good sports coach, helped to improve my work through his advice, challenging questions and inspirational ideas. Without his continuous support all this would have not been possible. Thank you.

Furthermore, I would like to express my gratitude to my colleagues and friends from the Computer Science department at Karlstad University, in particular the Distributed Systems and Communications Research (DISCO) group and my co-supervisor Anna Brunström. Their support as well as the enlightening and sometimes funny or even absurd discussions at the coffee table made Karlstad to a great place to work and live.

Also, I am grateful that Vasilios A. Siris accepted to take the role as opponent and that the members of the examination committee took the burden to travel to Karlstad during the dark winter months.

I would like to thank Deutsche Telekom Innovation Laboratories, in particular Hans Einsiedler, Nico Bayer and Christoph Peylo for their financial and technical support. In addition, some work presented in this thesis was financially supported by the Interreg IVB North Sea Region project E-CLIC and by STINT (Stiftelsen för internationalisering av högre utbildning och forskning), which I really appreciate.

Huge thanks go to my parents and my family for their support and encouragement throughout all the long years of study. Last but not least, I am indebted to my lovely wife Yao Qin, whose love, patience and understanding keeps me motivated from morning to evening.

A handwritten signature in black ink, appearing to read 'Peter Zeh', with a stylized flourish at the bottom.

Karlstad, October 2012

Publications

This dissertation is based on the work presented in the following publications:

- I. **Peter Dely**, Andreas Kassler, Dmitry Sivchenko. “Theoretical and Experimental Analysis of the Channel Busy Fraction in IEEE 802.11”, In *Proceedings of Future Network & Mobile Summit 2010*, Florence, Italy, June 2010.
- II. **Peter Dely**, Andreas Kassler, Nico Bayer, Hans Einsiedler and Christoph Peylo. “Optimization of WLAN associations considering handover costs”, In *EURASIP Journal on Wireless Communications and Networking*, 2012, 2012:255.
- III. **Peter Dely**, Fabio D’Andreagiovanni, Andreas Kassler. “Fair Optimization of Mesh-Connected WLAN Hotspots”, Submitted for publication to *Wiley Journal on Wireless Communications and Mobile Computing*, June 2012.
- IV. **Peter Dely**, Andreas Kassler, Nico Bayer, Hans Einsiedler, Dmitry Sivchenko. “Method and System for Centralized Control of User Associations in Wireless Mesh Networks”, *Patent Application*, EP 11175957.7-1525, Submitted to European Patent Office. May 2012.
- V. **Peter Dely**, Andreas Kassler, Nico Bayer. “OpenFlow for Wireless Mesh Networks”, In *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), Workshop on Wireless Mesh and Ad Hoc Networks*, Hawaii, USA, August 2011.
- VI. **Peter Dely**. “Towards an Architecture for OpenFlow and Wireless Mesh Networks”, Poster presentation at *CHANGE & OFELIA Summer School*, Berlin, Germany, November 2011.
- VII. **Peter Dely**, Andreas Kassler, Jonathan Vestin, Nico Bayer, Hans-Joachim Einsiedler, Christoph Peylo. “Method and system for the distribution of the control and data plane in Wireless Local Area Network Access Points”, *Patent Application*, Submitted to European Patent Office. August 2012.
- VIII. Jonathan Vestin, **Peter Dely**, Andreas Kassler, Nico Bayer, Hans J. Einsiedler, Christoph Peylo. “CloudMAC - Towards Software Defined WLANs”, *Poster Presentation at ACM Mobicom*, Istanbul, Turkey, August 2012.

- IX. **Peter Dely**, Andreas Kassler, Jonathan Vestin, Nico Bayer, Hans-Joachim Einsiedler, Christoph Peylo. “CloudMAC - An OpenFlow based Architecture for 802.11 MAC Layer Processing in the Cloud”, In *Proceedings of the IEEE Broadband Wireless Access Workshop*, held in conjunction with Globecom 2012, Anaheim, USA, December 2012.
- X. **Peter Dely**, Andreas Kassler, Nico Bayer, Hans Einsiedler, Christoph Peylo. “BEST-AP: Non-intrusive Estimation of Available Bandwidth and its Application for Dynamic Access Point Selection”, Submitted for publication in *Elsevier Computer Communications Journal*.
- XI. **Peter Dely**, Andreas Kassler, Lawrence Chow, Bradley Collins, Nick Bambos, Nico Bayer, Hans Einsiedler, Christoph Peylo, Daniel Mellado, Miguel Sanchez. “A Software Defined Networking Approach for Handover Management with Real-Time Video in WLANs”, In *Proceedings of the First International Workshop on High Mobility Wireless Communications*, Chengdu, China, November 2012.

Other Publications

In addition to the papers listed above, I have co-authored the following publications:

- Lawrence Chow, Bradley Collins, Nick Bambos, Christoph Peylo, Hans Einsiedler, Nico Bayer, **Peter Dely**, Andreas Kassler. “Channel Aware Rebuffering for Media Streaming with Handoff Control”, Submitted for publication in ICC 2013.
- Lawrence Chow, Bradley Collins, Nick Bambos, Nico Bayer, Hans Einsiedler, Christoph Peylo, **Peter Dely**, Andreas Kassler. “Playout-Buffer Aware Hand-Off Control for Wireless Video Streaming”, In *Proceedings of IEEE Global Communications Conference (GLOBE-COM) 2012*, Anaheim, USA, December 2012.
- Andreas Kassler, Lea Skorin-Kapov, Ognjen Dobrijević, Maja Matijašević, **Peter Dely**. “Towards QoE-driven Multimedia Service Negotiation and Path Optimization with Software Defined Networking”, In *Proceedings of the International Conference on Software, Telecommunications and Computer Networks (IEEE SOFTCOM)*, Split, Croatia, September 2012.

- Shuqiao Zhou, Ruixi Yuan, **Peter Dely**, Andreas Kassler. "Mitigating Control Channel Saturation in the Dynamic Channel Assignment Protocol." *JCIT: Journal of Convergence Information Technology*, no. 6 (2011): 271-281, 2011.
- **Peter Dely**, Andreas Kassler, Nico Bayer, Dmitry Sivchenko. "An Experimental Comparison of Burst Packet Transmission Schemes in IEEE 802.11-based Wireless Mesh Networks", In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM) 2010*, Miami, USA, December 2010.
- **Peter Dely**, Marcel C. Castro, Sina Soukhakian, Arild Moldsvor, Andreas Kassler. "Practical Considerations for Channel Assignment in Wireless Mesh Networks", In *Proceedings of IEEE Globecom 2010 Workshop on Broadband Wireless Access (BWA) 2010*, Miami, USA, December 2010.
- **Peter Dely**, Andreas Kassler, Nico Bayer, Hans-Joachim Einsiedler, Dmitry Sivchenko. "FUZPAG: A Fuzzy-Controlled Packet Aggregation Scheme for Wireless Mesh Networks" In *Proceedings International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) 2010*, Yantai, China, August 2010.
- **Peter Dely**, Andreas Kassler, Nico Bayer, Hans-Joachim Einsiedler, Dmitry Sivchenko. "Method and system for deriving an aggregation delay for packet aggregation in a wireless network", *European Patent Application*, Nr. EP10167525, June 2010.
- Barbara Staehle, Dirk Staehle, Rastin Pries, Matthias Hirth, **Peter Dely**, Andreas Kassler. "Measuring One-Way Delay in Wireless Mesh Networks - An Experimental Investigation". In *Proceedings of the 4th ACM PM2HW2N Workshop*, Tenerife, Spain, October 2009.
- Marcel C. Castro, **Peter Dely**, Andreas J. Kassler, Francesco Paolo D'elia, Stefano Avallone. "OLSR and Net-X as a Framework for Channel Assignment Experiments - Poster Presentation", In *Proceedings of the Fourth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH) 2009*, Beijing, China, September 2009.
- Marcel C. Castro, **Peter Dely**, Andreas J. Kassler, Nitin H. Vaidya. "QoS-Aware Channel Scheduling for Multi-Radio/Multi-Channel Wireless Mesh Networks", In *Proceedings of the Fourth ACM International Workshop on Wireless Network Testbeds, Experimental*

Evaluation and Characterization (WiNTECH) 2009, Beijing, China, September 2009.

- **Peter Dely**, Andreas Kassler. “KAUMesh Demo”, In *Proceedings of 9th Scandinavian Workshop on Wireless Ad-hoc & Sensor Networks*, Uppsala, Sweden, May 2009.
- Nico Bayer, Marcel C. Castro, **Peter Dely**, Andreas Kassler, Yevgeni Koucheryavy, Piotr Mitoraj, Dirk Staehle. “VoIP service performance optimization in pre-IEEE 802.11s Wireless Mesh Networks”, In *Proceedings of the IEEE International Conference on Circuits & Systems for Communications (ICCSC) 2008*, Shanghai, China, May 2008.
- Jonas Brolin, **Peter Dely**, Mikael Hedegren, Andreas Kassler. “Implementing Packet Aggregation in the Linux Kernel”, In *Proceedings of 8th Scandinavian Workshop on Wireless Ad-hoc & Sensor Networks*, Uppsala, Sweden, May 2008.
- **Peter Dely**, Andreas Kassler. “Adaptive Aggregation von VoIP Paketen in Wireless Mesh Networks”, In *Proceedings of WMAN FG 2008 (Ulmer Informatik Bericht)*, Ulm, Germany, February 2008.
- Marcel C. Castro, **Peter Dely**, Jonas Karlsson, Andreas Kassler. “Capacity Increase for Voice over IP through Packet Aggregation in Wireless Multihop Mesh Networks”, In *Proceedings of WAMSNET International Workshop on Wireless Ad Hoc, Mesh and Sensor Networks*, Jeju Island, South Korea, December 2007.
- Andreas Kassler, Marcel Castro, **Peter Dely**. “VoIP Packet Aggregation based on Link Quality Metric for Multihop Wireless Mesh Networks”, In *Proceedings of the Future Telecommunications Conference*, Beijing, China, October 2007.
- **Peter Dely**, Andreas J. Kassler. “On Packet Aggregation for VoIP in Wireless Meshed Networks”, In *Proceedings of 7th Scandinavian Workshop on Wireless Ad-hoc & Sensor Networks*, Stockholm, Sweden, May 2007.

Contents

1	Introduction	1
1.1	Research Questions	2
1.2	Research Method	3
1.3	Outline and Contributions	5
1.3.1	Chapter 3: Modeling the Channel Load in IEEE 802.11	5
1.3.2	Chapter 4: Optimization of WLAN Associations	6
1.3.3	Chapter 5: Fair Optimization of WMNs	6
1.3.4	Chapter 6: Optimizing WMNs with OpenFlow	6
1.3.5	Chapter 7: Distributed MAC for Software Defined WLANs	7
1.3.6	Chapter 8: Accurate Estimation of Link Quality and Fast Handovers	7
1.3.7	Chapter 9: Mobile Video Streaming with BEST-AP	8
2	Background	9
2.1	Wireless Local Area Networks	9
2.1.1	IEEE 802.11 WLAN System Architecture	9
2.1.2	Physical Layer	11
2.1.3	Medium Access Layer	13
2.1.4	Finding and Associating to Access Points	16
2.1.5	Mobility Management	18
2.1.6	Other Relevant IEEE 802.11 Standards	19
2.2	Wireless Mesh Networks	20
2.2.1	Architecture	21
2.2.2	Routing Protocols	21
2.3	Software Defined Networking	24

2.3.1	Towards Software Defined Networks	24
2.3.2	Architecture of Software Defined Networks	25
2.3.3	OpenFlow Protocol	26
2.4	Flow Optimization	28
2.4.1	Mathematical Optimization	29
2.4.2	Flow Optimization	30
2.4.2.1	Max-Flow Problem	30
2.4.2.2	Min-Cost Flow Problem	32
2.4.2.3	Applications	32
3	Modeling the Channel Load in IEEE 802.11	33
3.1	Introduction	33
3.1.1	Related Work	33
3.1.2	Problem Statement and Contributions	34
3.2	Analytical Model of the IEEE 802.11 MAC	35
3.2.1	IEEE 802.11 DCF under Saturation Conditions	35
3.2.2	IEEE 802.11 DCF under Non-Saturation Conditions	36
3.2.3	Modeling the Channel Busy Fraction	38
3.2.4	Discussion	39
3.2.5	Limitations of the Model	39
3.3	Validation of the Model	41
3.3.1	Experimental Setup	41
3.3.2	Channel Busy Fraction and Traffic Injection Rate	42
3.3.3	Linear Model of the Channel Busy Fraction	43
3.4	Conclusions	45
4	Optimization of WLAN Associations	47
4.1	Introduction	47
4.1.1	Related Work	50
4.1.2	Problem Statement and Contributions	51
4.2	Static Network Model	52
4.2.1	System Model and Notation	52
4.2.2	Variables	54
4.2.3	Model Constraints	54
4.2.4	Solving the Model	56
4.3	Dynamic Network Model	56
4.3.1	Parameters and Variables	57
4.3.2	Model Constraints	58
4.3.3	Objective Function	60
4.4	Static Optimization	60
4.4.1	Reconfiguration Strategies	61

4.4.1.1	Greedy	61
4.4.1.2	k-Handover	61
4.4.1.3	Hysteresis	61
4.4.2	Evaluation	62
4.4.2.1	Evaluation Settings	62
4.4.2.2	Evaluation Metric and Statistical Analysis	63
4.4.2.3	What is the Impact of User Mobility and Network Size?	64
4.4.2.4	What is the Impact of the Handover Cost?	65
4.4.2.5	What is the Impact of Hysteresis Parameter f ?	68
4.4.2.6	What is Impact of the Handover Limit k ?	69
4.4.3	Discussion	69
4.5	Sliding Window-Based Optimization	70
4.5.1	Sliding Window Method	70
4.5.2	Evaluation	71
4.6	Conclusions	74
5	Fair Optimization of WMNs	75
5.1	Introduction	75
5.1.1	Related Work	76
5.1.2	Problem Statement and Contributions	77
5.2	System Model	79
5.2.1	Basic Notation	79
5.2.2	Feasible Solution Set	79
5.2.3	Objective Function and Fairness Considerations	82
5.3	Solution Algorithms for MESHMAX	83
5.3.1	Optimal Max-Min Fair Rate Allocation (MESHMAX- OPT)	84
5.3.2	Relaxed Max-Min Fair Rate Allocation (MESHMAX- LP)	86
5.3.3	Heuristic Solution Algorithm (MESHMAX-FAST)	88
5.3.3.1	Sub-Problem I: Flow Maximization	89
5.3.3.2	Sub-Problem II: Establishing STA/MAP Assignments	90
5.3.3.3	Sub-Problem III: Increasing the Minimum STA Rate	92
5.3.3.4	Sub-Problem IV: Routing	94
5.3.3.5	Solution Algorithm	94
5.4	Numerical Performance Analysis	96
5.4.1	Evaluation Scenario	97

5.4.2	Throughput	97
5.4.3	Run-time	98
5.4.4	Discussion	101
5.5	Network Simulations	102
5.5.1	Scenario	102
5.5.2	Throughput Performance	103
5.5.3	Impact of Bottlenecks at Gateways	105
5.5.4	Increase in Network Scalability	106
5.5.5	Importance of Active STA Management	106
5.6	Conclusions	108
6	Optimizing WMNs with OpenFlow	111
6.1	Introduction	111
6.1.1	Related Work	112
6.1.2	Problem Statement and Contributions	113
6.2	An Architecture for OpenFlow in WMNs	114
6.2.1	OpenFlow-Enabled Mesh Routers	114
6.2.2	Core Network	115
6.2.3	Stations	117
6.3	Implementation	117
6.4	Micro-Benchmarks	118
6.4.1	Is there a Performance Penalty through OpenFlow Rule Processing?	118
6.4.2	What Control Traffic Overhead is Created by Open- Flow?	120
6.4.3	How Fast are Rules Activated?	121
6.5	Optimization of STA/MAP Associations with OpenFlow . .	122
6.5.1	Managing Station Handovers	122
6.5.2	Implementing the MESHMAX Algorithms	124
6.5.3	What is the Cost of a Handover?	125
6.5.4	What are the Performance Gains due to the MESH- MAX Algorithms?	126
6.6	Conclusions	128
7	Distributed MAC for Software Defined WLANs	131
7.1	Introduction	131
7.1.1	Related Work	132
7.1.2	Problem Statement and Contributions	133
7.2	Architecture	134
7.2.1	Overview	134
7.2.2	Data Frame Processing	135

7.2.3	Control Command Processing	137
7.3	Implementation	138
7.4	Performance Evaluation	139
7.5	CloudMAC Handovers	140
7.5.1	Evaluation	142
7.6	Other Potential Applications and Benefits	145
7.7	Conclusions	146
8	Accurate Estimation of Link Quality and Fast Handovers	147
8.1	Introduction	147
8.1.1	Related Work	148
8.1.1.1	Available Bandwidth Estimation	148
8.1.1.2	Access Point Selection	149
8.1.2	Problem Statement and Contributions	150
8.2	Motivating Examples	151
8.2.1	RSSI is not Suitable for Accurate Available Bandwidth Estimation	152
8.2.2	Packet Dispersion Measurements are too Slow for Continuous Estimation	153
8.2.2.1	Throughput with Varying Channel Load	153
8.2.2.2	Throughput with User Mobility	156
8.2.3	Discussion	158
8.3	Non-intrusive Bandwidth Estimation	159
8.3.1	Model for Estimation of Available Bandwidth with a Fixed MCS	159
8.3.2	Model for Estimation of Available Bandwidth with Rate Adaptation	160
8.4	Dynamic AP Selection Based on Bandwidth Estimation	163
8.4.1	Mobile Station	163
8.4.2	Pre-authentication and Pre-association	165
8.4.3	Handover Services	166
8.4.4	Scheduling AP Usage	167
8.4.5	Bandwidth Estimation	168
8.5	Implementation	169
8.6	Evaluation	170
8.6.1	Bandwidth Estimation	170
8.6.1.1	How Accurate is the Estimation under Constant Network Load?	170
8.6.1.2	How Fast Does the Estimation React to Changes in Network Load?	172
8.6.2	Dynamic AP Selection	173

8.6.2.1	What is the Impact of Different AP Scheduling Strategies?	173
8.6.2.2	Can BEST-AP Adapt to Changes in Available Bandwidth?	175
8.6.2.3	What Performance Gains are Possible under External Interference?	177
8.7	Conclusions	179
9	Mobile Video Streaming with BEST-AP	181
9.1	Introduction	181
9.1.1	Related Work	183
9.1.2	Problem Statement and Contributions	185
9.2	Making BEST-AP Mobile	185
9.3	Evaluation	186
9.3.1	Micro-Benchmarks	187
9.3.1.1	How well does BEST-AP Support Station Mobility?	187
9.3.1.2	How Long does it Take to Scan for new APs?	190
9.3.1.3	How Long does a Handover Take?	190
9.3.2	Video Streaming	192
9.3.2.1	How Smooth is the Video Playout?	192
9.3.2.2	Is a Dedicated Scanning Card Necessary?	194
9.3.2.3	What is a Good Playout Buffer Size?	195
9.4	Conclusions	195
10	Conclusions	197
10.1	Achievements and Contributions	197
10.2	Future Directions	198
10.3	Final Remarks	199
A	Notational Conventions	201
	Bibliography	203
	Abbreviations and Acronyms	225

List of Figures

2.1	Layering in the IEEE 802.11 standard. Source: [27]	10
2.2	Architecture of an infrastructure IEEE 802.11 network . . .	11
2.3	Architecture of an IEEE 802.11 Independent Basic Service Set	12
2.4	IEEE 802.11 frames exchanged when a station associates to an AP without encryption	17
2.5	IEEE 802.21 architecture. Reproduced from [25]	19
2.6	Architecture of an IEEE 802.11 mesh network	22
2.7	ONF architecture for Software Defined Networks	26
2.8	Architecture of an OpenFlow switch	27
2.9	Example graph. l denotes the capacity of the edge, c the cost of sending one unit of flow. s is source, t is sink.	30
3.1	Channel busy fraction and throughput as function of the aggregate offered load. The highest throughput is achieved before the network is saturated.	40
3.2	Channel busy fraction which gives peak performance. Both the network and packet size have relatively little impact. . .	40
3.3	Channel busy fraction and traffic offered load. The channel busy fraction grows almost linearly with the offered load until the network gets saturated. Then channel busy fraction grows faster due to the increasing collision probability. . . .	44
3.4	Channel busy fraction and average number of transmissions. The average number of transmissions is close to 1 as long as the network load is low. When the load increases, the collision probability and the average number of transmissions increase.	44

4.1	Likelihood of being in coverage range from several APs of the Karlstad University Campus WLAN. More than 45% of the stations can choose between two and more APs to connect to at 24 PHY rate.	48
4.2	Example of a user walking in a hotspot area with coverage from AP1-3. The user can either perform a handover as soon as a better AP is available (“Scheme A”) or after the connection breaks (“Scheme B”).	49
4.3	Map of the Computer Science Department at Karlstad University. 13 APs provide WLAN coverage in corridors, offices, meeting rooms and labs.	63
4.4	Impact of mobility on algorithm performance ($D = 3$, 40 STA). More mobile stations lead to a lower performance since handovers play a greater role.	64
4.5	Impact of network dynamicity on algorithm performance ($D = 3$). When stations move faster, the achievable performance decreases.	66
4.6	Impact of the handover cost on the algorithm performance with 10 STA and 0 m/s (left) and 1.5 m/s (right) station speed. With high speeds and high costs the Greedy scheme performs in particular bad.	67
4.7	Impact of Hysteresis factor f (30 STA, $D = 3$)	68
4.8	Influence of maximum allowed changes on the performance of k -Handover (with 30 STA, $D = 3$)	69
4.9	Time line of the sliding window algorithm	70
4.10	Performance of sliding window-based optimization. As the prediction window size increases, the performance also increases. Prediction errors decrease the performance.	73
4.11	Connection pattern of one STA with $W_p = 0, 10$ and 20. With a larger prediction window fewer handovers are necessary and higher loaded APs are avoided.	73
5.1	Example of RSSI association/minimum hop-count routing	76
5.2	Example of optimized association/routing	76
5.3	Network for which Algorithm 5.2 does not terminate	88
5.4	Feasible set for multi-path (left) and single-path routing (right)	89
5.5	Sequence diagram of the MESHMAX-FAST algorithm	90
5.6	Example: Finding a matching, where STA 1 can connect to MAP 1, STA 3 to MAP 2 and STA 2 to both ($\sigma = 2$).	92

5.7	Example re-association graph. u_1 can connect to m_1 , u_2 to m_1 and m_2 , u_3 to m_2 and m_3 . One can increase the association count at m_3 by moving u_2 and u_3	93
5.8	Minimum, 90-percentile, maximum and average throughput for 30 random networks	99
5.9	Empirical CDF of minimum throughput relative to optimum for 15 (left) and 25 (right) STA. MESHMAX-FAST* on average achieves 98% and 99% of the optimal performance. In more than 80% of the cases MESHMAX-FAST* computes the optimal solution.	99
5.10	Comparison of algorithm run-time (averaged over 30 random topologies). The run-times of MESHMAX-OPT and MESHMAX-LP can be several orders of magnitude higher than the run-times of MESHMAX-FAST and MESHMAX-FAST*.	100
5.11	Minimum (left) and average (right) UDP throughput in a random network. The simulation results match the analytical optimum very closely. The RSS-based association can lead to a starvation of users, while MESHMAX avoids this.	104
5.12	Minimum (left) and average (right) TCP throughput in a random network. The TCP throughput is lower than the analytical optimum. TCP connections in the RSS-based association scheme can be completely starved.	104
5.13	Impact of different gateway connection speeds	106
5.14	Increased network scalability through MESHMAX algorithms	107
5.15	Average minimum throughput when associations are controlled on a fraction of all STA	108
6.1	Overall system architecture	115
6.2	Architecture of an OpenFlow mesh node	116
6.3	Architecture of the core network	117
6.4	Forwarding performance with 1400 byte UDP datagrams	119
6.5	Total control traffic caused by OLSR and OpenFlow	120
6.6	Rule activation time on MAP3 (Average of 10 experiments, Coefficient of variation < 0.001 for all values)	122
6.7	Management of station connectivity and mobility	123
6.8	TCP throughput (averaged over 100 ms windows) from the core-network to the station during two handovers (after 15 and 30 sec.)	125
6.9	Testbed setup to evaluate the MESHMAX algorithm	127

6.10	Throughput with RSSI-based association, hop-count based association and the optimal association computed with MESHMAX and 3 Mbit/s gateway capacity	128
6.11	Throughput with RSSI-based association, hop-count based association and the optimal association computed with MESHMAX and 6 Mbit/s gateway capacity	129
7.1	Architecture of a CloudMAC based WLAN	135
7.2	Processing of MAC frames with CloudMAC	136
7.3	Processing of control commands with CloudMAC	137
7.4	Ping RTT for CloudMAC and the reference network. Cloud-MAC has a higher RTT due to the additional frame processing.	140
7.5	TCP throughput for different segment sizes. The throughput increases with the segment size (due to the lower overhead on the wireless transmission). With larger segment sizes the relative differences between CloudMAC and the reference network get smaller.	141
7.6	Scenario to test CloudMAC's ability to enable handovers of standard IEEE 802.11 stations. First all traffic between the VAP and the station is sent via WTP1 and then moved to WTP2.	142
7.7	ECDF of number of lost packets during an AP switch. Cloud-MAC significantly reduces the number of lost packets (send rate: 1000 packets/second).	144
7.8	TCP time/sequence diagram. With CloudMAC no disruption is visible, while with the reference network the throughput is 0 for 27 seconds.	144
8.1	Packet loss probability and RSSI at 12 Mbit/s obtained from measurements on the Karlstad University campus . . .	152
8.2	Histogram of channel busy fraction during 5 minutes	155
8.3	Sample autocorrelation function of the channel load	156
8.4	Sample autocorrelation function of the available bandwidth estimation, without (left) and with external interference (right)	157
8.5	PHY rate (black), frame success probability (gray) and available bandwidth estimation (red) for a user walking past an AP. The user is closest to the AP after 15 seconds. . . .	158
8.6	Architecture of a BEST-AP WLAN	164
8.7	Architecture of a mobile station	165

8.8	Sequence of messages exchanged during a handover from AP1 to AP2	167
8.9	States of APs and their transitions	168
8.10	Available bandwidth estimation with a fixed PHY rate and Minstrel Autorate with constant background load. Our estimation is more accurate than WBest in all cases. . . .	171
8.11	Available bandwidth estimation using BEST-AP (left) and WBest (right). The background load increases every 10 seconds.	173
8.12	Impact of the estimation duration. A longer time on the primary AP is beneficial for the throughput as fewer handovers are required.	175
8.13	Test setup to evaluate how well BEST-AP adapts to changes in available bandwidth. The load on channel 140 is constant, while the load on channel 52 varies according to an ON-OFF pattern.	176
8.14	Reaction to changes in bandwidth. The available estimation tracks the changes in load at AP1, while it stays constant for the constant load at AP2.	176
8.15	Selection of the primary AP. Shaded areas mark periods in which there is load on channel 52.	177
8.16	Performance under realistic external interference. Values greater than 1 mean that BEST-AP is beneficial. On average a 85% throughput increase is achieved with BEST-AP. . . .	178
9.1	Simplified architecture of a typical video player. Video frames are coming from the network and the buffer removes jitter before the decoder decodes the frames for playout. . .	182
9.2	Example scenario: the station is streaming a live video and moving from AP1 towards AP2. The station needs to perform a handover when it moves too far away from AP1.	183
9.3	Evolution of video buffer level while the station is moving from AP1 towards AP2.	184
9.4	Map of the testbed used for the video streaming tests. The mobile terminal moves from point A to point B and back. .	187
9.5	Test topology for mobility tests. The STA moves along the corridor from AP1 towards AP2.	188
9.6	TCP throughput with station mobility. With dynamic switching the primary AP is chosen according to the available bandwidth estimate. Linux default uses the RSSI to select the best AP.	189

9.7	TCP throughput during one test run. After 20 seconds AP2 is better and hence selected as primary AP.	189
9.8	Duration of a scan for new APs. Each channel takes approximately 8 ms to be scanned.	190
9.9	ECDF of handover and ACK duration. For a small fraction of tests signaling messages got lost and thus the handover duration increases by approximately 40 ms.	191
9.10	Distribution of freeze event durations. Standard systems have many and long freeze events while BEST-AP has few and short freeze events.	194

List of Tables

2.1	Overview of IEEE 802.11 PHY layer standards	14
3.1	Used symbols and description	37
3.2	Parameters of the model and the experiments	42
3.3	Summary of the linear regression for the Markov model data. $R^2 = 0.94$	46
3.4	Summary of the linear regression for the experimental data. $R^2 = 0.97$	46
4.1	Important notation used in this chapter	53
5.1	Summary of notation used in this chapter	80
6.1	Outage duration during a handover	125
8.1	Used symbols and default values for IEEE 802.11a	161
8.2	Minstrel retry chains. Source: [9]	162
9.1	Video freeze events	193

List of Algorithms

- 5.1 Max-min Fair bandwidth allocation for multi-commodity
single-path networks 85
- 5.2 Max-min fair bandwidth allocation for multi-commodity
multi-path networks 87
- 5.3 Increasing minimum STA rates by re-associations 93
- 5.4 Routing of user traffic 95
- 5.5 MESHMAX-FAST solution algorithm 96

Chapter 1

Introduction

Wireless Local Area Networks (WLANs) are ubiquitous today. WLANs are a simple and cheap way to connect laptops, tablet PCs, smart phones, digital cameras, TVs and set-top boxes etc. to the Internet in almost all places of daily life. WLANs can be found in private homes, universities, offices, restaurants and more recently even on airplanes, in cars or on trains. They can be used for web surfing, video streaming, telephony and many other applications. This versatility along with the cheap hardware and the use of license-free radio spectrum has led to an ever increasing demand for bandwidth. Industry and academia have managed to satisfy this demand and tremendously increased the speed of WLANs over the last 15 years. From the first release of the IEEE 802.11 standard in 1997 to the IEEE 802.11ad standard, which is expected in 2014, the transmission speed almost increased by four orders of magnitude. This means that on average the maximum transmission rate of the IEEE 802.11 standards has nearly risen by one order of magnitude every four years. This is even faster than the growth of CPU speeds, which according to Gordon Moore's famous law, double every 18 month in speed. Hence, they need approximately five years to speed up by one order of magnitude [125].

However, increasing the speed of WLANs often comes at the price of shorter communication ranges. Another famous law, Shannon's law [153], gives us a simple method to compute the fundamental capacity limit of a wireless link. It states that the capacity depends on two factors: the level of the received signal in relation to the noise and the channel bandwidth used for transmission. The capacity increase seen in WLANs to a large extent comes from more sophisticated wireless transceivers and antennas, which allow to use higher carrier frequencies, wider channel bandwidths and modulation schemes which are faster. All three factors lead to lower

transmission ranges. Higher carrier frequencies usually have worse radio propagation properties than lower frequencies. Wider channels are harder to transmit over larger distances [57]. Faster modulation schemes are susceptible to noise and hence are only suited for short links, which have high signal-to-noise ratios.

One can observe a clear trend that users demand higher speeds, which are only possible on short links. Fast WLANs covering large areas hence require a dense deployment of APs. Such a dense deployment creates new problems. One problem is how to achieve such a dense deployment in a cost-efficient way. The number of APs required to cover a certain area grows with the inverse square of the communication distance. This means, that by halving the communication distance, four times as many APs are required. To keep the costs low, AP hardware and software as well as network management needs to be simplified. Wireless mesh networks could also be a cost saver, since in wireless mesh networks APs can communicate with each other wirelessly and thus not all APs need expensive cabling.

When cell sizes are shrunk, the need for handovers increases. The network architecture and the algorithms for handover management of today's WLANs are not suited to address the challenges arising from densely deployed future WLANs. This thesis thus aims to design and evaluate new architectures that simplify the management and operation of future WLANs. We will also develop new algorithms and methods to optimize handovers in WLANs.

1.1 Research Questions

The following three main research questions are considered in this thesis:

1. *If a station is in the coverage area of several APs, when should the station use a given AP?*

In densely deployed networks the coverage areas of two APs often overlap. Hence, a station often can choose which AP to use. Selecting the optimal AP is a non-trivial task, as the optimal AP depends on many factors, such as a network-wide fairness policy and the costs of performing a handover from the current AP to the optimal AP. To answer this question, we first build a mathematical model of WLANs and then define several optimization problems. We then derive algorithms to compute solutions to the optimization problems.

2. *How can the current, closed architecture of WLANs and wireless mesh networks be opened up and evolved to allow the easy deployment of*

new applications to enhance mobility support and resource distribution fairness?

Current WLAN management systems are largely based on proprietary technologies and do not allow programmers to develop network applications in a vendor independent way. This makes it difficult to deploy new network applications, for example to allow wireless stations to roam around seamlessly or to distribute resources in a fair manner. To address this issue, we present a new architecture for the processing of MAC frames in IEEE 802.11 WLANs and a new architecture for controlling routing and rate allocation in wireless mesh networks. The architectures use ideas and protocols of Software Defined Networks and allow to deploy optimization algorithms, such as the ones developed in this thesis.

3. *How to estimate the quality of a link between an AP and a station in a fast way and how to enable seamless handovers between APs?*

Quality metrics of links between stations and APs are important input parameters to algorithms that optimize handovers and station/AP associations. Measuring the link quality is difficult in practice, as wireless links are subject to stochastic effects such as noise and fading. As a consequence, the link quality has to be estimated frequently, which is not possible with current probe-based quality estimation methods. We overcome this limitation with a new method, that uses regular data traffic to assess the quality of a link. In addition, we discuss why estimating the link quality requires a method for fast handovers between APs and present a system which enables such fast handovers. The performance of the system is demonstrated by streaming a video to a mobile user. The system continuously assesses the quality of surrounding APs and performs fast handovers when required.

1.2 Research Method

To answer the research questions posed above, the commonly used iterative research process of literature review, formulation of a problem statement, hypothesis formulation, hypothesis testing and analysis [145] is applied. In this process one step usually follows the previous one, but sometimes it may be required to revert to a previous step, for example to refine a hypothesis based on the results of the analysis phase.

The literature review helps to identify the state-of-the-art and relevant problems. It furthermore shows how other researchers have tackled a prob-

lem before and thereby allows to build on previous results. Subsequently, a research problem is stated and described, how a solution to this problem advances the state-of-the-art. Based on the knowledge gained from the literature review, a hypothesis is formulated. The hypothesis delivers a potential explanation of some aspect of the system under consideration and allows to make predictions. Sometimes hypothesis can be very formal, such as “a change of $x\%$ in variable y , leads to a change of $a\%$ in variable b with $c\%$ confidence”. Often it is just an informal description of an assumption derived from previous knowledge.

In the next step, the hypothesis is tested. In the performance analysis of computer systems the most common methods for hypothesis testing are analytical modeling, simulations and real-world experiments. An analytical model is a mathematical description of a system. In the process of formulating an analytical model, one needs to find a balance between the complexity of the model and the level of detail. Usually, a higher level of detail leads to more complexity, but makes the model more predictive. Computer simulations can include more details than analytical models, but still suffer from the same problem of finding a balance between the complexity of the simulator and the level of abstraction. Complex simulators are more likely to contain software bugs than simple ones. Thus, a higher level of complexity does not necessarily lead to more accuracy [95]. Also, the simulation run-time increases with the complexity of the simulator.

Real-world experiments have the lowest level of abstraction, but the system under investigation needs to exist and environmental factors are hard to control. Because of the broadcast nature of the wireless medium, the shared use of the wireless spectrum and the stochastic behavior of wireless links, real-world experiments with wireless systems are in particular difficult to control. In addition, usually only a small number of possible scenarios can be evaluated with real-world experiments, as otherwise the costs of the experiments would be too high. This makes it hard to draw general conclusions from them. Therefore, hybrid forms of hypothesis testing, such as network emulation, which combines aspects of real-world systems and simulation, are sometimes used. Each of the hypothesis testing methods has its advantages and disadvantages, which need to be considered when selecting the method. However, it is important to understand that neither of the methods should be solely used to test a hypothesis. As a best practice [95], all three methods should be used to validate each other’s results.

In this thesis we apply all three methods. We use analytical models to understand scaling behaviors, network simulations to validate analytical

models and real-world experiments with prototype implementations to see how a system behaves under realistic conditions. The results of the different methods are then analyzed using statistical and data visualization techniques. By successively applying those three methods we are able to identify whether the formulated models are a reasonably accurate representation of the real system and if ideas demonstrated in small real networks in principle should also work in larger networks.

1.3 Outline and Contributions

The rest of this dissertation is organized as follows. Chapter 2 gives background information about wireless local area networks, wireless mesh networks, software defined networks and network optimization. The background chapter is followed by Chapters 3-5, which study modeling aspects of WLANs and wireless mesh networks and develop new optimization algorithms. Hereafter, Chapters 6-9 present and evaluate concrete architectures that allow to implement such algorithms. Finally, Chapter 10 concludes the thesis with a summary and an outlook to future work. Each chapter, except for Chapters 2 and 10, presents

- a research problem,
- related work,
- research results and
- a summary and conclusions following from the research results.

Below, we will provide a more detailed overview of each chapter. Notational conventions and a list of abbreviations and acronyms are provided in the appendix of this thesis.

1.3.1 Chapter 3: Modeling the Channel Load in IEEE 802.11

In Chapter 3 we investigate how to model the IEEE 802.11 MAC layer and the utilization of WLAN channels. We develop a model to predict the channel utilization of an IEEE 802.11 WLAN and present testbed measurements to validate the model. The main insight is that IEEE 802.11 WLANs exhibit a linear behavior, as long as they are not operated under saturation conditions. This is an important insight, as linear systems are relatively easy to model and to control.

The findings of this chapter have been published in paper **I**. The author of this thesis was responsible for formulating the model, carrying out the experiments and analyzing the results.

1.3.2 Chapter 4: Optimization of WLAN Associations

Chapter 4 builds on the findings of Chapter 3 that in non-saturated WLANs the throughput is a linear function of the offered load. In Chapter 4, a mixed integer linear optimization problem is formulated, which allows to compute when a station should use which access point. With numerical simulations this model is used to study which factors have what influence on the optimal handover policy. The key insight of this chapter is that as the station mobility increases, a more efficient handover scheme which only causes minimal disruption, is necessary for good overall performance. If stations are static, even longer disruptions due to handovers do not lead to unacceptable performance degradations.

The model and the results of this chapter have been published in paper **II**. The author of this thesis formulated the model, carried out all simulations and analyzed the results.

1.3.3 Chapter 5: Fair Optimization of WMNs

In Chapter 5 we alter the problem setting of the previous chapter slightly. Here, we aim to compute the optimal station/AP associations, routing and resource allocation for APs that are connected to a wireless mesh network. The main emphasis of this chapter is how network resources should be distributed in a fair way and how a solution to the optimization problem can be computed fast. To this end we devise several fast heuristic solution algorithms and evaluate their efficiency with numerical and network simulations. The main outcome of this chapter is that an exact solution to the optimization problem is computationally too hard for online-network optimization, but our heuristic algorithms provide a good solution quality while at the same time they are fast enough for online network optimization.

The main contents of this chapter have been submitted for publication as paper **III**. Fabio D'Andreagiovanni and the author of this thesis jointly formulated the optimization problem. The solution algorithms were devised and evaluated by the author of this thesis.

1.3.4 Chapter 6: Optimizing WMNs with OpenFlow

In Chapter 6 an architecture within the Software Defined Networking framework is described that allows to exercise control of station associations,

routing and rate allocations in wireless mesh networks. The architecture was implemented and evaluated in a wireless mesh testbed. In a testbed implementation, the optimization algorithm of Chapter 5 computes the optimal routes and max-min fair rate allocations for associated clients. By using the OpenFlow protocol, the computed routes and rate allocations are enforced in the network. The evaluation shows that the proposed architecture enables centralized control of mesh networks at low overhead and that it is backward compatible to legacy protocols.

The contributions of this chapter have previously been published in papers **IV**, **V** and **VI**. The author of this thesis was responsible for the design, implementation and evaluation of the architecture.

1.3.5 Chapter 7: Distributed MAC for Software Defined WLANs

Chapter 7 discusses how to apply the idea of Software Defined Networks to distribute the IEEE 802.11 MAC layer. To this end, we present CloudMAC, which is an OpenFlow controlled distributed IEEE 802.11 MAC. The architecture allows to use OpenFlow and a centralized controller to manage important wireless transmission parameters such as the transmission power. Moreover, CloudMAC can for example be used to enable seamless handovers with unmodified IEEE 802.11 stations. The performance evaluation shows that the disruption caused by a CloudMAC handover is much shorter than the disruption experienced during a handover in a standard IEEE 802.11 WLAN.

The system design and evaluation results have been published in papers **VII**, **VIII** and **IX**. The author of this thesis is the initial developer of the system design. The prototype was implemented and evaluated by Jonathan Vestin under the supervision of the author of this thesis.

1.3.6 Chapter 8: Accurate Estimation of Link Quality and Fast Handovers

Chapter 8 introduces BEST-AP, a method and a system for estimating the bandwidth of links between a station and its surrounding access points. The method is based on statistics computed when transferring regular data traffic. A station can be associated to multiple APs simultaneously. The proposed system allows a station to pre-authenticate and pre-associate with new access points using the connection of the current access point. Thereby handovers between APs only require tuning the WLAN card to the correct channel, but no lengthy association procedures. This considerably speeds

up handovers. A station regularly transfers data over its surrounding APs to collect statistics for the bandwidth estimation. The AP with the highest available bandwidth is used longest. The main insight from this chapter is that the dynamic AP selection in BEST-AP allows to exploit temporal variations in available bandwidth by using the AP with the highest available bandwidth whenever possible.

The contributions of this chapter are submitted for publication as paper **X**. The author of the thesis has developed the system, implemented the prototype and evaluated it.

1.3.7 Chapter 9: Mobile Video Streaming with BEST-AP

In Chapter 9 the BEST-AP system of Chapter 8 is applied to optimize video streaming in WLANs and mobility management. The key challenge of mobile video streaming is that handovers must be fast enough to ensure that the video playout buffer is not emptied completely, as otherwise the video would freeze. In addition, in mobile scenarios the station needs to scan the WLAN channels regularly to detect new APs. We show that BEST-AP is capable of detecting APs fast enough and that the handovers are sufficiently quick to avoid video freezes. In our testbed evaluation BEST-AP significantly reduces the number of freeze events and their duration.

Parts of the content of this chapter have been published in paper **XI**. The author of the thesis has developed the system, implemented the prototype and evaluated it.

Chapter 2

Background

In this chapter we will first review basic terminology related to WLANs and the relevant standards. We will then explain what Wireless Mesh Networks are and how they can be used to extend WLANs. Thereafter, we will introduce the basic ideas and concepts of Software Defined Networking. Finally, we will give a short overview on network flow optimization.

2.1 Wireless Local Area Networks

According to [175], wireless networks can be categorized into three classes: *Wireless Personal Area Networks* (WPANs), *Wireless Local Area Networks* (WLANs) and *Wireless Metropolitan Area Networks* (WMANs). As the names indicate, the major difference between the network types is the geographical coverage. In this thesis we investigate WLANs, which provide local coverage, for example within buildings. Even though different types of WLANs have been proposed by standardization bodies such as the *Institute of Electrical and Electronics Engineers* (IEEE), only WLANs based on the IEEE 802.11 standard series have found major deployments. Hence, we limit the following discussion to IEEE 802.11-compliant WLANs.

2.1.1 IEEE 802.11 WLAN System Architecture

The IEEE 802.11 standards describe how wireless devices communicate with each other. As shown in Figure 2.1, the IEEE standards use the concept of network layers to abstract the architecture. The standards specify the lower two layers of the OSI reference model [27]: the *Physical layer* (PHY) and partially the *Data Link Control* layer (DLC). In IEEE

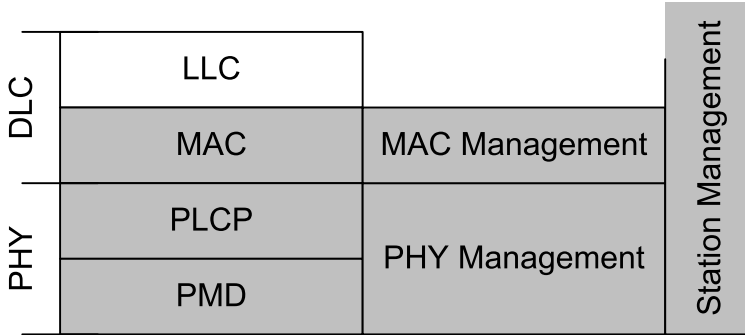


Figure 2.1: Layering in the IEEE 802.11 standard. Source: [27]

802.11, the PHY consists of the *Physical Medium Dependent Sublayer* (PMD) and the *Physical Layer Convergence Protocol* (PLCP). The PMD specifies the transmission type, which can be based on different technologies (e.g. infra-red or radio waves). The PLCP translates requests between the PMD and the *Medium Access Control* (MAC). The MAC is part of the DLC and controls access to the shared wireless medium. The DLC further includes the *Logical Link Control* (LLC), which is not specified in IEEE 802.11, but common to all IEEE 802 standards.

Besides the PHY and DLC sublayers, IEEE 802.11 furthermore describes control planes for PHY and MAC management. In addition, the standard further mentions an overall station management plane. However, the standard does not define any details of the station management plane. Therefore this component is implementation dependent.

Figure 2.2 depicts the architecture of an IEEE 802.11 WLAN: *Stations* (STA) communicate with an *Access Point* (AP), which is connected to a *Distribution System* (DS). APs forward *MAC Service Data Units* (MSDUs) between STA and the DS. The DS can be wired or wireless. The DS can be connected to a Portal, which allows to interconnect the DS with non-IEEE 802 networks. A set of STAs and one AP are grouped together to a *Basic Service Set* (BSS). Each BSS is identified by a unique *Basic Service Set Identifier* (BSSID). The BSSID is typically the MAC address of the AP. Several BSSs might be connected via a DS to form an *Extended Service Set* (ESS). The ESS is identified by an operator-given *Service Set Identifier* (SSID). The STAs and the AP of one BSS are controlled by a coordination function. In Section 2.1.3 we will discuss the role of the coordination function for medium access control. The AP provides authentication and privacy services (e.g. encryption key negotiation) for its BSS.

The IEEE 802.11 standard furthermore contains a simplified version

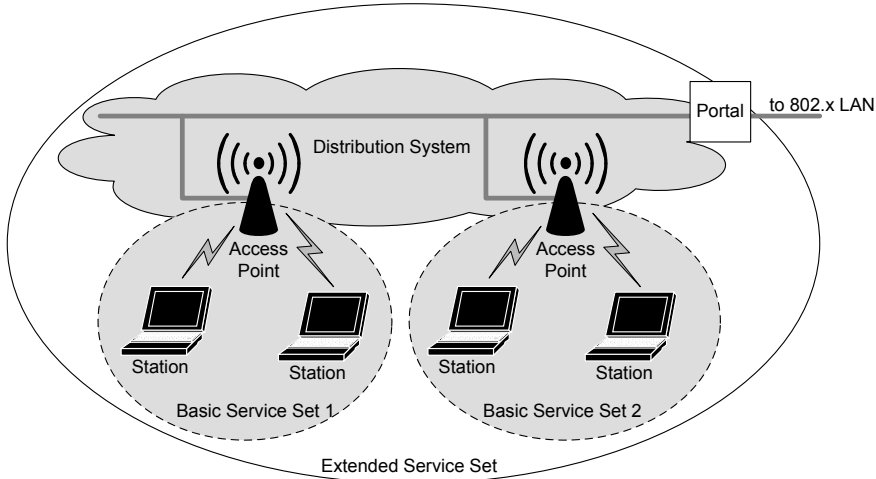


Figure 2.2: Architecture of an infrastructure IEEE 802.11 network

of a BSS, the *Independent Basic Service Set* (IBSS). As Figure 2.3 shows, an IBSS only consists of STAs. No other infrastructure, such as an AP, is required. This type of network can be formed spontaneously and hence is sometimes termed ad-hoc network.

2.1.2 Physical Layer

IEEE 802.11 specifies a number of different PHY layers, which are summarized in Table 2.1. The initial release of IEEE 802.11 [15] defined three PHY layers: *Infrared* (IR), *Frequency Hopping Spread Spectrum* (FHSS) and *Direct Sequence Spread Spectrum* (DSSS). The IR PHY layer uses infrared light for communications and was never widely adapted. FHSS and DSSS both apply spread spectrum modulation techniques. The main idea of spread spectrum is to spread the signal over a bandwidth much larger than the bandwidth required to transmit the information. Thereby the impact of narrow band interference and carrier selective fading can be mitigated. With FHSS, communication partners change carrier frequencies according to a known sequence. With DSSS, communication partners use the same carrier frequency, but spread the signal over a larger bandwidth using spreading codes. The initial release of IEEE 802.11 supports data rates of up to 2 Mbit/s and uses frequencies in the 2.4 GHz unlicensed band for *Industrial, Scientific and Medical* (ISM) use.

IEEE 802.11b [17] improved the initial IEEE 802.11 standard mainly with new coding and modulation schemes. IEEE 802.11b uses *Binary Phase Shift Keying* (BPSK) and *Quaternary Phase Shift Keying* (QPSK)

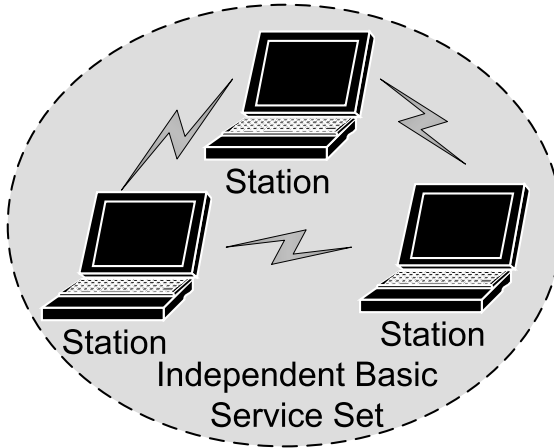


Figure 2.3: Architecture of an IEEE 802.11 Independent Basic Service Set

to modulate signals. With BPSK, the phase of the signal is used to encode one bit per symbol. QPSK in addition uses the amplitude of the signal and thereby can carry two bits per symbol. IEEE 802.11b moreover uses *Complementary Code Keying* (CCK) to encode the signal. The maximum data rate supported by IEEE 802.11b is 11 Mbit/s.

IEEE 802.11a [16] and IEEE 802.11g [19] further increased the maximum data rate by introducing new modulation and coding schemes. IEEE 802.11a/g use *Orthogonal Frequency Division Multiplexing* (OFDM) to divide a single data stream into multiple sub-streams. Those sub-streams are then modulated on sub-carriers. The bandwidth of each sub-carrier is much smaller than the total channel bandwidth. In IEEE 802.11a/g the total channel bandwidth is 20 MHz, while each of the 52 sub-carriers is only 312.5 kHz wide. Four sub-carriers transmit known pilot tones, which are used to detect and mitigate *Inter Carrier Interference* (ICI) and frequency shifts. The remaining 48 sub-carriers are used for data transmission. Data is modulated using BPSK, QPSK or *Quadrature Amplitude Modulation* (M-QAM). M-QAM is a generalization of QPSK, which allows to use several amplitude and phase levels. For example, 64-QAM distinguishes between 8 amplitude and 8 phase levels and thereby allows to encode 64 bits per symbol. IEEE 802.11a/g transmits 250k symbols per second, which are encoded using a convolutional encoder. The maximum data rate of IEEE 802.11a/g is 54 Mbit/s. IEEE 802.11a is operated in the 5 GHz *Unlicensed National Information Infrastructure* (U-NII) radio band and offers up to 19 non-overlapping channels, while IEEE 802.11g uses the 2.4 GHz ISM band and only provides 3-4 non-overlapping channels. The number of

available channels depends on the country in which the WLAN is operated. For example, in North America there are only 11 IEEE 802.11b channels available, while in Europa 13 channels can be used.

IEEE 802.11n [24] introduces *Multiple Input Multiple Output* (MIMO) to provide even higher data rates. MIMO systems use multiple transmit and receive antennas to exploit the diversity through multiple independent transmission paths. On each independent transmission path separate spatial data streams can be transmitted simultaneously and recovered by appropriate signal processing algorithms at the receiver. IEEE 802.11n supports at maximum 4 spatial streams. IEEE 802.11n furthermore allows a more flexible use of the available spectrum. Instead of using a fixed channel bandwidth of 20 MHz, IEEE 802.11n can bond together two adjacent channels and thereby can double the throughput.

IEEE 802.11ac [170] is currently being standardized. It will allow to use channels of up to 160 MHz bandwidth and up to 8 spatial streams. With Multi-User MIMO, multiple IEEE 802.11ac devices can simultaneously communicate via separate spatial streams. Those new technologies, together with higher order modulation schemes (up to 256-QAM) will enable data rates of more than 1 Gbit/s. IEEE 802.11ad [174] is the first WLAN standard, which will use the 60 GHz band, in which abundant spectrum is available and data rates of several Gbit/s will be possible. However, the propagation properties at 60 GHz only allow short communication distances.

2.1.3 Medium Access Layer

The medium access layer controls the access to the shared wireless medium. IEEE 802.11 includes two MAC protocols: the *Distributed Coordination Function* (DCF) and the *Point Coordination Function* (PCF).

The DCF is based on *Carrier Sensing Multiple Access Collision Avoidance* (CSMA/CA). When a STA would like to transmit a frame, it first senses the medium through the *Channel Clear Assessment* (CCA) function. The CCA indicates that the medium is busy, if the energy level on the wireless medium is below -82 dBm or below -62 dBm if a frame preamble has been detected. Before attempting to transmit, the STA needs to sense the medium idle for a period specified in the *Distributed Coordination Function Interframe Space* (DIFS). After the medium has been idle for DIFS, the STA randomly chooses a *Contention Window* (CW) in the interval $[0, CW_{min} - 1]$. Each time slot the medium is idle (slot length depends on the PHY), the STA decrements the CW. If during the countdown the medium gets busy again, the STA freezes the countdown and resumes the

	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n	IEEE 802.11ac	IEEE 802.11ad
Modulation	BPSK, QPSK, M-QAM, OFDM	BPSK, QPSK, DSSS	BPSK, QPSK, M-QAM, OFDM	BPSK, QPSK, M-QAM, OFDM	BPSK, QPSK, M-QAM, OFDM	BPSK, QPSK, M-QAM, OFDM
Coding	Convolutional coding	Barker, CCK	Convolutional coding	Convolutional coding, LDPC (opt.)	Convolutional coding, LDPC, STBC	LDPC
Max. data rate (Mbit/s)	54	11	54	300	3500	1000+
Frequency range (GHz)	5.15-5.25 (U-NII low), 5.25-5.35 (U-NII mid), 5.725-5.825 (U-NII upper)	2.4-2.4835 (ISM)	2.4-2.4835 (ISM)	2.4-2.4835 (ISM), 5.15-5.25 (U-NII low), 5.25-5.35 (U-NII mid), 5.725-5.825 (U-NII upper)	5.15-5.25 (U-NII low), 5.25-5.35 (U-NII mid), 5.725-5.825 (U-NII upper)	60
Num. of 20 MHz chan. in the EU (non-overlap.)	19 (19)	13 (4)	13 (4)	13 (4) in ISM and 19 (19) in U-NII	19 (19)	-
Channel width (MHz)	20	20	20	5, 10, 20, 40	5, 10, 20, 40, 80, 160	2116

Table 2.1: Overview of IEEE 802.11 PHY layer standards

countdown after the medium was idle for DIFS. When the countdown reaches 0, the STA transmits the frame. If the receiving STA correctly receives the frame, it waits for a *Short Interframe Space* (SIFS) and sends an *acknowledgment* (ACK) to the sending STA. If the frame reception failed, for example due to a collision or a too low *Signal-to-Noise Ratio* (SNR), the sending STA detects the loss because no ACK is received within a timeout period. The sender then chooses a new CW, this time from the interval $[0, 2 * CW_{min} - 1]$. With each failed transmission attempt the possible interval size is increased to $[0, \min(2^n CW_{min} - 1, CW_{max})]$, where n denotes the number of transmission attempts (starting from 0) and CW_{max} the maximum contention window size. In IEEE 802.11b, n is limited to 6, in IEEE 802.11a to 11. CW_{max} is usually set to 1023. If a frame cannot be transmitted successfully after the maximum allowable transmissions, it is dropped. If a frame is transmitted successfully, n starts at 0 again. IEEE 802.11 defines a variant of DCF, in which STAs exchange *Request to Send* (RTS) and *Clear to Send* (CTS) frames before transmitting the actual data frame. The RTS/CTS frames specify the transmission duration of the data frame. STAs overhearing such frames set their *Network Allocation Vector* (NAV) according to the duration field in the RTS/CTS frame. The NAV thereby provides additional means to determine the status of the medium.

The DCF is based on the random selection of contention windows and therefore cannot provide any *Quality of Service* (QoS) guarantees. Hence, IEEE 802.11 describes the Point Coordination Function which allows deterministic medium access. With PCF, the time is divided into super-frames, which consist of a *Contention Period* (CP) and a *Contention Free Period* (CFP). During the CP, STAs use the DCF to access the medium. During the CFP the *Point Coordinator* (PC), typically located at the AP, regulates the medium access. The PC sends CF-Poll frames to STAs, which respond with a data frame (if available). The PC can also piggy-back CF-Poll frames on normal data frames and thereby transmit data to a STA and request data from the STA at the same time. The PCF even works in the presence of co-located IEEE 802.11 STAs that do not participate in the BSS. Before transmitting a CF-Poll frame, the PC waits only for a Point Coordination Function Interframe Space (PIFS). As the PIFS is shorter than the DIFS, other CF-Poll frames get priority over data frames transmitted by co-located IEEE 802.11 STAs. Since the duration of the CFP and the CP is known and the PC has priority on medium access, guarantees on the transmission delay of a frame and on throughput can be given. While the DCF is available in both BSSs and IBSSs, the PCF is only supported by BSSs. The PCF is not found widely

in practice. Instead, the DCF is mostly used.

In order to provide better QoS support, IEEE 802.11e introduces the *Hybrid Coordination Function* (HCF). The HCF specifies two new MAC layers, the *Enhanced Distributed Channel Access* (EDCA) and the *HCF Controlled Access* (HCCA). EDCA is similar to the DCF, but includes several new features to prioritize traffic. IEEE 802.11e defines four *Access Categories* (ACs), which represent different classes of traffic (voice, video, background, best effort). CWmin, CWmax and Interframe Spacings are AC-specific parameters. In EDCA, a STA does not wait for DIFS before counting down the contention window, but for Arbitration Interframe Space (AIFS). ACs with higher priority have smaller AIFS values. IEEE 802.11e furthermore introduces the concept of *Transmission Opportunities* (TXOPs). Once a STA has gained access to the channel, it can use it for the time duration of a TXOP without being required to contend for the channel in-between. To increase the efficiency of the MAC layer, IEEE 802.11e supports block ACKs. Inside a TXOP, several data frames can be sent which are only acked by one block ACK at the end of the TXOP. The block ACK includes a bitmap of which data frames have been received correctly. A STA has transmission queues for each AC. A virtual contention resolution mechanism prioritizes the ACs inside the STA.

The enhanced version of the PCF is called HCCA. The main difference between the PCF and HCCA is the use of TXOPs. The Hybrid Coordinator (corresponds to the Point Coordinator in the PCF) can poll a STA and allow it to transmit several data frames inside a TXOP, instead of only one data frame as supported by the PCF.

2.1.4 Finding and Associating to Access Points

Before associating to an AP, a STA needs to detect the presence of the AP. IEEE 802.11 defines two procedures to find APs: passive and active scanning. With passive scanning, the STA stays on a channel for a while, e.g. 100 ms, and waits for periodic beacon frames broadcasted by APs. The beacon frames include the SSID of the AP and information about supported PHY and QoS settings. APs typically broadcast a beacon frame every 100 ms. A STA hence needs to stay at least 100 ms on a channel to retrieve the beacons of all APs. In order to detect APs with larger beacon intervals and to account for lost beacons, a STA might be required to listen to a channel even longer. This makes the passive scan procedure slow. In the alternative procedure, the active scanning, a STA broadcasts probe request messages and waits for an AP to reply with a probe response. If the probe request includes an SSID, only APs with this SSID answer. If

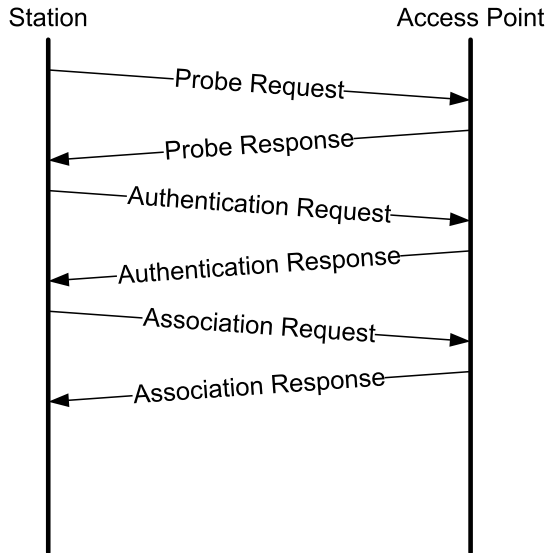


Figure 2.4: IEEE 802.11 frames exchanged when a station associates to an AP without encryption

a STA searches for multiple SSIDs, it sends one probe request for each SSID. The STA only needs to wait for a few milliseconds to get a probe response (the processing delay of the probe request and the transmission delay of the probe response). The active scanning procedure is faster than the passive one, but suffers from several drawbacks: active scanning has potential privacy implications, as it might reveal the list of previously used networks. Furthermore, active scanning might not be allowed on some channels in the 5 GHz band due to regulatory reasons.

Figure 2.4 illustrates the flow of messages when a station associates to an AP. Regardless of the scanning procedure used to detect the AP, stations usually first send a probe request message to the AP, which answers with a probe response. This is followed by the exchange of authentication messages. In the example shown in Figure 2.4 the AP does not require any authentication and encryption. Hence, only an authentication request, followed by an authentication response message needs to be transmitted. If IEEE 802.11i is enabled, additional messages are exchanged at this point to negotiate encryption keys. After a successful authentication, the station sends an association request, which is answered by the AP with an association response message. The station is now associated to the AP and data frames can be exchanged. Often, the association procedure is followed by a DHCP request for an IP address.

2.1.5 Mobility Management

When link layer encryption is used, the described association procedure is longer, because key negotiation messages are exchanged (see Section 2.1.6). In addition, a STA might request certain QoS parameters from the AP, which further prolongs the association procedure. IEEE 802.11r [21], the standard for fast BSS transition, aims to speed up the association procedure. For that purpose, IEEE 802.11r allows the exchange of key and QoS-parameter negotiation messages via the DS. While still connected to the old AP, a STA can already negotiate keys and QoS parameters for a new AP via the connectivity provided by the old AP and the DS. Once a STA hands off from the old AP to the new AP, it just needs to re-confirm the previously negotiated parameters. Alternatively, IEEE 802.11r allows a STA to piggy-back key and QoS parameter negotiation messages on Association and Authentication messages.

While IEEE 802.11r reduces the time required for a handover, it still requires the client to initiate the handover. In particular, handovers from one AP to another are triggered by the Station Management plane, which is not standardized. However, another IEEE standard, IEEE 802.21 [25], provides all primitives to perform handovers in WLANs and to other networking technologies. IEEE 802.21 defines an architecture and signaling messages to enable media independent, i.e. link layer technology independent, handovers. The focus of IEEE 802.21 are inter-technology handovers, e.g. from WLAN to UMTS, but in the case of WLAN also intra-technology handovers between APs are supported.

As Figure 2.5 shows, the *Media Independent Handover Function* (MIHF) is the central component of IEEE 802.21. The MIHF is a logical entity that facilitates handovers by providing information, event and command services. Those services enable Media Independent Handover users (MIH users) to obtain information about available networks, notifications about link layer events (e.g. link down) and to issue commands to trigger handovers. The MIHF, MIH users and the link layer communicate via abstract interfaces, the *Service Access Points* (SAPs). The MIHFs are located at STAs, but remote MIHFs are also possible. Remote MIHFs are part of the network infrastructure and communicate with the local MIHFs using layer 2 or 3 messages. IEEE 802.21 allows local and remote MIHFs. Thereby network and client initiated handovers are possible.

IEEE 802.21 alone is not sufficient to enable client mobility. The handover policy, i.e. when to handover to which network and *Point of Attachment* (PoA), is not part of the standard and needs to be implemented in a user-defined MIHF. Often handover policies are based on signal

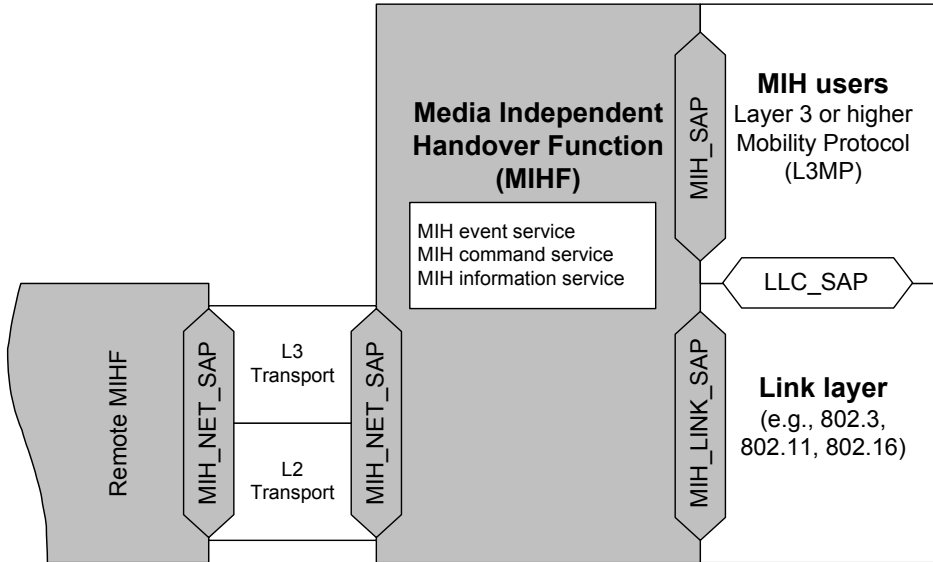


Figure 2.5: IEEE 802.21 architecture. Reproduced from [25]

strength measurements. For example, if the signal strength of the current PoA falls below a threshold, a handover to a new PoA is initiated. Another policy type hands off to a new PoA whenever the new PoA has a signal strength, which is higher than the current PoA's signal strength plus some hysteresis margin. The hysteresis margin avoids a ping-pong effect when two PoA have similar signal strengths.

IEEE 802.21 furthermore does not provide any mechanisms to reconfigure the binding between layer 2 and 3 addresses and routing after a handover. Additional protocols, such as Mobile IP [133], are required.

2.1.6 Other Relevant IEEE 802.11 Standards

Besides PHY/MAC layer and mobility management standards, the IEEE standards universe contains many other WLAN-related standards. Relevant for this thesis are mainly IEEE 802.11h, k and i. IEEE 802.11h [18] describes how to implement *Dynamic Frequency Selection* (DFS) and *Transmit Power Control* (TPC) with the aim to reduce interference for radars or satellite communication operating in the same portion of the 5 GHz spectrum. With IEEE 802.11h the AP tries to detect radars operated at the currently used channel. If a radar is detected, the AP changes its channel and requests the associated clients to also change the channel using a channel switch announcement message.

IEEE 802.11k [22] specifies frame formats and mechanisms for radio resource management in WLANs. An IEEE 802.11k station is capable of gathering data about the radio environment and exchange this data with other stations or access points. The stations measure channel properties such as channel load and noise level as well as statistics about transmitted packets and medium access delay. Those statistics can be either used locally or shared with neighboring stations. Thereby, IEEE 802.11k enables APs and stations to perform better decisions about which channel to use or when a station should trigger a handover to another AP.

IEEE 802.11i [20], or *Wi-Fi Protected Access II* (WPA2), contains procedures for station authentication, encryption key negotiation and data encryption. With IEEE 802.11i, the AP authenticates itself to the station and the station authenticates itself to the AP. To do so, the station and the AP first derive a *Pairwise Master Key* (PMK) and subsequently a *Pairwise Transient Key* (PTK). The PTK is then used to encrypt frame payloads using for example the Advanced Encryption Standard (AES). Deriving the PMK may involve communication between the AP and an external RADIUS server.

2.2 Wireless Mesh Networks

Usually, the transmission range of an IEEE 802.11 AP is in the order of several tens of meters. Higher transmission ranges might be possible if there are no obstructions between the AP and the STA and a low order modulation and coding scheme such as BPSK is used. In many scenarios a larger coverage area and higher PHY rates are needed, which can be achieved by deploying more APs. Such deployment of APs might be very costly though, in particular due to the cabling needed for the wired distribution system. In such a situation, a *Wireless Mesh Network* (WMN) might be an alternative. A WMN is a wireless multi-hop network. In a WMN only a subset of APs is connected to the wired DS, while other APs forward traffic wirelessly among each other. Different standards for implementing WMNs have been proposed. For example, IEEE 802.16 (also known as WiMAX) is a standard for metro-scale networks and also allows mesh operation. We limit the following discussion to IEEE 802.11-based WMNs, in particular IEEE 802.11s [26] WMNs, as such networks easily allow extending existing IEEE 802.11 WLANs.

2.2.1 Architecture

IEEE 802.11s is the IEEE standard for 802.11 WMNs. It uses existing PHY layers, but specifies extensions to the MAC layer and proposes new routing protocols to enable multi-hop operation. Figure 2.6 depicts an IEEE 802.11s WMN, which consists of *Mesh Access Points* (MAPs), *Mesh Points* (MPs), *Mesh Portals* (MPPs) and *Stations* (STA). MAPs/MPs can wirelessly forward MSDUs to other MAPs/MPs. A MAP in addition offers an AP service, which a STA can use to connect to the network. A Mesh Portal provides access to a wired DS. Just like in a regular IEEE 802.11 WLAN, STAs and a MAP form a BSS, but the BSSs are interconnected via a wireless DS instead of a wired DS. In the literature sometimes different terminology is used to describe the components of a WMN: MAPs/MPs are sometimes called mesh routers, mesh relay nodes or simply mesh nodes. Mesh Portals are also named gateways, and the wireless distribution system is called wireless backbone or backhaul.

In its most basic form, MAPs and MPs only use one wireless transceiver and one common channel to communicate with each other. However, this basic setup might only offer low performance, since a MP/MAP cannot receive and forward data simultaneously or neighboring wireless links might interfere with each other. It is possible to equip a MP/MAP with several wireless transceivers operated on non-interfering channels and thereby increase the capacity of the network. Such a more complicated setup requires careful assignment of channels and is not part of IEEE 802.11s. However, a considerable research effort has been undertaken to find efficient channel assignment strategies (surveyed for example in [64]).

2.2.2 Routing Protocols

Since in WMNs traffic can be forwarded via multiple-hops before reaching the final destination, routing protocols are required to find the route to the destination and set up the associated routing tables. A routing table contains `<destination, nexthop>` tuples. The MAP uses the table to decide which next hop to forward a packet to. Routing typically refers to forwarding layer 3 packets (mostly: IP packets). Sometimes the term is also applied to layer 2 forwarding. In the case of IEEE 802.11s, routing refers to forwarding MAC frames.

Two types of routing protocols are widely found in WMNs: *reactive* and *proactive* routing protocols. With a reactive-routing protocol, a route to a destination is only established upon the arrival of a packet for the destination. With this approach, nodes only need to maintain routes to destinations that are actually used. A disadvantage of reactive protocols

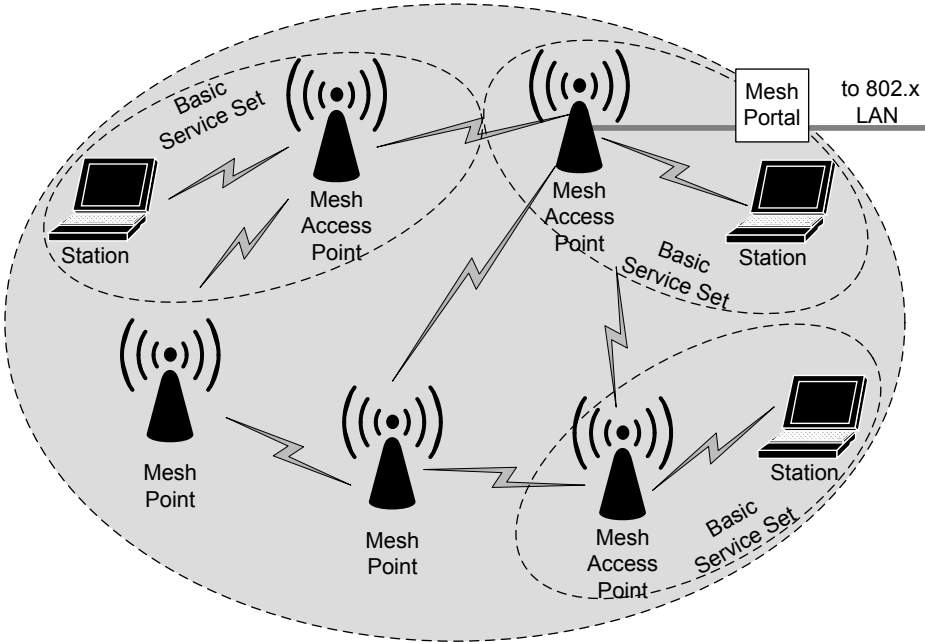


Figure 2.6: Architecture of an IEEE 802.11 mesh network

is the route discovery time: if there is no route available, the protocols first need to find a route, which can lead to significant delays for the first packets of a connection. In contrast, proactive routing protocols constantly maintain routes to all nodes in the network and therefore have a route available immediately. Which approach performs better depends on the situation. If there are only few communication pairs in the network and the network structure changes fast, e.g. due to mobility, reactive protocols are in general better. For static networks of limited size, proactive protocols are typically advantageous. If the network changes fast, proactive protocols may send many routing table updates which lead to a high overhead. To get the benefits of both worlds, hybrid routing protocols have been proposed. Hybrid protocols, such as the Zone Routing Protocol [85], are a mixture between reactive and proactive protocols. Some frequently used routes, for example to nodes within a local region, are maintained proactively, while other routes are only discovered when required.

Besides the method of finding routes, routing metrics are an important performance factor. In fixed networks, routes are often compared by their hop-count: routes with a lower hop-count generally are better. In WMNs the hop-count is usually a bad metric. A route might be composed by a few links with low data rate and high packet loss, while an alternative route

may be longer in terms of hop-count, but may only consist of high data rate links. In such a situation choosing the route with more hops would be beneficial. The *Expected Transmission Count* (ETX) [68] is one of the first routing metrics that explicitly takes into account the impact of packet loss. The ETX of a link is computed as $ETX = 1/(d_f \times d_r)$, where d_f and d_r denote the link delivery ratio in the forward and the reverse direction. The delivery ratios are usually measured with probe packets. One shortcoming of ETX is that it does not consider the speed of a link. Sometimes it can be favorable to use a fast link with higher packet loss ratio, instead of a slow link with a low packet loss ratio. The *Expected Transmission Time* (ETT) overcomes this drawback by estimating how long it takes to transfer one probe packet. It is computed as $ETT = ETX * (S/B)$, where S denotes the packet size and B the link speed. Following the example of ETT, several other routing metrics have been proposed that use the ETX or the ETT in combination with other information, for example about used channels and interference (e.g. WCETT [73] and iAWARE [162]). IEEE 802.11s [26] uses a link airtime metric. This metric computes how much airtime is consumed while transferring a packet over a specific link with a given packet loss rate. A link could for example have a low loss rate (thus it is favorable in terms of ETX), but use a slow modulation scheme. As a result, each transmission on that link would consume a lot of airtime and thereby waste resources. The airtime metric avoids to route traffic over such links. Instead it would send it over a link with a faster modulation scheme, while at the same time considering the packet loss probability.

Dozens of routing protocols for WMNs have been proposed (see for example [53]). We will restrict the following discussion to one reactive protocol, the *Adhoc On-Demand Distance Vector Routing Protocol* (AODV) [134] and one proactive protocol, *Optimized Link State Routing Protocol* (OLSR) [62]. Both protocols have found wide deployment and variants thereof are part of IEEE 802.11s. Furthermore, many other mesh routing protocols, such as [185] and [79], are variants of AODV and OLSR, for example with extensions for QoS support.

With AODV, a node broadcasts a *Route REQuest* (RREQ) message to find a path to a destination. RREQ messages include a RREQ ID and a source/destination sequence number as well as the IP addresses of the message originator, the message sender and the destination. When a node receives a RREQ message, it either answers directly with a *Route REPLY* (RREP) message if it has a fresh enough route to the destination or it rebroadcasts the RREQ message. Upon re-broadcasting the RREQ message, the node stores the RREQ ID and the source address. In addition, it sets up a reverse route to the originator via the RREQ source. A node

only re-broadcasts a RREQ message if it has not forwarded a request with the same originator address and RREQ ID before. This avoids loops of RREQ messages. The source/destination sequence numbers are monotonically increasing sequence numbers to determine the freshness of a route. When the destination receives the RREQ, it sends a RREP to the originator using the reverse route, which was set up previously when sending the RREQ.

OLSR takes a different approach to discover and maintain routes. As the name indicates, OLSR is a Link State Routing protocol. In this type of protocol, a node broadcasts the state of its links and re-broadcasts the state of its neighbors' links. As the link states are broadcasted through the network, each node gets to know the next-hop and distance for each destination in the network. Broadcasting the link state can lead to a significant amount of routing traffic. OLSR optimizes the broadcasting through the use of Multi-Point Relays (MPRs). Each OLSR node periodically broadcasts Hello messages. This allows a node to first discover all 1-hop neighbors. Subsequent Hello messages also include a list of all 1-hop neighbors of a node. Thereby, each node learns its 2-hop neighborhood. Among its one-hop neighbors, each node selects a set of MPRs so that the MPRs can directly reach all nodes in the 2-hop neighborhood of the node. A node that selects another node as MPR is called MPR selector. *Topology Control* (TC) messages are used to build routing tables beyond the two-hop neighborhood. Each node broadcasts TC messages, which include a list of its MPR selectors (instead of all its neighbors). When receiving a TC message, nodes update their routing tables.

2.3 Software Defined Networking

In this section we center our discussion around network architectures. We will introduce the idea of software defined networks, which is a new design paradigm for communication networks.

2.3.1 Towards Software Defined Networks

The computing industry has seen a tremendous progress over the past half century. This success can be in large parts attributed to open computing platforms such as the Personal Computer and the ease of writing and deploying software on those machines. Personal Computers offer an open interface, which allows for different operating systems. The operating systems themselves again provide programming interfaces, which allow application programmers to write their applications largely independent of

the hardware. This openness has fostered innovation and has led to the rise of the software industry.

Networking devices such as routers and switches, however, follow a different design strategy. In many network devices the hardware and software are tightly coupled and can only be extended by the hardware manufacturer. This design makes it hard to create new applications on top of those networking devices. The inability to re-program networking hardware has been recognized in [55] and [155] and subsequently caused the development of OpenFlow [165] and the concept of *Software Defined Networks* (SDNs).

The main idea of SDNs is to extend networking devices with standardized APIs that allow 3rd-party programmers to control the flow of data through the device and the network. In addition, SDNs provide higher level abstractions to network designers and programmers. Instead of re-implementing features like topology discovery and network access control for each application, abstractions at a higher level are provided to SDN programmers. SDNs promise to reduce the complexity of networks by making many specialized protocols obsolete. By providing networking programmers with a centralized, up-to-date view of the network, the development of network management applications is simplified significantly. Tedious tasks such as state distribution and synchronization are handled by a control layer provided by the SDN.

2.3.2 Architecture of Software Defined Networks

Different architectures for SDNs have been proposed in [28], [72] and [129]. In [55], a new architecture is proposed, in which routers and switches simply forward packets to a decision plane, which has a network view and can specify how packets are forwarded. [55] can be seen as a precursor to OpenFlow and the standardization activities of the *Open Networking Foundation* (ONF). Figure 2.7 depicts the SDN architecture envisioned by the ONF [28]. According to the ONF, an SDN can be divided into an application layer, a control layer and an infrastructure layer. The infrastructure layer contains all networking devices, which can be programmed via a control protocol such as OpenFlow. The ONF compares OpenFlow to the instruction set of a CPU as it specifies how an external application can program the forwarding logic of a networking device, just like a CPU instruction set allows to program a computer. SDN control software uses this API to implement networking services such as routing, access control or traffic engineering. The control layer then exposes APIs to the application layer. From the application layer perspective, the whole network

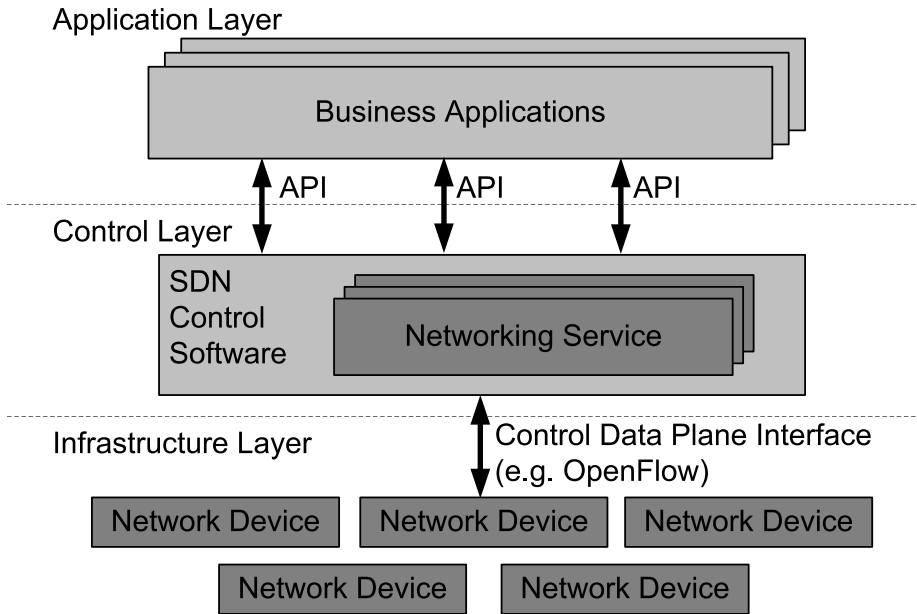


Figure 2.7: ONF architecture for Software Defined Networks

appears like a large logical switch. An application on the application layer thus does not need to know about the specifics of the network hardware or the protocols that are used to configure it.

The *Forwarding and Control Element Separation* (ForCES) Framework [72] is a proposed IETF standard to separate the forwarding and the control plane in routers. Its architecture is similar to OpenFlow and consists of control elements which remotely control the forwarding elements. [129] is a more recent Internet draft for SDN in data center networks. In this proposal, networking applications communicate with Orchestrators, the main elements on the networking control layer. Orchestrators, based on a policy and a location services database, translate application requests of the networking applications to a common protocol, which is sent to networking devices. The networking devices use plugins to transform the request into device specific or OpenFlow commands.

2.3.3 OpenFlow Protocol

An OpenFlow network (Figure 2.8) consists of OpenFlow switches and OpenFlow controllers that communicate with each other using the OpenFlow protocol. A normal switch usually has a data path and a control path.

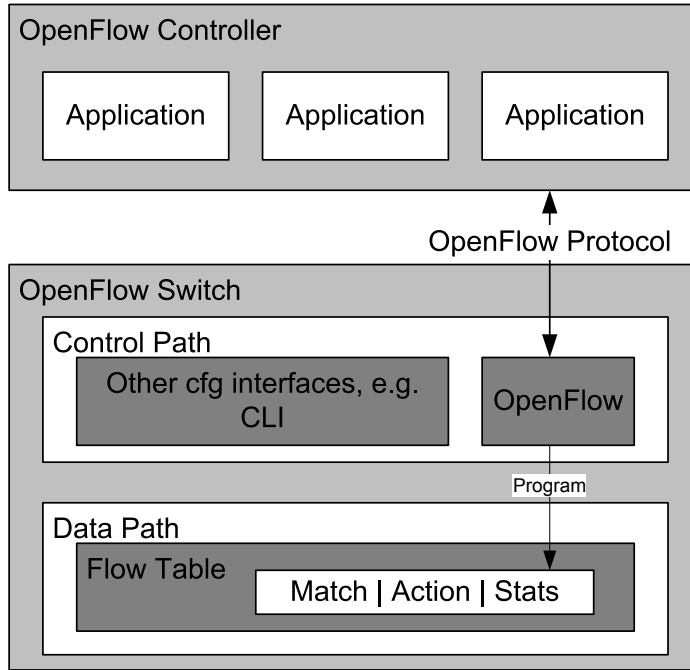


Figure 2.8: Architecture of an OpenFlow switch

The data path is responsible for forwarding packets. For performance reasons, it is typically implemented in an *Application-Specific Integrated Circuit* (ASIC). The ASIC uses a flow table to decide how to handle a packet, for example at which port to output a packet with a given destination MAC address. The control path is mostly implemented in software, often with specialized operating systems like Cisco IOS. OpenFlow switches in addition have a small control component in the data path, which allows OpenFlow controllers to configure the flow-table using the OpenFlow protocol.

The main task of the OpenFlow protocol is to configure and query flow table entries, so called rules. A rule consists of a match, an action and statistics. Matches are used to classify packets according to packet header values and wildcards, e.g. all packets with MAC source address 02:44:11:22:44:*. Possible actions are to output a packet at a specific port, to encapsulate and forward the packet to the controller, to drop the packet, to modify header fields or to send the packet to the normal switch processing pipeline. The statistics field keeps byte and packet counters about how often a rule has been matched. When a packet arrives at the switch, the flow table is processed sequentially, until a rule with

an appropriate match is found. The action is then executed and the statistics are updated. If no rule is found, the packet header is sent to the OpenFlow controller for further processing. There are both hardware OpenFlow switches from manufacturers such as Juniper and HP as well as software switches for Linux [135] available. Hardware switches have benefits when it comes to performance and scalability. They can forward data at line rate (10 GBit/s or more) and can have dozens of switch ports. In contrast, software switches can be run on cheap multi-purpose PCs or embedded hardware. With software switches, such as OpenVSwitch [135], the datapath is of course implemented in software instead of an ASIC, which often leads to lower performance when compared to hardware switches.

The OpenFlow controller hosts applications that use the OpenFlow protocol to program the network. OpenFlow applications can program the network in a reactive or a proactive mode. In the reactive mode, the application waits until it receives a packet from a switch (which had no rule for this packet). The application then decides how to handle the packet and installs a rule on the switch. In the proactive mode, an application installs rules independent of the actual traffic in the network. Each mode has its advantages and disadvantages and which mode should be used depends on the application requirements. For example, with the reactive mode, latencies in packet forwarding can occur, since a packet first needs to be sent to the controller, which then installs a rule for the packet. However, the newly installed rule usually matches all packets of the same data flow (e.g. TCP flow). Hence, the controller only needs to be consulted for the first packet of a flow. In the proactive mode, no such forwarding latencies occur, but it may be hard to know beforehand which rules will be required later. One OpenFlow controller can serve multiple switches and one switch can connect to multiple controllers. How to place controllers and how many controllers are required for a network is part of ongoing research ([168] and [90]). There are several OpenFlow controllers available, for example NOX [83], Floodlight [4] and Meastro [51].

2.4 Flow Optimization

In this section we introduce a mathematical tool, flow optimization, which allows us to compute the optimal paths that flows should take through a network. We first provide a brief introduction into mathematical optimization, which is the foundation of flow optimization. Then we will present different flow optimization problems.

2.4.1 Mathematical Optimization

Mathematical optimization is a sub-branch of applied mathematics, which studies problems of the following kind [59]:

$$\underset{\mathbf{x}}{\text{maximize}} \ g(\mathbf{x}) : \mathbf{x} \in \Omega$$

where $g(\cdot)$ is called *objective function*, \mathbf{x} is a vector of *decision variables* and Ω is the *feasible set*. The goal is to find a solution vector \mathbf{x} , which is in the feasible set Ω and maximizes the objective function $g(\cdot)$. Typically, Ω is described by a set of inequalities and a domain (e.g. an n -dimensional vector of real numbers, \mathbb{R}^n). Optimization problems are usually classified depending on the type of g and the structure of Ω .

Linear Programs (LPs) are the most simple type of optimization problems. An LP is an optimization problem in which the objective function is linear and the feasible region is described by a set of linear inequalities. An LP hence can be expressed in the following form:

$$\underset{\mathbf{x}}{\text{maximize}} \ g(\mathbf{x}) = \sum_{j \in \{1, \dots, n\}} c_j x_j \quad (2.1)$$

subject to

$$\sum_{j \in \{1, \dots, n\}} a_{ij} x_j \leq b_i \quad \forall i \in \{1, \dots, m\} \quad (2.2)$$

$$x_j \geq 0 \quad \forall j \in \{1, \dots, n\}. \quad (2.3)$$

This linear program has m constraints and n decision variables, a , b and c are constants. We aim to find a feasible solution vector $\mathbf{x} = (x_1, \dots, x_n)^T$ which maximizes the objective function $g(\mathbf{x})$. LPs have been studied widely since the 1950s, which has led to the development of efficient algorithms to solve LPs. On standard PCs, LPs with thousands of decision variables and constraints can be solved [124]. Many real world problems, such as scheduling, production planning and flow optimization problems, can be described as LPs. However, some real-world problems cannot be described with continuous decision variables, but require binary or integer decision variables. If all decision variables are constrained to be either 0 or 1, then the problem is called *Binary Integer Program* (BIP). If all decision variables are constrained to be integers, we call this problem an *Integer Program* (IP). If some decision variables are limited to be integers and the others are continuous variables, then the problem is called *Mixed Integer Linear Program* (MILP). Due to the integrality constraints of some variables,

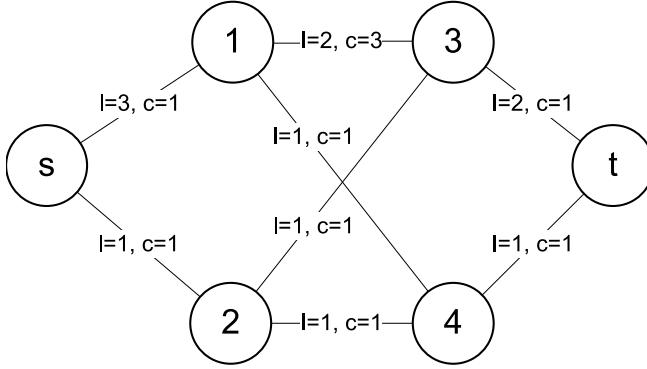


Figure 2.9: Example graph. l denotes the capacity of the edge, c the cost of sending one unit of flow. s is source, t is sink.

MILPs are usually harder to solve than LPs. While LPs are known to be solvable with polynomial time algorithms, MILPs are classified as NP-hard problems [78]. This means that most likely no algorithm can be found that solves MILPs in polynomial time. Nevertheless, many algorithms (e.g. Branch and Cut [59]) have been developed that are able to solve MILPs efficiently.

2.4.2 Flow Optimization

A graph $G(V, E)$ consists of a set of vertices V and a set of edges E . Each edge connects two vertices. In the example graph shown in Figure 2.9, $V = \{s, 1, 2, 3, 4, t\}$ and $E = \{(s, 1), (s, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, t), (4, t)\}$. In literature, vertices are sometimes also called nodes and edges are also occasionally named arcs or links. Graphs, or networks, have found a wide application to model real world systems, as for example road networks or data communication networks. Network elements such as routers and switches are vertices and the connections between them are edges. Flow optimization problems aim to optimize the flow of commodities or data through a network. The most studied flow optimization problems are the max-flow and the min-cost problem.

2.4.2.1 Max-Flow Problem

The aim of the max-flow problem is to find the maximum flow that can be sent from a source node s to a sink node t while respecting the capacity $l_{(i,j)}$ of all edges (i, j) . The optimization problem can then be written as the following LP:

$$\text{maximize } F \tag{2.4}$$

$$\sum_{j:(i,j) \in E} f_{(i,j)} - \sum_{j:(j,i) \in E} f_{(j,i)} = \begin{cases} F & \text{if } i = s \\ -F & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V \tag{2.5}$$

$$0 \leq f_{(i,j)} \leq l_{(i,j)} \quad \forall (i,j) \in E. \tag{2.6}$$

Equation 2.4 maximizes the flow F . Equation 2.5 is the flow conservation constraint, which ensures that the sum of the incoming and the outgoing flows is 0 for each node, unless the node is the source or the sink node. In this case, the net-flow needs to be equal to F or $-F$. Equation 2.6 states that the flow on an edge must be positive and cannot exceed the capacity of the edge.

For our example graph, the maximum s - t flow is 3. The solution for this problem instance is not unique. One potential solution is $f_{(s,1)} = 2$, $f_{(s,2)} = 1$, $f_{(1,3)} = 1$, $f_{(1,4)} = 1$, $f_{(2,3)} = 1$, $f_{(3,t)} = 2$ and $f_{(4,t)} = 1$. Another possible solution is $f_{(s,1)} = 3$, $f_{(1,3)} = 2$, $f_{(1,4)} = 1$, $f_{(3,t)} = 2$ and $f_{(4,t)} = 1$ (other flows are 0). The example also illustrates that it is not necessary to send the whole flow via one path. It is possible to split a flow up, as for example there are 3 incoming units of flow in node 1, which are split up into two units and one unit towards node 3 and node 4 respectively. The problem formulation above constitutes a linear program that can be solved with any standard solution algorithm, such as the Simplex algorithm [59]. Alternatively, specialized algorithms with a lower computational worst-case complexity than most LP solution algorithms can be used. For example, the Ford–Fulkerson or Edmonds–Karp algorithms are widely used to solve max-flow problems [32].

The problem above is sometimes also called $s - t$ max-flow problem to distinguish it from other maximum flow problems. Another prominent example of a max-flow problem is the multi-commodity max-flow problem [32]. In this problem, there are multiple types of flows, each with a distinct source and destination pair. The aim is then to maximize the sum of all flow types (commodities). Another variant is the unsplittable flow problem. Here an additional constraint is introduced, which forces all outgoing flows of one node to be sent via one edge. Even though this problem seems very similar to the original s - t max-flow problem, it is much harder solve, as it requires the use of integrality constraints, which make an LP formulation or the use of the aforementioned efficient algorithms impossible.

2.4.2.2 Min-Cost Flow Problem

The min-cost problem is similar to the max-flow problem. As the problem name indicates the aim is to minimize the cost of sending a flow through the network. In this problem, each edge has a capacity of $l_{(i,j)}$ and shipping one unit of flow costs $c_{(i,j)}$. The flow F this time is given and the link flow rates are computed by solving the following problem:

$$\text{minimize } \sum_{(i,j) \in E} f_{(i,j)} c_{(i,j)} \quad (2.7)$$

subject to Equation 2.5 and 2.6. For the example graph in Figure 2.9, the minimum cost flow for a flow of $F = 3$ is: $f_{(s,1)} = 2$, $f_{(s,2)} = 1$, $f_{(1,3)} = 1$, $f_{(1,4)} = 1$, $f_{(2,3)} = 1$, $f_{(3,t)} = 2$ and $f_{(4,t)} = 1$. Every s - t max-flow problem can be converted into a min-cost flow problem by setting the costs to 0 and by adding an $s - t$ edge with negative cost to the network. Hence, solution methods for the min-cost flow problem are also applicable to max-flow problems. The shortest-path problem, i.e. the problem of finding the shortest path between two nodes, is a special case of the min-cost flow problem, in which $F = 1$ and $l_{(i,j)} = 1$ for all $(i,j) \in E$.

2.4.2.3 Applications

Both the min-cost and the max-flow problem have found wide application in data communication networks, for example to optimize the routing in MPLS [147, 111] and in wireless mesh networks [33]. In addition, the max-flow problem can also be used to solve other related problems. For example, the max-flow problem can be used to solve the bipartite matching problem, in which one aims to find the largest collection of edges in a bipartite graph, such that no two edges share a node [59].

Modeling the Channel Load in IEEE 802.11 Networks

3.1 Introduction

In this chapter we investigate the relationship between the offered load in an IEEE 802.11 WLAN, the throughput of individual stations and the load of the wireless channel. We use the channel busy fraction, that is the fraction of time that the channel is busy, as a measure of the channel load. Understanding the relationship between the network and the channel load is a key to develop effective mechanisms for admission control, channel assignment and AP selection. Ideally, one would like to predict how the load of the wireless channel changes with the offered load in the network. Congested networks manifest themselves in high latency, jitter and packet loss, which should be avoided in order to provide predictable performance.

3.1.1 Related Work

Analytical models of the IEEE 802.11 MAC have attracted the attention of the research community for a long time. Bianchi's seminal paper [46] on the performance analysis of the IEEE 802.11 distributed coordination function for the first time presented a simple, but accurate model of the IEEE 802.11 MAC layer. Bianchi's model is based on Markov chains and allows to predict the throughput of a single-cell, saturated IEEE 802.11 WLAN. Bianchi's model was subsequently extended for example to compute frame delays and the throughput on lossy links. For instance, [54] uses Bianchi's model to compute the channel access delay and jitter in saturated IEEE 802.11 networks. Malone's model [118] is an interesting

extension of Bianchi's model that allows to compute the throughput and delay in IEEE 802.11 WLANs under non-saturation conditions. [177] proposes a model for the IEEE 802.11e MAC layer, which allows several transmission classes. [46], [54], [118] and [177] assume that collisions are the only cause of frame errors and loss. In WLANs, frames can also get corrupted if the signal strength is too low or other noise destroys the frame. This modeling limitation was addressed for example in [184] and [58], where frame losses due to bit-errors in saturated networks are considered. [67] extends Malone's model to incorporate the effects of noise related frame loss and capturing.

The aforementioned models are generic analytical models with no particular application in focus. [149] is a more specific application of a Bianchi-type model. It aims to find high throughput paths in IEEE 802.11 mesh networks. To identify such paths [149] relates the achievable throughput of a station with the load of the channel. The channel load is expressed as channel busy fraction, that is the fraction of time the channel is busy. The channel busy fraction is also used in [183], where an analytical model is derived to predict packet loss, delay and jitter for the IEEE 802.11 DCF. The authors observe that the channel busy fraction is a good indicator for the collision probability. However, their model does not allow to correlate the channel busy fraction with the traffic pattern and the MAC layer configuration. [30] uses the channel busy fraction in a rate-adaptation algorithm. The channel busy fraction helps to distinguish between bit-error related losses and losses due to collisions. [154] uses the channel busy fraction for admission control in wireless mesh networks.

3.1.2 Problem Statement and Contributions

The work reviewed above shows that the channel busy fraction is a good metric to link real world measurements with analytical models. However, current models do not allow to relate the channel busy fraction with key parameters such as the network size, the MAC layer configuration and the offered load. Such a model is desirable, as it allows to study the network behavior and channel busy fraction from a theoretical perspective, but also to apply this model in real systems which can measure some input parameters, such as the channel busy fraction. In this chapter we thus address this open research problem by extending Malone's model for IEEE 802.11 DCF under non-saturation conditions [118] to incorporate the channel busy fraction. By using the new model we can study the channel busy fraction as a function of different key parameters such a network size, frame lengths and offered load.

The key contributions of this chapter are:

- An **extension of Malone’s model** to compute the channel busy fraction for a given network size and load. The model allows to predict the channel load in IEEE 802.11 non-saturated IEEE 802.11 WLANs.
- The **validation of the model** through measurements in a wireless testbed. The measurements show a good correspondence with the model prediction. Moreover, using multiple linear regression analysis we show that a linear approximation of the model predictions is possible.

The remainder of the chapter is organized as follows. In Section 3.2 we first introduce Bianchi’s and Malone’s MAC models and extend them to compute the channel busy fraction. In Section 3.3 we validate the model predictions through testbed measurements. Finally, we conclude this chapter with Section 3.4.

3.2 Analytical Model of the IEEE 802.11 MAC

In this section we first review Bianchi’s model of the IEEE 802.11 DCF. We then discuss Malone’s model for non-saturated networks, in which we subsequently incorporate the channel busy fraction.

3.2.1 IEEE 802.11 DCF under Saturation Conditions

Bianchi’s seminal paper [46] presents an analytical model of the IEEE 802.11 DCF for a homogeneous network with n stations under saturation conditions. Saturated means that each station at any time has a packet available in its transmission queue. Homogeneous refers to the requirement that the MAC layer configuration of all stations is identical. The backoff process is modeled as a Markov-chain with states (i, k) , where i denotes the backoff stage and can take values from 0 to m . After each failed transmission, i is increased by 1. After a successful transmission i is set back to 0. The backoff window k is initially chosen randomly and uniformly from $[0, W_i - 1]$, where $W_i = 2^i W_0$ and $W_0 = CWmin$. When the channel is idle during a slot of length σ , k is decremented by 1. When k reaches 0, the station transmits.

Bianchi describes how to compute the steady-state probabilities $b(i, k)$ of the Markov chain. In particularly interesting are the probabilities of

the ready-to-send states $(i, 0)$, as they allow to compute the probability τ that a station transmits in a given slot:

$$\tau = \sum_{i=0}^m b(i, 0). \quad (3.1)$$

The probabilities $b(i, 0)$ are somewhat difficult to obtain, but taking into account the transition probabilities in the Markov chain one can show that τ can also be computed when only $b(0, 0)$ and the packet loss probability p are known. The transmission probability then is calculated as

$$\tau = \frac{b(0, 0)}{1 - p}. \quad (3.2)$$

What remains is to compute $b(0, 0)$ and p . Again, by analyzing the transition probabilities, one finds that $b(0, 0)$ can be expressed as

$$b(0, 0) = \frac{2(1 - 2p)(1 - p)}{(1 - 2p)(W_0 + 1) + pW_0(1 - (2p)^m)}. \quad (3.3)$$

A collision occurs if more than one station transmit in a given slot. Hence p is given by

$$p = 1 - (1 - \tau)^{n-1}. \quad (3.4)$$

Finally, equations 3.2 and 3.4 form a system of two equations, which can be solved for τ and p . The system throughput S is defined as the ratio of time spent on payload transmissions and time spent by successful transmissions, collisions or in an idle channel (see also Table 3.1). It can be written as

$$S = \frac{P_s P_{tr} E}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s)T_c}. \quad (3.5)$$

3.2.2 IEEE 802.11 DCF under Non-Saturation Conditions

Malone et al. [75, 118] extended Bianchi's model to also account for non-saturated and heterogeneous networks. For simplicity, we only present here how to apply the model to homogeneous networks. For details on the model and an extensive discussion, the reader is referred to [118].

The key idea is to introduce additional states $(0, k)_e$, which represent the case where a node has transmitted a packet and no further packet is waiting to be transmitted. The probability $b(0, 0)_e$ is of particular interest. In the stage $(0, 0)_e$ a node has completed the counter decrement, but no packet to transmit. The inverse of the steady-state probability of the state $(0, 0)_e$ is

Symbol	Description
$P_s = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}$	Probability of a successful transmission
$P_{tr} = 1 - (1 - \tau)^n$	Probability of a transmission by any station
$T_s = T_{difs} + T_{phy} + T_{mac} + T_{data} + T_{sifs} + T_{phy} + T_{ack} + 2T_{prop}$	Time to successfully send a packet
$T_c = T_{difs} + T_{phy} + T_{mac} + T_{data} + T_{prop} + T_{sifs} + T_{ACKtimeout}$	Time spent on the collision of a packet
$E = T_{data}$	Time to send the payload (incl. IP header)
$T_{phy}, T_{mac}, T_{ack}$ and T_{data}	Time to send the PLCP preamble, the MAC header, an ACK and the data payload
σ	MAC layer slot time
T_{difs}	DIFS duration
T_{sifs}	SIFS duration
T_{prop}	Propagation time
$T_{ACKtimeout}$	ACK timeout duration

Table 3.1: Used symbols and description

$$\begin{aligned}
1/b_{(0,0)e} = & (1-q) + \frac{q^2 W_0 (W_0 + 1)}{2(1 - (1-q)^{W_0})} + \frac{q(W_0 + 1)}{2(1-q)} \left(\frac{q^2 W_0}{1 - (1-q)^{W_0}} + \right. \\
& + p(1-q) - q(1-p)^2 \Big) + \frac{pq^2}{2(1-q)(1-p)} \left(\frac{W_0}{1 - (1-q)^{W_0}} - \right. \\
& \left. \left. - (1-p)^2 \right) (2W_0 \frac{1-p-p(2p)^{m-1}}{1-2p} + 1) \right). \tag{3.6}
\end{aligned}$$

In analogy to Equation 3.3 the transmission probability τ is

$$\tau = b_{(0,0)e} \frac{q^2}{1-q} \left(\frac{W_0}{(1-p)(1 - (1-q)^{W_0})} - (1-p) \right). \tag{3.7}$$

q denotes the constant probability, that at least one packet arrives within the expected duration of a state in the Markov chain. It is related with the input rate and the traffic pattern. For a Poisson arrival pattern and small buffers, q can be approximated by $q = 1 - e^{-\lambda T}$, with T being the expected duration of a state computed by the denominator of Equation 3.5 and λ being the packet input rate (pkts/s). [118] also provides a more precise definition of q , which does not utilize mean state durations, but requires a complicated calculation of mean MAC delay. Since simulation results show that the approximation is satisfactorily precise, we subsequently use the approximation. By substituting τ in Equation 3.4 with Equation 3.7 the collision probability p is obtained. Finally, Equation 3.5 gives the system throughput.

3.2.3 Modeling the Channel Busy Fraction

Based on Bianchi's and Malone's model we define the channel busy fraction. A channel is busy either due to successful transmissions for a period of $P_{tr}P_sT_s$ or due to collisions for a period of $P_{tr}(1-P_s)T_c$. We define the channel busy fraction as the ratio of the busy periods and all channel states (busy or idle), which can be expressed as

$$cbf = \frac{P_{tr}P_sT_s + P_{tr}(1-P_s)T_c}{(1-P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}(1-P_s)T_c}. \tag{3.8}$$

A node senses both its own transmissions and other transmissions. If all nodes are within the same sensing range, all sense the same channel busy fraction. In contrast to [183], which approximates the channel busy fraction by only counting successful transmissions as busy, Equation 3.8 also takes into account medium usage by collisions. By that, Equation 3.8 is more accurate, in particular for higher loaded networks.

3.2.4 Discussion

Next, we apply the model (parameters see Table 3.2) to study the channel busy fraction in different settings. Figure 3.1 shows the aggregate throughput (primary y-axis) and the busy fraction (secondary y-axis) for 4, 8 and 12 nodes and 1400 byte UDP packets. In the rest of the chapter packet sizes denote the UDP payload length (without UDP/IP header) and the aggregate throughput is the sum of the UDP throughputs of all nodes. As already known from [118], the peak performance is reached before saturation. Going beyond this point towards a saturated network will result in more collisions, but not more throughput. Instead, the throughput decreases, especially for larger networks.

While the channel busy fraction at which the peak performance is achieved is relatively insensitive to the network size, the packet size plays a more important role. Figure 3.2 displays the busy fraction which gives the maximum throughput for 200, 500 and 1400 byte UDP packets. For 1400 byte packets the peak performance is reached at the channel busy fraction of about 95%, almost irrespective of the network size. For 200 byte packets the performance peak is already reached at 92% busy fraction. With smaller packets the stations access the wireless channel more frequently and thus have a higher chance of creating a collision. Due to the higher collision probability and the increased overhead for sending small packets, the overall throughput decreases when small packets are sent.

3.2.5 Limitations of the Model

The model is based on two assumptions: an error-free channel and perfect carrier sensing. The first assumption implies that packets are only lost due to collisions, not due to bit errors caused by channel noise. Following the argumentation in [182], a crude approximation of the effect is possible by introducing a new probability p_e , which denotes that a data packet is corrupted by bit errors. The probability of a successful transmission is then given as the joint probability of exactly one transmission and no bit error:

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}(1-p_e). \quad (3.9)$$

The second assumption is only valid if there are no hidden terminals. A hidden terminal is a node that cannot sense an ongoing transmission of another node and hence might start its own transmission and thereby destroy the ongoing transmission. Hidden nodes are a well known problem of CSMA/CA networks. However, it is unclear to what extent hidden terminals can be found in practical network deployments. The use of

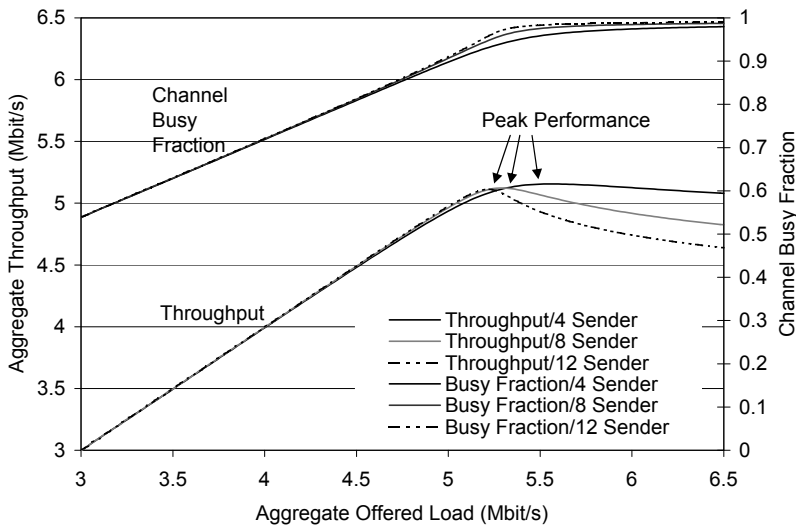


Figure 3.1: Channel busy fraction and throughput as function of the aggregate offered load. The highest throughput is achieved before the network is saturated.

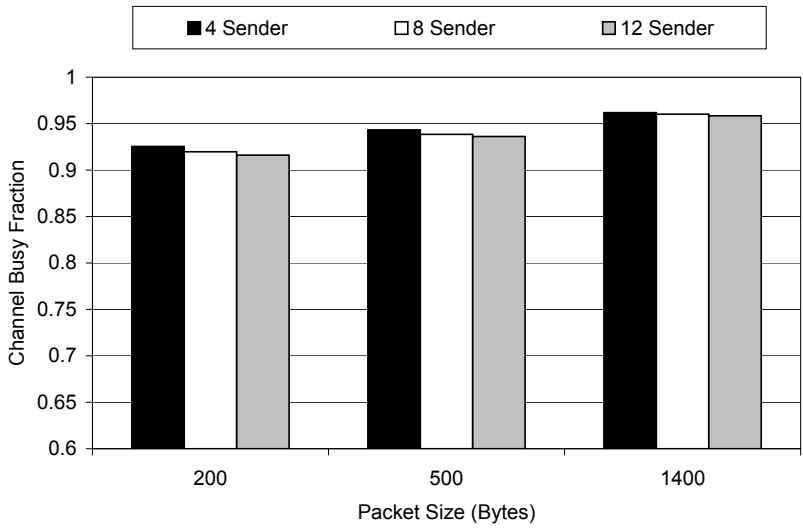


Figure 3.2: Channel busy fraction which gives peak performance. Both the network and packet size have relatively little impact.

RTS/CTS, transmission power control and tuning of the carrier sensing threshold can theoretically alleviate the problem. One explanation why those mechanisms are not widely deployed is that hidden terminals do not occur too frequently or their negative impact is low.

Considering hidden terminals requires a different, more complicated class of analytical models, such as [137]. Therefore, we do not consider hidden terminals in our model. In addition, even if the model predictions get inaccurate by this design choice, the model can still be useful. For example, one could apply the model and use information about a significant difference between the model prediction and real measurements for example as an indicator of hidden nodes.

3.3 Validation of the Model

In this section we validate the model by comparing the model predictions to measurements obtained from a testbed.

3.3.1 Experimental Setup

WLAN cards that implement the IEEE 802.11k standard [22] allow extracting the channel busy fraction from the channel load report. In our testbed we used cards with the wide-spread Atheros 5212 802.11a/b/g chipset. Those cards do not support IEEE 802.11k, but the hardware registers `PROFCNT_RXCLR` (0x80f4) and `PROFCNT_CYCLE` (0x80f8) contain information about the number of timeslots that were sensed busy due to the Clear Channel Assessment (CCA) and the total number of timeslots that have passed. The length of a timeslot depends on the card clock and is approximately 2.5×10^{-8} seconds for Atheros 5212. Note, that those timeslots are not related to the MAC layer timeslots of length σ .

According to IEEE 802.11a [16] the CCA reports the channel idle if the energy level is below a threshold (-82 dBm if a preamble was detected, -62 dBm otherwise). We extended the driver to export the content of the two registers to the `/proc`-filesystem. A userspace application reads the values every 500 ms. The channel busy fraction is `PROFCNT_RXCLR/PROFCNT_CYCLE`. The model assumes that the medium is also busy during periods of SIFS or DIFS. Since the CCA detects those periods as idle we correct the measured `PROFCNT_RXCLR` by a T_{sifs} and T_{difs} for each overheard or sent packet (using antenna statistics from the `athstats` utility [10]). In contrast to IEEE 802.11k, the NAV from RTS/CTS packets is not considered in the proprietary Atheros interface for determining the channel busy fraction.

Parameter	Value
T_{sifs}	16×10^{-6} sec.
T_{difs}	34×10^{-6} sec.
Backoff slot time σ	9×10^{-6} sec.
Data rate/Basic rate	6 Mbit/s
ACK timeout $T_{ACKTimeout}$	25×10^{-6} sec.
CW_{min}/W_o	15
CW_{max}	1023
Max. number of retransmissions	5
Payload length (incl. UDP/IP header)	1428 bytes
RTS/CTS	Disabled

Table 3.2: Parameters of the model and the experiments

We have validated the model with five Cambria GW2358-4 network computers placed on a desk in a lab room. Each node is equipped with three network cards. The nodes run Linux 2.6.22 and MadWifi 0.9.4. One node acted as receiver, four nodes generated UDP traffic with mgen [11]. Since the backoff processes run independently on all cards, four nodes are sufficient to emulate 12 senders. We sent UDP packets with 1400 bytes payload according to a Poisson arrival process. In Figures 3.3 and 3.4 each datapoint represents the average of a 4 minute testrun. Further settings are listed in Table 3.2.

3.3.2 Channel Busy Fraction and Traffic Injection Rate

We now compare the predictions of the model with the measurements from our testbed. Figure 3.3 depicts the channel busy fraction as a function of the offered load. For aggregate offered loads smaller than 5 Mbit/s the channel busy fraction grows roughly linearly with the offered load. In these low loaded conditions the impact of collisions is low and the numerator of Equation 3.8 is governed by $P_{tr}P_sT_s$, while $P_{tr}(1 - P_s)T_c$ is small. After 5 Mbit/s the busy fraction increases faster than the offered load, since collisions are more likely then. The model slightly underestimates this disproportionate growth and the maximum achievable channel busy fraction. As we will discuss below, this is because the model underestimates the number of lost packets.

Next, Figure 3.4 shows the average number of transmissions as a function of the busy fraction. We read the average number of packet transmissions from the file `/proc/net/madwifi/athX/ratestats_1600`.

In the model it is obtained using Equation 3.4 as input to the function $avg(p) := \sum_{i=1}^{m+1} p^{i-1}(1-p)i$. We use $avg(p)$ to calculate the average number of transmissions for a packet and compare it with the value from the `/proc`-file. As in [46], we assume the collision probability p to be the same for each of the at maximum m transmission attempts and that the attempts are independent of each other.

As mentioned above, for lightly loaded networks the average number of transmissions is close to one. Only when the load and thus the busy fraction increases, the average number of transmissions rises sharply at a busy fraction of about 90%. The model underestimates the average number of collisions and is more accurate for small networks (4 nodes). While in the model the transmission number is insensitive to the network size, in the experiments there is a difference between small and large networks.

Despite great efforts, we were not able to find the cause why the quantitative predictions of the model are inaccurate in the transition from a non-saturated to a saturated network. We assume one or a combination of the following reasons: queuing effects, traffic generation and a non-standard backoff process. The modeled relation between q and λ is an approximation for very short queues. In the experiment we used a small hardware buffer (5 frames) to reduce effects caused by variations in the time from packet generation until delivery to the hardware. Using simulations, [74] has shown that the buffering process has a significant impact on the throughput and collisions. Also, due to randomly occurring interrupts the arrival process of packets at the hardware is not exactly the Poisson one (although generated in Poisson manner with `mgen` [11]). Finally, according to [47], many IEEE 802.11 cards do not back-off according to the IEEE 802.11 standard on which the model is based.

Nevertheless, the model is sufficiently precise to provide useful qualitative predictions. For low loaded networks, also the quantitative predictions are accurate. Although the characterization of the transition into the saturation state is an interesting theoretical problem, real networks should be operated under non-saturation conditions anyhow. In this situation, the model matches the testbed measurements quite well.

3.3.3 Linear Model of the Channel Busy Fraction

The model and the experimental data in large parts show linear relationships between the offered load, the number of senders and the channel busy fraction. The question arises, if it is possible to find a simple, linear model that produces accurate predictions of the more complicated Markov model described above or the experimental data. To investigate this question we

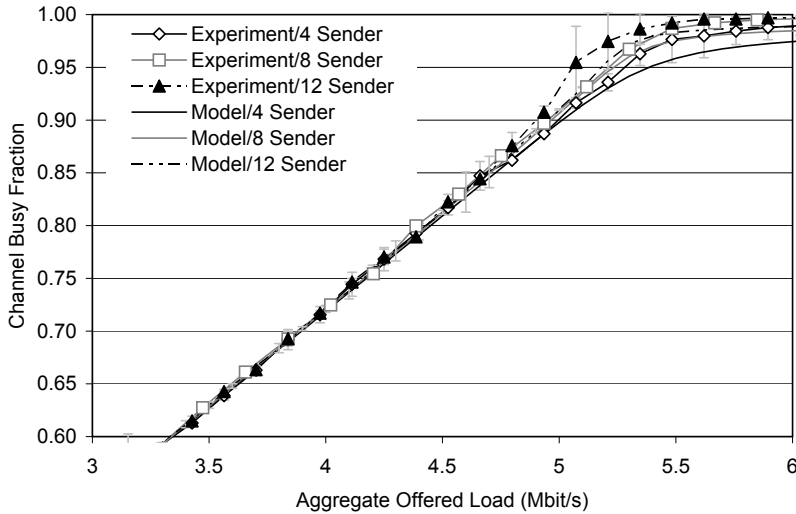


Figure 3.3: Channel busy fraction and traffic offered load. The channel busy fraction grows almost linearly with the offered load until the network gets saturated. Then channel busy fraction grows faster due to the increasing collision probability.

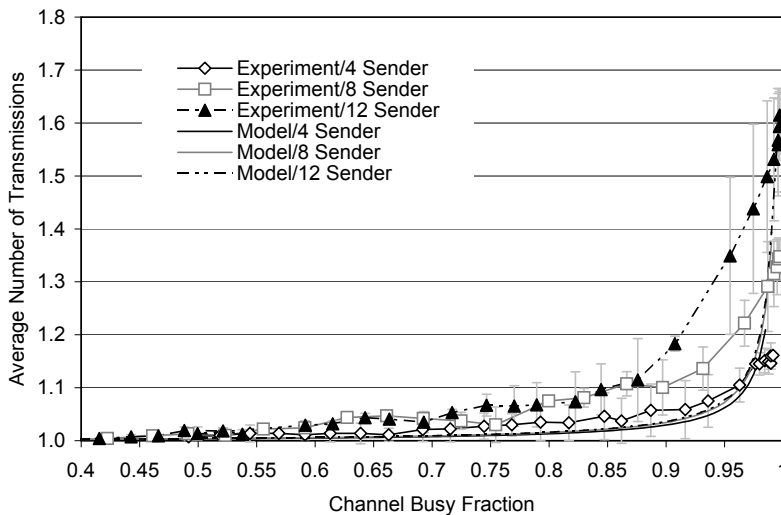


Figure 3.4: Channel busy fraction and average number of transmissions. The average number of transmissions is close to 1 as long as the network load is low. When the load increases, the collision probability and the average number of transmissions increase.

fitted a *Multiple Linear Regression* (MLR) model to the data. A MLR model in general has the form

$$y = \alpha + \sum_k \beta_k x_k + \epsilon, \quad (3.10)$$

where y is the dependent variable and $x_1 \dots x_k$ denote explanatory variables. The parameters to be estimated are the coefficients $\beta_1 \dots \beta_k$ and the intercept α . With ϵ we denote an error term. Our explanatory variables are the offered load and the number of senders. Our dependent variable y is the channel busy fraction. We used the R programming language [139] to estimate the parameters and to analyze the goodness-of-fit of the model.

Tables 3.3 and 3.4 show the parameters estimations, their standard errors and t and p values of the linear model predictions. The parameter estimations explain how much each explanatory variable contributes to the dependent variable. The t value is computed as the estimate divided by the standard error. The p-value, in the last column of the table, allows us to determine which of the explanatory variables in the MLR model are statistically significant. The p value is the probability of rejecting the null hypothesis (H0) when that hypothesis is true. In our case the null hypothesis is $\beta_k = 0$. Typically, significance levels of 0.05 are used when computing the p value. This means, if the p value of a explanatory variable is smaller than 0.05, then the variable is statistically significant and should therefore be included in the model. As can be observed from the tables, both variables are statistically significant in the Markov model data, while in the experimental data only the offered load is significant. For the Markov model data we in addition tested and confirmed the frame size as a significant variable (not shown in the table). For both data sets the R^2 statistic is close to one, which indicates a close agreement of the model predictions and the data sets.

The analysis shows that a simple linear model, that only takes into account the offered load, can compute a relatively good approximation of the channel busy fraction. This is also in line with the observations presented in [66], which show a linear relationship between the offered load and the throughput.

3.4 Conclusions

In this chapter we have explored the channel busy fraction by an analytical model and by testbed experiments. We have shown that our model predicts the testbed results qualitatively and quantitatively reasonable well in lightly loaded networks. In networks with higher load the model provides correct

	Estimate	Std. Error	t value	Pr(>t)
Intercept	0.1449	0.0082	17.582	0.0000
Offered Load	0.1416	0.0015	93.778	0.0000
Number of Senders	0.0011	0.0004	2.499	0.0128

Table 3.3: Summary of the linear regression for the Markov model data. $R^2 = 0.94$.

	Estimate	Std. Error	t value	Pr(>t)
Intercept	0.0518	0.0176	2.94	0.0043
Offered Load	0.1637	0.0032	50.89	0.0000
Number of Senders	0.0008	0.0011	0.73	0.4685

Table 3.4: Summary of the linear regression for the experimental data. $R^2 = 0.97$.

qualitative predictions. We further showed that for lightly loaded networks the channel busy fraction can be approximated well with a simple linear model. We will leverage this insight in the following chapters to build (mixed) linear optimization models for IEEE 802.11 networks, which will use the channel busy fraction as a key component.

Optimization of Associations in WLANs Considering Handover Costs

4.1 Introduction

Many commercial WLANs are deployed with a considerable overlap between the coverage areas of adjacent APs. Consequently, users can often choose which AP to connect to. For example, Figure 4.1 shows the results of a site-survey on the Karlstad University campus, which was made using the Cisco Prime Network Control System software [2]. With IEEE 802.11g and a PHY rate of 24 Mbit/s, roughly 55% of the users in the coverage area can choose from only one AP. However, at 45% of the locations, users can choose between 2, 3 or even 4 APs. With 12 Mbit/s even more choices are possible, as this *Modulation and Coding Scheme* (MCS) requires a lower signal strength for successful decoding.

In current systems, end users select an AP to associate with typically using the Received Signal Strength Indicator (RSSI). This leads to unequal resource usage and poor performance. Recently, especially in enterprise WLAN deployments, centralized management schemes have become more and more interesting as they allow to exercise more control on the STA/AP associations. However, finding the best AP for a user station (STA) is non-trivial, as it depends on many factors such as signal strength, interference and the load of the AP. Furthermore, the best AP for a STA might change over time, for example due to mobility or time-variant interference of other users.

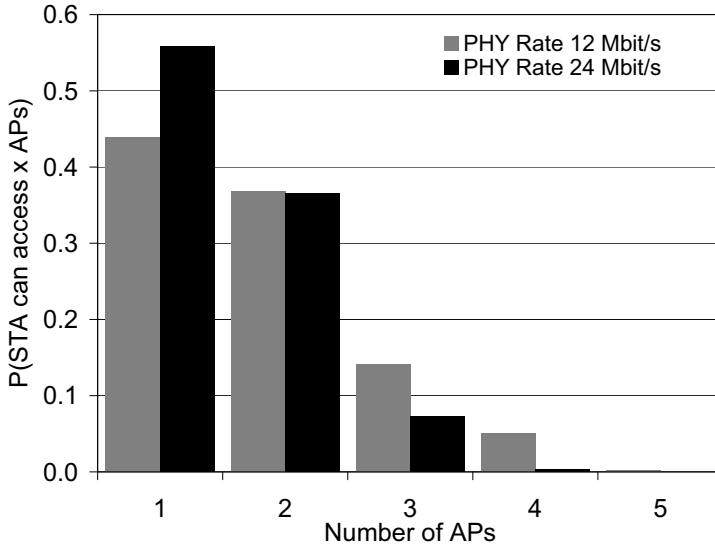


Figure 4.1: Likelihood of being in coverage range from several APs of the Karlstad University Campus WLAN. More than 45% of the stations can choose between two and more APs to connect to at 24 PHY rate.

Finding the best STA/AP selection has been studied extensively [110, 42, 186, 49, 115, 29]. However, those optimization models do not consider the *cost of reconfiguring* the network: If a STA needs to handover from one AP to another, the user might experience a temporary disruption of service during the handover. In addition, signaling messages need to be exchanged and create overhead. In networks with high dynamicity, reoptimizing the network at every change might lead to high costs through network reconfiguration and to low long term user download rates. Recent measurements have shown that in particular public WLAN hotspots exhibit a high dynamicity due to short user inter-arrival times and session durations [80]. User mobility, which is increasing due to the popularity of small WLAN devices such as smart phones and tablets, is another cause of changes in the network.

In Figure 4.2 we illustrate the problem of too frequent reconfiguration with a simple example. A user moves inside an area that is covered by three access points AP1, AP2 and AP3. The user would like to download data from the Internet with the highest possible speed. The signal strength and hence the feasible download rate decreases with the distance from the AP. A common optimization strategy (“Scheme A”) is to handover to a new AP, as soon as the new AP offers a better signal strength than the

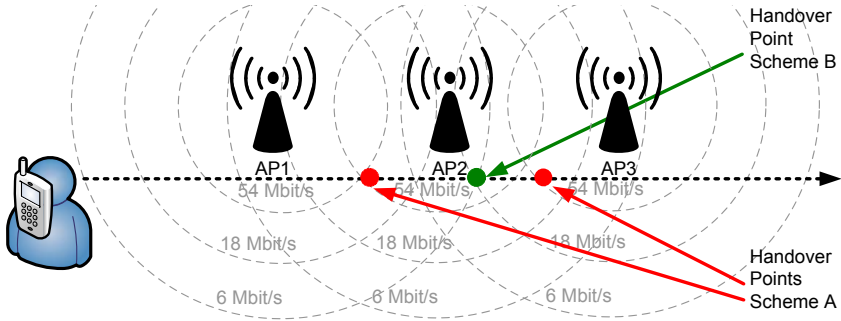


Figure 4.2: Example of a user walking in a hotspot area with coverage from AP1-3. The user can either perform a handover as soon as a better AP is available (“Scheme A”) or after the connection breaks (“Scheme B”).

current AP. Sometimes, a hysteresis is introduced into this strategy by only performing a handover if the signal strength of the new AP is x dB higher.

In our example and using “Scheme A”, AP1 is used until the user reaches the 54 Mbit/s zone of AP2. From then, AP2 is used until it reaches the 54 Mbit/s zone of AP3, where the next handover is executed. However, this strategy can result in overall low performance, if switching from one AP to another AP generates some cost, e.g. due to service interruption, because the WLAN client needs to authenticate itself to the network, the channel needs to be switched or TCP sessions have a timeout and need to start in the slow-start phase again. This can take several seconds in standard WLANs [120] and should be optimized (see Chapter 8). If the user walks fast, the station might be out of the 54 Mbit/s zone of AP2 before the handover is completed and the download can be resumed again.

Under such circumstances it could be better to not use AP2 at all. Instead, a make-after-break strategy (“Scheme B”) could be better. With this strategy, a handover is only initiated if the station moves out of the coverage area of its current AP. Thereby the number of handovers can be reduced and the overall throughput may increase.

This example demonstrates that the optimal handover policy (when to handover to which AP) depends on many factors, such as the service disruption duration, the network topology, the distance and throughput between APs and STAs and the connection opportunities. Clearly, one difficulty of finding the optimal handover policy is that the best decision in the present depends on the unknown future state of the network (e.g. which AP is in reach at what time).

4.1.1 Related Work

Optimizing STA/AP associations has been investigated in a number of works. For example, [49] attempts to characterize the capacity region of multi-channel WLANs under different association policies. The authors conclude that the PHY rate and the load dependent throughput must be considered to achieve high performance. A general framework to study the impact of STA/AP associations while considering the IEEE 802.11e MAC layer and multi-rate networks is presented in [158]. The framework is applied to evaluate a cell-breathing method in WLANs and allows to use different utility functions, for example to maximize fairness or to minimize delay. The framework does not incorporate the dynamics of mobile users and the costs of handovers. [186] presents a user-centric framework to select an AP and its operational channel. STAs exchange information with APs, which then periodically compute the optimal channel and associations. The authors remark that too frequent reoptimization *results in frequent reassociation which influences the user experience due to the hard breakdown in the reassociation process*. However, [186] does not aim to derive how often to reoptimize. In their simulations they reoptimize every 600s, which seems to be very long in dynamic networks. [43] proposes a constant-factor approximation scheme for max-min fair bandwidth allocation in WLANs. For the online optimization of networks with STAs joining and leaving the authors adopt a Hysteresis approach. [42] applies an approach in which a reoptimization is only performed when a time or a load threshold is exceeded. [110] proposes an NP-hard, non-linear optimization problem and a heuristic solution algorithm for computing proportional fair AP association in multi-rate WLANs. [38] presents a MILP formulation of the STA/AP association problem and implements an optimization system adapting cell-breathing concepts known from cellular networks. [144] proposes a multi-objective optimization problem that tries to avoid unnecessary handovers. However, none of the approaches considers the costs of handovers in their optimization models. [172], [173], [31], [103] and [127] propose systems for controlling STA/AP associations using simple heuristics.

Besides deciding when to handover to which AP, optimizing the actual handover procedure has been the focus of several works and technical standards (see also Chapter 8). For example, [122] investigates how to optimize the scanning procedure for new APs. In Chapter 2 we have already surveyed several IEEE standards related to handover management. In particular IEEE 802.11r [21], 802.21[25] and 802.11h [18] are relevant for the practical implementation of STA/AP selection schemes and handovers.

4.1.2 Problem Statement and Contributions

The literature review shows that a key question to solve for the practical application of STA/AP association optimization algorithms is *when* and *how often* to invoke the optimization and then reconfigure the network. This question has not been adequately addressed in previous works such as [49, 186, 110, 43, 38]. In this chapter we therefore aim to answer this question with the following *key contributions*:

- **Formulation of an optimization model to derive the optimal association strategy** for STAs. The system consists of a collection of APs and STAs. The system state is described by service requests, link capacities and link interference conditions. We start by formulating a Mixed Integer Linear model, which allows to maximize the throughput of users for a given network state (later referred to as “Static network model”). Based on this static network model we discuss three simple and commonly found myopic optimization schemes (variants of [103] and [42]). By myopic we mean that the schemes do not consider costs of future handovers and only try to optimize the present network state. The first algorithm reoptimizes the network at every state change. The second scheme additionally allows to restrict the number of handovers at each reoptimization step. The third algorithm implements a classical hysteresis scheme, where a reoptimization is only applied if the throughput is improved by a configurable amount.
- **Development of a dynamic model** that assumes that the future network state is known: By violating the non-anticipativity constraint (i.e. using future state information), too frequent handovers, or handovers to APs that will soon be used by other STAs can be avoided. In a practical setting, it is of course not possible to know the future network state exactly, as the state depends on random user activity. However, in simulations, where the user activity is determined a-priori, the model provides an upper bound on the solution quality of the three simple schemes that do not require exact information about the future. With extensive numerical simulations we show that with respect to the upper bound the simple schemes perform reasonably well if there is little dynamicity in the network. However, if the network state changes often, e.g. due to user mobility, the schemes all exhibit low performance.
- **Use of network state predictions** to improve performance: We show that the use of a simple interpolation from the present network

state already greatly improves the performance compared to the above mentioned schemes when combined with our optimization model. Our optimization model thus provides valuable insights for the design of centralized WLAN management systems. The aim of this chapter is not to show how such estimates of the future can be obtained (for example by using mobility predication), but to show that even if those predictions are inaccurate they can help to improve performance.

The rest of this chapter is organized as follows: In Section 4.2 we model the problem of finding optimal associations and download rates in a static network setting. In Section 4.3 we extend this model to incorporate temporal network state changes, such as re-associations induced by user mobility. In Section 4.4 we discuss in detail the impact of disregarding handover costs in the optimization model. Section 4.5 uses the insights of Section 4.4 to devise a new sliding window based optimization model. Finally, we conclude the chapter with Section 4.6.

4.2 Static Network Model

In this section we develop an optimization model of a wireless access network, which considers the network state at a given point of time, but not the dynamicity of changes. In Section 4.3 we extend this model to a dynamic model to incorporate changes over time.

4.2.1 System Model and Notation

In our model (the notation is summarized in Table 4.1), the network consists of STAs and APs, which are connected to the Internet. We model the network as a directed graph $G(V, E)$. The set of nodes V is the union of the set of user stations U , the set of APs A and a node t , representing the Internet. The set of edges E includes one element for each communication link between nodes in V . Each edge $(i, j) \in E$ is associated with a non-negative value $p_{(i,j)} \geq 0$ representing the bit rate of the link. We assume, that for the wired links between APs and the Internet node t the bit rate is determined by the access technology (e.g. Ethernet).

For the wireless links, a node can choose a PHY-layer bit rate from a set of bit rates. Each bit rate p corresponds to a modulation and coding scheme, for which a minimum *Signal-to-Interference-plus-Noise Ratio* (SINR) of γ_p is required. Given a wireless link (i, j) , a transmitter i then chooses the

Symbol	Description	Type
U	Set of user devices (stations)	Parameter
A	Set of APs	Parameter
I	Set of interfering links	Parameter
T	Set of time slots	Parameter
(i, j)	Link between node i and j	Parameter
t	Fictitious Internet node	Parameter
$p_{(i,j)}$	PHY rate on link (i, j)	Parameter
$q_u(t)$	Usage indicator, 1 if a STA u requests a download in slot t	Parameter
D	Handover cost	Parameter
$f_{(i,j)}^u$	Flow rate on link (i, j) generated by user u	Continuous variable
$x_{(i,j)}^u$	Binary selection variable if link (i, j) is used by user u	Binary variable
Γ	Objective function	

Table 4.1: Important notation used in this chapter

highest bit rate p for which the following inequality holds:

$$\frac{P_i G_{(i,j)}}{N + P_k G_{(k,j)}} \geq \gamma_p \quad \forall ((i, j), (k, l)) \in I, \quad (4.1)$$

where P_i denotes the transmission power of node i , $G_{(i,j)}$ is the channel gain ratio on link (i, j) , N is the thermal noise and I is the set of links that cannot be active at the same time. In other words, I describes the *collision domains* [104, 60] of the network.

In the case of the IEEE 802.11 standard, the collision domain is determined implicitly by carrier sensing. Each node i has a carrier sensing threshold δ_i . If the received signal strength is above the sensing threshold, the node backs off and does not transmit. In order to ensure error free reception on a given link (i, j) , the following inequality must hold:

$$P_i G_{ik} \geq \delta_k \quad \forall ((i, j), (k, l)) \in I. \quad (4.2)$$

If a link (k, l) is in the collision domain of link (i, j) , the carrier sensing threshold of node k must be low enough to detect a transmission on link (i, j) . This ensures that there are no hidden nodes, i.e. nodes that cannot hear transmission which they could interfere with. Clearly, one would like to choose the highest possible bit rate p (to maximize throughput) and

the lowest possible carrier sensing threshold δ (to increase spatial reuse). From a practical point of view, changing δ is not possible on most wireless cards. Hence, we assume that δ is fixed and the bit rate p is set so that Equations 4.1 and 4.2 are full-filled for all links.

We remark that *collision domains* represent a relaxed resource sharing condition, as the collision domain model requires *all* links of a collision domain to be inactive. A more refined modeling of the resource sharing condition can be obtained through the *clique model* [161]. However, this model requires to compute maximal independent sets, which is an NP-hard problem. Moreover, the *collision domain model* has been shown to be reasonably accurate [97] and is thus widely used (e.g. [104], [171], [92], [157]).

4.2.2 Variables

Our model aims to compute 1.) which STA should use which AP and 2.) at what rate a STA can download from the Internet via the chosen AP. Therefore, we introduce a binary variable $x \in \{0, 1\}$ that models the connection between a STA and an AP as follows:

$$x_{(a,u)}^u = \begin{cases} 1 & \text{if STA } u \text{ is connected to AP } a \\ 0 & \text{otherwise.} \end{cases} \quad (4.3)$$

Furthermore, we denote the *download rate* that STA u uses when retrieving data from the Internet via AP a as $f_{(a,u)}^u \in \mathbb{R}^+$. With download rate we refer to the rate that a user can download data with (not considering protocol overheads) and not the PHY rate. In practice, such download rates can be enforced by rate shaping at the APs and routers and/or adapting MAC layer parameters [101].

4.2.3 Model Constraints

The network is described with the following set of integer-linear constraints:

$$\sum_{a \in A} x_{(a,u)}^u \leq 1 \quad \forall u \in U \quad (4.4)$$

$$f_{(a,u)}^u \leq x_{(a,u)}^u M \quad \forall u \in U, \forall a \in A \quad (4.5)$$

$$\sum_{u \in U} f_{(a,u)}^u \leq p_{(t,a)} \quad \forall a \in A \quad (4.6)$$

$$f_{(a,u)}^u = f_{(t,a)}^u \quad \forall u \in U, \forall a \in A \quad (4.7)$$

$$\frac{f_{(a,u)}^u}{p_{(a,u)}} + \sum_{((a,u),(a',u')) \in I} \frac{f_{(a',u')}^u}{p_{(a',u')}} \leq \eta \quad \forall u \in U, \forall a \in A \quad (4.8)$$

$$x_{(a,u)}^u \in \{0, 1\} \quad \forall u \in U, \forall a \in A \quad (4.9)$$

$$f_{(a,u)}^u \geq 0 \quad \forall u \in U, \forall a \in A. \quad (4.10)$$

Equation 4.4 ensures that a STA is connected to at maximum one AP. Equation 4.5 ensures that a station can only download when it is connected. M is a large number (greater than the download rate of any STA). Equation 4.6 makes sure that all STAs connected to an AP cannot download more than the connection of the AP to the Internet allows. With Equation 4.7 we ensure that all flow coming from the Internet node t is forwarded to the user u . Equation 4.8 states that the normalized data rate of a link and the links in its collision domain cannot exceed η and thereby guarantees schedulable rates. η models the efficiency of the MAC layer protocol and typically is smaller than or equal to 1 (we use $\eta = 1$ in the remainder of this chapter). Equations 4.9 and 4.10 specify the domain of the decision variables.

Constraint 4.8 is based on the assumption that contending links can share resources arbitrarily, for example by time-division. It further assumes that the efficiency of the channel access is independent of the traffic in the network. [104] shows how to find a schedule of finite length for arbitrary legal flow rates in TDMA networks under such assumptions. Most WLANs however adopt the IEEE 802.11 MAC layer, which uses random access. Unfortunately, computing η and the achievable throughput for IEEE 802.11 requires lengthy computations, for which no closed-form solution is available [75]. In particular, when the network load is close to the capacity of the network, there are many collisions, which requires a complicated model. However, as we have argued in Chapter 3, for a lower network load a linear approximation of the achievable rates is possible. Hence, constraint 4.8 is also a suitable approximation for IEEE 802.11 networks.

4.2.4 Solving the Model

We aim to maximize the *download rate* of each STA. We hence are confronted with a multi-objective optimization problem, in which the rates of the STAs are the objectives. A standard method for solving such problem is to construct a single *Aggregate Objective Function* (AOF) and maximize this function [119]. The AOF has great impact on fairness and the efficiency of the resource allocation. The often used weighted max-sum AOF might lead to unfair resource allocation and starvation of individual users. In order to enforce fairness, we define the following AOF:

$$\textbf{maximize } \alpha + \kappa \sum_{u \in U} \sum_{a \in A} f_{(a,u)}^u, \quad (4.11)$$

where κ is a fairness parameter and α is a continuous variable described through the following additional constraint:

$$-\sum_{a \in A} f_{(a,u)}^u + \alpha \leq 0 \quad \forall u \in U. \quad (4.12)$$

Equation 4.12 states that each STA must receive at least a rate of α . When κ is set to 0, the minimum download rate is maximized. However, by the definition of Equation 4.11, it might occur that some download rates are not maximized beyond α , even if they could be increased without decreasing α . By increasing κ , more focus is put on overall network performance and less on fairness. Hence, α might be lower then. In the rest of the chapter we set $\kappa = 10^{-8}$ to enforce a high level of fairness and to make sure that download rates are maximized beyond α . A smaller κ could lead to numerical instabilities in the solving process and floating point rounding errors, while a larger one could reduce fairness.

Equations 4.4-4.12 constitute a Mixed Integer Linear Program (MILP) which can be solved with MILP solvers such as CPLEX [6]. We have implemented the model in CPLEX and seen that even for a relatively large network (13 APs and 40 STAs) the problem can be solved within reasonable time on a normal PC (2.26 GHz Intel Core2 Duo, 4 GB RAM). The model results will be analyzed in Chapter 4.4.2, but before we will extend it to a dynamic model in the next section.

4.3 Dynamic Network Model

We proceed by extending the static network model to a dynamic model. The main difference between the static and the dynamic model is that the dynamic model incorporates a temporal view on the network. For example,

the dynamic model considers when a STA joins the network, how the link speed changes over time and when the STA leaves the network again.

4.3.1 Parameters and Variables

We assume that the time of interest is divided into slots of arbitrary, but equal length. Changes in the model parameters and variables only occur at the boundary between two slots. The set of slots is denoted with T . Given a slot $t \in T$, $t + 1$ refers to the slot following t .

Typically, WLAN hotspot users do not want to download data continuously. Users instead download e.g. a website and wait a while before issuing a new request. This user activity is modeled with a parameter u :

$$q_u(t) = \begin{cases} 1 & \text{if STA } s \text{ would like to download in slot } t \\ 0 & \text{otherwise.} \end{cases} \quad (4.13)$$

Furthermore, the parameters p and I are now time dependent. We write $p_{(a,u)}(t)$ to describe the PHY rate on link (a, u) in slot t . Similarly, $I(t)$ now specifies the set of interfering links in slot t .

When a station connects to an AP, it cannot download data immediately. First, control messages for authentication, encryption key negotiation and address assignment need to be exchanged (see Sections 2.1.4 and 4.1). Consequently, we distinguish between the two states “*connecting*” and “*connected*”. The corresponding binary variables \hat{x} and x are hence given as:

$$\hat{x}_{(a,u)}^u(t) = \begin{cases} 1 & \text{if STA } u \text{ is connecting to AP } a \text{ in slot } t \\ 0 & \text{otherwise} \end{cases} \quad (4.14)$$

and

$$x_{(a,u)}^u(t) = \begin{cases} 1 & \text{if STA } u \text{ is connected to AP } a \text{ in slot } t \\ 0 & \text{otherwise.} \end{cases} \quad (4.15)$$

A STA u can only download data when it is in the *connected* state. A STA can only enter the *connected* state after it has been in the *connecting* state for $D_a^{(u)}$ time slots. In other words, $D_a^{(u)}$ models the service interruption duration (in time slots) when a STA u performs a handover to AP a .

4.3.2 Model Constraints

At slot t , a STA u is connected to AP a , if and only if it has been connecting to AP a in $D_a^{(u)}$ slots before t , i.e.

$$x_{(a,u)}^u(t) = 1 \iff \hat{x}_{(a,u)}^u(t') = 1 \quad \forall t' \in \{t - D_a^{(u)} - 1, \dots, t\}, t \in T \setminus \{0, \dots, D_a^{(u)}\}. \quad (4.16)$$

As a STA cannot be in connected state of one AP and connecting state of an other AP simultaneously, we enforce that the connection state also implies the connecting state. We will address this technical limitation of IEEE 802.11 in Chapter 8, but for the rest of the chapter we assume standard IEEE 802.11 operations. Equation 4.16 is not a linear constraint. Therefore, we reformulate Equation 4.16 by replacing the equivalence operator with two logical implications and the set expressions with sums:

$$\begin{aligned} \sum_{d \in \{0 \dots D_a^{(s)}\}} \hat{x}_{(a,u)}^u(t-d) = D_a^{(u)} &\implies x_{(a,u)}^u(t) = 1 \\ &\wedge \\ x_{(a,u)}^u(t) = 1 &\implies \sum_{d \in \{0 \dots D_a^{(u)}\}} \hat{x}_{(a,u)}^u(t-d) = D_a^{(u)}. \end{aligned} \quad (4.17)$$

By using Boolean logic we can reformulate Equation 4.17 to:

$$\begin{aligned} \sum_{d \in \{0 \dots D_a^{(u)}\}} \hat{x}_{(a,u)}^u(t-d) \geq D_a^{(u)} \vee x_{(a,u)}^u(t) < 1 \\ &\wedge \\ \sum_{d \in \{0 \dots D_a^{(u)}\}} \hat{x}_{(a,u)}^u(t-d) < D_a^{(u)} \vee x_{(a,u)}^u(t) \geq 1. \end{aligned} \quad (4.18)$$

We add the two binary variables y and z and rewrite Equation 4.18 as:

$$\sum_{d \in \{0 \dots D_a^{(u)}\}} \hat{x}_{(a,u)}^u(t-d) \geq D_a^{(u)}(1 - y_{(a,u)}(t)) \quad \forall a \in A, u \in U, t \in T \quad (4.19)$$

$$x_{(a,u)}^u(t) \leq 1 - y_{(a,u)}(t) \quad \forall a \in A, u \in U, t \in T \quad (4.20)$$

$$\sum_{d \in \{0 \dots D_a^{(u)}\}} \hat{x}_{(a,u)}^u(t-d) \leq D_a^{(u)}(1 + z_{(a,u)}(t)) + \epsilon \quad \forall a \in A, u \in U, t \in T \quad (4.21)$$

$$x_{(a,s)}^u(t) \geq z_{(a,u)}(t) \quad \forall a \in A, u \in U, t \in T \quad (4.22)$$

$$y_{(a,u)}(t), z_{(a,u)}(t) \in \{0, 1\} \quad \forall a \in A, u \in U, t \in T. \quad (4.23)$$

The variables y and z ensure that at least one of the conditions in each OR statement of Equation 4.18 is fulfilled. ϵ is a small number, $1/M$ in our case. The AND condition is modeled implicitly, as a feasible solution needs to fulfill all constraints. Reformulating Equations 4.4 - 4.10 to take into account changes over time and the state of a STA results in the following set of constraints:

$$\sum_{u \in U} f_{(a,u)}^u(t) \leq p_{(t,a)} \quad \forall a \in A, \forall t \in T \quad (4.24)$$

$$f_{(a,u)}^u(t) \leq x_{(a,u)}^u(t)M \quad \forall a \in A, u \in U, t \in T \quad (4.25)$$

$$\frac{f_{(a,u)}^u(t)}{p_{(a,u)}(t)} + \sum_{((a,u), (a',u')) \in I(t)} \frac{f_{(a',u')}^u(t)}{p_{(a',u')}(t)} \leq \eta \quad \forall a \in A, u \in U, t \in T \quad (4.26)$$

$$\sum_{a \in A} x_{(a,u)}^u(t) \leq 1 \quad \forall u \in U, t \in T \quad (4.27)$$

$$\sum_{a \in A} \hat{x}_{(a,u)}^u(t) \leq 1 \quad \forall u \in U, t \in T \quad (4.28)$$

$$\hat{x}_{(a,u)}^u(t) \leq q_u(t) \quad \forall a \in A, u \in U, t \in T \quad (4.29)$$

$$\hat{x}_{(a,u)}^u(t) \in \{0, 1\}, x_{(a,u)}^u(t) \in \{0, 1\} \quad \forall a \in A, u \in U, t \in T \quad (4.30)$$

$$f_{(a,u)}^u(t) \geq 0 \quad \forall a \in A, u \in U, t \in T. \quad (4.31)$$

Equation 4.24 ensures that the capacity of the Internet link is not exceeded. Equation 4.25 ensures that only connected STAs can download.

Equation 4.26 is the capacity constraint of the wireless channel. Equations 4.27 and 4.28 state that a STA can only be associated and connected to at maximum one AP in each slot. Furthermore, a STA can only attempt to connect to an AP, if the user is requesting a service (Equation 4.29). Finally, Equations 4.30 and 4.31 describe the domain of the decision variables.

4.3.3 Objective Function

As we are interested in data downloads, the instantaneous download rate of a STA is not so important. Instead, the average rate that a STA can achieve during the time it requests the service should be maximized. Therefore, we specify the following objective function $\Gamma(u)$ for each STA u :

$$\Gamma(u) = \frac{\sum_{t \in T} \sum_{a \in A} f_{(a,u)}^u(t)}{\sum_{t \in T} q_u(t)}. \quad (4.32)$$

Since we would like to maximize $\Gamma(u)$ for each $u \in U$, we again face a multi-objective optimization problem, which we solve by maximizing a simple aggregate objective function:

$$\textbf{maximize } \alpha + \kappa \sum_{u \in U} \Gamma(u), \quad (4.33)$$

where κ is a fairness parameter and α is a continuous variable described through the following additional constraint:

$$-\Gamma(u) + \alpha \leq 0 \quad \forall u \in U. \quad (4.34)$$

Equation 4.19-4.31 and 4.34 are now constraints to a standard MILP with Equation 4.33 as objective function. By solving this MILP we can compute the optimal download rates and handover patterns in each time slot, given we know the PHY rates, collision domains and service requests for the whole system run-time.

4.4 Static Optimization

As the optimization models and goals of [49, 186, 110, 43, 38] differ considerable, our goal is here not to compare those approaches directly. We will instead describe three approaches of *when* to invoke the optimization and reconfigure the network. We apply our static model with those approaches and compare the performance to the upper bound provided by the dynamic model (which assumes perfect knowledge of the future).

4.4.1 Reconfiguration Strategies

4.4.1.1 Greedy

The *Greedy* scheme computes the solution to the static model in every time slot. It does not consider the current state of a STA (connected or not). It greedily tries to optimize the network configuration in the present state, not considering any implications on the future performance of the network. If the computed optimal network configuration differs from the current configuration, the required changes to implement the optimal configuration are applied accordingly. This invocation strategy is for example proposed in [103].

In the example network depicted in Figure 4.2, the Greedy scheme produces the same results as “Scheme A”. No interference from other STAs is present and therefore according to the Greedy scheme it is best to download from the AP with the highest PHY rate.

4.4.1.2 k-Handover

The *k-Handover* scheme extends the *Greedy* scheme by adding an additional constraint that specifies that at maximum k handovers can be performed during one slot. As handovers cause service disruption, it might be beneficial to limit the number of handovers.

4.4.1.3 Hysteresis

The *Hysteresis* scheme aims to avoid flapping of configurations and re-configurations that might only yield minor improvements. In this scheme, for every slot the solution quality of the current network configuration $\hat{\alpha}$ (without changing associations) and the optimal solution α^* are computed. The optimal solution is then applied if $\alpha^* > \hat{\alpha}/f$. Typically, f is chosen in the interval $]0...1]$. A value close to 0 requires a large improvement over the current solution in order to be applied. This might lead to a small number of handovers, but might operate the network in a suboptimal configuration. In contrast, a value close to 1 might cause a larger number of handovers. A variant of this invocation strategy is for example used in [42].

In the example network of Figure 4.2, if the user starts walking from the 18 Mbit/s zone of AP1 to the intersection of the 6 Mbit/s zone of AP1 and the 18 Mbit/s zone of AP2, the solutions are $\alpha^* = 18$ and $\hat{\alpha} = 6$. For $f < 1/3$, a handover to AP2 would be triggered.

In most real implementations the hysteresis is computed based on RSSI samples, not the data rate. As in our model the aim is to maximize the throughput and not the RSSI, we apply a hysteresis on the rate.

4.4.2 Evaluation

Next we evaluate the performance of the invocation strategies presented above and the impact of different parameters such as user mobility considering reconfiguration costs. Our key findings are that

- User mobility has a significant impact on the performance of the invocation strategies.
- With low mobility, a Hysteresis based scheme performs well.
- The impact of the handover cost on performance depends on the user mobility.

4.4.2.1 Evaluation Settings

We used CPLEX [6] and a set of custom-made simulation scripts to numerically evaluate the performance of the different schemes. In each time slot, the static optimization problem is solved and the solution is applied according to the invocation strategy under investigation. The size of the dynamic model grows proportionally with the number of time slots. In order to solve the model fast and to be able to run a large set of different scenarios, the *number of time* slots should not be too high. A short *slot length* is a more accurate representation of the reality, in which the network state changes continuously and not only at slot boundaries. However, a short slot length leads to a large number of slots when simulating a long time period. In our simulations, time slots are *1 second* long and the network is simulated for *120 slots*. During this time, each user randomly generates one traffic request within the first 30 seconds and aims to download data from the Internet via an AP for at least 50 seconds. Each simulation was repeated 30 times with different random STA positions and mobility patterns.

In our model, when a STA arrives at the network, it first connects to the AP with the highest signal strength. Only after connecting, it can receive instructions to handover to a new AP. Since we aim to maximize the minimum throughput, the performance comparison metric is the minimum average throughput, i.e. α in Equation 4.34. The STA mobility follows the random way-point model with fixed way points: STAs move along the corridors and when they arrive at a junction, they decide randomly which corridor and direction to follow. With this model, synthetic mobility traces were created by randomly placing STAs on the map (Figure 4.3) of the Computer Science Department of Karlstad University. A total of 13 APs are positioned according to the real deployment and assumed to have

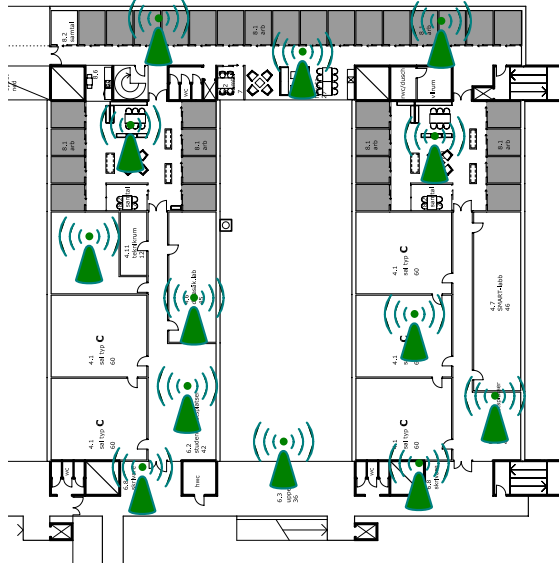


Figure 4.3: Map of the Computer Science Department at Karlstad University. 13 APs provide WLAN coverage in corridors, offices, meeting rooms and labs.

Fast Ethernet connections to the Internet (100 Mbit/s). With the Cisco Prime Network Control System software [2] we determined the achievable PHY rates between STAs and APs at each location of the map. In a real network, STAs and APs should have an autorate mechanism in place, which allows them to adjust the PHY rate according to environmental changes. By adjusting the speed of the mobile STAs and the fraction of mobile STAs, the dynamicity of the network can be varied.

4.4.2.2 Evaluation Metric and Statistical Analysis

Our main interest is to compare the different invocation schemes with respect to the upper bound provided by the model in Section 4.3. Hence, we use the *normalized minimum throughput* $\tilde{\alpha}$ as a performance metric. Formally, $\tilde{\alpha}$ is defined as

$$\tilde{\alpha} = \alpha / \alpha^*. \quad (4.35)$$

Recall that α denotes the minimum throughput of all stations and the optimal value of α computed with the dynamic network model is called α^* . Hence, the normalized performance ranges between 0 and 1, where a value of 1 means that the respective heuristic is as good as the optimum solution.

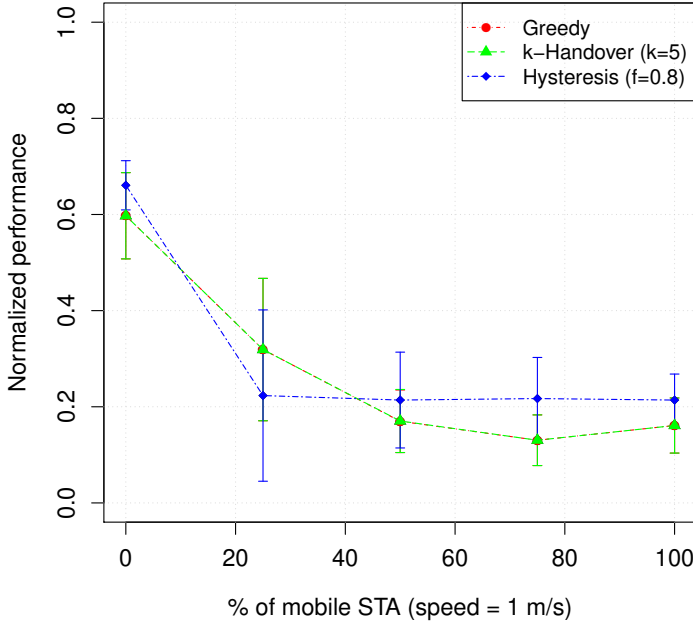


Figure 4.4: Impact of mobility on algorithm performance ($D = 3$, 40 STA). More mobile stations lead to a lower performance since handovers play a greater role.

The plots below show the average (error-bars are standard deviation) of the 30 repetitions.

4.4.2.3 What is the Impact of User Mobility and Network Size?

Handovers of STAs are typically necessary due to user mobility and due to newly arriving STA. To evaluate the impact of both effects, we first simulated a network with 40 STA, of which 0, 10, 20, 30 or 40 STAs are moving at a speed of 1 m/s and the rest are static. The handover cost D is 3 for all handovers, i.e. a handover results in 3 time slots where no data can be downloaded.

Figure 4.4 shows the normalized performance $\tilde{\alpha}$ as a function of the fraction of mobile stations. As the figure reveals, even in absence of user mobility (0% mobile users), the different invocation strategies on average only achieve 60%-65% of the optimal solution. As the fraction of mobile

users increases, the solution quality of the heuristics drops below 23%. The Hysteresis scheme is better than the Greedy and the k-Handover scheme in most cases. However, as the large standard deviations show, the differences are not significant. The Hysteresis scheme is less aggressive when triggering handovers and hence stations are not in the connecting state so often, which is often beneficial for performance.

With no STA mobility, the gap between the heuristics and the optimum is caused by two effects: first, even in absence of mobility, handovers might be required when STAs join or leave the network. The heuristics do not find the best points in time for those handovers. Second, the heuristics maximize the fair throughput in each time slot. In order to maximize the long term average fair throughput, the dynamic model allows temporary unfairness. As mobility increases the timing of handovers gets more important and hence the performance of the heuristics drops.

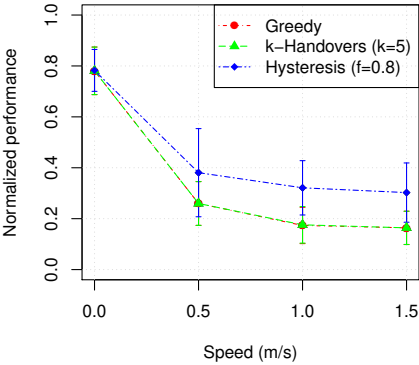
The performance of the Greedy and the k-Handover scheme is identical. We found that the k-Handover scheme does not really avoid handovers, it just delays them to the next time slot (if there are already k handovers in the current time slot).

Next, we investigate the impact of network dynamicity on the performance. The network dynamicity (i.e. the rate of changes in the network) depends on the number of STA arrivals/departures and mobility. We now assume that all STAs move with the same speed. Figure 4.5 shows that an increase in speed results in a decrease of performance. This trend is also due to the higher number of state changes because of faster mobility and the resulting handovers. Generally, if the user mobility gets too high, none of the schemes performs very well. A direct comparison between the schemes shows that the Hysteresis scheme outperforms the k-Handover and the Greedy scheme in 75% of all simulation instances.

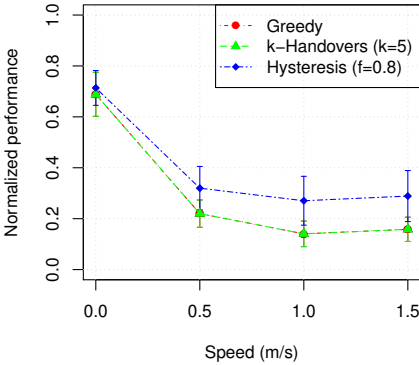
Figure 4.5 furthermore shows that for a fixed speed (e.g. 0 m/s), the performance decreases if the number of STAs in the network is increased. A larger number of STAs causes more dynamicity in the network, e.g. through new STA arrivals. Each time the state of the network changes, the discussed invocation strategies might trigger a handover, even if it is better to remain in the current network configuration for a while and only change the STA/AP associations later.

4.4.2.4 What is the Impact of the Handover Cost?

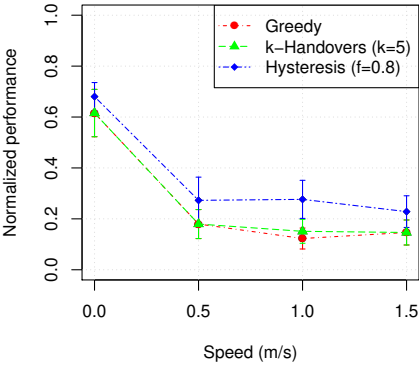
The interruption due to handovers depends on many factors, such as the exchanged protocol messages and the used hardware and encryption scheme. For commercial enterprise WLANs or hotspots the interruption is



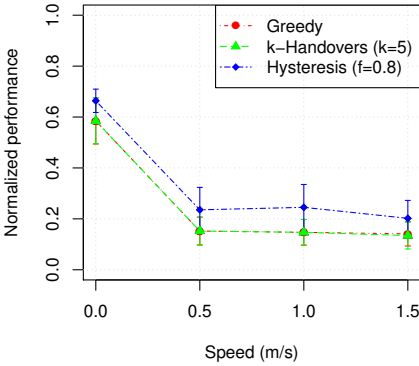
(a) 10 stations



(b) 20 stations



(c) 30 stations



(d) 40 stations

Figure 4.5: Impact of network dynamicity on algorithm performance ($D = 3$). When stations move faster, the achievable performance decreases.

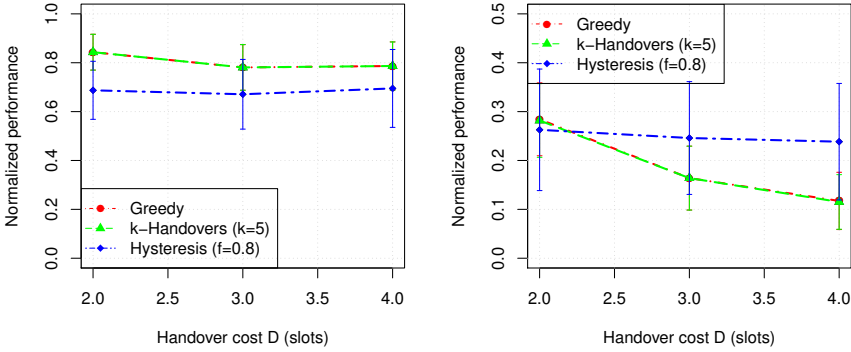


Figure 4.6: Impact of the handover cost on the algorithm performance with 10 STA and 0 m/s (left) and 1.5 m/s (right) station speed. With high speeds and high costs the Greedy scheme performs in particular bad.

in the order of a few hundred milliseconds to a few seconds [123, 120, 100]. When further taking into account the interruption due to TCP timeouts and packet losses, 2-4 seconds are a realistic range [178].

The impact of the handover cost D depends on the dynamicity of the network. As Figure 4.6a shows, for handover costs between 2-4 slots and static stations, there is almost no impact due to higher handover cost. If stations are not moving, the number of handovers is small and hence the cost of handovers plays no role. However, if stations move (Figure 4.6a), the handover cost has a considerable impact on the performance. In particular with the Greedy and the k-Handovers scheme the normalized performance decreases from 28% to 12%. With those schemes, many unnecessary handovers are triggered and stations spend a lot of time performing handovers and connecting, instead of downloading data. The Hysteresis performs better in some cases as it avoids a flapping between access points which provide similar performance.

We would like to note that a constant normalized performance as seen with the Hysteresis scheme does not necessarily reflect a constant absolute throughput. For example, in the case of 1.5 m/s station speed, the normalized throughput remains almost constant regardless of the handover cost. The absolute throughput however drops from 2.2 to 1.5 Mbit/s. This is because with higher handover costs few slots are available for data transfers.

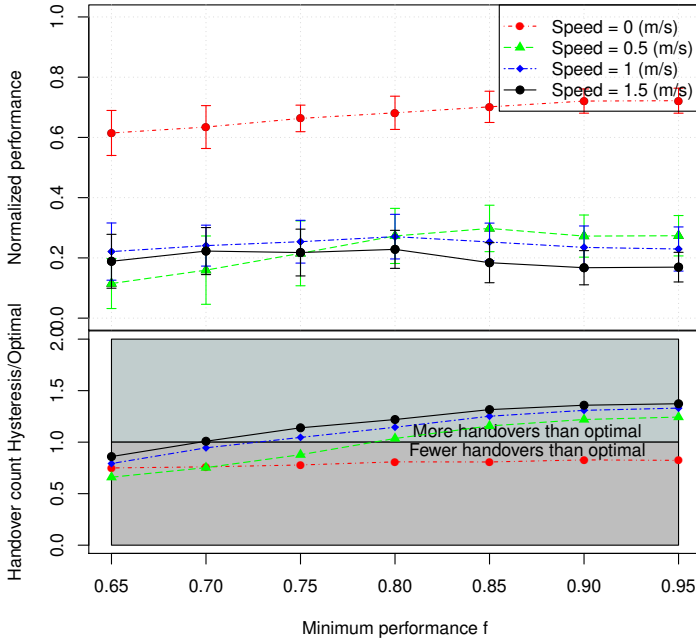


Figure 4.7: Impact of Hysteresis factor f (30 STA, $D = 3$)

4.4.2.5 What is the Impact of Hysteresis Parameter f ?

Recall, that in the Hysteresis scheme we apply the optimal solution, if it is better than the current solution divided by f . Figure 4.7 shows the impact of f on networks with different node mobility. The figure confirms the results provided in [113, 86]: the optimal Hysteresis margin depends on many factors such as network traffic and channel conditions.

If there is no mobility in the network (speed 0 m/s), only a few handovers are required due to station arrivals. Thus, even a small improvement due to a handover (which are infrequent in this setting) should be exploited as the network state is stable for a longer time afterward. Hence a large f is better in such a situation. With higher mobility the opposite is true: smaller values for f are better. For example, when nodes move with 1.5 m/s, $f = 0.7$ gives best performance and chooses the optimal amount of handovers. However, the normalized performance then does not exceed 0.2, showing that not only the number of handovers matters. The rate allocation and the choice of STA/AP associations are more important.

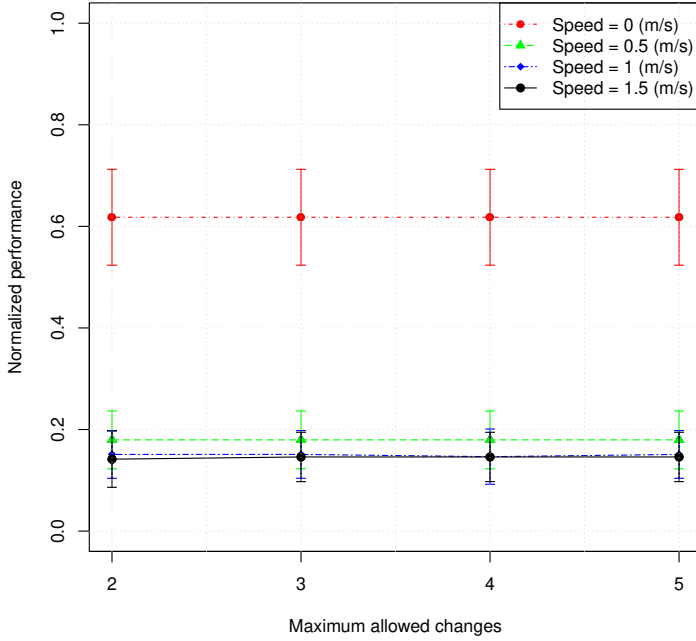


Figure 4.8: Influence of maximum allowed changes on the performance of k -Handover (with 30 STA, $D = 3$)

4.4.2.6 What is Impact of the Handover Limit k ?

The parameter k determines how many handovers can be performed at maximum in each time slot. We evaluate the impact of this parameter for different user mobility patterns. Figure 4.8 shows that there is almost no influence of k on the performance. This is not surprising, as the k -Handover scheme only delays handovers to the next time slot (if there are already k handovers in the current time slot). Hence the result of the different k -s is almost identical.

4.4.3 Discussion

The numerical evaluation has shown that the proposed invocation strategies work well as long as there is no user mobility. In that case, 70-80% of the bound given by the dynamic model are achievable. However, when the users are mobile, the performance quickly drops below 20%. A detailed analysis of the handover patterns has revealed that indeed the reasons

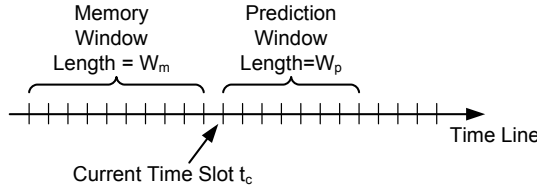


Figure 4.9: Time line of the sliding window algorithm

for this low performance are too frequent handovers (as illustrated in the motivating example of Section 4.1) or handovers to APs that will soon be used by other STA. Sometimes it is better, if a STA does not immediately handover to the AP with the highest signal strength, but remains at the current one (even if the signal strength and the resulting MCS are lower). We apply this insight in the next section, where we develop a sliding window scheme that estimates the immediate future networks states and incorporates this in the handover decisions.

4.5 Sliding Window-Based Optimization

In this section we develop and evaluate a sliding window-based scheme. The key idea of this scheme is to use predictions of the immediate future and to consider amount of data a STA has already downloaded in the past. This allows to avoid too frequent handovers and to compute a better rate allocation. The scheme does not include or depend on any specific method to predict the future network state. We show that already a simple prediction method of the future network state is useful, even if the predictions are erroneous. Developing more sophisticated estimation methods is out of the scope of this thesis.

4.5.1 Sliding Window Method

We denote the current time slot as t_c . As shown in Figure 4.9, we define two windows. The memory window includes W_m time slots in the past, the prediction window includes W_p time slots in the future. Furthermore, we define the set of slots $T = \{t_c, \dots, t_c + W_p + 1\}$ and $T_m = \{t_c - W_m - 1, \dots, t_c - 1\}$. $T \setminus t_c$ denotes the time slots in the prediction window, T_m the slots in the memory window. We replace Equation 4.32 with Equation 4.36 to maximize the utility during T , taking into account the already

downloaded data during T_m .

$$\Gamma(u) = \frac{\sum_{t \in T_m} \sum_{a \in A} f_{(a,u)}^u(t) + \sum_{t \in T} \sum_{a \in A} f_{(a,u)}^u(t)}{\sum_{t \in T_m} q_u(t) + \sum_{t \in T} q_u(t)}. \quad (4.36)$$

With $f_{(a,u)}^u(t)$ (where $t \in T_m$) we denote the download rate of STA u from AP a during time slots prior to the current time slot. Hence, it is not a variable (since we cannot change the past), but a parameter. The parameters $p_{(a,u)}(t)$ and $q_u(t)$ for times $t_c + W_p + 1 > t > t_c$ are not known and need to be estimated. Different techniques are available to estimate those parameters. Each one comes at different cost and achieves different accuracy. For example, one could utilize mobility prediction techniques [159] or machine learning techniques such as Support Vector Machines [77] for the parameter estimation. We would like to point out that our approach is independent of the prediction technique used.

4.5.2 Evaluation

We have evaluated the sliding window method using the setup of Section 8.6 in a network with 10 STA moving at 1.5 m/s. Under those settings the invocation strategies of Section 8.6 reach at maximum 35% of the upper bound. We compare the performance of the sliding window method with different prediction window sizes W_p (W_m is set to 120 for all simulations) and prediction errors. We assume that parameters can be estimated with higher accuracy in the immediate future than in the distant future. Hence, the probability that a predicted parameter at slot $t \geq t_c$ is not equal to the actual parameter can be described as $1 - (1 - e)^{t - t_c}$, where $e \in [0...1]$ models the accuracy of the prediction. Furthermore, we implement a simple, but potentially inaccurate prediction method, where we set $p_{(a,u)}(t) = p_{(a,u)}(t_c)$ and $q_u(t) = q_u(t_c)$, i.e. we assume that the present network state will remain unchanged during the whole prediction window. We call this estimation “*Simple Estimation*”.

Figure 4.10 depicts the performance for different window sizes and error rates. Note that $W_p = 120$ and $e = 0$ is equal to the dynamic model, as a window of 120 covers the whole simulation duration. For $W_p = 0$, all approaches perform equally good, as no prediction is done. As shown in the previous section, the simple Hysteresis achieves 0.35 under the same conditions. The increase from 0.35 to 0.5 is due to the memory window, which takes into account how much data has already been downloaded and hence the available resources are distributed among STAs better. For example, if a STA has already downloaded at a high average rate in the past, its download rate can be decreased in the present and thereby

allowing other STAs to download faster. When increasing the prediction window, the performance increases most of the time. Consider the case of $e = 0$. There is a significant increase between $W_p = 0$ and $W_p = 2, 3$ or 4 . A larger prediction window allows the station to remain connected to an AP with weak signal strength and not handover immediately. Furthermore, it avoids to handover to an AP which is used by another STA in the next slots.

Figure 4.11 shows those effects with the example of one STA and prediction window sizes of 0, 10 and 20. The figure shows to which AP the STA is connected in which time slot and how many STAs in total are connected to the same AP. With a prediction window of 0, the STA tries to connect to an AP several times, but needs to change again before it can download (because another AP is better meanwhile). For example, in slot 8 and 9 the STA downloads from AP 9. Then, after 9 slots, the STA tries to connect to other APs, and only at slot 16 it is connected to AP 7. With larger prediction windows, e.g. $W_p = 10$, the STA does not attempt other connections and hence is already connected to AP 7 in slot 13. The example also shows that the load is balanced better with larger prediction windows. In this example, with prediction window sizes 10 and 20, the STA never shares an AP with other STAs. In contrast, with a prediction window size of 0, the STA needs to share an AP during 4 time slots. The example furthermore shows that with $W_p = 20$, fewer handovers are required than with $W_p = 0$ (no prediction) or $W_p = 10$ and that the STA is in the connected state longer.

Surprisingly, a larger window is not always better. For example, on average $W_p = 10$ is better than $W_p = 20$. A larger window sometimes results in handovers to accommodate for a change in the network state in future (e.g. after 17 slots), which is not optimal compared to a very large window. However, the smaller window cannot “see” that network state change, as it is outside the prediction window.

Introducing errors to the prediction makes the performance worse. However, even with a large error probability in the prediction ($e = 0.20$ means that at 5 slots in the future the state is wrongly estimated with 67% likelihood) good performance gains can be achieved. Nevertheless, it seems not to be beneficial to extend the prediction window to more than 5 slots. For larger window sizes the error probability gets too high.

The simple estimation method works relatively well for small prediction windows. With $W_p = 5$ approximately 65% of the normalized performance can be achieved. This is an increase of 15% compared to $W_p = 0$ and 30% compared to the Hysteresis scheme.

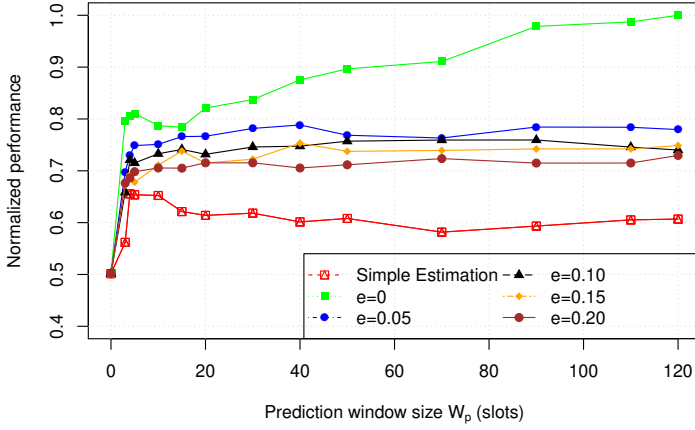


Figure 4.10: Performance of sliding window-based optimization. As the prediction window size increases, the performance also increases. Prediction errors decrease the performance.

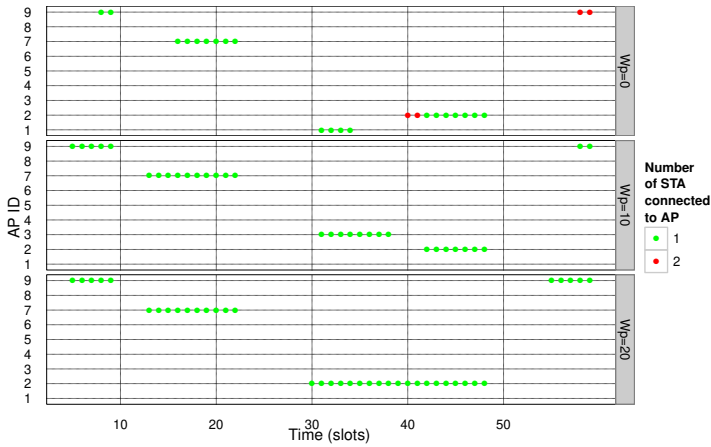


Figure 4.11: Connection pattern of one STA with $W_p = 0, 10$ and 20 . With a larger prediction window fewer handovers are necessary and higher loaded APs are avoided.

4.6 Conclusions

In this chapter we have investigated the STA/AP selection and rate control problem under dynamic network conditions. As the main result of this chapter, we have shown that optimizing associations of mobile stations while disregarding the costs of handovers and network reconfiguration results in performance degradations of up to 80% compared to an optimal scheme. If there is little dynamicity in the network, invoking the re-optimization at each change in the network configuration is possible without major performance degradations. However, in situations with high dynamicity the costs of handovers and the question when to re-optimize the network are important performance factors.

Potential solutions to performance problems under such circumstances are 1.) to consider the handover costs when doing the re-optimization and 2.) to reduce the cost of handovers. In this chapter we have proposed a scheme that, by estimating future network states and by considering the costs of handovers, shows better performance than an always-optimize scheme. One of the main limitations of the model is that it does not consider the possibility of failed handovers. Handover failures are common in real networks (see also Chapter 9) and can lead to severe performance problems. A future extension of the model therefore should take into account the handover failures, for example by modeling the handover cost D as a random variable.

In Chapters 7 and 8 we develop systems that reduce handover costs. In addition, in Chapter 8 we address the question of how to obtain information about the network state in a timely manner. However, before turning to those more practical questions, in the next chapter we discuss how to optimize STA/AP associations in wireless mesh networks. This problem differs from the one discussed in this chapter in several aspects. While in enterprise WLANs the performance bottleneck is usually in the wireless link and not the wired distribution system, in mesh networks the wireless distribution system or the mesh gateway is often the performance bottleneck. Hence, for good performance it is not sufficient to just consider the STA/AP associations, but also other aspects such as routing within the mesh network need to be considered.

Fair Optimization of Associations in Mesh-Connected WLANs

5.1 Introduction

In the previous chapter we have attempted to answer the question when to connect to which AP in a normal enterprise WLAN. In the past years, Wireless Mesh Networks (WMNs) have emerged as a suitable technology to extend the coverage area of WLANs in a cost efficient way. By deploying Mesh Nodes (MNs) to wirelessly forward traffic on behalf of other MNs or Mesh Access Points (MAPs) from or towards Mesh Gateway nodes (MGWs), the cabling and installation costs can be reduced considerably.

Compared to normal enterprise WLANs, in WMNs it is more difficult for a STA to select the best AP. While in an enterprise WLAN the available bandwidth of the wireless link between the STA and the AP is typically the performance bottleneck, in a WMN also the wireless backhaul and the routing needs to be considered for good performance and the fair distribution of resources.

Figure 5.1 illustrates this problem for a small network. The entire traffic of one STA needs to be routed via one AP and the mesh to a *Network Operations Center* (NOC), where access control and accounting functions are performed (see also the management architecture in Chapter 6). In the given setup, the classical RSSI-based STA/MAP association strategy and minimum-hop count routing results in three stations using MAP1 as AP and MGW1 as gateway, while MAP2 and MGW2 are only used by one station. This leads to unfairness and sub-optimal performance for STA1, STA2 and STA3. As the three stations need to share resources

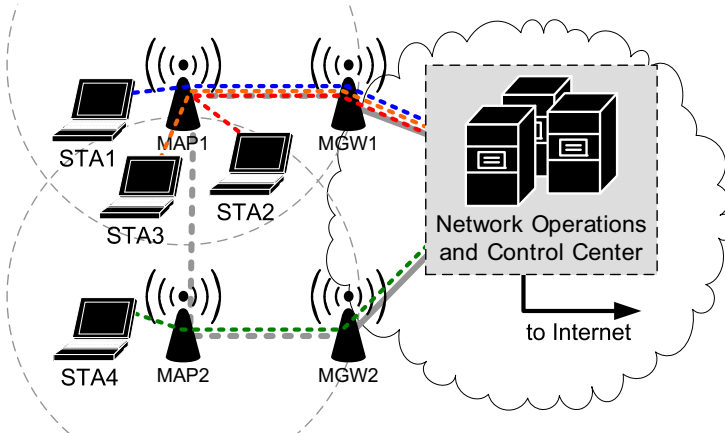


Figure 5.1: Example of RSSI association/minimum hop-count routing

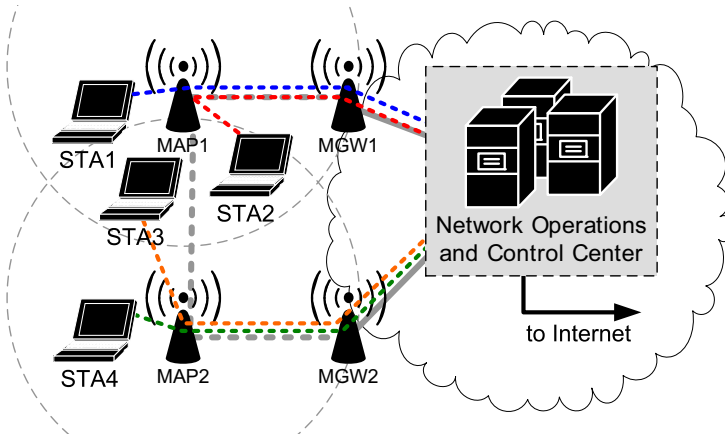


Figure 5.2: Example of optimized association/routing

of MAP1 and MGW1, they might receive a much lower throughput than STA4, which exclusively uses MAP2 and MGW2.

By intelligently controlling the STA/MAP associations and the routing in the wireless backhaul one can improve fairness and performance. One alternative network configuration, which provides better fairness, is depicted in Figure 5.2. STA1-3 now receive higher flow rates, as resources are only shared among two instead of three users.

5.1.1 Related Work

The related work surveyed in the previous chapter considers the optimization of STA/AP associations in WLANs without mesh backhaul. The

results of those works can only partially be translated to scenarios with mesh connected WLANs. As mentioned above, in mesh scenarios the backhaul should be jointly optimized with the STA/AP associations. Such joint optimization is for example proposed in [99], [116] and [37]. In [37], stations incorporate information about the capacity of the wireless backhaul, which is broadcasted by the MAPs, when selecting a MAP to associate to. [99] and [116] formulate the STA/MAP association problem as a non-linear optimization problem. While [99] states that there is a trade-off between fairness and throughput, [116] claims *to improve throughput while enhancing fairness*. [117] proposes a heuristic algorithm to solve the association and routing problem. The three latter works use simple fairness concepts. In [99] and [117] the bit rate of any two STAs cannot differ by more than a fairness ratio. However, such ratio may limit the rate of one STA, even if it does not compete with a second STA that receives a lower rate. While [116] claims to achieve max-min fairness, it does not compute the lexicographically maximum rate vector, as required by the classical textbook definition [45].

[71], [164], [108], [142] and [136] compute max-min fair rate allocations according to textbook definition. For example, [71] jointly optimizes routing and rate allocations, but relies on *clients* to forward traffic on behalf of other clients. This is typically not possible in WMNs. [164] studies max-min fair rate allocation and routing in WMNs. However, [164] might require multi-path routing, which often leads to practical problems such as packet re-ordering and higher jitter. [108] proposes a convex optimization model for max-min fairness in WMNs based on the IEEE 802.11e standard. [142] proposes a congestion control algorithm to achieve max-min fairness in WMNs. Finally, [136] presents a mixed-integer linear programming model for max-min fair rate allocation in WMNs. The latter papers optimize the flow rate allocation, but use fixed, pre-computed routes.

5.1.2 Problem Statement and Contributions

To the best of our knowledge, no other optimization model and a corresponding fast heuristic for WMNs has been proposed previously, that jointly considers single-path routing, AP selection and rate allocation under max-min fairness. Under practical considerations such a model would be desirable, as single-path routing avoids packet re-ordering and max-min fairness does not require a-priori knowledge of traffic demands.

As discussed, some previous works only take into account a subset of those aspects and do not consider an integrated optimization problem.

While other works jointly optimize all those aspects, they rely on simple heuristics and do not compare them against the solutions obtained by exact methods. It is thus not possible to evaluate the quality of the produced solutions.

The main objective of this chapter is thus to close the gaps in literature that we highlighted and to propose a fast heuristic for the integrated planning of all the considered aspects of the mesh optimization problem. Additionally, we assess the quality of the obtained solutions with respect to those obtained by an exact approach.

The *key contributions* of this chapter are:

- **A compact mathematical formulation** (i.e., the number of constraints is polynomial in the size of the input) for the problem: specifically, the set of solutions to the problem is defined through a Mixed Integer Linear Programming (MILP) formulation. The resulting optimization problem can in principle be solved by an effective commercial solver, though in practice the run-time may be unacceptably long.
- **A fast heuristic algorithm:** due to the long time required to find an optimal solution, the exact approach is not suitable for online optimization. We hence develop a novel heuristic solution algorithm, which iteratively solves the STA/MAP association and the routing and rate allocation problem. For medium to large size networks (up to 300 users) the run-time does not exceed 10 seconds, which makes it suitable and competitive for online network optimization. For a number of random networks we show that the minimum per STA rate computed by our heuristic roughly matches 98% of the rate computed by the exact solution algorithm.
- **Extensive evaluation with network simulations:** we extensively evaluate the solution algorithms and compare them to two simple heuristics, which use the RSS and the distance of a MAP to the next MGW (similarly to what is done in [115]). We show the superiority of our heuristic over the other heuristics. In particular for TCP connections fairness improves greatly.

The rest of the chapter is organized as follows: Sections 5.2 and 5.3 describe in detail the optimization problem and solution algorithms. Section 5.4 evaluates the proposed solution algorithms. In Section 5.5 we compare the performance of the solution algorithms with two simple distributed heuristics using ns-2 simulations. Finally, in Section 5.6 we derive some conclusions.

5.2 System Model

5.2.1 Basic Notation

Throughout this chapter, we reuse the notation that was introduced in Chapter 4 and add new definitions as required by the problem under consideration. The mesh network is modeled as a directed graph $G(V, E)$. The set of nodes V is the union of three sets: 1.) a set U , that includes one element for each STA 2.) a set M , that includes one element for each mesh node 3.) a set W , that includes one element for each mesh gateway. An additional node t is also introduced to represent a wired connection to the NOC and the Internet. Thus, $V = U \cup M \cup W \cup \{t\}$. The set of edges E includes one element for each potential communication link between nodes in V and each edge $(u, v) \in E$ is associated with a non-negative value $b_{(u,v)} \geq 0$ representing the maximum network layer bit rate (i.e. the rate usable by IP connections, taking into account MAC and PHY layer overheads) that link (u, v) can support. The achievable network layer bit rate depends on PHY rate and the interference model and is determined as follows: First, we compute the PHY rate p_j as shown in the previous chapter (Equations 4.1 and 4.2). We then calculate the network layer bit rate required including all overheads for link (u, v) with PHY rate p_j as

$$b_{(u,v)} = \sum_{i=1}^{\text{MTUSIZE}} \pi_i (i / (OH_{fixed} + \frac{OH_{var} + i}{p_j})), \quad (5.1)$$

where i is the payload length (in bits), MTUSIZE is the maximum transferable payload length (in bits), π_i is the probability of sending a packet of length i , OH_{fixed} are the fixed duration components of the overhead that are not dependent on the PHY rate (e.g. channel access or PHY preamble) and OH_{var} accounts for PHY rate dependent overhead (e.g. MAC header). Considering the network layer capacity instead of the link layer capacity, as done in Chapter 4, slightly increases the practical applicability of the model. The vector \mathbf{b} contains the bit rates of all links.

As before, we use the collision domain model and the set of interfering links I to represent the resource sharing relations. A summary of the notation used throughout the rest this chapter is given in Table 5.1.

5.2.2 Feasible Solution Set

Each STA $u \in U$ downloads data from node $v_u \in V$ (typically the Internet t) via a MAP that is connected to the mesh network. The data transmission request generated by each STA can thus be identified by a tuple (u, v_u) ,

Symbol	Description	Type
U	Set of STA	Parameter
M	Set of non-gateway mesh nodes	Parameter
W	Set of mesh gateways	Parameter
t	Virtual node representing the Internet	Parameter
V	Set of nodes $V = U \cup M \cup W \cup t$	Parameter
E	Set of links	Parameter
G	Connectivity graph $G(V, E)$	Parameter
I	Set of interfering links	Parameter
v_u	Source of traffic demand for STA u	Parameter
$b_{(i,j)}$	Network layer capacity of link (i, j)	Parameter
$f_{(i,j)}^u$	Flow of STA u on link (i, j)	Continuous variable
$x_{(i,j)}^u$	Indicator variable if STA u uses link (i, j)	Binary variable
$F^u(j)$	Total flow of node j by STA u	Function of variables
$X^u(j)$	Total number of links at node j by STA u	Function of variables
R	Set of feasible rate allocation vectors	Set

Table 5.1: Summary of notation used in this chapter

that indicates the destination and source nodes of the transmission in the graph G . We assume a single-path routing protocol, which imposes that the entire traffic flow of each request must be routed on a single path.

The problem that we study can be described as follows: given the mesh network graph $G(V, E)$ including a set U of STA and a set M of mesh nodes, the set of interfering links I and the vector of edge bit rate capacity $\mathbf{b} \in \mathbb{Z}_+^{|E|}$, the *Mesh Max-Routing Problem (MESHMAX)* is the one of establishing a single routing path for each STA so as to maximize the flow rates sent on the network, while respecting the capacity of each link and the resource sharing relations imposed by the set of interfering links. Note that by computing a path for a STA, we implicitly compute the STA/MAP association, as the STA and the MAP are the first two nodes among the ones constituting the path.

The MESHMAX problem can be naturally modeled as a variant of the *Unsplittable Multicommodity Flow Problem* [32], where the flow models the

transfer rate associated to each STA. In particular, we refer to a formulation based on edge flows, where we introduce a non-negative continuous variable $f_{(i,j)}^u$ to represent the bit rate of STA $u \in U$ on link $(i, j) \in E$ and a binary variable $x_{(i,j)}^u$ that is equal to 1 if the entire traffic of u is routed on edge (i, j) and 0 otherwise. The latter variable is needed to model the unsplittable flow requirement. The unsplittable multicommodity flow problem has no constraints on fairness and hence can lead to the starvation of users. Therefore, later when solving the MESHMAX problem, we impose additional fairness constraints.

$$\sum_{(i,j) \in E} f_{(i,j)}^u - \sum_{(j,i) \in E} f_{(j,i)}^u = F^u(j)$$

$$\text{with } F^u(j) = \begin{cases} \sum_{(v_u,i) \in E} f_{(v_u,i)}^u & j = u \\ 0 & \text{otherwise} \\ -\sum_{(u,j) \in E} f_{(u,j)}^u & j = v_u \end{cases} \quad \forall u \in U, j \in V \quad (5.2)$$

$$\sum_{(i,j) \in E} x_{(i,j)}^u - \sum_{(j,i) \in E} x_{(j,i)}^u = X^u(j)$$

$$\text{with } X^u(j) = \begin{cases} 1 & j = u \\ 0 & \text{otherwise} \\ -1 & j = v_u \end{cases} \quad \forall u \in U, j \in V \quad (5.3)$$

$$\sum_{u \in U} \frac{f_{(i,j)}^u}{b_{(i,j)}} + \sum_{((i,j),(k,l)) \in I} \sum_{u \in U} \frac{f_{(k,l)}^u}{b_{(k,l)}} \leq \eta \quad \forall (i, j) \in E \quad (5.4)$$

$$f_{(i,j)}^u \leq x_{(i,j)}^u M \quad \forall u \in U, (i, j) \in E \quad (5.5)$$

$$f_{(i,j)}^u \geq 0 \quad \forall u \in U, (i, j) \in E \quad (5.6)$$

$$x_{(i,j)}^u \in \{0, 1\} \quad \forall u \in U, (i, j) \in E \quad (5.7)$$

In Equations 5.2-5.7 we define the set of feasible solutions for the MESHMAX problem. Equation 5.2 models the flow conservation constraint. For each node j and each STA u , the amount of flow $F^u(j)$ associated with STA u and entering node j must be equal to the amount exiting the node, except in the origin node v_u and destination node u . Equation 5.3 ensures that each STA routes its entire traffic on a single path between u and v_u . The capacity constraint 5.4 ensures that the sum of flows sent on a link does not exceed the capacity of the collision domain of the corresponding

link. Specifically, the overall flow that is considered in the left-hand-side of each constraint 5.4 includes: 1.) the sum of the flows of all STAs sent on edge $(i, j) \in E$, plus 2.) the sum of flows of all STAs sent on edges $(l, k) \in E$ that share the bandwidth of (i, j) according to the set of colliding links I . The efficiency of the channel access is given by the constant η . We normalize the flow rates by $b_{(i,j)}$, i.e. flows sent on faster links occupy the wireless channel less. Constraint 5.5 ensures that, if edge (i, j) is not used to route the traffic of u , then no flow of u can be sent over it (M is again a big number). Finally, Equations 5.6 and 5.7 define the decision variables of the problem.

5.2.3 Objective Function and Fairness Considerations

It is reasonable to assume that the user utility increases as the flow rate increases. Hence we aim to maximize the flow rate of each STA. For notational convenience, we define a rate allocation vector $\mathbf{r} = (r_1, \dots, r_{|U|})$, which denotes the flow rate of each STA, i.e.

$$r_u = \sum_{(v,u) \in E} f_{(v,u)}^u. \quad (5.8)$$

The set of feasible rate allocation vectors \mathbf{R} is described by Equations 5.2-5.7. We would like to maximize \mathbf{r} over all $\mathbf{r} \in \mathbf{R}$. Hence, the following optimization problem needs to be solved:

$$\mathbf{maximize} \{ \mathbf{r} : \mathbf{r} \in \mathbf{R} \}. \quad (5.9)$$

As in Chapter 4, we are confronted with a multi-objective optimization problem. A standard approach to solve such an optimization problem is to define an aggregate objective function $f(\mathbf{r}) : \mathbb{R}^{|U|} \rightarrow \mathbb{R}$ and then to maximize $f(\mathbf{r})$. One of the simplest aggregation functions is the so-called max-sum:

$$\mathbf{maximize} \{ \sum_{u \in U} r_u : \mathbf{r} \in \mathbf{R} \}. \quad (5.10)$$

The drawback of max-sum optimization is the complete lack of fairness. Some STAs may receive high rates, while other STAs are completely deprived of bandwidth. Consider for example a simple network consisting of two mesh nodes v_1 and v_2 and two STAs u_1 and u_2 , such that u_1 can only be associated to v_1 and u_2 to v_2 . We additionally assume that all links are in one collision domain and that the network layer bit rate of the links is 10 Mbit/s. In this case, the max-sum optimization could allocate 10 Mbit/s to u_1 and 0 Mbit/s to u_2 - a very unfair assignment.

To avoid such deprivation, fairness needs to be introduced into the optimization model. Instead of simply re-writing Equation 5.10 as done in Chapter 4, we apply a more formal fairness concept in this chapter. Specifically, we want to compute a *max-min fair* rate allocation. Such rate allocation is in particular useful in scenarios, in which traffic demands of users are not known *a priori*.

Definition 1: By \vec{r} we denote the bit rate allocation vector \mathbf{r} with its entries sorted in non-decreasing order. Following the classical definition of [45], we call a rate allocation \vec{r} *max-min fair*, if $\mathbf{r} \in \mathbf{R}$, i.e. it is feasible and if for any other feasible allocation \vec{s} , if $\vec{s}_i > \vec{r}_i$, there exists some j such that $\vec{r}_j < \vec{s}_j$ and $\vec{s}_j < \vec{r}_j$. In other words, if \mathbf{r} is max-min fair, it is not possible to increase one entry of \mathbf{r} without decreasing a smaller entry.

Finding a max-min fair allocation is closely related to the *lexicographic ordering* of rate vectors. [128] defines the lexicographic order of two vectors as follows:

Definition 2: We call a vector $\mathbf{r} \in \mathbb{R}^m$ lexicographically greater than vector $\mathbf{s} \in \mathbb{R}^m$, $\mathbf{r} \succ \mathbf{s}$, if there exists an index $i \in \{1, \dots, m-1\}$, such that $r_j = s_j$ for all $j = \{1, \dots, i\}$ and $r_{i+1} > s_{i+1}$. If $\mathbf{r} \succ \mathbf{s}$ or $\mathbf{r} = \mathbf{s}$, we write $\mathbf{r} \succeq \mathbf{s}$.

Definition 3: A lexicographic maximization $\text{lex max } \{\mathbf{r} : \mathbf{r} \in \mathbf{R}\}$ finds the lexicographically greatest vector \mathbf{r} in the feasible set \mathbf{R} .

Intuitively, a vector $\vec{r} \in \mathbf{R}$ is max-min fair, if there exists no other vector $\mathbf{s} \succeq \mathbf{r}$. We will use this relationship in the following algorithms.

5.3 Solution Algorithms for MESHMAX

Our aim is to solve $\text{lex max } \{\mathbf{r} : \mathbf{r} \in \mathbf{R}\}$. As such a problem cannot be solved directly with standard MILP solution algorithms, we proceed with presenting three algorithms for solving the MESHMAX problem (i.e., the MILP defined by Equations 5.2- 5.7) with the max-min fairness objective. We describe one exact solution algorithm and two faster heuristics.

The MESHMAX problem needs to be solved each time a user enters or leaves the network or the network topology changes. The first algorithm (MESHMAX-OPT) that we present computes the optimal max-min fair allocation in single path networks. This is computationally very intensive and only feasible for small networks. The second algorithm solves a relaxed version of the original MESHMAX problem, by allowing to split a flow as it traverses the network (MESHMAX-LP). The outcome of this algorithm is then used to derive a rounded single-path version.

The third algorithm (MESHMAX-FAST) takes into account the two-

tier nature of WMNs, with infrastructure nodes and STAs that generate service requests. Instead of solving the MESHMAX problem for all STAs, it solves it for MAPs and then uses a matching algorithm to associate STAs to MAPs.

5.3.1 Optimal Max-Min Fair Rate Allocation (MESHMAX-OPT)

We use the idea of *conditional means*, which was introduced in [131], to compute a max-min fair resource allocation over non-convex sets. The basic idea of this approach is to define $\tau_k = \sum_{i=1}^k \vec{r}_i$, which represents the k -th cumulative ordered value and can be computed by solving the following optimization problem:

$$\tau_k = \mathbf{minimize} \sum_{u \in U} r_u w_{uk} \quad (5.11)$$

$$\sum_{u \in U} w_{uk} = k \quad (5.12)$$

$$0 \leq w_{uk} \leq 1 \quad \forall u \in U. \quad (5.13)$$

With w we denote a continuous variable to weigh the importance of each individual r . Note that this problem is non-linear, as both w and r are variables. Taking the dual of the program leads to the following linear formulation, in which τ can be computed:

$$\tau_k = \mathbf{maximize} k\beta_k - \sum_{u \in U} \lambda_{ku} \quad (5.14)$$

$$\beta_k - r_u \leq \lambda_{ku} \quad \forall u \in U \quad (5.15)$$

$$\lambda_{ku} \geq 0 \quad \forall u \in U. \quad (5.16)$$

According to [131], solving $\text{lex max } \mathbf{r}$ is equivalent to solving a lexicographic maximization of the conditional means, i.e. $\text{lex max } \{\tau_1, \dots, \tau_{|U|}\}$. The latter lexicographic maximization problem is then solved with opti-

mization problem (OP1):

$$\text{maximize } \tau_k \quad (5.17)$$

$$\tau_k \leq k\beta_k - \sum_{u \in U} \lambda_{ku} \quad (5.18)$$

$$\tau_l^* \leq l\beta_l - \sum_{u \in U} \lambda_{lu} \quad \forall l \in \{1, \dots, k-1\} \quad (5.19)$$

$$\beta_l - r_u \leq \lambda_{lu} \quad \forall l \in \{1, \dots, k-1\}, u \in U \quad (5.20)$$

$$\lambda_{iu} \geq 0 \quad (5.21)$$

$$\{r_1, \dots, r_{|U|}\} \in \mathbf{R}, \quad (5.22)$$

where β is an unbounded continuous variable and τ_l^* is the optimal solution of iteration l . Constraints 5.18-5.21 and variables β and λ are introduced for calculating the conditional mean. For details on how transforming a non-convex lexicographic optimization problem into the form above, we refer the reader to [131] or [128]. OP1 cannot be used to compute τ_1^* (it is not defined for $k = 1$). Instead, we compute τ_1^* by solving the following problem (OP2):

$$\text{maximize } \tau_1 \quad (5.23)$$

$$-r_u + \tau_1 \leq 0 \quad \forall u \in U \quad (5.24)$$

$$\{r_1, \dots, r_{|U|}\} \in \mathbf{R} \quad (5.25)$$

The max-min fair rate allocation vector \mathbf{r} is then computed with Algorithm 5.1.

Algorithm 5.1 Max-min Fair bandwidth allocation for multi-commodity single-path networks

Input : Set of feasible rate allocations \mathbf{R}

Output : Max-min fair rate allocation vector \mathbf{r}^*

- (1) $r_1^* := \tau_1^* := \text{Solve OP2}$
 - (2) **foreach** $k \in \{2, \dots, |U|\}$
 - (3) $\tau_k := \text{Solve OP1}$
 - (4) $r_k^* := \tau_k - \tau_{k-1}$
 - (5) **end**
-

5.3.2 Relaxed Max-Min Fair Rate Allocation (MESHMAX-LP)

MESHMAX-OPT is a particular case of an unsplittable maximum flow problem. With respect to the specific application of online network optimization, it requires a too long time to be solved in reasonable quality. We thus consider faster solution approaches, starting by considering a relaxed version of the problem that can be solved more easily. Specifically, for the moment we drop the single-path routing requirement, thus allowing the flow to be split among multiple paths between every origin-destination pair. We remark that solving the splittable version of the max-flow problem is computationally easier and the solution provides an upper bound on the unsplittable version. The solution to the relaxed splittable problem is then given as input to a fast heuristic that derives a (in general non-optimal) unsplittable solution.

$$\text{maximize } \alpha \tag{5.26}$$

$$\begin{aligned} & \sum_{(i,j) \in E} f_{(i,j)}^u - \sum_{(j,i) \in E} f_{(j,i)}^u = F^u(j) \\ \text{with } F^u(j) = & \begin{cases} \alpha & j = u \wedge u \in U \setminus U' \\ -\alpha & j = v_u \wedge u \in U \setminus U' \\ 0 & \text{otherwise} \\ d_u & j = u \wedge u \in U' \\ -d_u & j = v_u \wedge u \in U' \end{cases} \quad \forall j \in V, u \in U \end{aligned} \tag{5.27}$$

$$\sum_{u \in U} \frac{f_{(i,j)}^u}{b_{(i,j)}} + \sum_{((i,j),(k,l)) \in I} \sum_{u \in U} \frac{f_{(k,l)}^u}{b_{(k,l)}} \leq \eta \quad \forall (i,j) \in E \tag{5.28}$$

$$f_{(i,j)}^u \geq 0 \quad \forall u \in U, (i,j) \in E \tag{5.29}$$

To solve the problem of max-min fair rate allocation with splittable flows, we first reformulate OP2 to OP3 (Equations 5.26-5.29) and then use the well-known water-filling approach [140].

OP3 maximizes α , the rate of non-saturated STAs, subject to rates already computed for saturated STAs. For this purpose, we define the set U' , which contains all saturated STAs. Furthermore, we define a parameter d_u ($u \in U'$), which denotes the traffic demand of a saturated STA u . Equation 5.27 enforces a rate α for each non-saturated STA and a rate d_u for each saturated STA. Constraints 5.28 and 5.29 enforce the

capacity constraints and positive flow rates.

We then iteratively compute d_u with Algorithm 5.2: at each iteration, newly saturated STAs are identified and added to set U' , until all STAs are saturated. To check if a STA u is saturated, one can solve OP3 and fix the demands of all STAs except for u . If the rate r_u can not be increased, STA u is saturated.

Algorithm 5.2 Max-min fair bandwidth allocation for multi-commodity multi-path networks

Input : Network graph G and set of users U

Output : Max-min fair rate allocation vector \mathbf{r}

```

(1)  $\mathbf{d} := 0, U' := \emptyset$ 
(2) while (Set of saturated users  $U'$  does not contain all users) do
(3)    $\alpha :=$  Compute minimum rate by solving OP3
(4)    $U'_{new} :=$  Identify saturated demands
(5)    $U' := U' \cup U'_{new}$ 
(6)   foreach  $u \in U'_{new}$ 
(7)      $d_u := \alpha$ 
(8)   end
(9) end
(10)  $\mathbf{r} := \mathbf{d}$ 

```

We use the results of Algorithm 5.2 to compute a single-path routing solution for the max-flow problem by consecutively selecting the maximum capacity path for each STA. The maximum capacity path for a STA can be found by constructing a new graph from all edges with positive flow (computed by Algorithm 5.2) for the corresponding STA. On this graph a single-commodity unsplittable flow problem is solved. To our experience this new graph tends to be small (as we only consider edges that carry flow of the respective STA). For this reason, the maximum capacity path can be found very fast by solving the MILP formulation through CPLEX. We remark that strictly polynomial time solution algorithms for this problem exist (e.g., [132], [91]). For example, [91] requires solving a Minimum Spanning Tree (MST) problem. The MST has a lower theoretical worst-case computational complexity than the MILP formulation. However, when solving the MILP with a state-of-the-art solver such as CPLEX, which has been refined over years and has reached extremely high level of efficiency, it can practically beat also specialized non-commercial prototypal implementation of polynomial time algorithms. We tested the performance of both approaches with a network consisting of 100 edges and 39 vertices. CPLEX solved the MILP formulation in 0.01 seconds, while the MST

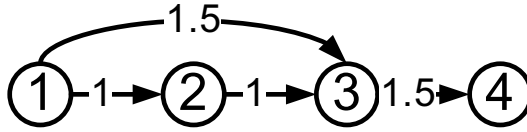


Figure 5.3: Network for which Algorithm 5.2 does not terminate

algorithm required 0.33 seconds. Hence, we use the MILP formulation for all experiments.

Algorithm 5.2 does not require to introduce the new variables β and λ . On the basis of our computational experience, Algorithm 5.2 performs better than Algorithm 5.1. However, Algorithm 5.2 cannot be applied to the original MESHMAX problem (single-path) as there is the possibility that a saturated user is not found and hence the algorithm may not terminate. This can be shown by the simple example depicted in Figure 5.3 and the corresponding feasible region as depicted in Figure 5.4: we consider two users, with demands associated with origin-destination pairs (1, 3) and (1, 4) respectively. In the first iteration of Algorithm 5.2 each user gets assigned 1 Mbit/s. Now, as the paths of both users are not fixed, each demand can potentially be routed via link (1, 3) and none of the users is saturated, as either one of them could potentially be improved to 1.5 Mbit/s. Hence, the termination condition (have only saturated users) of the loop in Algorithm 5.2 will never be fulfilled.

Graphically, Algorithm 5.2 can be interpreted as follows: it first finds a point on the efficient frontier of Figure 5.4 at which both users have the same rate. It then checks if it can improve the rate of one user without decreasing the rate of the other. In the case of single-path routing, the algorithm first finds point (1, 1), but then cannot decide in which direction to progress.

In contrast, when applying Algorithm 5.2 to multi-path networks, both users get assigned 1.25 Mbit/s in the first iteration and all links are saturated. For this reason, Algorithm 5.1 needs to be used for a non-convex feasible region, which is typical for single path routing.

5.3.3 Heuristic Solution Algorithm (MESHMAX-FAST)

In the case of real network instances, solving MESHMAX-LP corresponds to solving a large LP, making it too complex for online optimization. We can reduce the problem complexity, by considering that 1.) in mesh connected WLANs, most of the traffic is exchanged between the STAs and the Internet (and not among mesh nodes or STAs), and 2.) that typically

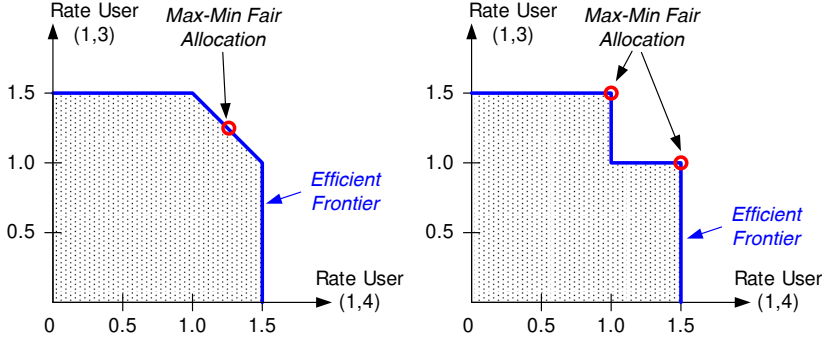


Figure 5.4: Feasible set for multi-path (left) and single-path routing (right)

there are many STAs, but only a few MAPs. In the following, we decompose the original MESHMAX problem into a number of sub-problems, which can be individually solved fast:

1. Flow-maximization: find the maximum flow (under fairness considerations) between STAs and the Internet
2. (Initial) STA/MAP assignment: find a MAP for each STA
3. STA/MAP re-assignment: move STA between MAPs, in order to increase the minimum STA rate
4. Routing: compute a feasible routing for each STA.

As shown in Figure 5.5, the sub-problems are solved iteratively, until the minimum per STA rate cannot be improved anymore. Instead of directly facing a multi-commodity max-flow problem that requires time to get solutions of reasonable quality, we first compute a single-commodity max-flow problem with the Internet as source and the MAPs as sinks and an overall STA/MAP assignment (computationally simpler). We then use the results to find a feasible solution with respect to Equations 5.26-5.29, i.e. the original MESHMAX problem.

This approach works particularly well when the flow traversing a MAP is divided into flows for many STAs and when most traffic is between the STAs and the Internet.

5.3.3.1 Sub-Problem I: Flow Maximization

With the mesh nodes M , gateways W and the fictitious nodes t and t' , we build a reduced connectivity graph $G'(V', E')$, where $V' = M \cup W \cup \{t, t'\}$.

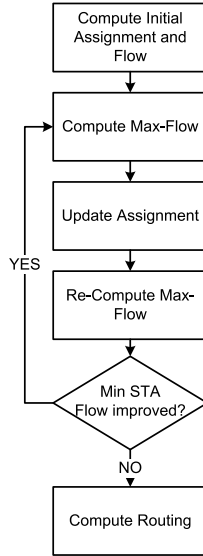


Figure 5.5: Sequence diagram of the MESHMAX-FAST algorithm

The set of edges E' includes the original mesh backbone links from E as well as edges from all MAPs to t' and from all gateways to t . With t we model a consolidated source node representing the Internet and t' is a consolidated sink node. I' is the set of conflicting links and is created as described in Section 5.2.

We aim to find \mathbf{f} , which maximizes the flow from the Internet that is passing through MAPs, weighted by the parameter \mathbf{a} , which is the number of STA associated to a MAP. The corresponding linear optimization problem (OP4) is defined in Equations 5.30-5.35.

Similar to Section 5.2, Equations 5.31 and 5.32 ensure flow conservation and capacity constraints. Equation 5.33 and 5.35 make sure that STA rates do not differ by more than a configurable factor q ($q = 1$ enforces equality) and that each STA u receives a flow rate of at least d_u . With Equation 5.35 the flow passing through a MAP is distributed to its connected STA equally. Note that LP4 has no integer constraints and that its size does not depend on the number of STA, which makes it much easier to solve than OP1-OP3.

5.3.3.2 Sub-Problem II: Establishing STA/MAP Assignments

Each STA should be associated to exactly one MAP. We formulate the STA/MAP assignment problem as a maximum cardinality bipartite matching problem [32]. In this problem, a graph is partitioned into two sets of

$$\text{maximize } \sum_{j \in V'} a_j f_{(j,t')} \quad (5.30)$$

$$\sum_{(i,j) \in E'} f_{(i,j)} - \sum_{(j,i) \in E'} f_{(j,i)} = F(j)$$

$$\text{with } F(j) = \begin{cases} \sum_{(i,j) \in E'} f_{(i,j)} & j = t \\ 0 & \text{otherwise} \\ -\sum_{(i,j) \in E'} f_{(i,j)} & j = t' \end{cases} \quad \forall j \in V' \quad (5.31)$$

$$\frac{f_{(i,j)}}{b_{(i,j)}} + \sum_{((i,j),(k,l)) \in I'} \frac{f_{(k,l)}}{b_{(k,l)}} \leq \eta \quad \forall (i,j) \in E' \quad (5.32)$$

$$qa_j f_{(i,t')} \leq a_i f_{(j,t')} \quad \forall i, j \in V' : a_i, a_j > 0 \quad (5.33)$$

$$f_{(i,j)} \geq 0 \quad \forall (i,j) \in E' \quad (5.34)$$

$$\sum_{(j,t') \in E'} f_{(j,t')} \geq a_j d_j \quad \forall j \in V' \quad (5.35)$$

vertices U and T , so that all edges have one endpoint in U and the other endpoint in T . A matching is a set of edges K , such that every vertex is endpoint of at most one edge in K . Finding the set K of maximum cardinality is called maximum matching problem. This corresponds to finding the max-flow in an augmented graph - a task that can be accomplished for example by the Hopcroft-Karp algorithm [32] or by solving an LP. In our implementation we relied on the direct use of CPLEX to solve the problem, instead of using our implementation of the Hopcraft-Karp (in our preliminary tests, the latter found the optimal solution to the STA/MAP assignment with 30 clients in 0.35, against the 0.1 seconds needed by CPLEX).

The STA/MAP assignment is calculated on a graph with the vertex set $U \cup T$, where U is the set of STAs and T is a set of virtual service slot nodes. Each MAP m is split into σ_m service slots nodes. σ depends on the capacity of a MAP and will be calculated in Algorithm 5.5. Edges between nodes in U and T are created according to connectivity given by G . Every $e \in K$ is an association between a MAP and a STA. Figure 5.6 depicts an example graph for finding a matching.

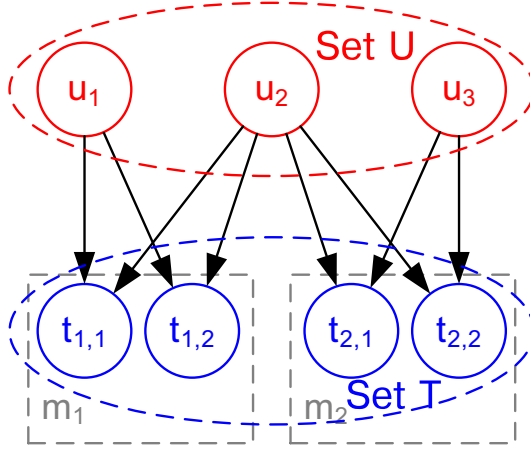


Figure 5.6: Example: Finding a matching, where STA 1 can connect to MAP 1, STA 3 to MAP 2 and STA 2 to both ($\sigma = 2$).

5.3.3.3 Sub-Problem III: Increasing the Minimum STA Rate

Given are the flow rate vector \mathbf{f} , graph G and a matching K , we would like to find new matching K' and a corresponding association vector $\mathbf{a} = \{a_1, \dots, a_{|V|}\}$, which maximize the minimum STA rate. This problem could be formulated as a MILP. As integrality constraints make the problem hard to be solved in reasonable time for larger problem instances, we develop a simple heuristic algorithm. The algorithm is based on a re-association graph $R(V, Z)$. An edge $(u, v) \in Z$ is created if there exists a STA that can be associated to both u and v . If there exists a path $p^{u \rightarrow v}$, it is possible to increase the number of STAs associated to u and decrease the one of v by re-associating STAs, potentially requiring STAs on other MAPs to re-associate.

For example, in Figure 5.7, u_1 and u_2 are first connected to m_1 and u_3 is connected to m_2 , m_3 has no STA connected. There is no STA on m_1 that could be moved directly to m_3 . But following the graph on the reassociation graph, one can see that u_2 can be moved to m_2 and u_3 can be moved to m_3 . Hence, we can decrease the association count a_1 of m_1 , by increasing the association count on m_3 .

Algorithm 5.3 checks all MAPs pairwise to see if by moving a STA along the re-association graph the minimum STA rate can be increased, while considering the link capacity and the interference according to the contention graph. If this is possible, K is updated accordingly.

Algorithm 5.3 Increasing minimum STA rates by re-associations

Input : Flow rate vector \mathbf{f} , association count vector \mathbf{a} , re-association graph R and initial matching K

Output : Updated matching K'

- (1) $asc :=$ Sort MAPs by per STA rate, ascending
- (2) $dsc :=$ Sort MAPs by per STA rate, descending
- (3) foreach $h \in dsc$
- (4) foreach $l \in asc$
- (5) if (Re-association graph has a path from l to h *and*
- (6) Moving a STA from h to l increases the min rate *and*
- (7) according to R capacity is available) *then*
- (8) Update K and \mathbf{a}
- (9) fi
- (10) end
- (11) end
- (12) $K' := K$

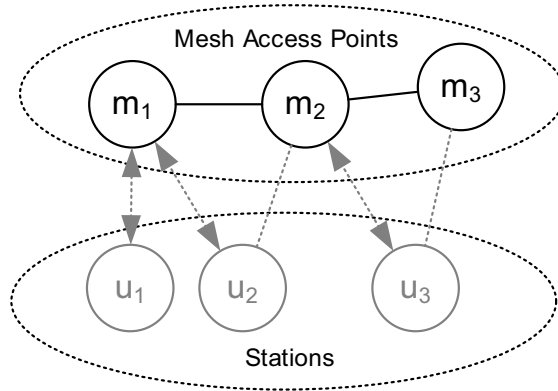


Figure 5.7: Example re-association graph. u_1 can connect to m_1 , u_2 to m_1 and m_2 , u_3 to m_2 and m_3 . One can increase the association count at m_3 by moving u_2 and u_3 .

5.3.3.4 Sub-Problem IV: Routing

Given the link flow rates \mathbf{f} for the $t \rightarrow t'$ max-flow and the matching K , we would like to find for each STA $u \in U$ associated at MAP m a path $p^{t \rightarrow u} = \{(t, i), \dots, (m, u)\}$ capable of forwarding $f_{(m, t')}/a_m$ flow without violating capacity constraints imposed by the link flow rates. In general, the $t \rightarrow t'$ max flow is only feasible for split flows. To obtain unsplit flows for individual users, we make use of the following observation: the total flow of a MAP m is comprised of a_m individual flows, which provides a natural way to split up the total flow of a MAP without splitting traffic of an individual user. We create a routing graph RG from G' by removing all edges with zero flow. Further, let $c_{(i,j)}$ denote the available capacity on edge (i, j) . Initially, $c_{(i,j)}$ equals $f_{(i,j)}$. The cost of each edge (i, j) is set to $(1/c_{(i,j)})^{10}$, so that the use of links with low capacity is penalized.

Algorithm 5.4 finds a path for each user u that does not violate flow constraints for a rate r'_u . In the first two nested loops (lines 3-16) each user gets assigned a path to t . The path is found using a minimum cost route algorithm for graphs (Dijkstra's algorithm in our implementation). Each time a user gets assigned path, the capacity $c_{(i,j)}$ of a link (i, j) on the user's path is reduced by the requested user rate r'_u . The link costs $cost_{(i,j)}$ are updated accordingly, so that links with lower available capacity will have a higher cost.

After assigning the initial flow rates to the users, it might happen that some users share the same path, but one user gets a higher rate than the others (since it was assigned to the path earlier). Therefore, in the last step, the rates of all users sharing the same path (except for the last hop, of course) are divided equally.

5.3.3.5 Solution Algorithm

We introduce Algorithm 5.5 to iteratively solve sub-problems I-IV, aiming to maximize the minimum STA rate. Line 1 initializes *low*, which denotes the minimum throughput that any of the STAs receives. In lines 2-3 an initial matching K is calculated and the number of associated STA for the given matching is computed. We call the algorithm **MESHMAX-FAST** if the initial matching is computed by associating each STA to the MAP with the lowest hop count to the next gateway. For **MESHMAX-FAST*** we compute the maximum flow $t \rightarrow t'$ on a graph with edges $E \cup \{(u, t') \forall u \in U\} \cup \{(t, v) \forall v \in M\}$ and vertices $V \cup \{t, t'\}$ (i.e. the original graph G is augmented with a consolidated source node t and a sink node t' , which are connected to the gateways and STA respectively). For each STA we select the MAP from which it receives the highest flow.

Algorithm 5.4 Routing of user traffic

Input : User rate vector \mathbf{r} , link costs \mathbf{c} and routing graph RG **Output** : Set of paths P and adjusted rate vector \mathbf{r}'

- (1) Sort users U by rate, ascending
 - (2) Initialize cost of all links to $1/c_{(i,j)}^{10}$
 - (3) foreach $u \in U$
 - (4) $map :=$ Get MAP of user u based on matching K
 - (5) $p^{t \rightarrow u} :=$ Find minimum cost path from t to u according to $cost$
 - (6) $r'_u :=$ Minimum of user rate r_u and the available capacity on link $p^{t \rightarrow u}$
 - (7) foreach $(i, j) \in p^{t \rightarrow u}$
 - (8) $c_{(i,j)} :=$ Update the available capacity to $c_{(i,j)} - r'_u$
 - (9) if $c_{(i,j)} > 0$ then
 - (10) $cost_{(i,j)} :=$ Update the cost of the link to $1/c_{(i,j)}^{10}$
 - (11) else
 - (12) $cost_{(i,j)} :=$ Update the cost of the link to ∞
 - (13) fi
 - (14) end
 - (15) Add $p^{t \rightarrow u}$ to the set of paths P
 - (16) end
 - (17) foreach $u \in U$
 - (18) $S :=$ Find all other stations $\in U$ with same path as u
 - (19) Divide the flow rates equally among all stations in S and update r'
 - (20) end
-

Algorithm 5.5 MESHMAX-FAST solution algorithm**Input** : Network graph G , set of interfering links I and set of users U **Output** : User rate allocation vector \mathbf{r} and set paths \mathbf{P}

-
- (1) $low_{old} := -1$
 - (2) $K :=$ Get an initial matching
 - (3) $\mathbf{a} :=$ Get the association count vector for the matching K
 - (4) $\mathbf{f} :=$ Compute the max-flow by solving OP4 with ($q := 1, d := 0, \mathbf{a}$)
 - (5) $low_{cur} :=$ Get the smallest per STA flow as $\min(\mathbf{f}/\mathbf{a})$
 - (6) while $low_{cur} > low_{old}$ do
 - (7) $low_{old} := low_{cur}$
 - (8) $\mathbf{f} :=$ Compute max-flow by solving OP4 with ($q := 0, d := \mathbf{a} * low_{cur}, \mathbf{a}$)
 - (9) $K :=$ Use Algorithm 5.3 to update the matching using the new flow rates
 - (10) $\mathbf{a} :=$ Get the association count vector for the matching K
 - (11) $low_{cur} :=$ Get the smallest per STA flow as $\min(\mathbf{f}/\mathbf{a})$
 - (12) $\mathbf{f} :=$ Compute max-flow using OP4 with ($q := 0, d := \mathbf{a} * low_{cur}, \mathbf{a}$)
 - (13) $low_{cur} :=$ Get the smallest per STA flow as $\min(\mathbf{f}/\mathbf{a})$
 - (14) end
 - (15) $\mathbf{P}, \mathbf{r} :=$ Get paths and updated rate allocation vector using Algorithm 5.4
-

In lines 4-5 the per-STA flow rates are computed and the smallest one is assigned to low .

Lines 6-14 constitute the main loop of the algorithm. In line 8 the link flow rates \mathbf{f} are computed by solving OP4. Equality of all STA is enforced ($q = 1$) and no minimum rates are required ($m = 0$). Lines 9-11 update the matching K by calling Algorithm 5.3 and re-compute low_{cur} ¹. In lines 12-13 the flow rates are computed again, this time requiring no fairness ($q = 0$), but guaranteeing at least low_{cur} to each STA. The main loop is repeated until the lowest STA rate does not increase anymore. Finally, routes and adjusted rates are computed using the algorithm 5.4. The user rate vector \mathbf{r} and be computed from the flow rate vector \mathbf{f} using Equation 5.8.

5.4 Numerical Performance Analysis

In this section, we compare the performance of the three previously described MESHMAX algorithms. All algorithms are implemented in the OPL modeling language and the Linear Programming problems were solved by IBM ILOG CPLEX 12.3 [6]. The tests were performed on a Linux

¹Note: \mathbf{f}/\mathbf{a} is a piece-wise division of a flow rate vector \mathbf{f} and association count vector \mathbf{a}

server with an Intel E5606 quad-core CPU (2.13 GHz) and 4 GB RAM.

Specifically, the main results that we will show are:

- When the time limit imposed by an online mesh problem is considered, even a state-of-the-art LP solver like CPLEX is not able to find solutions of reasonable quality for the MESHMAX problem within the time limit.
- In contrast, our heuristic approach MESHMAX-FAST, based on the four sub algorithms, provides a solution at high quality within the time limits required for online optimization of networks.

5.4.1 Evaluation Scenario

We compared the solution quality in 30 randomly generated topologies. In each topology, 20 mesh nodes were uniformly scattered over an area of 500x500 m. Non-connected networks were discarded. In each network, 4 gateway nodes were randomly placed and connected to the Internet with a 100 Mbit/s Ethernet connection. The mesh backbone was operated at 54 Mbit/s PHY rate (yielding approx 32 Mbit/s network layer rate), the STA/MAP links at 18 Mbit/s (yielding approx. 10 Mbit/s network layer rate). STAs were randomly dropped over the coverage area of the network.

5.4.2 Throughput

Figure 5.8 depicts the minimum, 90-percentile, maximum and average throughput as a mean value over 30 random networks. The minimum and 90-percentile throughput show how well the worst-off users are supported, while the average and maximum throughput indicate how well the overall network resources are utilized.

MESHMAX-OPT represents an upper bound for the minimum throughput, since OP2 maximizes the throughput of the worst STA (see Section 5.3). This upper bound is almost achieved with MESHMAX-FAST*. On average 98% of the optimum rate are possible. This surprisingly good performance can be explained as follows: optimizing split flows from the Internet to each MAP is a good approximation of optimizing unsplit flows from the Internet to each STA, as each MAP aggregates traffic of many STAs. The approach of MESHMAX-LP to first optimize split flows for each STA individually and then approximate an unsplit version works slightly worse. For a few STAs it might occur that the difference between the split and approximated unsplit throughput is high. However,

as a look at the 90%-ile throughput reveals, this does not happen too frequently. In terms of 90%-ile throughput, MESHMAX-LP almost achieves the performance of MESHMAX-OPT.

Interestingly, in terms of minimum throughput, MESHMAX-FAST performs much worse than MESHMAX-FAST*. This shows that using a good initial matching in Algorithm 5.5 is critical for the performance. If the algorithm starts from a bad initial matching, the reshuffle operation might not attempt to move STAs to MAPs which are required for a good solution.

The ratio between the maximum and the minimum rate is a simple measure for how fairly resources are distributed. Among all the Pareto-efficient rate allocations, the max-min fair rate allocation minimizes this ratio. Therefore, it also measures how well a rate allocation approximates a max-min fair rate allocation. For MESHMAX-OPT the ratio was on average 3.19 (max. 12.03). Instead, it is 3.60 (max. 15.00) for MESHMAX-LP, 3.48 (max. 24.06) for MESHMAX-FAST* and 6.32 (max 29.06) for MESHMAX-FAST. This means, that MESHMAX-LP provides a more fair allocation of rates than MESHMAX-FAST and MESHMAX-FAST*. However, with regard to the minimum rate, MESHMAX-FAST* still outperforms MESHMAX-LP.

For two topologies with a total of 15 and 25 STAs, we made deeper investigations on the relative performance of MESHMAX-LP, MESHMAX-FAST and MESHMAX-FAST*. In Figure 5.9 we plot the CDF of the minimum throughput relative to MESHMAX-OPT, which was obtained by considering 30 random STA distributions over the area. A relative performance of 1 is equal to the optimal solution. For all three heuristic algorithms the relative performance depends on the location of the STA. For example, with MESHMAX-LP, the minimum rate can be equal to 49% up to 100% of the optimal solution. A similarly large difference can be observed for MESHMAX-FAST. With MESHMAX-FAST* the performance is between 70% and 100%, with an average of 98% and a median of 100%. In general, it is desirable that the worst case performance of a heuristic is as high as possible. MESHMAX-FAST* fulfills this requirement best with a worst case performance of 70%.

5.4.3 Run-time

As the algorithm is used for online optimization of the network and executed each time a STA joins or leaves the network or the association opportunities change due to user mobility, the run-time is critical. The rate of such changes depends on the network size and on the usage. [80] studied more

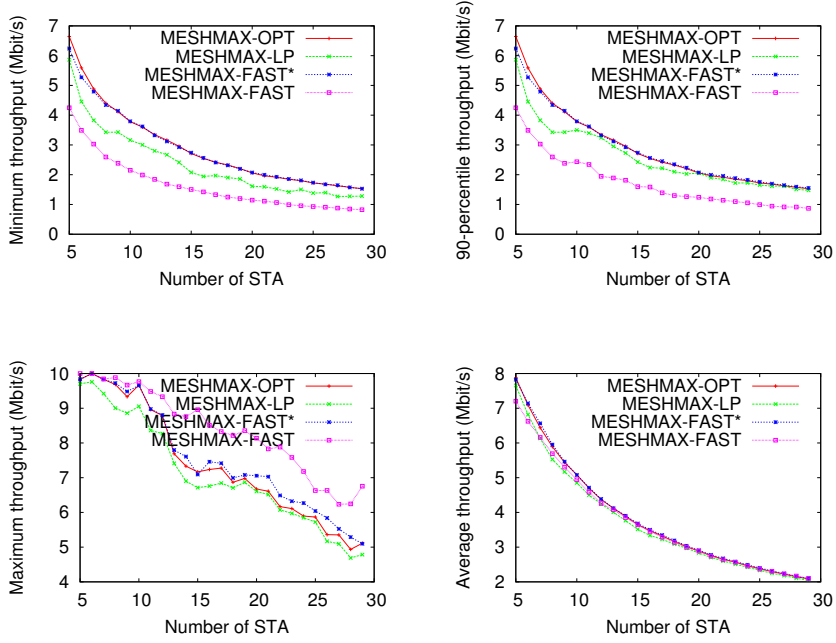


Figure 5.8: Minimum, 90-percentile, maximum and average throughput for 30 random networks

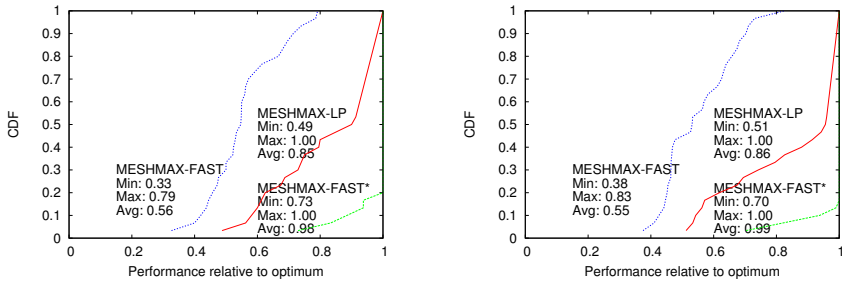


Figure 5.9: Empirical CDF of minimum throughput relative to optimum for 15 (left) and 25 (right) STA. MESHMAX-FAST* on average achieves 98% and 99% of the optimal performance. In more than 80% of the cases MESHMAX-FAST* computes the optimal solution.

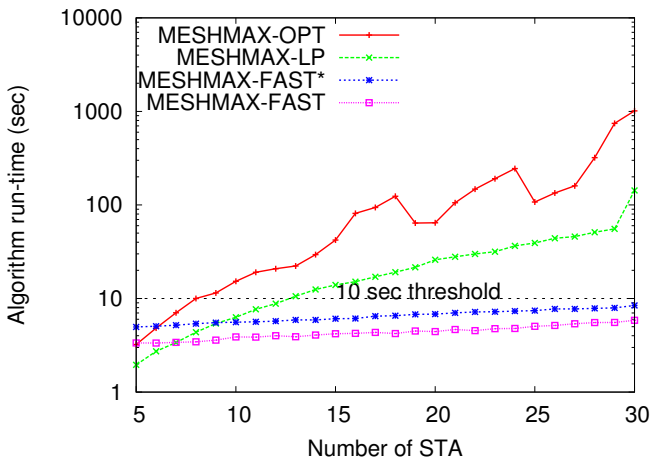


Figure 5.10: Comparison of algorithm run-time (averaged over 30 random topologies). The run-times of MESHMAX-OPT and MESHMAX-LP can be several orders of magnitude higher than the run-times of MESHMAX-FAST and MESHMAX-FAST*.

than 1.3 million connections in public WLAN hotspots. In high load situations the average arrival rate is roughly one customer per minute. To account for bursty arrivals, the run-time of the algorithm should thus be considerably shorter than one minute, e.g. 10 seconds.

Figure 5.10 compares the average run-time of the three algorithms in a random topology with 20 mesh nodes and 5-30 STA (observe the log-scale on the y-axis). As expected, the run-time of MESHMAX-OPT and MESHMAX-LP increases at (almost) exponential rate. MESHMAX-OPT and MESHMAX-LP can be used for online optimization (when the threshold is 10 seconds) in the evaluated networks only when up to 8 and 13 STAs are present respectively. The run-time growth of MESHMAX-FAST and MESHMAX-FAST* is much slower.

When a number of STAs between 5 and 15 is present, MESHMAX-FAST* is slower than MESHMAX-FAST, but gets relatively faster when more STAs are present. This is because initial matching of MESHMAX-FAST* takes longer, but then fewer iterations of the main loop of the algorithm are required.

In [80] the mean number of concurrent STA per AP during busy hours does not exceed 10. Further assuming a network size of 20 MAPs, a target network size of 200 STAs should be solved within the 10 seconds constraint. For 30 random networks of such size the average run-time for MESHMAX-FAST was 4.40 seconds. Even with 300 STAs the run-time is only 8.36 seconds.

Larger networks often can be partitioned into smaller logical units. For example, instead of optimizing a network that spans several floors of a building, one can optimize each floor individually. When there is no interference between the logical units and STAs of one unit cannot use MAPs of another unit, optimizing those partitions results in a global optimum.

5.4.4 Discussion

Based on the results of the numerical analysis we make the following key observations: 1.) in terms of solution quality and run-time, the direct solution of a straightforward relaxation (MESHMAX-LP) of the original MILP problem (MESHMAX-OPT) is not as good as an algorithm that considers the structure of the network (MESHMAX-FAST*); 2.) solving MESHMAX-OPT requires too much time to be solved during the online optimization of networks of practical size. However, its solution can be used by system designers to establish a bound on the quality of the solution obtained by heuristics; 3.) MESHMAX-FAST* is a practical heuristic,

that reveals to be very useful to find solutions of good quality within the time constraint associated with the application.

5.5 Network Simulations

The numerical analysis provides valuable insights on the performance of the algorithms. However, the algorithms are based on simplified models of the wireless channel and user traffic. We hence conducted a series of ns-2 simulations to investigate how suited the simplified model of the wireless channel and MAC is for the MESHMAX algorithms. Furthermore, the ns-2 simulations enable us to compare the performance of the MESHMAX algorithms to several state-of-the-art heuristics for AP selection.

5.5.1 Scenario

All simulations were conducted with ns-2 2.33 and the ns-miracle extensions [39]. Each mesh node was equipped with two radios, of which one is operated using IEEE 802.11a at 5 GHz and solely used in the mesh backbone. The other radio were tuned to 2.4 GHz and provides an AP interface to the stations using IEEE 802.11g. On the backbone, all the radio devices used the same channel, while for the STA access the MAPs were assigned non-interfering channels. The PHY rates were fixed to 36 Mbit/s. Errors in the wireless transmission were introduced randomly. Frames with a small SINR are more likely to be erroneous and thus dropped than frames with a high SINR.

All MGWs are connected to a core router (as in Figures 5.1 and 5.2), which represents the Network Operations Center (NOC) and acts as exchange point to the Internet. If not stated otherwise, the fixed line from the MGW to the core-router is operated at 100 Mbit/s. The NOC collects monitoring information from the network, executes the optimization algorithm and configures the forwarding tables at the mesh nodes and the rate-shapers at the core-router and mesh nodes. Such functionality could easily be implemented in real deployments using the architecture which will be presented in the next chapter. The rate-shapers are configured according to the flow rates computed by the optimization algorithm.

The comparison consists of two network topologies: the first network consists of 15 mesh nodes (of which 4 are MGWs) and 15 STA deployed in a random way. The other network resembles a subset of a real WMN deployed in Chaska [1]. It consists of 67 mesh nodes (16 MGWs) and 50 randomly dropped STA.

The MESHMAX algorithms are compared to two schemes which can be implemented in a fully distributed way: an RSS-based association policy and a minimum hop-count scheme. In the RSS-based scheme, a STA associates to the MAP with the highest signal strength (default policy in most modern operating systems). In the hop-count scheme, each MAP broadcasts its hop-count towards the next gateway in periodic beacon messages. A STA then associates to the MAP with lowest hop-count.

Each simulation was run for 180s and repeated 30 times with different random STA drops. The error-bars in the plots below are the standard deviation of the 30 random network instantiations. TCP throughput was measured with TCP New Reno, the Selective ACK option and one ACK per segment (no delayed ACKs). As a comparison, we also measured the UDP throughput with backlogged CBR traffic and 1400 bytes datagram length. This allows us to evaluate the impact of the TCP congestion control.

5.5.2 Throughput Performance

In Figures 5.11 and 5.12 we plot the UDP and TCP throughput for the random network. For the MESHMAX algorithms we additionally plot the analytical optimum. MESHMAX-FAST* and MESHMAX-OPT are about equal, followed by MESHMAX-LP. With the RSS-based scheme the minimum throughput is about 20 times lower than with the MESHMAX schemes. The hop-count scheme is much better, as it avoids having STA using MAPs far from the next gateway. The performance of the MESHMAX algorithms very closely matches the predictions made by the model. On average 99% of the model predictions are reached. This shows that the simple linear approximation of the MAC layer characteristics in our system model indeed is sufficient to achieve a good correspondence between the simple model and the more complex simulation assumptions.

With TCP the picture changes slightly: TCP implements a congestion control algorithm to determine the send rate and requires ACKs, which both reduce the available rate and can exacerbate unfairness issues. With TCP, the RSS-based scheme leads to a almost complete starvation of some flows. It is a well known behavior of TCP that flows close to the gateway can starve other, longer flows (see for example [114]). The hop-count scheme improves the situation, since STAs are associated close to the GWs. The MESHMAX algorithms in addition shape the rates, so that some flows cannot completely capture all resources.

The analytical optimum and the TCP throughput diverge more than the UDP throughput. The simulations reach on average 84% of the analytical

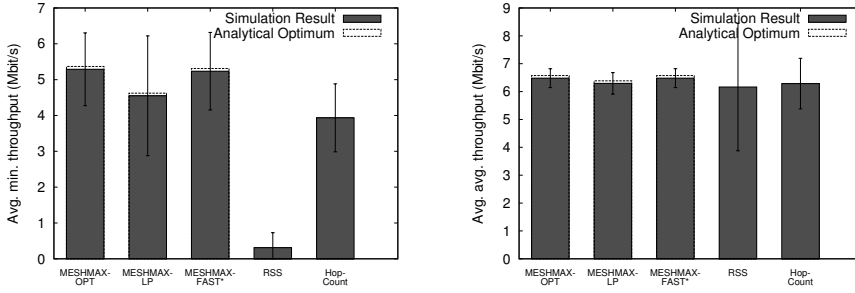


Figure 5.11: Minimum (left) and average (right) UDP throughput in a random network. The simulation results match the analytical optimum very closely. The RSS-based association can lead to a starvation of users, while MESHMAX avoids this.

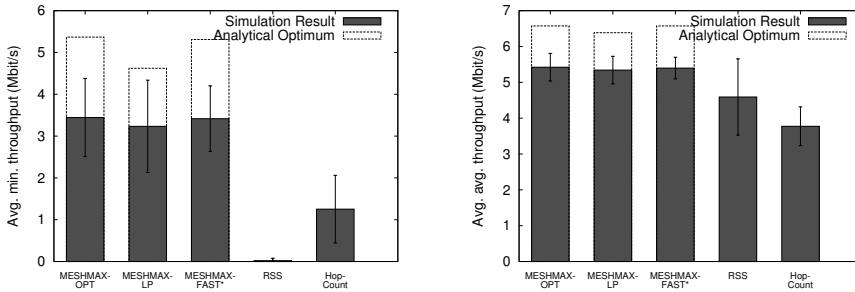


Figure 5.12: Minimum (left) and average (right) TCP throughput in a random network. The TCP throughput is lower than the analytical optimum. TCP connections in the RSS-based association scheme can be completely starved.

throughput. This is not surprising, as the congestion control reacts to losses of DATA or ACK frames by reducing the send rate. If TCP frames get lost, it takes a while until TCP recovers to the rate computed by the MESHMAX algorithm. Hence, the average rate over the whole simulation time is lower than the maximum reachable by a UDP connection.

The Chaska topology is too large for MESHMAX-OPT and MESHMAX-LP to be solved in acceptable time. We therefore only compared the performance of MESHMAX-FAST* and the RSS and hop-count policy. For both, UDP and TCP, MESHMAX-FAST* performs better than the RSS-based and the hop-count based scheme. With UDP traffic and MESHMAX-FAST*, each STA receives on average 5.6 Mbit/s, which is 22% more than with the RSS-based scheme and 27% more than with the hop-count based scheme. With TCP-traffic, MESHMAX-FAST* on average provides 4.5 Mbit/s for each STA, which is 20% and 40% higher than the RSS and hop-count scheme. With the RSS and the hop-count scheme some TCP flows are completely starved, resulting in zero throughput. MESHMAX-FAST* at least provides on average 0.44 Mbit/s to the worst-off STA.

5.5.3 Impact of Bottlenecks at Gateways

Many WLAN hotspots are connected to the Internet via an xDSL line, which might be slower than the wireless backhaul and hence present a performance bottleneck. We analyzed how sensitive the performance of the MESHMAX algorithms is to different gateway connection speeds. Figure 5.13 plots the minimum UDP throughput relative to the optimum given by MESHMAX-OPT as an average of 30 simulations using random topologies.

The measurements reveal that there is an impact for all algorithms, albeit no clear trend can be seen for some of them. MESHMAX-LP performs better with higher gateway speeds, because the difference between split and unsplit flow rate is lower in that case. MESHMAX-FAST* shows no clear trend. However, it is always outperforming the other schemes. The RSS-based scheme is in particular bad when the gateway speed is high. This is because in the RSS scheme, stations may associate to MAPs which are many hops from the gateway. Transmissions to such distant stations are starved by transmissions to nodes closer to the gateway. When the gateway speed is low, the wireless backhaul might not be saturated and therefore the impact of this competition on for resources on the wireless backhaul plays no major role. With high gateway speeds this competition is leads to starvation of some nodes.

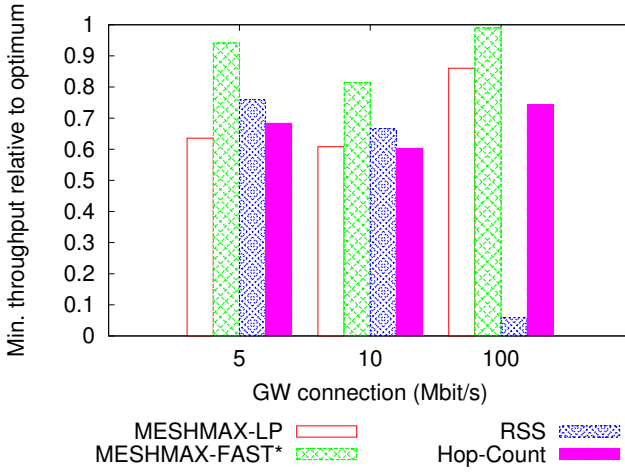


Figure 5.13: Impact of different gateway connection speeds

5.5.4 Increase in Network Scalability

From a network operators point of view it is interesting how many customers can be supported with a minimum rate. In Figure 5.14 we plot the simulated average minimum throughput value for each STA over 30 random topologies for MESHMAX-FAST* and the RSS-based association. If the network operator would like to provide on average at least 1 Mbit/s rate for each STA, then the default scheme (RSS) can only support 7 concurrent STA. With MESHMAX-FAST* 39 users can be supported. Figure 5.14 shows the actual value of the proposed optimization algorithm: with a given network infrastructure and rate-requirement a much higher number of customers can be served. For a 1 Mbit/s rate requirement, a 5.5 fold improvement in the number of supported users is possible. Or conversely, for a given user population of e.g. 15, the minimum rate can be increased from 200 kbit/s to 2.6 Mbit/s.

5.5.5 Importance of Active STA Management

We call a STA *actively managed*, if it can inform the optimization algorithm which MAPs are in reach and if it can follow a handover request. In practice, not all STA are actively managed, as the IEEE 802.11 standard does not provide mechanisms for monitoring reachability of MAPs and

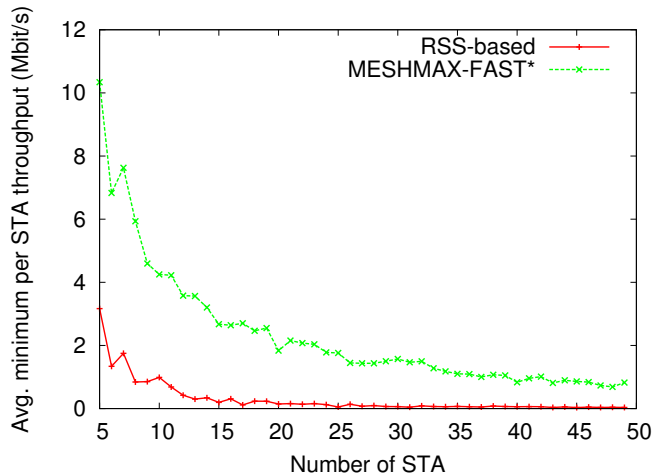


Figure 5.14: Increased network scalability through MESHMAX algorithms

performing handovers. Only newer standards, such as IEEE 802.11k [23] and IEEE 802.21 [25] enable such features. The MESHMAX algorithms do not distinguish between actively- or non-actively-managed STA. However, for non-actively managed STA the MAP is selected by the STA in a (potentially) non-optimal way and cannot be changed by the algorithm, even if better MAPs might be in reach.

In Figure 5.15 we plot the average minimum UDP throughput of 30 random topologies with 50 STAs, given that only a fraction of the STAs is actively managed and the rest uses the default RSS-based association scheme. As a comparison baseline we also provide the throughput, if all STAs use the RSS-based association and no rate-shaping is done. Not surprisingly, an increase of actively managed STAs allows the algorithm to find a better network configuration and hence the throughput is higher. The throughput increases from 0.25 to 0.5 Mbit/s, when going from 0 to 100% of actively managed STA.

The comparison of MESHMAX-FAST* with 0% actively managed STA and the RSS-based scheme (dashed line) shows that limiting flow rates and optimizing the routing doubles the throughput from 0.125 to 0.25 Mbit/s. This shows that even in a network with 100% legacy IEEE 802.11 STA, major performance improvements are possible.

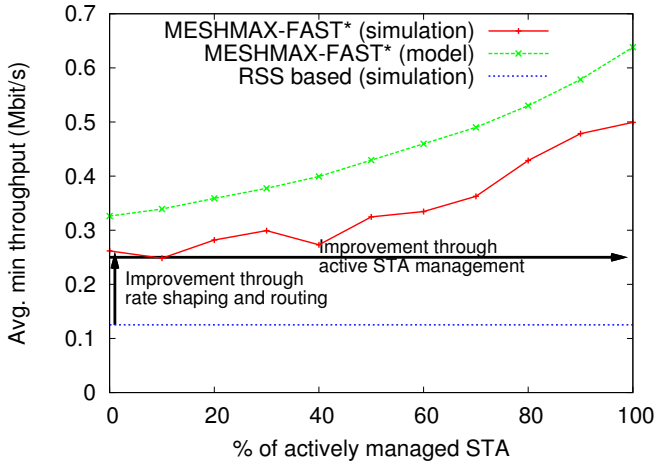


Figure 5.15: Average minimum throughput when associations are controlled on a fraction of all STA

5.6 Conclusions

In this chapter we have presented and compared several different approaches for optimizing the fair throughput of STAs in mesh connected WLANs. The presented network simulations show that the proposed optimization algorithms yield major improvements of flow rates, in particular for users that otherwise would be starved. For network operators this translates into more happy users. The exact solution algorithm provides an upper performance bound, which can be used as a benchmark for heuristics. The MESHMAX-FAST* heuristic can be implemented and used for online optimization of networks.

One limitation of the proposed models is that they do not consider the cost of handovers and the dynamicity of the network, as has been done in Chapter 4. As shown in Chapter 4, disregarding the handover costs is not problematic, as long as the handover costs and the network dynamicity are low. In contrast, in highly dynamic environments, handover costs should be considered. However, the MESHMAX algorithms are still useful in such scenarios, as they can be used as a building block for example for a Hysteresis based invocation strategy (see Section 4.4.1.3).

Another shortcoming of the presented approach is that flow rates are assigned statically to the stations, neglecting the fact most stations are

silent most of the time and do not want to download data. Thus, a future extension of the work should investigate how other stations can “borrow” capacity from a station that is currently not requesting any resources.

To manage a real network with one of the MESHMAX algorithms, appropriate signaling primitives to query the network resources and to configure the network according to the output of the algorithm are required. Standard mesh routing protocols are not suited for MESHMAX, as they do not allow a centralized computation and configuration of host-based routes and traffic flows. To enable such fine grained control over the network, a new management architecture is necessary, which we will introduce in the next chapter.

Optimizing Wireless Mesh Networks with OpenFlow

6.1 Introduction

In the previous chapters we have discussed the possibility of improving the performance of WLANs and WMNs by actively controlling the user associations and rate allocations. The presented optimization models require a centralized controller to obtain monitoring information from the network and to apply the outcomes of the optimization model by adapting the routing, modifying rate allocations and by triggering handovers.

Typical WMNs do not provide such functionality. For example, the wide-spread mesh routing protocols AODV [134], B.A.T.M.A.N. [96] and OLSR [62] do not allow fine grained flow-based routing. The idea of flow-based routing is, that instead of using only one path for a given destination, a destination can be reached on different paths, depending on the characteristics of the flow. Moreover, traditional routing protocols are not optimized to handle handovers of stations between APs. In order to maintain end-to-end connectivity, a station should not be required to change its IP address due to a handover. Therefore, when the point of attachment (i.e. the AP) of a station changes, the routing protocol needs to ensure that the routing tables of the mesh routers are updated, so that the traffic towards the station is routed to the new AP. As most routing protocols do not support such fast routing table updates, in practice often tunnel-based solutions such as Mobile IP [133] are deployed. Such tunnel based solutions create an overlay routing which is controlled by an external entities such as Home Agents.

Even though tunnel protocols such as Mobile IP are in principle suited to enable station mobility in mesh networks, they lack flexibility, introduce overhead at each transmission and do not integrate well with the routing process of the underlying network. Similar shortcomings have been observed in the architecture of fixed networks and have led to the development of Software Defined Networks (SDNs) and the OpenFlow protocol [121].

As already described in more detail in Chapter 2, the key idea of SDNs is to move the forwarding intelligence to a typically centralized control layer, while keeping the routers or switches simple. The control layer uses the OpenFlow protocol to program routers and switches. This architecture allows many applications, such as mobility management, to be implemented efficiently. In addition, OpenFlow can co-exist with legacy routing protocols and offers the possibility of network virtualization. This allows to run multiple logical networks on top of one physical network, for example to enable virtual network operators.

6.1.1 Related Work

SDNs and OpenFlow have been applied in wireless networks in different contexts. For example, PhoneNet [93] uses OpenFlow to enable efficient file-sharing applications between wireless terminals. The main idea behind PhoneNet is to create multi-cast trees between smart phones and thereby reduce the data transfer volume. The OpenRoads project [180] mainly investigates how to provide mobility management in heterogeneous wireless networks. However, OpenRoads is restricted to single-hop wireless communications and there is no integration with existing routing protocols. [126] and [181] propose to use OpenFlow to manage home networks. While the latter works do not consider multi-hop networks, [89] explicitly discusses the possibility of using SDNs to manage rural multi-hop wireless networks. [89] formulates several research challenges in this context, such as how to design SDN management tools for human operators with limited expertise. Concrete technical solutions to those problems are yet to be developed.

Besides SDNs, also other approaches to enable client mobility and flow based routing in WMNs have been proposed. For example, the CARMEN project developed an architecture for carrier grade mesh networks, which supports mobility by using layer-3 tunnels [40]. [34] uses multi-cast trees and tunnels for speeding up handovers. OLSR-FastSync [160] is an extension to OLSR, which allows fast route-updates after handovers. The CARS algorithm [169] allows flow-based routing in mesh networks. The main aim of the algorithm is to balance load between gateways and not to enable client mobility. Compared to the SDN solutions, a drawback of

those proposals is that they are not integrated in a broader management architecture. Thus, a reuse of functionality in other contexts and software tools is not possible easily.

6.1.2 Problem Statement and Contributions

The flexibility to integrate with existing networks and the centralized control make OpenFlow and SDNs an ideal candidate to manage station associations and mobility in WMNs. As the nature of wired links and networks is very different from wireless networks, in particular wireless mesh networks, the following aspects should be addressed when using OpenFlow for route and association optimization in WMNs: Due to variations in link qualities and nodes joining and leaving the network, the network topology changes at a much higher pace than in wired networks. Along with the self-configuration requirement of WMNs, this necessitates an autonomous topology discovery and the ability to react swiftly on changes in the network. In addition, as wireless networks do not have the clear notion of a point-to-point link, neighbor and topology discovery need to be adapted to wireless networks. Moreover, the capacity of WMNs is one or two orders of magnitude smaller than the one of fixed networks. Hence, the overhead introduced by communication between OpenFlow-enabled mesh nodes and the network controller deserves special attention. Furthermore, when compared to fixed networks, mesh routers typically have low processing power. As a result, performance bottlenecks could arise from the deployment of OpenFlow.

Previous works such as [180] have investigated how OpenFlow can enable mobility management in wireless networks. However, those works are limited to single-hop networks only and therefore do not deal with important WMN issues such as routing. Thus, in this chapter we aim to develop and evaluate an architecture for mobility management with OpenFlow in WMNs.

The *key contributions* presented in this chapter are:

- An **architecture** that allows for the flexible and efficient use of OpenFlow in WMNs. By integrating a legacy routing protocol, our architecture allows for a self-configuration of the network and an assessment of the link quality that is tailored to wireless networks.
- The **identification of important performance** factors and their evaluation in a real testbed. In particular, we show that on low-end mesh routers performance degradations due to large rule-sets are

possible. Furthermore, our measurements suggest that for small networks the overhead due to OpenFlow protocol messages is negligible.

- The application of the proposed architecture to **mobility management and performance optimization**. With the MESHMAX algorithms we compute the optimal routes, rates and associations and apply this configuration in the network with the help of OpenFlow.

The rest of this chapter is organized as follows: In Section 6.2 we outline the new architecture for OpenFlow-based management of WMNs. An implementation is described in Section 6.3. Section 6.4 evaluates performance characteristics of typical OpenFlow components in WMNs. In Section 6.5 we demonstrate how to use the proposed architecture to handle client mobility and to implement the MESHMAX algorithms. In Section 6.6 we summarize the chapter, discuss limitations of the proposed architecture and propose potential remedies.

6.2 An Architecture for OpenFlow in WMNs

In our architecture, which is depicted in Figure 6.1, a WMN consists of OpenFlow-enabled mesh routers and mesh gateways. Via a gateway, the WMN is connected to a core network, in which the OpenFlow Control Server and the Monitoring and Control Server are positioned. Regular stations can connect to the Internet via the mesh routers, which wirelessly forward the traffic to the gateway.

6.2.1 OpenFlow-Enabled Mesh Routers

As shown in Figure 6.2, a mesh node is equipped with one or multiple physical wireless cards. By using multiple wireless cards, the network capacity [102] may be increased. Each physical wireless interface is split into two virtual interfaces. One virtual interface is used for control traffic and the other one for data traffic.

Using multiple SSIDs, the mesh node can distinguish between traffic of the two interfaces. Multi-hop IP connectivity between the virtual control interfaces is enabled using any normal mesh routing protocol. In our specific implementation, which we will describe in detail below, OLSR [167] is used to setup the routing. The data interfaces are connected to the OpenFlow data path. The data path uses local sockets to communicate with the control path component. The control path component connects via the control interface and the secure channel to the OpenFlow controller

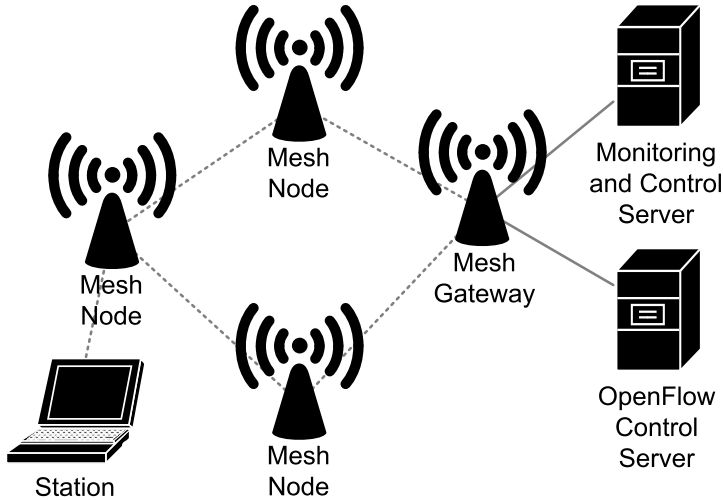


Figure 6.1: Overall system architecture

(e.g. NOX [83]), which is located in the core network. Connectivity to outside the mesh is achieved by using one or multiple mesh gateways. A mesh gateway is a normal mesh node, which in addition is connected to a fixed network.

On each mesh node a monitoring agent collects information about link quality and channel utilization. This monitoring agent can be queried by an external monitoring and control server.

When using multi-radio multi-channel mesh networks, the initial network setup requires a channel assignment to the radios and links in order to maintain connectivity. Our architecture does not rely on a specific mechanism. For example the approach in [141] can be used. However, when performing network optimizations, the channel assignment needs to be known to the network controller, since it determines the interference.

6.2.2 Core Network

The core network, which is shown in Figure 6.3 is composed of a *Monitoring and Control Server* (MCS) and an *OpenFlow Control Server* (OCS). The MCS queries information from the mesh routers and clients and builds a topology, an association and an association opportunity database. The topology database contains a network graph, which is built from the connectivity information of the mesh routing protocol. If a link-state routing protocol such as OLSR is used, the network connectivity graph

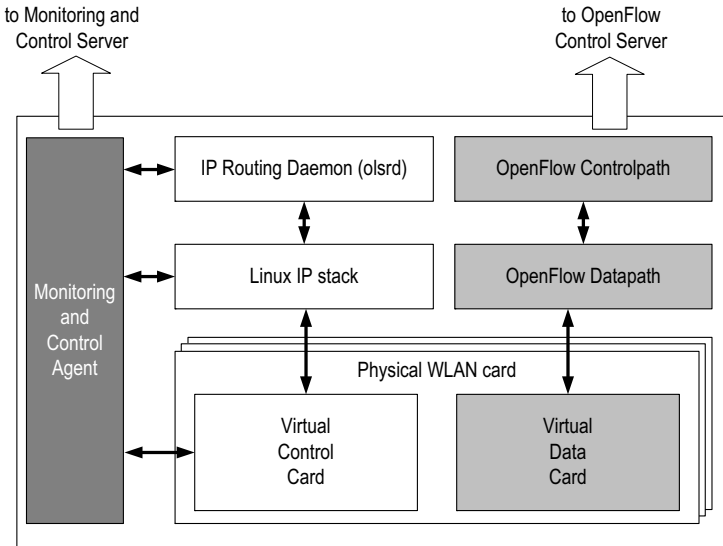


Figure 6.2: Architecture of an OpenFlow mesh node

can be constructed from the routing tables of the nodes. In addition, the network graph can be annotated with link quality metrics, if provided by the routing protocol. The association database contains a list of stations and the MAPs they are associated to and is built from the system statistics of the MAPs. During idle periods, the stations scan the network to find close-by MAPs and report this to the monitoring server, which builds the association opportunity database.

The main purpose of the OCS is to perform routing related tasks, such as setting forwarding tables, handling node mobility and managing network addresses. To this end, the OCS contains the routing and mobility application. The OCS can access all databases maintained by the MCS. When the topology database is changed (e.g. because of a link failure detected through periodic polling of the OLSR link tables at the gateways), the MCS informs the OCS, which then updates the routes based on the new network graph. When a station joins the network and issues a DHCP request, the network controller uses the topology database to calculate a path between the station's MAP and a gateway and then installs flow-rules via the OCS. The ARP application and the DHCP application manage IP addresses of stations. In Section 6.5 we will describe in more detail how the MCS and the OCS implement the MESHMAX algorithms.

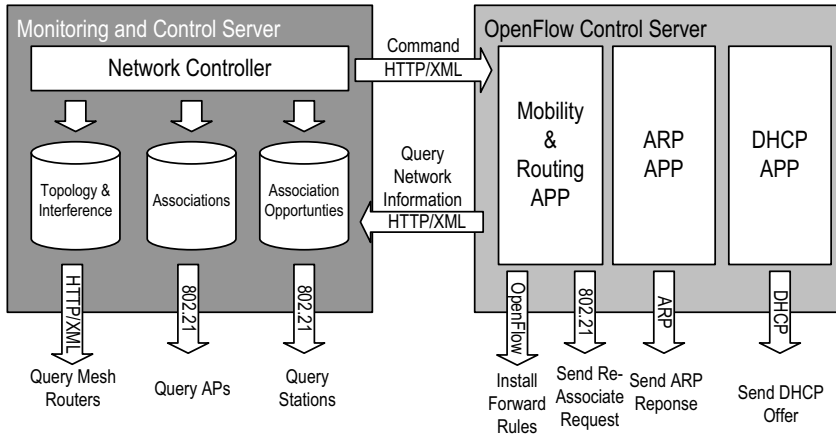


Figure 6.3: Architecture of the core network

6.2.3 Stations

The stations comply with the IEEE 802.11 standard when associating to a MAP. A monitoring and control agent may be installed on the station, which allows the MCS and the OCS to query information to build the association opportunity database and to trigger handovers using the IEEE 802.21 [25] command set. Legacy clients without the agent can still associate to the network, but cannot participate actively in the network management, as the OCS cannot trigger actions on such clients, for example to initiate a handover.

6.3 Implementation

We have implemented the proposed architecture in the KAUMesh testbed at Karlstad University [70] and the MultiRAT testbed of Telekom Innovation Laboratories [41]. In the KAUMESH testbed, the mesh routers are built upon the Cambria GW2358-4 platform that uses an Intel XScale IXP435 667 MHz CPU. The routers use OpenWrt Backfire [13] (Linux 2.6.31) and version 1.0.0 of the OpenFlow reference implementation for the control and data path [63]. The network virtualization is implemented using the multi-SSID feature of mac80211 and the ATH5K wireless driver [9]. We use NOX 0.5 [83] as the OpenFlow controller, which are installed on a Linux server with a 2.8 GHz Intel Xeon CPU and 2 GB of RAM. All performance measurements, except for the measurements with the MESHMAX algorithm, which will be presented in Section 6.5.4, were performed in the KAUMesh testbed. The measurements in Section 6.5.4

were performed in the MultiRAT testbed. This testbed is deployed in an office building in Berlin and comprises several Saxnet III mesh nodes. The software configuration is identical to the KAUMesh testbed, except for the operating system, which is Debian Linux instead of OpenWrt.

The communication among the individual components of the network architecture is based on standard protocols. The OpenFlow protocol is used for setting up the flow tables. The MCS and OCS communicate with each other using an XML-like protocol. The mobility application can create IEEE 802.21-like [25] messages (we do not follow the whole IEEE 802.21 specification) to trigger handovers at the stations. The stations run a rudimentary IEEE 802.21 implementation, which can trigger a handover upon the reception of such messages. The control network routing is set up using `olsrd` [167]. The `olsr-txtinfo` plugin provides the network topology information to the MCS via a Telnet based interface. The monitoring is performed with Nagios [106]. Furthermore, the mesh nodes and clients contain custom Nagios plugins, which are queried by the Nagios Remote Plugin Executor [106]. The association and topology databases are stored in a MySQL database.

6.4 Micro-Benchmarks

In this section, we evaluate the principle feasibility of OpenFlow in WMNs. We focus on general performance features of the architecture and the software components and evaluate the forwarding performance, the rule activation time and control traffic overhead.

6.4.1 Is there a Performance Penalty through OpenFlow Rule Processing?

As the data path on the mesh routers is implemented in software, it might become a performance bottleneck. In the used OpenFlow reference implementation, the data path runs in user space, which might create additional problems. To evaluate the achievable performance, we set up a string of three nodes, in which the leftmost node transmits data and the middle node forwards the data to the rightmost node using two separate network cards. The links between the leftmost and the middle and the middle and the rightmost node were tuned to orthogonal channels and use fixed PHY rates of 36 Mbit/s in the 5 GHz band. We made sure that no external traffic disturbed the experiments.

We measured the throughput of a backlogged UDP-stream sent over the two hops for 60 seconds. The achievable throughput using normal

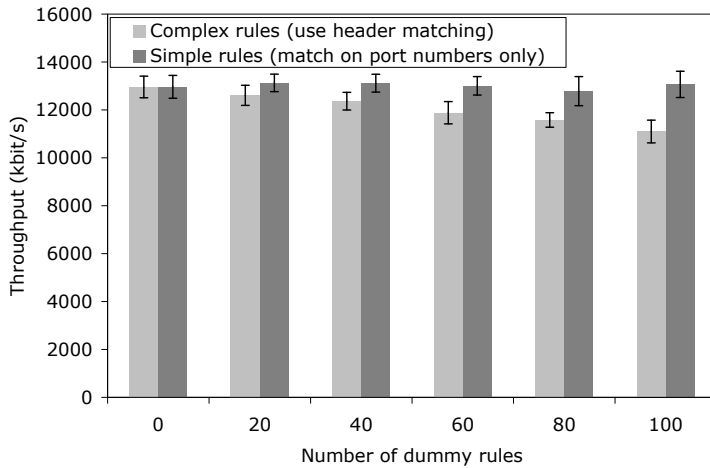


Figure 6.4: Forwarding performance with 1400 byte UDP datagrams

Linux IP routing without the OpenFlow components was approximately 20 Mbit/s. We repeated the measurements using the OpenFlow data path for packet forwarding. To see the impact of the rule set size, between 0 and 100 dummy rules were active and needed to be processed before the actual rule to forward the data is matched. Two kinds of rules were used: simple rules that match by the incoming port and complex rules that also match MAC and IP addresses (and hence need more processing power).

Figure 6.4 shows that the throughput degrades by about 15% when 100 complex rules need to be processed for each packet. For simple rules the parsing process is very fast and thus does not have impact on the throughput. For routing, IP header matching is required though. Those results show that the present implementation of the data path can indeed create a performance bottleneck on slow mesh-router devices. The results also suggest that keeping the number of rules low is beneficial for performance if user-mode forwarding is activated. The CPU utilization in the OpenFlow tests was higher than with the standard Linux IP forwarding, but did not reach 100%. We therefore think that the CPU speeds itself is not the performance bottleneck, but other factors such as the memory bandwidth degrade the throughput. A more efficient data path implementation in the OS-kernel might improve performance and is therefore very desirable. However, regardless of the size of the rule set, the throughput of the OpenFlow data path was always considerable lower than the normal Linux IP stack.

In control measurements on MultiRAT mesh nodes, which use a more

powerful AMD Geode CPU, no performance degradations compared to the normal Linux IP stack were observable. This shows that low-end mesh routers in combination with a non-optimal software implementation may be too slow. With more powerful devices performance degradations are not to be expected.

6.4.2 What Control Traffic Overhead is Created by OpenFlow?

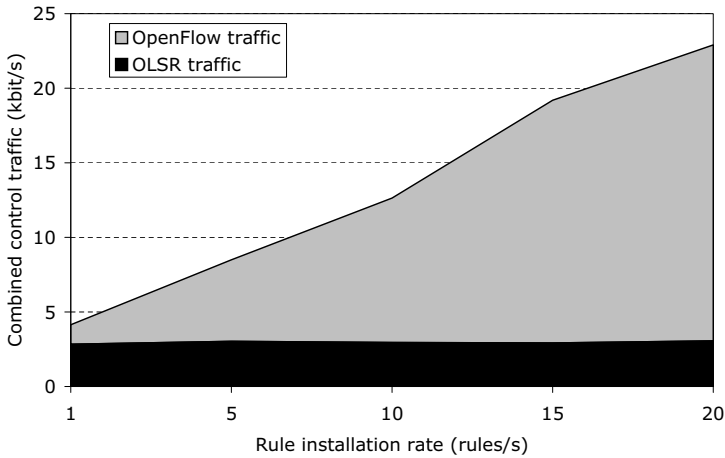


Figure 6.5: Total control traffic caused by OLSR and OpenFlow

OpenFlow creates control traffic when new rules are installed or statistics are queried. In addition, routers send heart beat signals to the OCS, which is located in the fixed part of the network. In wireless networks with potentially low capacity, the control traffic overhead could consume a significant share of the available resources and leave only a small portion to the actual data traffic.

We measured the amount of control traffic that is created when installing rules at different rates at random nodes in the network depicted in Figure 6.7. As traffic is relayed over multiple wireless hops, it is counted each time it is transmitted. In Figure 6.5 we compare the OpenFlow control traffic rates to the control traffic created by OLSR, which provides the basic routing infrastructure. As expected, the OpenFlow control traffic increases as the rule installation rate increases, while OLSR traffic stays constant. With 20 new rules per second, the additional control traffic introduced by OpenFlow is about 20 kbit/s and the total control traffic

is about 10 times higher compared to a case where only OLSR is used. However, compared to capacity of IEEE 802.11 mesh networks, the control traffic volume is still low. Moreover, for certain scenarios such as load balancing, much lower rule installation rates can be anticipated. Compared to a pure OLSR network, OpenFlow adds some extra control traffic, but the amount is relatively small.

Scalability is a major concern when using centralized schemes such as OpenFlow. As the network size increases, more heart beat signals are generated and potentially more rules need to be installed. The results from the small test network show that the amount of control traffic generated by each mesh router is in the order of a few kbit/s. As long as the rule installation rate is low, the amount of control traffic should also stay moderate for larger networks.

6.4.3 How Fast are Rules Activated?

The rule activation time is the duration from when the first packet of a new flow arrives at a node until it is emitted again for forwarding. For new flows, the first packet is encapsulated and sent to the NOX, which then installs a rule on the OpenFlow data path. We measured the rule activation time in the small demo network depicted in Figure 6.7. We created new flow arrivals at MAP3 and then correlated packet transmission and arrival times on MAP3 and the NOX to calculate the rule activation time. The PHY rates for this and all following experiments of all links were fixed to 6 Mbit/s.

Figure 6.6 plots the rule activation time for different network loads at MAP3. The NOX processing time denotes the time to parse the packet at the NOX and create a new rule. The network delay is the time it takes for the encapsulated packet to travel to the NOX and the time to send the rule to the mesh router. The data forwarding delay denotes the node traversal delay for packets when a rule has been established.

When the background traffic load is low, the rule activation time is smaller than 10 ms. However, as the network load gets high, it takes longer time to transmit the packet to the NOX and therefore the total activation time increases. The processing time at the NOX is in the order of 1.5 ms and might be decreased by a more efficient implementation of the packet-processing application (e.g. by using C++ instead of Python, as in our case) or more powerful server hardware. The network delay for higher loads could be decreased by prioritizing channel access for control packets with IEEE 802.11e. Also, in the present setup the NOX is close to the mesh gateway and therefore the delay in the core network plays a

minor role. However, in a real deployment the NOX might be placed in a data-center which is physically and logically far from the mesh network and hence the delay in the core network can be considerably higher.

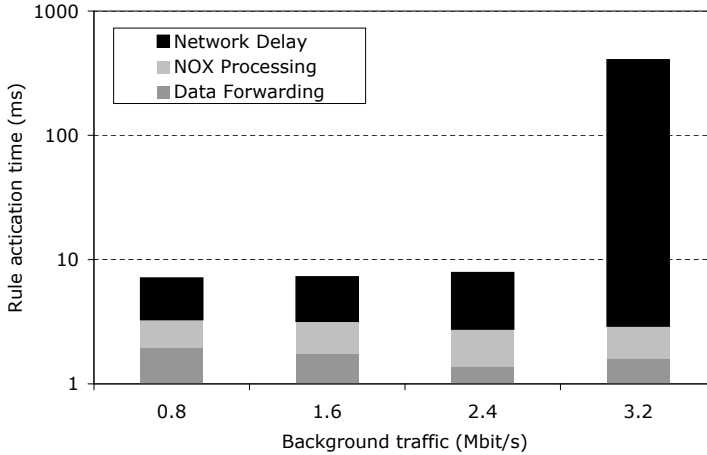


Figure 6.6: Rule activation time on MAP3 (Average of 10 experiments, Coefficient of variation < 0.001 for all values)

6.5 Optimization of STA/MAP Associations with OpenFlow

In this section we describe how the architecture enables station handovers and load balancing algorithms such as the MESHMAX algorithms described in Chapter 5. The MESHMAX algorithm requires a centralized controller to 1.) instruct stations to perform handovers, to 2.) install routes on the mesh routers and 3.) configure rate shapers.

6.5.1 Managing Station Handovers

Based on the input from the network controller, the mobility management application triggers handovers and sets forwarding rules on the mesh routers to allow stations to access the wired network via the mesh network and the gateways. The clients implement a rudimental IEEE 802.21 Media Independent Handover Function, which is capable to trigger a handover between MAPs upon receiving a *MIH Net HO Commit* message from the NOX. In addition, the NOX runs an application to respond to ARP-queries.

When a station resolves the IP address of its default gateway, the NOX will answer with the MAC address of the MAP the station is associated to.

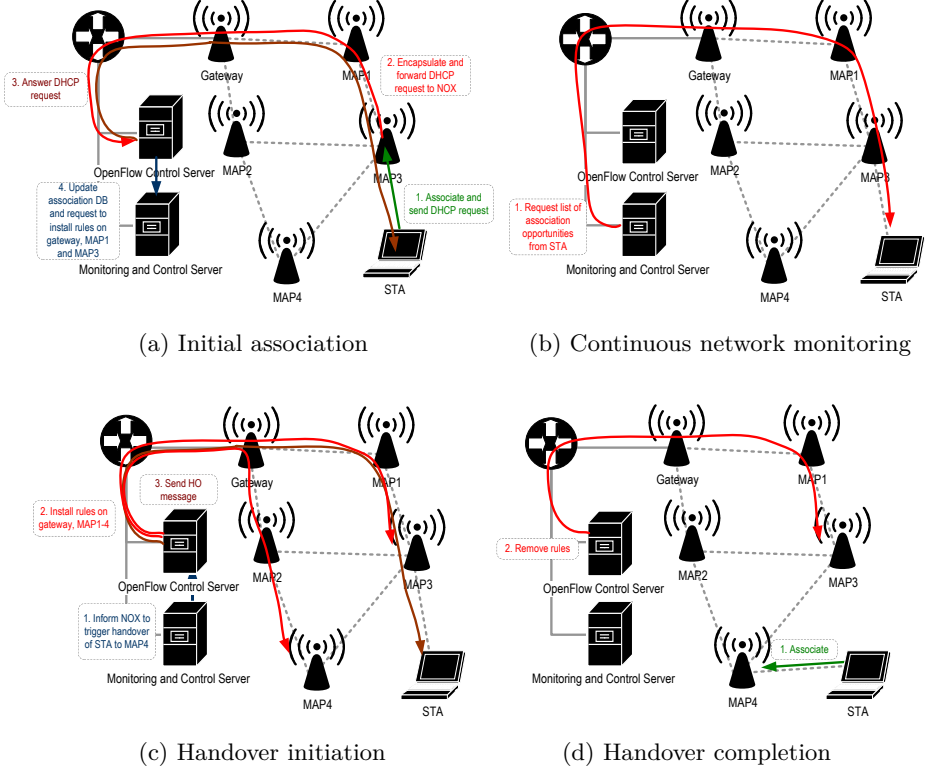


Figure 6.7: Management of station connectivity and mobility

The procedure to enable client mobility is shown in Figure 6.7 and works as follows: First (Figure 6.7a), when the station initially connects to the network, it uses the default IEEE 802.11 association procedure to connect to the close-by MAP MAP3. It then issues a DHCP-request, which is forwarded to and answered by the NOX. The NOX registers the client in the association database at the MCS. Hereafter, the control application computes routes and rate allocations with the MESHMAX algorithm and instructs the NOX to install rules on the intermediate mesh routers MAP1, MAP3 and the gateway, which enable a station to access the Internet via the mesh network. Rate allocations are enforced with rate shaping queues installed at the MAPs and the gateways.

In the second step (Figure 6.7b), the MCS continuously monitors the network. If the need arises (for example because the capacity of

a MAP3 falls below a threshold), the MCS requests a list of potential association opportunities from each STA, which STAs create by scanning its environment when no data is transmitted. Then the MESHMAX algorithm decides which STA should be associated to which AP. If for any STA a handover is required, the monitoring server proceeds to the third step.

In the third step (Figure 6.7c), the MCS informs the NOX to initiate a handover of the station from MAP3 to MAP4. The NOX sets temporary rules, which forward traffic to both MAPs. The NOX invokes the Media Independent Handover Command Function at the station to trigger a handover to the new MAP. The handover request includes MAC and SSID of the new MAP, optionally the channel of the new MAP.

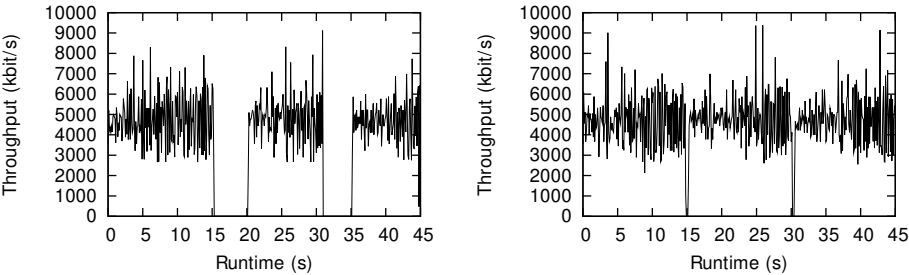
In the final phase (Figure 6.7d), the STA disassociates from MAP3 and associates to MAP4. The NOX removes the temporary rules - the traffic is now only forwarded to the STA via the MAP4. The association database is updated and the handover is completed.

6.5.2 Implementing the MESHMAX Algorithms

The input to the MESHMAX algorithms is computed from the topology, association and association opportunity database. In the topology database we store a graph of the network connectivity, the PHY rate of the wireless links and their ETX values. The PHY rate is obtained directly from the Linux wireless subsystem. The ETX values and the topology graph are provided by the OLSR instance on the control network. In order to ensure reliable transmissions, wireless links with high ETX values are not considered in the connectivity graph.

For licensing reasons, our testbed implementation solves the MILPs with LPSolve [44] instead of CPLEX, as done with Chapter 5. LPSolve is available under the LGPL licensing scheme, whereas CPLEX is a commercial product. LPSolve is typically slower than CPLEX. However, given the relatively small problem size in our testbed experiments, no performance problems could be observed.

A set of Perl scripts converts the topology, the association and the association opportunity database into an LPSolve compatible input format. The MESHMAX algorithms compute the optimal MAP for each connected station, a path between the station and the Internet and what rate can be used. For each station, which is not connected to its optimal MAP, the script calls the OCS to trigger a handover to the optimal MAP. Furthermore, the script requests the OCS to install forwarding rules on the mesh routers, so that traffic between each station and the Internet is sent via the computed path. Finally, the script instructs the OCS to configure the rate shaper



(a) Handover-message does not include the channel of the new MAP (b) Handover-message includes channel of the new MAP

Figure 6.8: TCP throughput (averaged over 100 ms windows) from the core-network to the station during two handovers (after 15 and 30 sec.)

according to the computed rates.

6.5.3 What is the Cost of a Handover?

To better understand what the costs of a handover in a real network are, we used iperf [7] and measured the TCP throughput from a station to the wired network during handovers from MAP3 to MAP4 and back. Figure 6.8 shows the throughput during a handover. When the handover message does not contain any information about the channel of the new MAP, during a period of almost 5 seconds the TCP throughput drops to 0 (Figure 6.8a). During this time the STA scans the network to find the new MAP. However, when the handover message is augmented with the channel of the new MAP, this period reduces to roughly 200 ms. In this case, the STA does not need to scan for the new AP and therefore can perform the handover much faster. As the delay bandwidth products on both paths are small (RTTs ~ 30 ms), TCP recovers fast after the handover. This shows, that if the handover message contains all required information to connect to the new AP (SSID and channel), the handover cost is relatively low.

Duration (seconds)	Without channel information	With channel information
Average	1.87	0.21
Minimum	0.06	0.05
Maximum	2.65	0.27

Table 6.1: Outage duration during a handover

In the second test, the outage duration during the handover was analyzed closer. Here, a node in the wired network transmitted a UDP datagram every 10 ms to the STA. Table 6.1 shows the minimum, maximum and average outage duration (i.e. packet inter-arrival time after handover - 10 ms). The results also show that providing channel information in the handover message to the STA is key to decrease the outage duration. However, even with channel information the outage is on average larger than 200 ms. This might lead to severe quality degradations in real-time services such as Voice over IP. The 200 ms are a sum of the dissociation time, the channel switching latency of the wireless NIC and the association time at the new MAP.

During the handover, the OCS and intermediate mesh routers are not aware if the station is still associated to the old MAP or already associated to the new one. Hence, the OCS can install rules to forward traffic to both MAPs. Using this temporary packet duplication, the number of packets lost during the handover can be slightly reduced from 25 to 18. Yet, the biggest optimization potential lies in a faster handover procedure, which we will introduce in Chapters 8 and 9.

6.5.4 What are the Performance Gains due to the MESH-MAX Algorithms?

In the MultiRAT testbed, we verified that the MESHMAX algorithms in combination with our architecture can deliver better performance than a standard mesh network. The network topology is shown in Figure 6.9 and consists of 7 mesh nodes, of which 3 are MGWs, 2 are MAPs and two are pure MRs, which do not provide any AP services and just forward data. The mesh nodes use multiple radios and orthogonal channels on the respective links. The MGWs are connected to a corresponding host (not shown in Figure 6.9) that represents the Internet using wired links of 3 Mbit/s and 6 Mbit/s speed (to simulate an ADSL connected mesh network). In the testbed area 10 stations were deployed that download data from the corresponding host using iperf with TCP. Some stations have multiple association opportunities (e.g. station 5 can connect to MAP E and F), while station 10 can only connect to MAP F.

The evaluation compares three different schemes:

- RSSI-based: the station connects to the MAP with the highest RSSI and paths are computed using minimum hop-count routing.
- Hop-count: the station connects to the AP which is closest to the next MGW (in terms of wireless hops). Paths are also computed

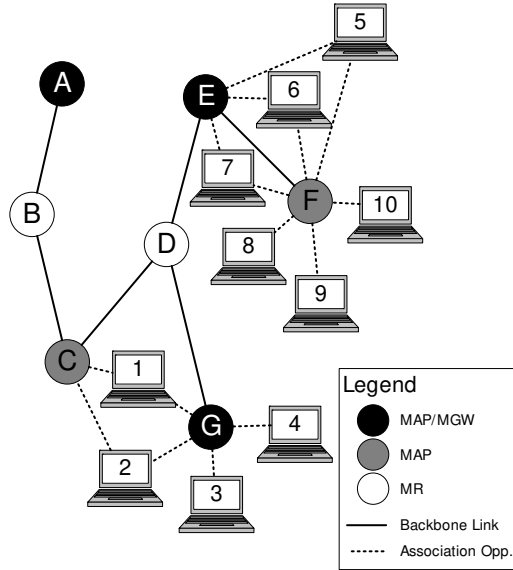


Figure 6.9: Testbed setup to evaluate the MESHMAX algorithm

using minimum hop-count routing.

- MESHMAX: The MESHMAX scheme uses the associations, routes and rates computed by MESHMAX-FAST*.

In the first two schemes, the rates are not explicitly limited by rate shapers.

Figures 6.10 and 6.11 show the distribution of the TCP throughput with 3 and 6 Mbit/s gateway capacity respectively. For both scenarios, the network configuration computed with the MESHMAX algorithms provides higher median and mean throughput than the hop-count and RSSI-based scheme. This improvement is mainly due to the better distribution of traffic to the gateways and APs. MESHMAX avoids to concentrate traffic on one gateway or AP, while with the two other schemes more stations share the resources of one AP and gateway. For example, with the hop-count and the RSSI-based scheme, all stations either use gateways “E” and “G”, but not gateway “A”. In contrast, HOTMESH routes the traffic of three stations via gateway “A” and thereby balances the load better (at the cost of longer paths for some stations).

The achievable TCP throughput is close to rates that MESHMAX computed and configured in the rate shapers. However, in the 6 Mbit/s scenario, two stations receive a much lower throughput than their allocated rate (191 and 562 kbit/s instead of 1800 kbit/s). When performing the same

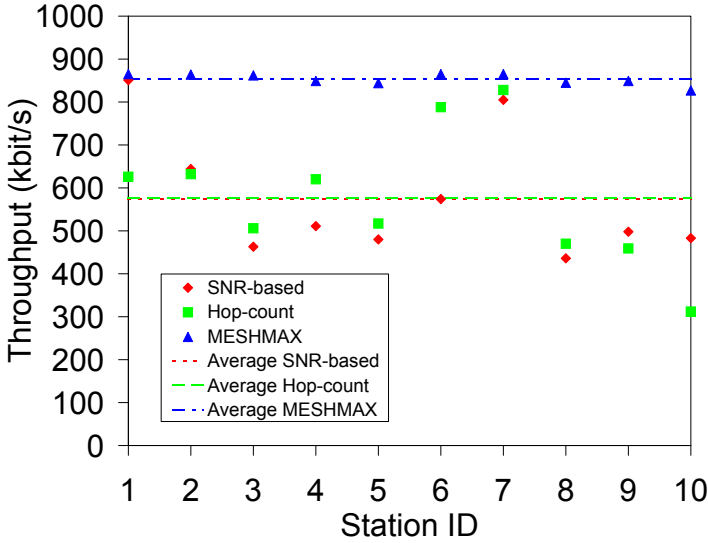


Figure 6.10: Throughput with RSSI-based association, hop-count based association and the optimal association computed with MESHMAX and 3 Mbit/s gateway capacity

experiment with UDP instead of TCP, all stations received the throughput that was computed by MESHMAX (within a small error bound). This shows, that the network in principle is capable of delivering the rates that are computed with MESHMAX, but TCP is sometimes not able to full utilize the available capacity.

6.6 Conclusions

In this chapter we have proposed an OpenFlow-based architecture which allows flexible control of packet routing in WMNs. The architecture combines the benefits of OpenFlow (flexible packet forwarding) and WMNs (self-configuration and error-resilience). The micro-benchmarks show that it is possible to use OpenFlow on low power mesh devices. However, a too large rule table may penalize performance. Our architecture can be used to implement the MESHMAX algorithms in a real network. The evaluation shows that the intelligent association management and routing with the MESHMAX algorithms translate into a higher performance in real networks.

Due to limitations in testbed size and availability, larger tests to evaluate

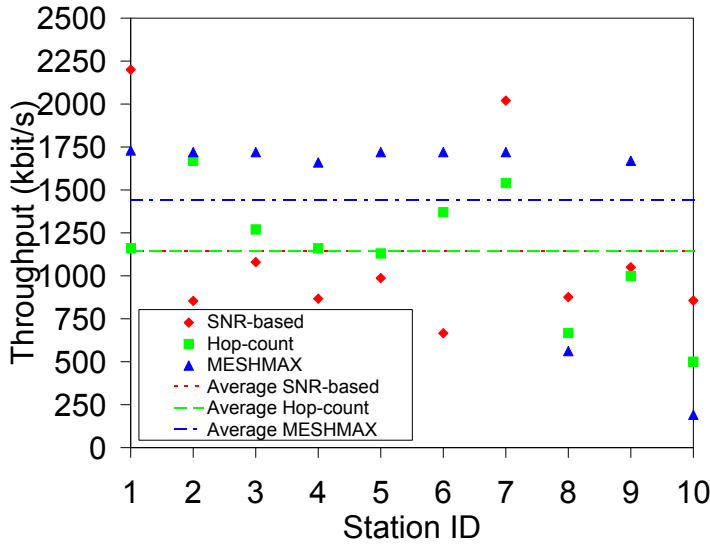


Figure 6.11: Throughput with RSSI-based association, hop-count based association and the optimal association computed with MESHMAX and 6 Mbit/s gateway capacity

the scalability of the architecture could not be performed. In Chapter 5 we have shown through simulations that the MESHMAX algorithms are scalable. As simulation results do not necessarily reflect the behavior observed in real networks, a large scale testbed evaluation is desirable in the future. Such large scale tests could also cast light on the role of signaling overhead in large networks. The experiments have also shown that TCP does not always fully utilize the available capacity. The role of TCP congestion control in fully utilizing the available capacity in multi-hop wireless networks is outside the scope of this thesis, but should be investigated further in future works.

The proposed architecture is a first step in controlling wireless networks with OpenFlow. However, it does not take full advantage of the potential of OpenFlow and SDNs to steer specifics of the wireless transmission. Hence, we continue in the next section to investigate how a deeper integration of OpenFlow with the control of the wireless transmissions can improve the management of wireless networks.

Distributed MAC for Software Defined WLANs

7.1 Introduction

A dense deployment of APs is required to fully utilize the potential of future WLANs. Traditionally, WLAN APs were devices with low processing power and little intelligence. However, this design paradigm has gradually been abandoned and APs are getting more powerful and are equipped with more functionality.

There are multiple reasons for increasingly powerful APs: First, the PHY layer rates of WLANs have increased considerably over the past years, reaching several gigabits per second with the new IEEE 802.11ac standard. Such high speed PHYs need powerful CPUs on the APs in order to support fast frame processing, which is largely done in software. Second, new functional requirements such as Quality of Service and mobility management need software support on the AP. A larger software stack on the AP requires more CPU power and memory to execute the software. For example, the number of software packages included in the current release (version July 2012) of the popular open source AP firmware OpenWRT increased by more than 800% over the past five years [13]. The operating system image size of the widespread Cisco 500 series APs increased by more than 150% from 2007 to 2011 [3].

Typically, a dense network deployment also leads to high *Operational Expenditures* (OPEX) and *CAPital Expenditures* (CAPEX). To reduce the OPEX and CAPEX, network virtualization and infrastructure sharing are essential. The main rationale behind network virtualization is that several virtual networks, by potentially different network operators, are provisioned

on top of one physical network. This allows to utilize the available hardware better, but requires isolation between the virtual networks.

To address the aforementioned issues, industry and standardization bodies have introduced new management protocols, which partially offload functionality to dedicated control servers. For example, CAPWAP [82] allows to offload authentication services partially to external control servers. However, the functionality of those protocols is limited and extensions are difficult to implement. Furthermore, isolation of virtual networks is hard to achieve with those protocols. It is thus desirable to have a new WLAN management architecture, which 1.) enables the deployment of network applications in a vendor independent way 2.) allows to offload non-time-critical processing to external servers, 3.) is scalable and can process MAC layer traffic at high rates 4.) supports virtual networks.

As already discussed in previous chapters, SDNs in principle can fulfill those requirements. However, the architecture proposed in the previous chapter is limited in the amount of control that an SDN application can exercise over the wireless transmission. Furthermore, in this architecture MAC frames are still processed at the APs or MAPs and virtualization of MAC processing is not considered.

7.1.1 Related Work

Attempts to create a new SDN-based management architecture for WLANs can be found in [126], [181] and [163]. [126] and [181] propose to use OpenFlow to monitor traffic flows and to let end-users control traffic flows via a GUI. [181] explicitly addresses network virtualization and suggests to slice networks based on IP layer and application characteristics. For example, a video streaming service could create its own network slice with specific QoS settings. [181] does not allow to control the IEEE 802.11 MAC layer. Cell-SDN [112] is recent proposal to use SDNs for manage the control path in cellular networks. The key idea is to decouple the control plane from the radio hardware and to use controllers in remote data centers to for example perform radio resource management or mobility management.

With Odin [163], the physical APs run software agents that provide authentication services and generate beacon frames. Odin allows to initiate handovers by migrating the state of such an agent between APs using a self-defined protocol. OpenFlow is used to update forwarding tables in the network switches. Similar to CAPWAP, association states in Odin are kept both on a central server and on the AP itself. Within Odin, MAC frames are processed at the AP and OpenFlow cannot control the MAC.

In [88] and [50] virtual APs are created by using an OS hypervisor on the APs. Such full AP virtualization increases flexibility, but requires a powerful x86-based AP, which is not typical for normal WLAN deployments. Furthermore, [88] and [50] do not have a central control platform, as it is provided by the OpenFlow approaches discussed above.

CAPWAP [82] and its predecessor LWAPP [52] are IETF standards for splitting MAC processing in IEEE 802.11 WLANs. CAPWAP uses a centralized controller to discover APs, configure them and to provide authentication and authorization services to stations. CAPWAP implements a split MAC, in which some MAC frames are generated by the AP controller and others by the AP. However, CAPWAP does not have a standardized platform and API to deploy new network applications. Moreover, CAPWAP requires still relatively complex MAC processing at the AP.

7.1.2 Problem Statement and Contributions

While all of the surveyed work addresses one or more of the requirements outlined above, none fulfills all four requirements. In this chapter, we introduce CloudMAC, a new management architecture for WLANs. The key idea of CloudMAC is to split up a WLAN AP into a physical AP, which just forwards MAC frames, and a virtual AP, which is hosted in a virtual machine in a data center or the cloud (thus the name CloudMAC) and contains all functionality such as MAC frame generation and authentication services. The virtual and the physical APs are connected via an OpenFlow [165] enabled network. The OpenFlow switches can manipulate control header information, which is attached to frames sent from the virtual to the physical AP, according to the flow table contained in the switch. This control header information allows to specify important aspects of the wireless transmission, such as the coding scheme or the transmission power. The flow table is programmed by external applications using the OpenFlow protocol. Thereby, CloudMAC allows to offload processing from the APs to powerful data centers, to leverage the fast packet processing in hardware switches and to deploy new applications in an open and vendor independent way. Furthermore, with CloudMAC the network operator can trigger seamless handovers of stations, without any active cooperation of the station.

The *key contributions* presented in this chapter are:

- The **CloudMAC architecture** and its implementation in a testbed. We specify the components of a CloudMAC WLAN and how they

interact with each other. Furthermore, we describe how the architecture can be implemented using standard SDN components and APs.

- The **evaluation of the architecture through micro benchmarks** and the discussion of potential applications enabled by CloudMAC. The micro benchmarks show that the distributed data processing in CloudMAC achieves similar performance as normal WLAN APs. However, CloudMAC is more flexible when deploying new applications when compared to standard WLANs.
- The **evaluation of CloudMAC initiated handovers**. The centralized storage of the association state allows CloudMAC to trigger handovers simply by reconfiguring OpenFlow rules. The evaluation shows that such handovers are more seamless than a normal break and re-connect handover in a standard WLAN.

The remainder of this chapter is structured as follows: In Sections 7.2 and 7.3 we introduce the architecture of the CloudMAC system and discuss implementation issues. Section 7.4 presents micro benchmarks performed in the testbed implementation. In Section 7.5 we describe how CloudMAC can enable station handovers. Section 7.6 presents other potential applications and benefits of CloudMAC. In Section 7.7 we summarize the chapter and discusses limitations of CloudMAC.

7.2 Architecture

7.2.1 Overview

CloudMAC is a distributed architecture in which 802.11 WLAN MAC processing is partially performed in data centers on virtual machines connected by OpenFlow. This is in contrast to traditional approaches, in which the complete MAC layer is located at the local AP. Our approach simplifies the management of WLAN deployments and allows a rapid development of new functionality using software modifications. A CloudMAC based WLAN deployment consists of Virtual APs (VAPs), Wireless Termination Points (WTPs), an OpenFlow switch, an OpenFlow controller and tunnels to connect the entities (Figure 7.1).

VAPs are operating system (OS) instances running on a virtualization host, such as Xen or VSphere Center. Each VAP has one or several virtual WLAN cards. A virtual WLAN card is a driver that appears to the OS and user space applications like a normal physical WLAN card.

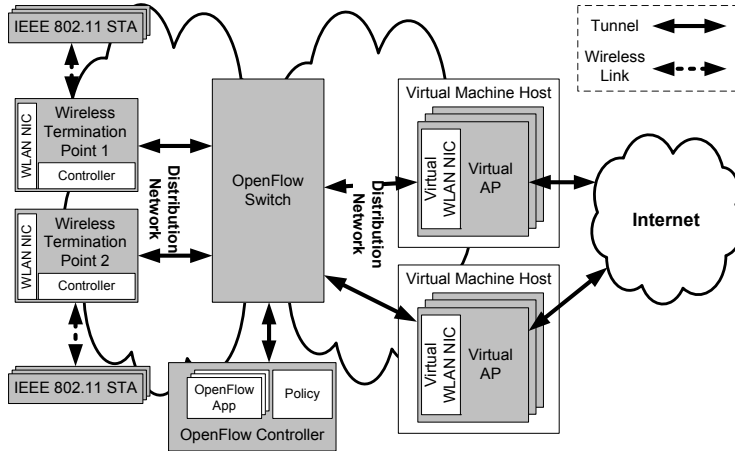


Figure 7.1: Architecture of a CloudMAC based WLAN

Standard WLAN management tools can be used to set parameters of virtual WLAN cards. VAPs run access point management software, for example to generate beacon frames, or respond to Association/Authentication MAC frames. As the virtual WLAN card appears like a real card, standard software such as `hostapd` [9] can be used. This allows an increased flexibility through CloudMAC, while at the same time using well known standard software. One VAP can comprise many virtual WLAN cards, which are connected to physical cards on different WTPs. In the extreme case, one enterprise WLAN is only one VAP or there is one dedicated VAP for each station.

WTPs are slim APs equipped with one or several physical WLAN NICs. A WTP can transmit and receive MAC frames and forwards them to the VAP for further processing. In addition, WTPs contain a control daemon, which allows the OpenFlow controller and the VAP to set configuration parameters of the physical WLAN NIC, such as the operating channel.

7.2.2 Data Frame Processing

Figure 7.2 illustrates how MAC frames are processed in CloudMAC. Here, the user station transmits a frame to the VAP. The user station first transmits a data frame to the WLAN NIC of the WTP (1) using the IEEE 802.11 PHY layer standard. To comply with the IEEE 802.11 protocol, the WLAN NIC has to almost immediately (typically within a SIFS) answer with an ACK frame upon the successful reception of a frame (2). The WLAN NIC then forwards the MAC data frame to the tunnel endpoint (3),

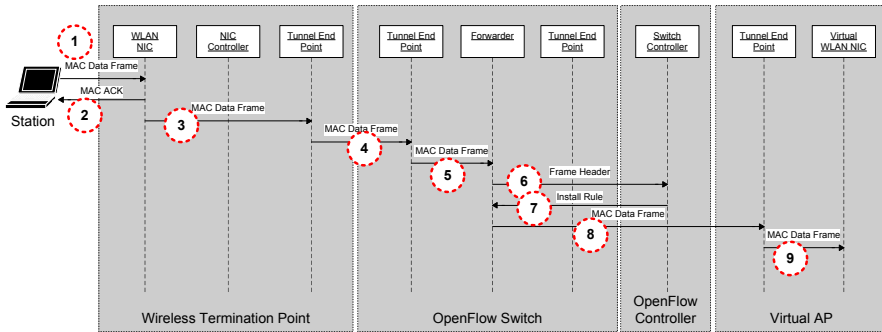


Figure 7.2: Processing of MAC frames with CloudMAC

which encapsulates the frame and transmits it via the distribution network to the OpenFlow switch (4). The tunnel is required, since the frame format (Radiotap + IEEE 802.11) does not comply with a standard Ethernet frame format and normal intermediary switches might fail to forward such frames. The frame is decapsulated (i.e. tunnel protocol headers are removed) at the tunnel endpoint of the OpenFlow switch (5) and sent to a forwarding unit. The forwarding unit searches for appropriate rules in the forwarding table. If the forwarding table has a rule for handling the frame, this rule is executed. If not, the frame is sent to the OpenFlow controller (6). The OpenFlow controller inspects the frame header and decides on which tunnel endpoint the frame needs to be sent. It is evident here, that this decision creates the binding between the virtual NIC at the VAP and the WLAN NIC at the WTP. A rule is then installed on the forwarder (7) and the packet is sent to the VAP (8) via a tunnel. For subsequent frames with the same VAP/WTP binding, an appropriate rule is now established and the OpenFlow controller does not need to be consulted in future. The VAP decapsulates the frame and hands it over to the virtual WLAN NIC (9). The virtual WLAN NIC calls the respective operating system routines to process the frame further (e.g. to send it to a user space application etc.).

Sending a frame from the VAP to a user station follows the same sequence as above, just in reverse order. The VAP generates a MAC frame and sends it via tunnels and the OpenFlow switch to the WTP. The WTP then transmits this frame wirelessly to the user station using its WLAN card. The VAP can be connected to the Internet just like a normal AP and stations can connect to the Internet by using the VAP as bridge or router.

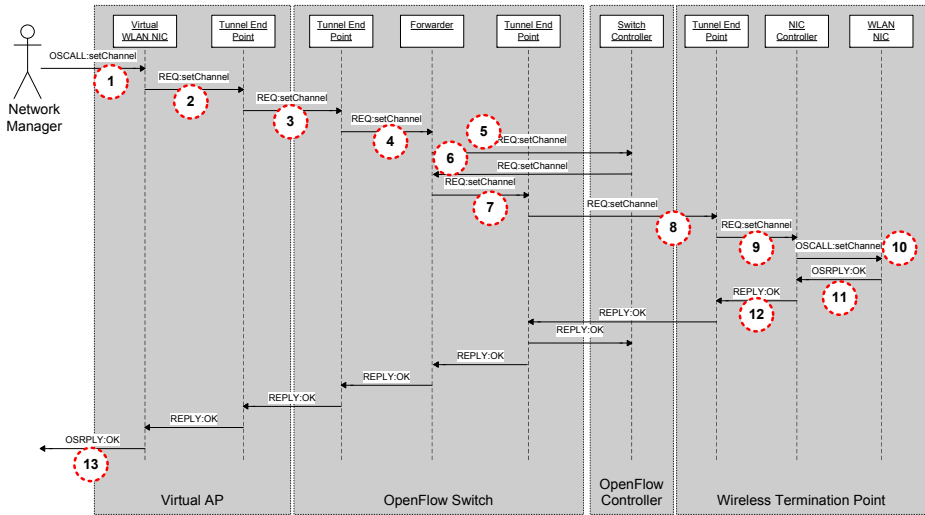


Figure 7.3: Processing of control commands with CloudMAC

The process of forwarding frames between the WTP and the VAP is oblivious to encryption that might be applied on the link layer, such as WEP (Wired Equivalent Privacy) or IEEE 802.11i. With those encryption standards, frame payloads are encrypted and decrypted by the user station and the VAP. The forwarding decisions taken by the OpenFlow switch are based on unencrypted header information, such as source and destination addresses.

7.2.3 Control Command Processing

Besides forwarding MAC data, control and management frames, CloudMAC allows fine grained control over configuration commands. Those commands are used to configure the WLAN card and typically issued by a user space application in a VAP. For example, if a user space application requests the virtual WLAN card to change its channel, the request is intercepted in the virtual WLAN card driver.

Figure 7.3 depicts in more detail how control commands are processed. A control command request is issued by a network management application or the network administrator using standard operating system tools (1). Control command requests for example specify the channel or the transmission power of the AP. The request is sent to the virtual network WLAN NIC driver via standard operating system routines (for example I/O control

calls or Netlink messages). The WLAN NIC driver receives the message (e.g. an OSCALL setChannel to set the channel of a card). The driver then encapsulates the message into a control frame (REQ:setChannel) (2) and forwards it to the tunnel endpoint. The tunnel endpoint encapsulates the control frame into an IP packet (3) and sends it via the distribution network to the OpenFlow switch (4). At the tunnel end point of the OpenFlow switch the message is de-capsulated and forwarded to the OpenFlow controller (5). The OpenFlow controller determines, based on its policy, if the command specified in the control message is permitted. If it is permitted, the command is returned to the switch (6) which then forwards it via a tunnel (7 and 8) to the appropriate control daemon on the WTP (9). The controller subsequently translates the message to an operating system call and sends this operating system call to the WLAN NIC driver (10). The WLAN NIC driver executes the command (e.g. setChannel) and returns the result to the controller (11). The result is then encapsulated into a control frame, which is returned back to the virtual WLAN NIC (12). The virtual WLAN NIC updates its status (sets the “virtual” channel) and returns an OK answer to the network manager via an OS call. If several virtual WLAN NICs are connected to the respective physical WLAN NIC, the OpenFlow switch command informs all connected cards about the change. If at step (5) the OpenFlow controller determines that the command is not permitted, it sends an error message back to the VAP.

7.3 Implementation

We have implemented a prototype of CloudMAC on the KAUMesh testbed described in Chapter 6. The WTPs are the Cambria GW2358-4 WLAN nodes. The WTPs use OpenWRT Backfire and ATH5K as WLAN driver. The WLAN card is in monitor mode, which allows to transmit and receive raw MAC frames. We extended the driver to transmit frames at the PHY rate specified in the radio-tap header generated by the VAP. The WTPs utilize the multi-BSSID feature provided by Atheros chipsets and many other WLAN cards: the card includes a hardware register that controls which MAC addresses are used by the card and hence which MAC frames are acknowledged. The register can be configured from the OpenFlow controller via a control daemon that is part of the WTP.

The VAPs are Debian 6.0 VMs on a VSphere Center installation. The VAPs use `hostapd` [9] (version 0.6) as Station Management Entity (SME). The SME generates beacon message and provides authentication and authorization services. The virtual WLAN card driver is based on

the `mac80211_hwsim` driver [9], which is a software simulator of an IEEE 802.11 device used for testing MAC functionality and user space tools such as `hostapd`. We modified `mac80211_hwsim` in order to allow reading and injecting RAW IEEE 802.11 frames.

Packets between the VAP, the switch and the WTP are tunneled using the Capsulator tool from OpenFlow project [63]. As we have no hardware OpenFlow switch to our disposal, we use OpenVSwitch 1.3.0 instead. OpenVSwitch [135] is an OpenFlow-compatible software switch in the Linux kernel. The switch runs in a VM on the same VSphere installation and is controlled by custom made OpenFlow switch controller implemented in Python. To bind together a WTP and a VAP, the switch controller configures a rule on the switch, so that traffic is simply forwarded between the tunnels.

7.4 Performance Evaluation

The split MAC processing done by CloudMAC could lead to performance degradations, because of the additional frame processing steps involved in CloudMAC. We hence compare the performance of CloudMAC with a normal WLAN AP (using the same hardware) that is connected to an IP router via FastEthernet.

Using ping and iperf we measured the Round Trip Time (RTT) and the TCP throughput between a station and the VAP/IP router. Figures 7.4 and 7.5 show the Empirical Cumulative Distribution Function (ECDF) of the RTT and the throughput for different TCP segment sizes. When CloudMAC is used, the RTT (min/median/max) increases from (1.30/1.79/12.17) ms for a standard WLAN system to (1.60/2.28/14.62) ms. This is due to additional processing at the OpenFlow switch and due to delay added by the tunnels. However, our experiments showed that time critical MAC frames like association response messages are delivered fast enough to allow standard clients (we tested with Windows XP, Linux and MacOS X) to associate to the CloudMAC VAP. We verified that the time for the whole association procedure was similar on both systems (around 1.44 sec). While CloudMAC shows small additional latencies due to the tunneling/OpenFlow overhead, the processing time for such control frames is significantly reduced due to the more powerful processing at the VAP. In our testbed, the VAP and the WTPs are on the same LAN and hence the RTTs are small. However, one might envision a setup, in which VAPs are located in some remote data centers or connected via a mesh network. In such a setup the RTTs could be too large though, as some MAC frames

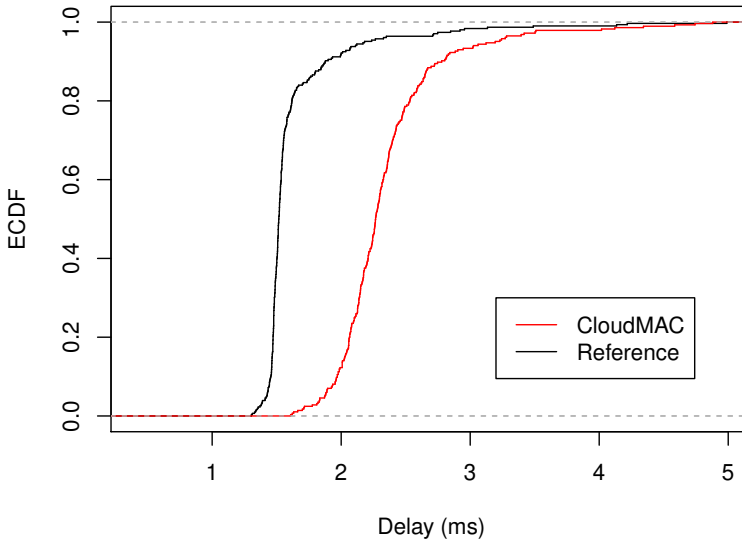


Figure 7.4: Ping RTT for CloudMAC and the reference network. CloudMAC has a higher RTT due to the additional frame processing.

like association response and probe response messages need to be delivered to stations within a few milliseconds. In the current Linux kernel (3.2.20), authentication messages need to be answered within 200 ms and probe messages within 500 ms. In addition, the power saving mechanism in IEEE 802.11 relies on a relatively precisely timed transmission of beacon message, which can contain the *Delivery Traffic Indication Map* (DTIM). To allow stations to schedule wake-up periods when beacons are sent, the inter-arrival time of beacon messages should have a low jitter.

Similarly, the TCP throughput is decreased slightly due to the additional components added by CloudMAC. For large TCP segments the performance decreases by approximately 8.5%. This performance decrease is due to the tunnels, which in our current implementation run in user space and therefore require context switching. In future work we plan to investigate the possibility of using kernel-space tunnels, which should result in improved performance.

7.5 CloudMAC Handovers

In CloudMAC, the association and authentication state is stored in the VAP only. Therefore, initiating a handover from one WTP to another

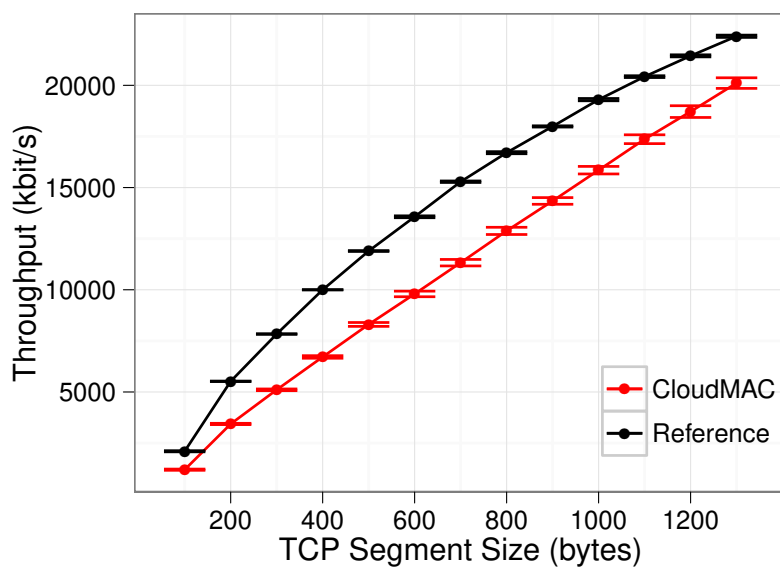


Figure 7.5: TCP throughput for different segment sizes. The throughput increases with the segment size (due to the lower overhead on the wireless transmission). With larger segment sizes the relative differences between CloudMAC and the reference network get smaller.

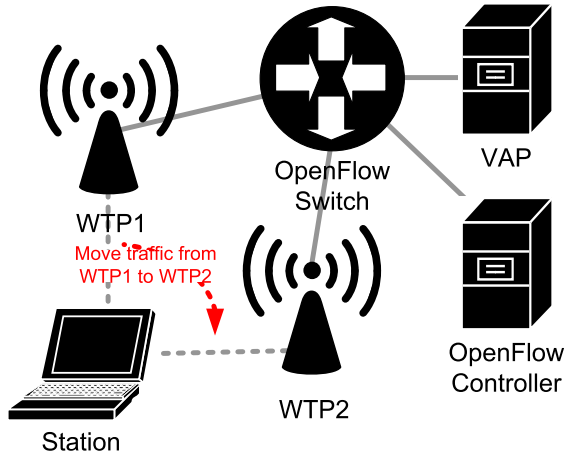


Figure 7.6: Scenario to test CloudMAC’s ability to enable handovers of standard IEEE 802.11 stations. First all traffic between the VAP and the station is sent via WTP1 and then moved to WTP2.

is relatively simple: by changing the forwarding rules in the OpenFlow switch, traffic can be moved from one WTP to another. Since a station stays associated to the same VAP, such a handover does not require any cooperation by the station. If traffic should be balanced on a per station basis, each station needs its own VAP.

Such handovers can be utilized in different scenarios. For example, the load balancing and mobility management optimizations discussed in the previous chapters can be implemented using CloudMAC handovers. As no cooperation by the station is required, CloudMAC handovers work with standard IEEE 802.11 clients. Besides load balancing and mobility management, handovers can also be used for the energy efficient operation of WLANs. The network operator can save energy by temporarily switching off some APs/WTPs that cover overlapping areas and have low load. An energy optimizer can thus decide to switch off a lightly loaded WTP and trigger a handover of all associated STA to neighboring WTPs.

7.5.1 Evaluation

Figure 7.6 depicts the test scenario, which in the CloudMAC case consists of two WTPs, one VAP and one standard IEEE 802.11 STA. The reference network consists of two regular APs (AP1 and AP2) that are connected to an external server via an Ethernet switch. The STA runs Linux and uses `wpa_supplicant` (version 0.8) [9] to find networks and associate to them.

In the first test, the station connects to the network via WTP1 and the VAP/external server sends 1000 UDP datagrams per second to the STA. After a few seconds, the forwarding rules in the OpenFlow switch are changed and the register of MAC addresses is updated so that all traffic from WTP1 is moved to WTP2. Then WTP1 is switched off. In contrast, in the reference network the STA connects to AP1. After a few seconds `hostapd` sends a disassociation message to the STA and `hostapd` is terminated. `wpa_supplicant` searches for new APs and eventually connects to AP2, which then allows receiving UDP datagrams again. Figure 7.7 shows the ECDF of the number of lost packets during the AP switch (20 trials). With CloudMAC on average 3.5 packets are lost (min: 2, max: 104). Those packets are in the transmit queue of the WTP during the AP switch and are lost during the switch. With the reference network, the AP/STA association is broken and the STA needs to find a new AP and connect to it. This results in much larger number of lost packets. On average 10780 packets are lost. Figure 7.7 also shows that the distribution of lost packets is bi-modal: In about 50% of the tests approximately 3200 packets were lost, while in the other 50% of the tests approximately 17400 packets were lost. `wpa_supplicant` sometimes uses a cached list of available APs to select the next AP to associate to. Sometimes however, `wpa_supplicant` initiates a new scan for AP, which takes much longer time and results in the high number of lost packets.

In addition to finding and associating to a new AP, the dynamics of the transport layer can even cause a larger disruption to the user. We repeated the two tests with TCP downloads and show the corresponding time/sequence graphs in Figure 7.5. With CloudMAC, only a very small number of TCP segments gets lost during the switch and hence the time/sequence graph is smooth and steady. In the reference network though, there is no increase in sequence numbers (i.e. no throughput) between second 3 and 30. During that time, the link layer connection needs to be re-established and the TCP send rate recovers.

Our tests show that with CloudMAC the network controller can initiate seamless handovers, while with a normal AP architecture this is not possible. We remark that with CloudMAC and standard IEEE 802.11 STAs such a switch is even possible if the WTPs are operated on different channels. The IEEE 802.11h standard, which is mandatory for IEEE 802.11a/n devices, includes a Channel Switch Announcement (CSA) frame, in which the AP informs the STA to switch its channel. If WTP1 and WTP2 are using different channels, the OpenFlow controller just needs to generate such a CSA to request all STAs to switch channels before moving traffic from WTP1 to WTP2.

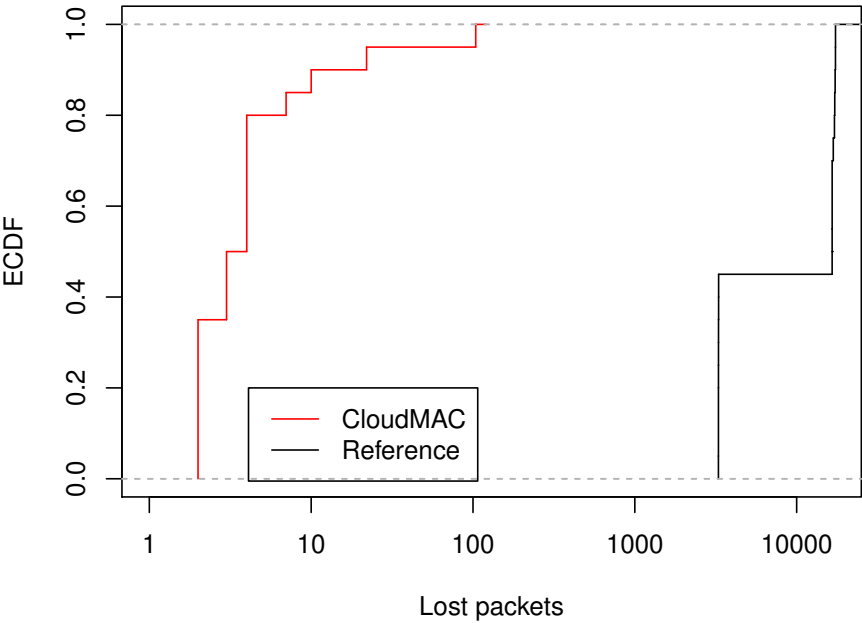


Figure 7.7: ECDF of number of lost packets during an AP switch. CloudMAC significantly reduces the number of lost packets (send rate: 1000 packets/second).

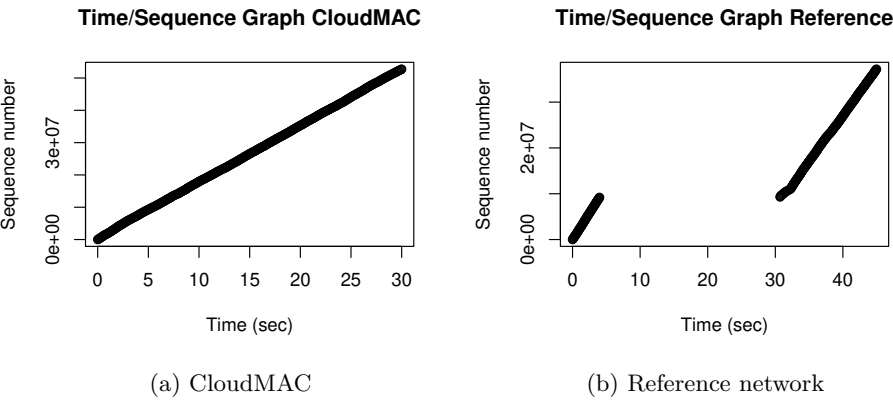


Figure 7.8: TCP time/sequence diagram. With CloudMAC no disruption is visible, while with the reference network the throughput is 0 for 27 seconds.

7.6 Other Potential Applications and Benefits

Besides handovers, CloudMAC enables a range of new applications, such as:

Dynamic Spectrum Use: In CloudMAC, one virtual WLAN card can be connected to several physical WLAN cards. This enables a scenario, where one WTP contains several physical cards, operating with the same MAC address, but on different channels. The physical cards periodically monitor channel utilization and provide this information to the network controller. If a station is currently using a channel with high external interference, the OpenFlow controller sends an IEEE 802.11h Channel Switch Announcement message to the station to instruct it to switch to a less used channel. The station does not experience any interruption, as it can continue to communicate on the new channel with another physical card of the same WTP. Since the station is associated with the VAP (and not the WTP), no re-association is required. This procedure does not require any modification on the client, as long as it supports IEEE 802.11h (mandated by IEEE 802.11a/n).

On-Demand AP: In today's virtualized WLANs, one AP might broadcast the SSID for dozens of networks. As each SSID requires one beacon frame, an AP might broadcast hundreds of beacon frames per second thereby reducing available capacity for data transfers. CloudMAC enables a scenario, where the OpenFlow switch per default does not forward beacon frames from a VAP to the connected WTP. When a new user arrives and sends a probe request (that sometimes includes the SSID of the desired network), an application on the OpenFlow controller inspects the probe request and dynamically enables the beacons. The new user now receives the beacon and can connect to the network. If the probe request does not include the SSID, historical usage data and the users MAC address can be used to identify the desired network. Thereby, the number of beacon messages on the wireless medium can be reduced.

Downlink Scheduling: All traffic between the VAPs and the WTPs passes through an OpenFlow switch. The switch hence can be used for downlink scheduling either by simple rate shaping as provided by OpenFlow or by time division. For time division scheduling, the OpenFlow controller instructs the switch to only forward the packet of one WTP, while putting the packets of interfering WTPs in a queue on the switch. After one time slot, the switch rules are changed, so that packets of another WTP are released from the queue and forwarded. A system similar to CENTAUR [156] could be implemented.

7.7 Conclusions

In this chapter we have presented CloudMAC, a new WLAN management architecture, in which most processing and management functionality is concentrated on VAPs that can be provided via existing cloud infrastructure. The physical APs are simple devices that relay MAC frames between VAPs and mobile stations via an OpenFlow controlled network. Such OpenFlow-based architecture provides many benefits for rapidly creating new services. For example, CloudMAC allows network initiated handovers and requires no modifications on the stations.

The main goal of this chapter was to introduce the CloudMAC architecture and to show how it can be used to control station associations. Several other interesting aspects, which have not been discussed, remain to be investigated further. For example, the evaluation of the prototype implementation showed that in some situations CloudMAC is slightly slower than traditional WLANs. By replacing the user-space tunnels with more efficient kernel-space counterparts, CloudMAC's performance should closely match traditional WLANs. Furthermore, in this chapter scalability issues have not been addressed. Future work therefore should evaluate the scalability of CloudMAC, for example by determining how many VAPs or stations can be supported on typical hardware. Moreover, a precise characterization of the requirements on the backhaul network between the WTPs and VAPs could answer the question if VAPs can be provided by public cloud service providers on the Internet, such as the Amazon EC2.

As those aspects do not directly contribute to the research questions of this thesis, we continue in the next chapter with investigating how OpenFlow helps the station to better estimate the quality of a wireless link and how it can increase the speed of handovers.

Accurate Estimation of Link Quality and Fast Handovers

8.1 Introduction

In the previous chapters we have discussed optimization models which allow to find the best STA/AP associations and systems that enable to control the network from a central controller. To compute the optimal association pattern, the central controller needs to know the link quality of each station to its surrounding APs. Standard IEEE 802.11 WLANs do not provide such information. IEEE 802.11k [22] is an attempt to exchange management information between APs and STAs. The standard allows APs and STAs to get a more comprehensive view about the network. While IEEE 802.11k provides messaging primitives to exchange such information, the key question to be answered is still which metric characterizes link quality in a good way.

Traditionally, the Received Signal Strength Indicator (RSSI) or the Signal to Noise Ratio (SNR) are used as quality metrics. The rationale behind using the RSSI is that a high RSSI should enable high data rates and low packet loss and thus satisfied users. However, in practice the RSSI is not always a good metric: first, the RSSI does not reflect the load of the AP and channel. Even if the RSSI of one AP is high, the load of this AP or channel could be high as well and hence the user would experience low performance. Second, it is well understood by practitioners that there is no clear relationship between the RSSI and packet loss probability as predicted by theoretical models [87]. Measurements from the KAUMesh indoor testbed (Figure 8.1) show that the transition region of links with

almost no loss and 100% loss can span around 10 dB, making predictions of packet loss rates based on RSSI difficult.

Optimization models often need information about the achievable throughput on a link or the airtime consumption of a transmission. However, estimating those factors from RSSI samples can be very inaccurate. When in addition considering other effects such as hidden terminals and rate adaption, the RSSI is an even less suitable metric.

An alternative method to determine the quality of a wireless link is to use tools such as WBest [109] or BART [76], which use probe packets to estimate available bandwidth of a link. The available bandwidth estimate can then be used to select the best AP among all APs in range. Unfortunately, this method is not scalable as it requires frequent probing to all available APs. Moreover, active bandwidth estimation methods often take a long time to converge [109], during which the available bandwidth may have changed and a once optimal AP becomes sub-optimal again.

While RSSI-based link quality estimation is inaccurate, probe based approaches are more accurate, but can only determine the quality of an AP with an active association. To probe a new AP, a station would have to perform a handover from its current AP to the new AP. When APs are probed frequently, the long handover durations in standard IEEE 802.11 systems would cause severe disruptions in the service. Thus it is desirable to have a system, which 1.) provides accurate probe-based link quality estimation, 2.) supports fast handovers or simultaneous connections to multiple APs and 3.) allows station mobility. Such a system could be used both to enable seamless station mobility and load balancing, as we will show in this and the next chapter.

8.1.1 Related Work

8.1.1.1 Available Bandwidth Estimation

A number of different tools and techniques have been proposed which allow to estimate the *Available Bandwidth* (AB). However, most techniques were designed for wired networks and cannot provide accurate information for wireless networks and have a long convergence time. This is because in wireless environments, the available capacity can vary dramatically, wireless links behave non-deterministically and non-linearly especially under heavy load.

We can distinguish between passive and active methods to estimate the AB. Passive methods are typically based on monitoring certain transmission characteristics in a non-intrusive way. [84] presents one example of a passive

estimation method, which estimates the available bandwidth based on passively obtained measurements of the channel idle time.

In contrast, active methods typically send probing packets to determine the AB, usually applying some form of packet dispersion measurements or self-loading of the network. The main idea of packet dispersion measurements is to send a pair or a train of packets back-to-back and measure the time-difference of arrival. Thereby, the available bandwidth can be estimated.

ProbeGap [105] is one example of an active estimation method, which estimates the available bandwidth in WLAN deployments by observing the idle time fraction using one-way delay samples. Pathload [94] and pathChirp [146] probe the end-to-end network path using multiple traffic rates. Dynamic PHY rate adaptation is a challenge when estimating the AB. WBest [109] was designed to provide accurate measurements even with dynamic PHY rate adaptation. WBest uses a combination of the packet pair technique to estimate effective capacity over a path where the last hop is a wireless link and a packet train method which estimates achievable throughput to infer the available bandwidth.

The discussed mechanisms provide relatively accurate AB estimation, but fail if continuous information about available bandwidth is required, because the constant probing would result in an unacceptable overhead. BART [76] and its multi-rate wireless extension MR-BART [152] aim to overcome this limitation. BART allows a *quasi-continuous* estimation of the available bandwidth by applying a Kalman filter, which gradually improves the accuracy of the estimate. BART was designed to introduce only little probe traffic. Nevertheless, BART is not suitable for the purpose of AP selection, since the station would still have to successively connect to the available APs to estimate the bandwidth. This would lead to disruptions due to the handovers.

8.1.1.2 Access Point Selection

There has been significant interest to develop systems which support multiple APs. The most simple way is to use separate wireless cards to connect to different APs or base stations. An example of such system is PERM [166], which uses multiple residential ISPs and assigns network flows to different cards based on latency estimates. A system which uses multiple cellular connections and providers is Horde [138]. Systems using just one card are more challenging to design because different APs may be on different channels. A prominent example is FatVAP [98], which uses a single WLAN card to associate and exchange data with multiple

APs. A scheduler assigns TCP/UDP flows to the APs and, based on bandwidth estimations, schedules when to request buffered data from which AP. FatVAP provides performance improvements in networks with a slow wired backhaul. Here, the APs can pre-buffer downstream data, before the station requests it. THEMIS [81] is a system similar to FatVAP, but was explicitly designed to improve fairness among WLAN users. Finally, MultiNet [56] allows a single WLAN card to appear as multiple virtual WLAN interfaces, which can then be configured independently to connect to different wireless networks. However, MultiNet does not define how long a given user should remain connected to an AP to maximize its performance. Station mobility cannot be handled properly with [166], [98], [81] and [56], since those systems use one IP address for each AP connection and mobility would break ongoing connections.

8.1.2 Problem Statement and Contributions

There is an interesting relationship between AB estimation, fast handovers and AP selection. To select the best AP, the available bandwidth to surrounding APs needs to be estimated. Probe-based estimation methods are accurate, but require the station to be associated to the AP, which should be probed. If there are multiple APs around, the station needs to frequently associate to all of the surrounding APs to determine the best AP. To avoid noticeable disruptions in ongoing data transfers, the handover between APs should be fast or simultaneous connections to multiple APs should be supported.

None of the reviewed proposals meets all the three outlined requirements, i.e. accurate available bandwidth estimation, simultaneous connections and fast handovers and mobility support. Thus, in this chapter we develop and evaluate BEST-AP, a system for available bandwidth estimation and fast handovers. In this chapter we focus on dynamic AP selection in absence of mobility. How BEST-AP supports station mobility will be analyzed in Chapter 9.

The *key contributions* presented in this chapter are:

- A new **method for the estimation of the available bandwidth**: The available bandwidth estimation is based on the observation, that the two main factors determining the available bandwidth are the channel load and the packet loss rate. The available bandwidth for each STA is estimated at the AP based on a mathematical model, which combines available air time measurements with statistics on individual frame loss and PHY rates using normal data traffic. APs send bandwidth estimates to a monitoring server, which combines

estimates from several APs and informs STAs about available bandwidth to each AP periodically.

- **A system for fast handovers:** A scheduler in the STA decides how long to communicate with each AP. The best AP is used longest, while surrounding APs on potentially other channels are used for shorter durations to obtain bandwidth estimates. STAs can pre-associate and pre-authenticate to APs and thereby increase the handover speed. BEST-AP works with a single IEEE 802.11 interface and requires only small changes of the APs and the WLAN drivers on the stations. It does not require changes to the IEEE 802.11 MAC layer and works together with automatic rate control and all other standard features of the driver. The design of BEST-AP uses ideas and protocols from the Software Defined Networking framework.
- **An extensive evaluation** of our prototype in the KAUMesh testbed. The evaluation shows that our bandwidth estimation is robust to cross-traffic and interference. In addition, it achieves a rapid convergence time. In all relevant criteria, it beats current state-of-the-art wireless bandwidth estimation tools such as WBest [109]. The system for the dynamic scheduling of APs achieves a median throughput gain of 85% under interference.

The rest of the chapter is organized as follows: Section 8.2 presents extensive measurements to motivate why a frequent estimation of available bandwidth is necessary. In Section 8.3 we use the insights gained from the measurements to develop a new model and method for estimating the available bandwidth of a WLAN link. In Section 8.4 we describe how the available bandwidth estimation can be used for dynamic AP selection. Section 8.6 evaluates the accuracy of the available bandwidth estimation and the performance gains obtained through the dynamic AP selection. In Section 8.7 we conclude the chapter with a discussion of limitations and possible future extensions of BEST-AP.

8.2 Motivating Examples

In the following we will illustrate the shortcomings of current approaches for estimating the available bandwidth in WLANs, in particular with the aim to find the AP which provides the highest available bandwidth.

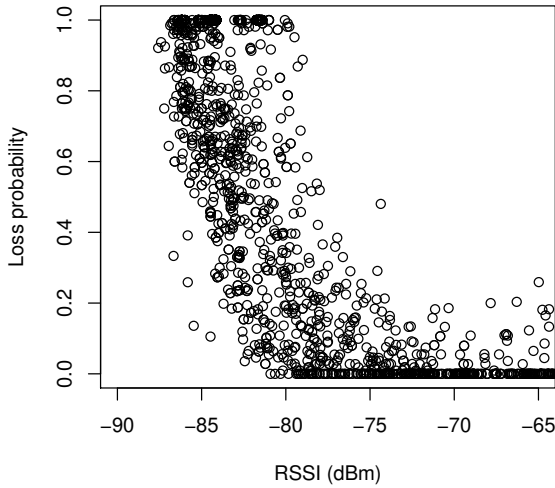


Figure 8.1: Packet loss probability and RSSI at 12 Mbit/s obtained from measurements on the Karlstad University campus

8.2.1 RSSI is not Suitable for Accurate Available Bandwidth Estimation

A common approach to assess the quality of an AP and to estimate the available bandwidth is to measure the signal strength of a link. In theory, the Bit Error Rate (BER) can be computed for a given Signal-to-Noise Ratio (SNR) [143]. Using the BER, the frame error rate and therefore the throughput can be computed. However, in practice such computations lead to large estimation errors for several reasons. First, a WLAN card can only measure the superposition of the signal and the noise, but not the SNR. The noise level needs to be estimated and is subject to error [148]. Thus, the value reported by the WLAN card, usually called RSSI, does not necessarily reflect the true SNR. Second, the theoretical SNR/BER models assume certain noise characteristics, such as white noise [143]. In real deployments the noise characteristics are affected by multi-path propagation and inter-symbol interference, which depend on the TX/RX location and the propagation environment. In addition, as a third source of uncertainty, the RSSI does not consider the actual load on the channel. Even for a high RSSI link the available bandwidth may be small, when other users compete for the same channel.

Because of those factors for given RSSI the loss probability is not fixed, but can differ significantly. We measured the RSSI of beacon messages broadcasted by the WLAN on the campus of Karlstad University in different locations during evening hours, in which the traffic volume generated by students and staff is low. As the beacon broadcast interval is fixed at 100 ms, it is easy to identify lost frames. For each received beacon message, the RSSI is reported by the ath5k WLAN driver and logged into a file. When a frame is not received correctly, the RSSI is assumed to be identical to the last received frame. This assumption is reasonable, as the channel gain over short periods in almost static environment does not change significantly. Thus the RSSI of the previous frame is also a good estimation of the reception strength of the current frame. Figure 8.1 shows that the transition region between a nearly loss-free and a 100% lossy link is quite wide. As a consequence, for a given RSSI, the loss rate can vary significantly. For this reason, the RSSI only allows a coarse estimation of the link quality.

8.2.2 Packet Dispersion Measurements are too Slow for Continuous Estimation

A second common approach to estimate the available bandwidth is to measure the packet dispersion, as for example implemented by WBest [109]. As in WLANs the packet transmission is subject to a probabilistic MAC protocol, a relatively large number of probe packets need to be sent to get an accurate estimate. As a result, packet dispersion measurements can take several seconds to produce an estimate. However, in many WLAN scenarios the available bandwidth fluctuates fast due to variations in background load and user mobility. In the following, we will present results from a series of measurements of channel load and link quality. Channel load and link quality are the two main factors impacting the available bandwidth. Our measurements show that those factors and hence the available bandwidth in WLANs fluctuates much faster than typical packet dispersion measurement systems can track.

8.2.2.1 Throughput with Varying Channel Load

The channel load in WLANs varies due to random user activity. The channel busy fraction is the fraction of time the channel is busy and represents a common metric for the channel load. We measured the channel busy fraction using the method of Chapter 3. In addition, we implemented a measurement application that reads the value of both

hardware registers approximately 1500 times per second and stores them along with a time stamp in a logfile. We measured the channel load in three different locations, which represent possible application areas for BEST-AP: in an office environment in downtown Berlin, on the Karlstad University library and in a hotspot environment. The hotspot environment was simulated by replaying traces from the SIGCOMM 2008 conference [151] in our indoor testbed.

The channel busy fraction is measured and computed within a certain time window. This naturally leads to the question what a meaningful window size is. In the extreme case, a window could just be one OFDM symbol duration ($3.2 \mu\text{s}$ in IEEE 802.11a) long. In such a case, there are only two possible values for the channel load: 0 and 1. Either the channel is indicated busy or not. Such small time scales are clearly not of interest for available bandwidth estimation. Instead, time scales in the order of several frame transmissions durations should be considered. Figure 8.2 depicts histograms of the channel busy fraction at the Karlstad University library for different averaging windows. With 0.001 seconds window size, the histogram peaks at 0 and 0.7. When the window size is increased, the histogram gets narrower and finally converges to one peak, which is the average channel busy fraction of the whole measurement period.

Even though Figure 8.2 suggests that changes in channel load occur on small time scales, the channel load may still exhibit a high autocorrelation. If the available bandwidth can be estimated (or approximated) as a linear function of the channel busy fraction, a high autocorrelation of the busy fraction would also imply a high autocorrelation of the available bandwidth. In such a scenario, estimations can be obtained infrequently, as the current estimate and a future estimate most likely will not differ much. However, our measurements show that the autocorrelation is low and hence frequent estimations of the available bandwidth are required. More specifically, Figure 8.3 plots the sample autocorrelation function for our three datasets with 10 ms windows. Typically, the coherence time of a wireless channel is defined as the time in which the autocorrelation of the channel signal response is larger than 0.5 [143]. We apply the same definition to the channel load and see that the coherence time of the load for the library and office set is smaller than 20 ms and 10 ms seconds respectively. Interestingly, in the hotspot data set the autocorrelation decays slower and the coherence time is approximately 5.3 sec. This is because in the hotspot data set during long periods there is almost no load on the channel, while during some periods the load is constantly very high. The office data set furthermore shows 10 spikes per second, which are a result of the periodic beacon frames broadcasted by the APs at 1 Mbit/s. In the library data set beacon

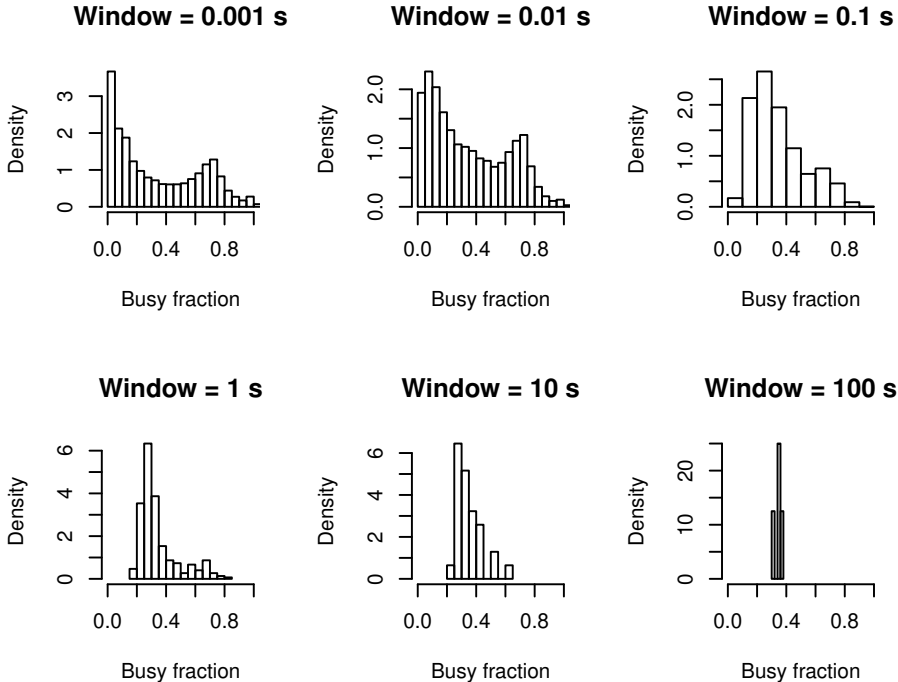


Figure 8.2: Histogram of channel busy fraction during 5 minutes

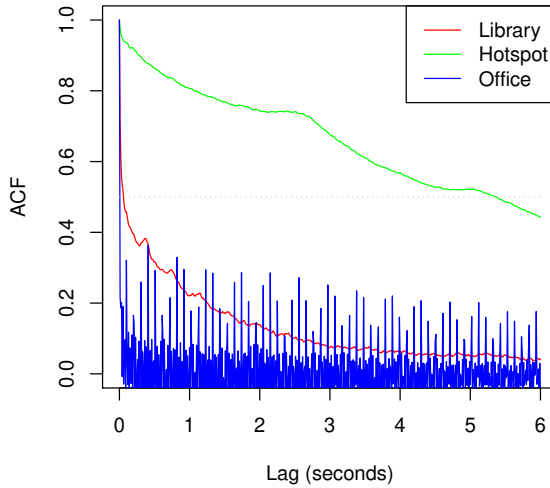


Figure 8.3: Sample autocorrelation function of the channel load

frames are broadcasted at 6 Mbit/s and therefore impact the channel load much less.

The results show that the channel load changes fast and can exhibit low autocorrelation. Hence, if one would like to exploit changes in channel load, for example to select a lightly loaded AP, those selections should preferably be done on very short time scales, e.g. every 10-100 ms. Because then it is possible to use a channel with very low load. In contrast, when adapting to changes only every couple of minutes, those short term variations cannot be exploited.

8.2.2.2 Throughput with User Mobility

User mobility leads to changes in the link quality and consequently to changes in available capacity. For example, as the distance between the user station and the AP increases, more robust but slower coding and modulation schemes need to be used. We studied the temporal characteristics of the channel by transmitting 1000 UDP packets per second from an AP to a laptop. The AP uses the Minstrel rate adaptation algorithm [9], which continuously aims to select the best Modulation and Coding Scheme (MCS). While walking around with the laptop and receiving the UDP packets, the AP recorded statistics about frame loss

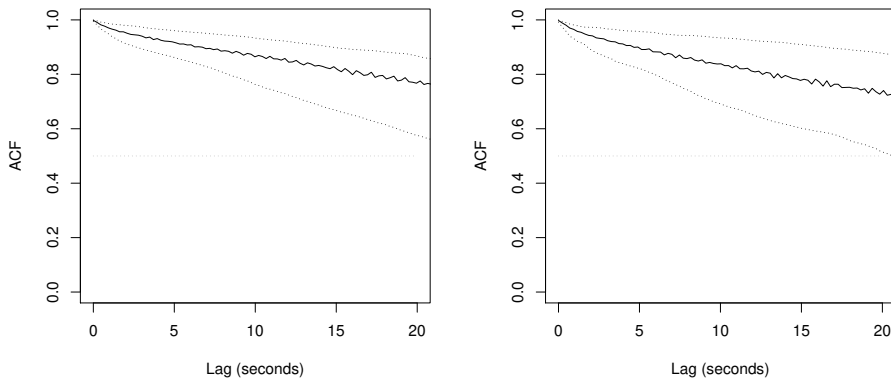


Figure 8.4: Sample autocorrelation function of the available bandwidth estimation, without (left) and with external interference (right)

probability and the used PHY rate. We then computed a simple estimate of the available bandwidth as $(1 - F) \times P \times \eta$, where F denotes the frame loss probability, P is the PHY rate and η is a discount factor to account for transmission overhead. This estimation does not consider yet the impact of load generated by other users of the same WLAN channel. We will present a more precise model in Section 8.3.

Figure 8.4 plots the mean (solid line) as well as the 1-st and the 99-th percentile of the sample autocorrelation function of the bandwidth estimation obtained for nine different walks with the test laptop, without and with external interference. The external interference was created by five stations, which continuously downloaded the start page (with images, CSS files etc.) of popular websites such as google.com or amazon.com. In both cases the autocorrelation function decreases surprisingly slow. External interference may create frame loss due to collisions and cause a slightly faster decrease of the autocorrelation. In both cases however, even the 1-st percentile autocorrelation does not drop below 0.5 even after 20 seconds. This shows that the available bandwidth is relatively stable over long periods, even when the station is mobile.

Figure 8.5 helps to explain *why* the autocorrelation drops only slowly. The PHY rate (black line) is adapted frequently by Minstrel and shows discrete jumps (there is only a finite set of possible PHY rates to choose from). Similarly, the frame success probability (gray line) also is subject to high variations. Since the rate adaption mechanism reacts to high frame loss by choosing a more robust MCS, the combined effect of the

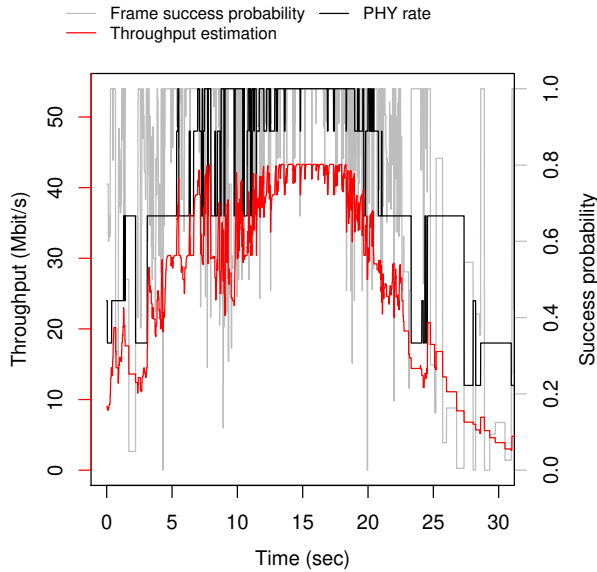


Figure 8.5: PHY rate (black), frame success probability (gray) and available bandwidth estimation (red) for a user walking past an AP. The user is closest to the AP after 15 seconds.

variations in PHY rate and frame loss probability leads to a relatively stable estimate of the available bandwidth (red line). The estimate of the available bandwidth increases as the laptop moves towards the AP and decreases as it moves away, but does not have any sharp drops from e.g. 40 to 0 Mbit/s.

8.2.3 Discussion

The examples above illustrate why current approaches for available bandwidth estimation in WLANs are not sufficient, in particular if short-term variations should be tracked. Signal strength measurements are inaccurate and packet dispersion measurements are too slow. The channel load varies much faster than the convergence time of packet dispersion measurement systems, which is between 0.5 and 30 seconds [109]. In the next two sections we will therefore develop BEST-AP, a method and a system to estimate the available bandwidth in a fast non-intrusive manner. BEST-AP allows stations to be connected to several APs in parallel and switch fast between them, while using the AP with the highest available bandwidth most of the time and others less frequently. The estimation exploits the insight,

that the available bandwidth is mainly a product of the PHY rate, frame loss rate and the channel load. The first two factors are hard to measure and vary slowly over time, while the channel load varies fast, but is more easy to measure.

8.3 Non-intrusive Bandwidth Estimation

We proceed with describing a simple model for available bandwidth estimation. We define the available bandwidth of a link as the maximum data rate which can be sent over the link, while not deteriorating already ongoing connections ([48] calls this *surplus available bandwidth*). This is in contrast to the *fair share bandwidth*, which would allow to deteriorate already ongoing connections to a certain degree. The available bandwidth and the fair share bandwidth can be seen lower and upper bounds of the actual bandwidth an application can use. The gap between those bounds depends on many factors such as the network configuration and topology. In the following we will estimate the available bandwidth, since such estimation provides information about the worst case performance and it is less sensitive to topology changes.

8.3.1 Model for Estimation of Available Bandwidth with a Fixed MCS

The transmission duration T_{tx} of an IEEE 802.11 unicast data frame with fixed length is a function of the backoff level n and the PHY rate r . In Equation 8.1 we compute T_{tx} as the sum of a waiting period T_{difs} , a random backoff period $T_{backoff}$, a data transmission period T_{data} , followed by T_{sifs} and transmission time of an ACK frame T_{ack} (default values for IEEE 802.11a are listed in Table 8.1). Additionally, the data and the ACK frames have a transmission delay of T_{prop} .

The Physical Layer Protocol Data Unit (PPDU) of a data frame is composed of a PLCP preamble, a PHY header and the MAC Protocol Data Unit (MPDU). The PLCP preamble and the PHY header require T_{pre} and T_{phy} time for transmission. The transmission duration of the MPDU depends on the symbol time T_{sym} , the length of the data L_{data} and the number of symbols per second N_{dbps} . In addition, check-sums and a MAC header of length L_{hdr} need to be added. This then results in a transmission time of T_{data} , which is defined in Equation 8.2. Similarly, the transmission time of an ACK, which is sent at rate r_l , is defined in Equation 8.3. IEEE 802.11 uses a binary exponential backoff mechanism in which the back-off time for the n -th transmission attempt is chosen

$$\mathbb{E}[T_{tx}(r, n)] = T_{difs} + T_{backoff}(n) + T_{data}(r) + T_{sifs} + T_{ack} + 2T_{prop} \quad (8.1)$$

$$T_{data}(r) = T_{pre} + T_{phy} + T_{sym} \lceil \frac{16 + 6 + 8L_{hdr} + 8L_{data}}{N_{dbps}(r)} \rceil \quad (8.2)$$

$$T_{ack} = T_{pre} + T_{phy} + T_{sym} \lceil \frac{16 + 6 + 8L_{ack}}{N_{dbps}(r_l)} \rceil \quad (8.3)$$

$$\mathbb{E}[T_{backoff}(n)] = T_{slot} \frac{\min(2^n CW_{min}, CW_{max})}{2} \quad (8.4)$$

$$\mathbb{E}[T_{total}(r)] = \sum_{i=0}^k (1 - p_{suc}(r))^i p_{suc}(r) T_{tx}(r, i) \quad (8.5)$$

$$\mathbb{E}[avbw(r, b)] = L_{data} \frac{1}{\mathbb{E}[T_{total}(r)]} (1 - b) \quad (8.6)$$

randomly between $[0, \min(2^n CW_{min}, CW_{max}) - 1]$ slots of length T_{slot} . We compute the expected back-off time for the n -th transmission attempt in Equation 8.4. As in [46], we assume that the probability of a successful frame transmission, p_{suc} , is constant and independent of the backoff stage. The expected transmission time for a frame, including re-transmission attempts is denoted as $\mathbb{E}[T_{total}]$ and computed by Equation 8.5. We are able to transmit $1/\mathbb{E}[T_{total}]$ frames per second on an idle channel. Finally, for a channel busy fraction b and a PHY rate r , the available bandwidth is computed in Equation 8.6.

Equation 8.6 provides a simple way to estimate the available bandwidth for a known channel busy fraction and packet success probability. However, Equation 8.6 assumes that a fixed PHY rate is used (i.e. N_{dbps} is fixed), which is often not the case in real networks. Instead, many WLAN drivers use a rate control algorithm, which chooses the best PHY rate for the current channel conditions. Therefore, we extend the model for the auto-rate case in the next section.

8.3.2 Model for Estimation of Available Bandwidth with Rate Adaptation

Minstrel is the default rate control algorithm in the Linux mac80211 subsystem [9]. Minstrel aims to select the PHY rate which gives the best throughput by utilizing retry chains. A retry chain specifies at what PHY rate to send a MAC frame at what transmission attempt. For example, a

Symbol	Default value for IEEE 802.11a	Description
T_{pre}	16 μ s	PLCP preamble duration
T_{phy}	4 μ s	PHY header duration
T_{sym}	4 μ s	OFDM symbol duration
T_{slot}	9 μ s	MAC layer slot duration
T_{difs}	34 μ s	DIFS duration
T_{sifs}	16 μ s	SIFS duration
T_{prop}	typically < 0.1 μ s	Propagation duration
$N_{dbps}(r)$	up to 6 bits	Number of bits per symbol
CW_{min}	15	Minimum contention window
CW_{max}	1023	Maximum contention window
k	11	Maximum back-off stage
r	6-54	PHY rate
L_{ack}	14 bytes	Acknowledgment length
L_{hdr}	32 bytes	MAC header and checksum length
L_{data}	max. 2312 bytes	Data length

Table 8.1: Used symbols and default values for IEEE 802.11a

retry chain can specify to send a frame at 54 Mbit/s, and in case of an error re-transmit at a lower rate on the second try and an even lower rate on the third try.

For each PHY rate, Minstrel keeps statistics about the frame success probability. Minstrel estimates the achievable throughput of a PHY rate based on the time it takes to transmit a packet and its success probability with a model similar to the one described above. In the normal retry chain, a frame is first transmitted at the PHY rate with the best throughput prediction (e.g. 54 Mbit/s), then with the second best one (e.g. 48 Mbit/s), then at the PHY with the highest success probability (e.g. 6 Mbit/s) and finally at the lowest base rate (e.g. 6 Mbit/s). If the frame is not received in the final transmission attempt, it is discarded and a new transmission with a new frame is started. In order to obtain statistics about all PHY rates, Minstrel sends 10% of the frames with an explore chain. The explore chains include random rates. Table 8.2 shows the retry chains of the current Minstrel implementation in the Linux Kernel (version 3.0.3).

As Minstrel sends frames at different PHY rates, estimating the throughput is not as simple as in Section 8.3.1. We first define five different PHY rates:

Try	Explore chains		Normal chain
	random < best	random > best	
1	Best throughput	Random rate	Best throughput
2	Random rate	Best throughput	Next best throughput
3	Best probability	Best probability	Best probability
4	Lowest baserate	Lowest baserate	Lowest baserate

Table 8.2: Minstrel retry chains. Source: [9]

1. r_b : Rate which provides the best throughput according to the estimation of Minstrel
2. r_n : Rate with the second best throughput
3. r_p : Rate with the highest success probability
4. r_l : The lowest base-rate according to the used PHY standard (6 Mbit/s in IEEE 802.11a).
5. r_r : A random rate chosen by Minstrel

Based on those rates, we define

- the normal chain as $\mathbf{c}_n = \langle r_b, r_n, r_p, r_l \rangle$,
- the first explore chain as $\mathbf{c}_{e1} = \langle r_b, r_r, r_p, r_l \rangle$
- and the second explore chain as $\mathbf{c}_{e2} = \langle r_r, r_b, r_p, r_l \rangle$.

The first explore chain is used if $r_r < r_b$, the second is used otherwise. With $[\mathbf{c}]_i$ we denote the i -th element of a vector (starting from 1). The expected time to transmit a frame with rate r at backoff stage n is defined in Equation 8.7. Assuming independence between the transmission attempts, the expected transmission time of a frame with the retry chain \mathbf{c} can therefore be computed as the weighted sum of transmission times and success probabilities for the rates of the chain (Equation 8.8).

With \mathbf{r} we denote the vector of all PHY rates supported by the WLAN card. $H(x)$ is the heavy side step function, which is 0 if x is strictly negative and 1 otherwise. Minstrel uses the normal chain $\alpha\%$ (90% by default) of the time. Equation 8.9 computes expected transmission time of a frame with Minstrel as sum of the transmission times for the three retry chains weighted by the probability of using a chain. Finally, the expected available bandwidth with Minstrel rate control and a channel busy fraction b is computed in Equation 8.10. We will evaluate the accuracy of the model in Section 8.6.

$$\mathbb{E}[T_{tx}(r, n)] = T_{difs} + T_{backoff}(n) + T_{data}(r) + T_{sifs} + T_{ack} + 2T_{prop} \quad (8.7)$$

$$\mathbb{E}[tx(\mathbf{c})] = p_{suc}([\mathbf{c}]_1)T_{rx}([\mathbf{c}]_1, 1) + \sum_{i=2}^4 T_{tx}([\mathbf{c}]_i, k)p_{suc}([\mathbf{c}]_i) \prod_{k=1}^{i-1} (1 - p_{suc}([\mathbf{c}]_k)) \quad (8.8)$$

$$\begin{aligned} \mathbb{E}[T_{total}] &= \alpha tx(\mathbf{c}_n) + (1 - \alpha) \left(\frac{1}{|\mathbf{r}|} \sum_{r_r \in \mathbf{r}} H(r_b - r_r) tx(< r_b, r_r, r_p, r_l >) + \right. \\ &\quad \left. + H(r_r - r_b) tx(< r_r, r_b, r_p, r_l >) \right) \end{aligned} \quad (8.9)$$

$$\mathbb{E}[avbw(b)] = L_{data} \frac{1}{\mathbb{E}[T_{total}]} (1 - b) \quad (8.10)$$

8.4 Dynamic AP Selection Based on Bandwidth Estimation

In this section we present the BEST-AP system, which measures the parameters required for the available bandwidth estimation, computes the estimation for all APs in reach of a station and sends the estimates to the stations. The aim of the system is to allow a station to determine and use the AP which currently offers the highest available bandwidth. A BEST-AP WLAN is a standard WLAN, which in addition provides services for bandwidth estimation and station handovers. Figure 8.6 shows an overview of a BEST-AP WLAN, which consists of mobile user stations, APs interconnected with an Ethernet network, an IP router or OpenFlow switch and the BEST-AP server.

8.4.1 Mobile Station

The mobile station is equipped with one IEEE 802.11 WLAN card and optionally with one dedicated card for scanning for new APs. On top of the physical card, several virtual WLAN cards are created. Each virtual WLAN card has a unique MAC address and can be associated to one AP. As one station has multiple virtual WLAN cards, it can be connected to several APs simultaneously. It is not required that all APs are operated on the same channel, but the physical WLAN card needs to be able to

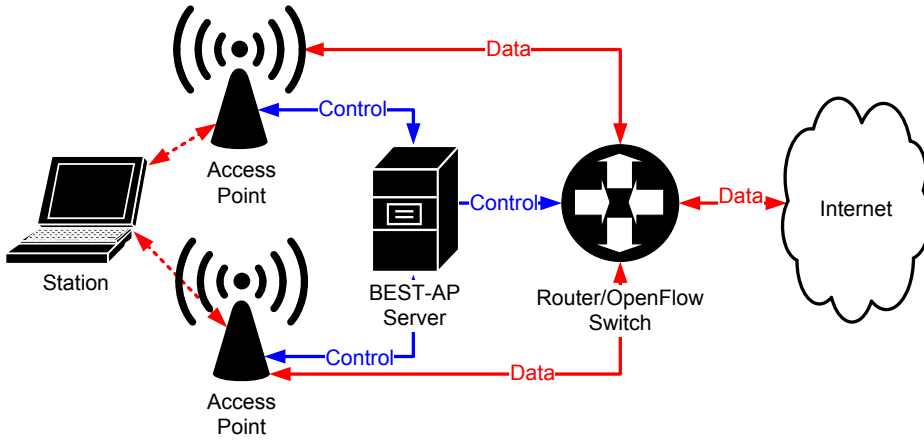


Figure 8.6: Architecture of a BEST-AP WLAN

switch fast between available channels.

The virtual WLAN cards are connected to a software switch, the OpenFlow switching element. In addition, the software switch is connected to a super virtual WLAN device and a transmission buffer. The super virtual WLAN device is the interface to the OS networking stack and has a device-wide IP and MAC address. The transmission buffer can be set into a *blocked* state, in which all data packets (not control packets) from higher layers are stored in the buffer, but not forwarded to the OpenFlow switching element. If the transmission buffer is in the unblocked state, it simply passes packets down to the switching element. Blocking traffic at the transmission buffer ensures that no outgoing packets are lost when the station is performing a handover.

The switching element contains rules that specify 1.) to which virtual WLAN card outgoing packets should be forwarded to and 2.) what source MAC address should be used for outgoing packets. Those rules are configured from the switching controller application using OpenFlow. As each virtual WLAN card is associated to one AP, choosing a virtual WLAN card for transmission results in a transmission to a specific AP. Rewriting the source MAC address is required, since MAC frames generated by the operating system carry the MAC address of the super virtual WLAN device as source. To be transparent to the underlying network, the MAC address of the respective virtual WLAN card is used instead.

The station contains the handover manager, which consists of a scheduler and a scanning module. The scheduler decides when to transmit or receive frames via which AP and when to scan for new APs. In other

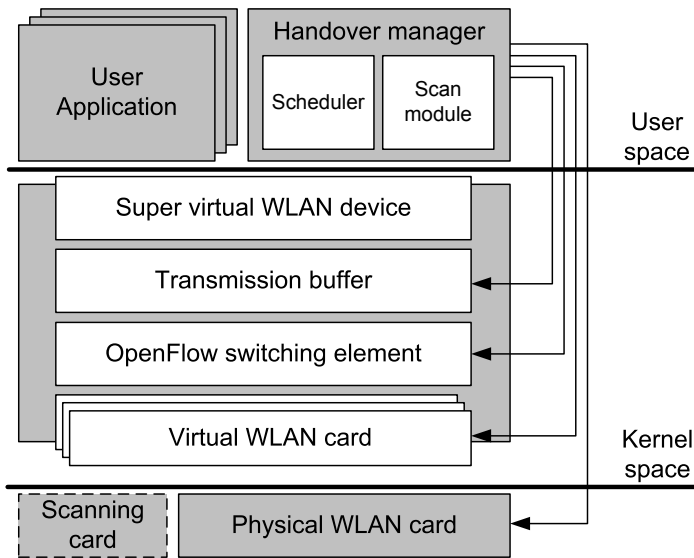


Figure 8.7: Architecture of a mobile station

words, the scheduler decides when to perform a handover. In addition, the handover manager generates the signaling messages to reconfigure the routing in the backhaul network for up- and downstream packets (see Section 8.4.3). The decision when to schedule a handover or a scan for new APs can use information provided by user applications.

8.4.2 Pre-authentication and Pre-association

In a normal IEEE 802.11 WLAN a station can only associate to one AP at a given time. The station typically needs to perform the complete association procedure when doing a handover to a new AP. As a result, normal WLAN handovers are slow. With BEST-AP, a station can associate to an AP outside its coverage area using out-of-band signaling. Using an existing wireless connection and the wired backhaul, it can exchange association and authentication messages with the BEST-AP server, which relays them to the respective AP. This mechanism allows a station to associate to all APs of a limited geographic area, before the station arrives at this area, which can significantly speed up handovers. Note, that our approach of pre-authentication and pre-association is different from IEEE 802.11r, as we allow to exchange messages via an AP which is not in reach using the connection of a close-by AP. This can be performed *before* the handover. Therefore, the number of authentication message exchanged

and the duration is not a performance limiting factor in our scheme.

8.4.3 Handover Services

The BEST-AP scheduler at the station decides when to initiate a handover to which AP. Figure 8.8 shows a sequence diagram of the messages exchanged during a handover. First, the station puts its transmission buffer into the blocked state. Hereafter, the station transmits a “Handover Initiate” message to its currently used AP (AP1), which forwards the message to the BEST-AP server. Optionally, the BEST-AP server then answers to the station with an ACK message and sends a pause message to the switch. If the network supports OpenFlow, the BEST-AP server uses the OpenFlow protocol to update the forwarding table of the switch to send future downstream data packets via AP2. If the network does not support OpenFlow, the BEST-AP server updates the ARP table of the router. Since each virtual WLAN card has its own AP-specific MAC address, updating the ARP table allows to direct IP packets to the station via the correct AP.

As soon as the station has received the ACK message, it tunes its WLAN card to the channel of the new AP. In case no ACK is received within a timeout period, the station retries up to five times. If no retry is successful, for example because AP1 is already out of reach, the station nevertheless attempts to change the channel and sends a “Handover Initiate” message via AP2. The OpenFlow switch, just like the station, contains a per station transmission buffer, which is set into the blocking state upon the reception of the pause transmission command. Once the station has tuned its card to the new channel, it sends a “Handover Complete” message to the controller via AP2 and unblocks its transmission buffer. The controller then resumes the downlink traffic by unblocking its transmission buffer.

In scenarios where the station is not mobile and the network does not support OpenFlow, a station can ensure that no packets are lost using the Power Save Mode (PSM) of IEEE 802.11. More specifically, when a station would like to communicate with an AP that is not on its current channel, the control application first informs all APs on the current channel that it enters PSM. This causes the APs to stop transmitting frames to the station and to buffer them. The station then changes the channel. Subsequently it informs the AP on the new channel that it is available for packet reception by sending a null frame with the PS-POLL flag. The AP then sends all buffered frames to the station. The station then can communicate with the AP.

The handover between the APs does not require the exchange of

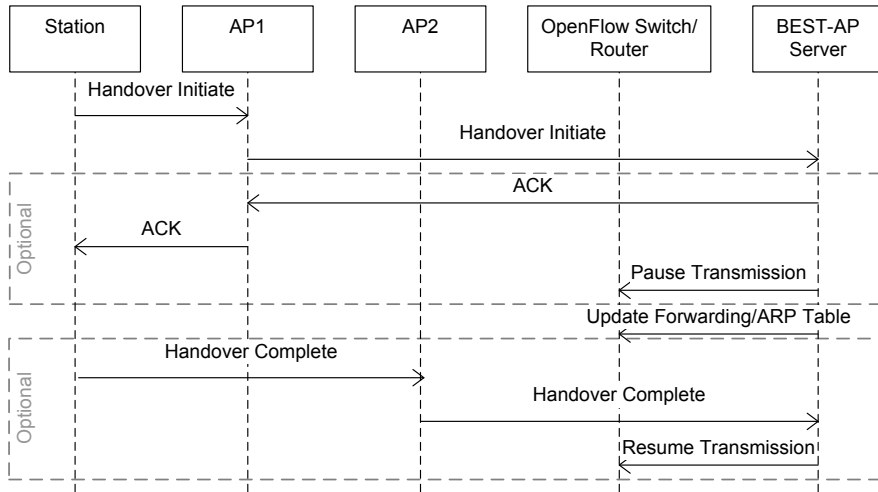


Figure 8.8: Sequence of messages exchanged during a handover from AP1 to AP2

any additional IEEE 802.11 control messages, as the station is already associated to the new AP by means of out-of-band pre-authentication and pre-association before the actual handover has started.

8.4.4 Scheduling AP Usage

In this section we describe how a station decides when to use which AP. Each station maintains a list of APs along with their RSSI and bandwidth estimation. Figure 8.9 shows the states an AP can have and the corresponding state transitions:

1. Not detected: No recent activity of the AP has been recorded.
2. Detected: An AP moves from the “Not detected” state to the “Detected” state, once it has been discovered through scanning.
3. Secondary AP: the AP is used occasionally in order to collect statistics about the frame loss probability. A “Detected AP” is promoted to a “Secondary AP”, if its RSSI is among the top k values for all APs (e.g. the top 3 values). We distinguish between “Detected APs” and “Secondary APs”, because a station may see many APs, of which most may have a low available bandwidth. Sending data messages via those APs may not contribute to finding the AP with the highest available bandwidth. Moreover, a handover to each secondary AP

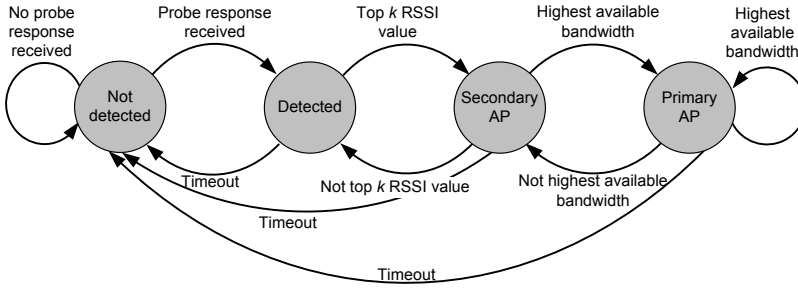


Figure 8.9: States of APs and their transitions

needs to be performed every once in a while, in order to update the frame loss statistics.

4. Primary AP: Through the collected statistics and the method described below, the available bandwidth of each AP is estimated. The AP with the highest estimation is the primary AP.

The control application on the station includes a scheduler, which decides when to use which AP for how long. The scheduler schedules the primary AP for T_{prim} (e.g. 2 seconds) and only transmits normal data packets via secondary APs for T_{second} to update the packet loss statistics (e.g. every 2 seconds for 50 ms).

In this chapter, we only consider non-mobile stations. Without mobility, the detection of APs through scanning only plays a minor role. The station can detect its neighboring APs once using normal driver routines and then use this information within BEST-AP. Thus, the transition between secondary and primary AP is most relevant in static networks. In Section 8.6.2 we will therefore analyze the impact of different strategies to schedule primary and secondary APs on the system throughput. In the next chapter however, we will extend our analysis to mobile stations, where scanning plays a more crucial role.

8.4.5 Bandwidth Estimation

APs implement a monitoring service, that collects statistics from the WLAN driver which are needed to compute the available bandwidth estimation. The monitoring service measures the channel busy fraction b as described in Section 8.2.2.1 by reading the channel load statistics from the appropriate hardware registers. In addition, to account for SIFS and DIFS periods, we add SIFS and DIFS as busy period for each unicast frame overheard

by the WLAN card. Our method of computing the channel busy fraction requires special hardware support to obtain statistics from the CCA. If the WLAN card does not offer such support, all traffic could be captured and the channel load could be computed by taking into account the airtime consumed by each frame (similar to [84]). This approach works with most WLAN cards, but increases the processing load at the AP.

APs estimate the available bandwidth for each station periodically using Equation 8.10 and send tuples of (AP-BSSID, STA-MAC, AB-ESTIMATION) to the BEST-AP server, which stores the tuples in a database. For each station and every 100 ms, or if available bandwidth has changed by more than a configurable factor, the BEST-AP server sends the estimates of the best k APs to each station. In our experiments send updates if the bandwidth has changed by more than 10%. One nice feature of BEST-AP is that it can detect if the available bandwidth has changed, even if the station is currently not using the AP. If the channel load and thereby the available bandwidth changes, the BEST-AP server informs the station. The station then can react to the change, for example by selecting a new primary AP. In order to achieve global fairness objectives, the BEST-AP server could also report wrong estimates to a station and thereby force the station to use a specific (non-optimal) primary AP.

We remark that the available bandwidth of a link may depend on the sending direction (uplink or downlink). While models presented in Section 8.3 are suited for both directions, our implementation only estimates the available bandwidth on the downlink, as in typical WLANs the majority of traffic is download traffic and the increased complexity by estimating both directions only leads to minor improvements in accuracy [98].

8.5 Implementation

We have implemented BEST-AP in Linux. All control applications (the BEST-AP server, the control application on station and the measurement application on the AP) were written in Python. In addition, several modifications to the WLAN drivers were required. By default, Linux does not allow virtual interfaces to be used on different channels, which we changed accordingly. Furthermore, we added an interface to the ath5k and ath9k driver, which allows to initiate a fast channel switch. We have implemented the pre-authentication/association scheme by extending the mac80211-subsystem and hostapd. A new control command was added to hostapd, which allows to create an association for a given station MAC address and capability list (PHY rates, 40 MHz channels etc.). Similarly,

the mac80211-subsystem was modified to create an association without exchanging IEEE 802.11 authentication/association frames over the air.

We extended the Linux mac80211-subsystem to measure how much airtime each station consumes and expose those statistics via the debug file system. This information is used to compute the available bandwidth of a station, while considering the bandwidth it is already using. The frame success probabilities p_{suc} for each PHY rate are obtained from the statistics of the rate control algorithm, which are also exposed by the mac80211 debug file system.

8.6 Evaluation

We have evaluated the accuracy and responsiveness of our bandwidth estimation method and the performance improvements through the dynamic AP selection system in the KAUMesh testbed. Our key-findings are:

- Our bandwidth estimation system is more accurate than WBest, while at the same time it does not require any artificial probe traffic.
- The estimation system reacts fast to changes in available bandwidth in below 70 ms.
- The dynamic AP selection achieves a median throughput gain of 85% under interference.

8.6.1 Bandwidth Estimation

8.6.1.1 How Accurate is the Estimation under Constant Network Load?

First, we evaluated how accurate the available bandwidth estimation under constant network load is. For this test, two stations were connected to one AP. The primary station estimates the available bandwidth, while the secondary station uses mgen [11] to transmit UDP data at a constant rate to the AP (“Offered Load”). Recall, that the available bandwidth of a link is defined as the maximum data rate which can be sent over this link without deteriorating already ongoing connections. To obtain the real available bandwidth we gradually increased the send-rate over the AP-primary station link, until the reception rate of the secondary station dropped 2% below the send rate of the AP for one second. A 2% throughput degradation of the secondary link is allowed to account for random frame losses unrelated to the primary link traffic and for inaccuracies in the frame

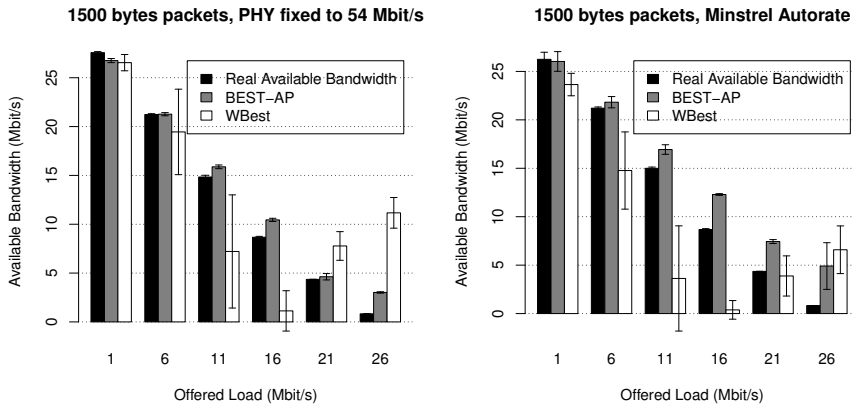


Figure 8.10: Available bandwidth estimation with a fixed PHY rate and Minstrel Autorate with constant background load. Our estimation is more accurate than WBest in all cases.

generation timing. To compute the bandwidth estimation with our method it is required to send a few packets over the link and to obtain statistics about the packet loss rate. In an operational network normal data traffic can be used.

In Figure 8.10 we compare the real available bandwidth with the estimation computed by BEST-AP and WBest for a link with 54 Mbit/s PHY rate and one link with Minstrel Autorate. The figure shows the average and the standard deviation of the 30 repetitions. With the fixed PHY rate, BEST-AP is very accurate, typically within 1 Mbit/s of the real available bandwidth. In contrast, WBest is less accurate, in particular under high network load. The WBest results also have a high variability and therefore several measurements may be required in practice to obtain an accurate estimation.

Automatic rate adaption increases the difficulty of estimating the available bandwidth. With packet pairing techniques, different PHY rates may lead to very inaccurate estimates. BEST-AP explicitly takes into account the impact of rate adaption and is still reasonably accurate in situations where autorate is enabled. The comparison baseline, WBest, was designed to cope with the variations in the link-layer speed caused by autorate algorithms. Our evaluation results show, that BEST-AP is more accurate than WBest. While BEST-AP tends to over-estimate the available bandwidth, WBest tends to underestimate it.

8.6.1.2 How Fast Does the Estimation React to Changes in Network Load?

BEST-AP delivers accurate estimations under constant channel load and conditions. In real WLANs however the channel is subject to constant change. A key feature for available bandwidth estimation in WLANs, besides accuracy, is therefore the ability to react fast to changes in network load. Using the same setup as above, a secondary station added a new traffic flow on the secondary link every 10 seconds. This results in a lower available bandwidth on the primary link.

In Figure 8.11 the estimations with BEST-AP and WBest are plotted. The vertical dotted lines show when the first packet of a new flow is generated. With BEST-AP, the AP computes the bandwidth estimation 100 times per second based on the packet loss rate statistics and the observed channel load. The APs send their estimation to the BEST-AP server, which then sends it to the station every 100 ms or if the change in available bandwidth is larger than 10%. This process leads to some delay before the station receives the current estimation. To analyze the impact of this delay, we therefore plot both estimations, the estimation at the STA and the estimation at the AP. With WBest, the client application is run on the station and the server on the AP. Both client and server show the same estimations and therefore only one estimate is plotted for WBest.

Figure 8.11 shows that BEST-AP reacts fast to changes in available bandwidth. The station is informed about the reduced available bandwidth within approximately 70 ms after a new flow has been started. This delay could further be reduced by computing the available bandwidth more often and by allowing the BEST-AP server to send updates to the station more often. However, this would result in increased system load at the AP and in increased bandwidth use by the control traffic. We would like to remark, that in our implementation the updates from the BEST-AP server are sent to the station in dedicated control packets. However, it would be possible to piggy-back the estimations on normal data packets, for example by using the Options-field in the IP header of regular data packets. We abstain from such an extension of the IP header, as it would increase the implementation complexity and the overhead of the control packets (approximately 7 packets/sec) is negligibly low. The estimation at the STA shows more fluctuations than the estimation at the AP, since the STA only gets a new estimate from the BEST-AP server if the available bandwidth changes by more than 10%.

To get a constant flow of estimations with WBest, we waited until WBest converged and then started a new measurement process. It takes

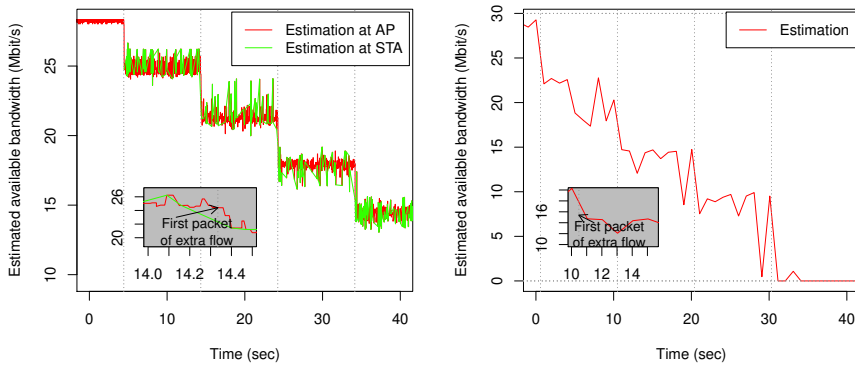


Figure 8.11: Available bandwidth estimation using BEST-AP (left) and WBest (right). The background load increases every 10 seconds.

approximately one second for WBest to converge and hence about one estimation per second is possible. This leads to much slower reaction time than BEST-AP. It can take up to two seconds before changes in the available bandwidth due to a new flow are detected reliably. Furthermore, the fluctuations in the estimation which were present under constant load are also visible under varying load.

8.6.2 Dynamic AP Selection

BEST-AP constantly measures the available bandwidth to surrounding APs. One application of BEST-AP is dynamic AP selection. The idea is that a station selects the AP with the highest available bandwidth as its primary AP, i.e. the AP it uses most of the time. Thereby the station can avoid APs with temporarily high load and mitigate mobility related variations in channel quality.

8.6.2.1 What is the Impact of Different AP Scheduling Strategies?

To estimate the available bandwidth of a secondary AP, data between the secondary AP and the station needs to be transmitted to update the packet loss statistics. There is a trade-off between how often the available bandwidth on secondary APs is estimated and the cost of the estimation. On the one hand, when a station switches between two APs, in-flight packets may be lost or arrive out of order, which subsequently can lead to a throughput degradation. In addition, the secondary APs may provide a

lower available bandwidth and as a result the throughput is decreased if the secondary APs are used. On the other hand, if the available bandwidth of secondary APs is estimated infrequently or for a too short period (i.e. only a few packets are used to update the loss statistics), the estimation may be inaccurate and the wrong AP might be chosen as primary AP.

To characterize this trade-off, we measured throughput of a 60 seconds long TCP-flow, while the station uses the primary and the secondary AP for different durations. To force the station to remain with the same primary AP for the whole experiment, background load was generated on the channel of the secondary AP, which reduced the available bandwidth for the secondary AP. Figure 8.12 shows that a longer time spent on the primary AP leads to a higher throughput. The longer the time on the primary AP, the less often the secondary AP is probed and therefore the “cost” of switching between APs becomes less significant. The duration spent on the secondary AP has less impact on performance. Several positive and negative effects on performance almost cancel each other. With more time on the secondary AP, fewer AP switches are required. In addition, the TCP session has more time to recover from possible packet losses. In the other direction, more time on the “bad” secondary AP reduces throughput.

An optimal schedule of primary and secondary AP use is hard to find in practice, since it depends on many factors. If the channel qualities are very stable, because there is no mobility, then the secondary AP should be used less frequently. How long time should be spent on the secondary AP depends on the packet send rate. If the packet send rate is high, then the packet loss statistics are updated fast and the secondary AP only needs to be used for a short duration. The measurements presented in Section 8.2.2.2 showed that even with user mobility the channel is relatively stable within time horizons of several seconds. Therefore, we advocate to probe the secondary AP every 5-10 seconds for 100-500 ms, as such strategy provides the best performance and allows to detect changes in available bandwidth even with moderate mobility. Unless otherwise stated, we use the primary AP for 10 seconds and the secondary AP for 100 ms in the remaining experiments.

In the discussed experiment only two APs (one primary and one secondary) were present. In some situations more APs might be available. In this case different strategies are possible. One can have multiple secondary APs and probe them in round robin fashion. This leads to additional costs for obtaining the estimations. Another strategy is to select two APs among all APs based on RSSI measurements. As discussed in Section 8.2, RSSI measurements are easy to obtain, but can only provide a coarse estimation of available bandwidth. Such coarse estimations can be used to pre-select

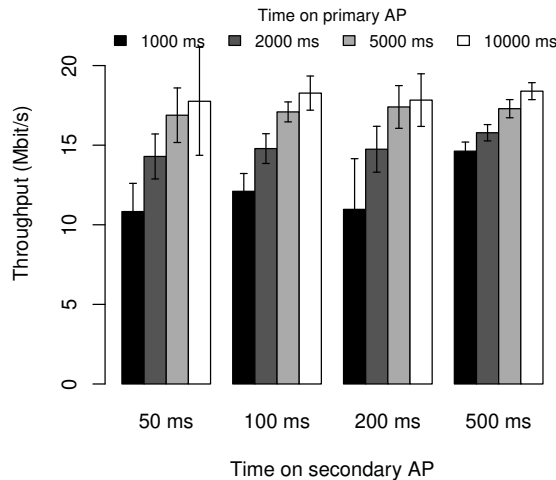


Figure 8.12: Impact of the estimation duration. A longer time on the primary AP is beneficial for the throughput as fewer handovers are required.

two APs among all available APs. The available bandwidth of the two APs is then estimated with BEST-AP.

8.6.2.2 Can BEST-AP Adapt to Changes in Available Bandwidth?

We verified that BEST-AP can detect such changes in available bandwidth and select the primary AP accordingly with the simple setup illustrated in Figure 8.13. The station is using AP1 and AP2 to download data with TCP. At the same time, external stations create load on the channels, which are used by the APs. The load on AP2's channel is constant with 5 Mbit/s, while the load on AP1's channel follows an on-off pattern, with 20 Mbit/s load during 10 second on-periods and no load during 10 second off-periods.

Figure 8.14 shows that BEST-AP tracks those changes in available bandwidth accurately and selects the primary AP according to the channel load. During periods in which AP1's channel experiences load from the external interferer (indicated by the shaded area in Figure 8.15), AP2 is selected as primary AP.

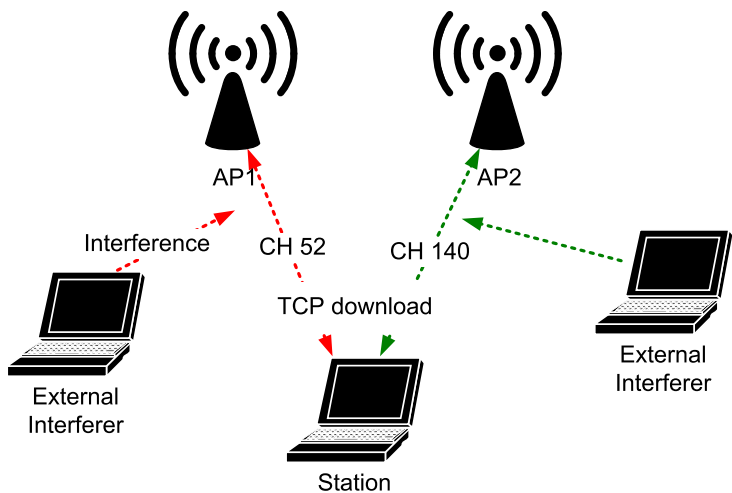


Figure 8.13: Test setup to evaluate how well BEST-AP adapts to changes in available bandwidth. The load on channel 140 is constant, while the load on channel 52 varies according to an ON-OFF pattern.

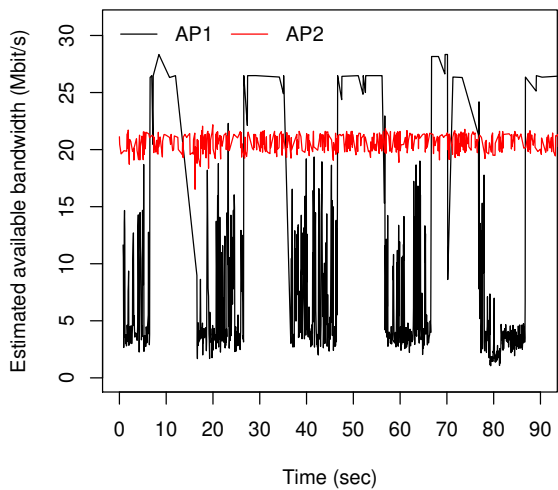


Figure 8.14: Reaction to changes in bandwidth. The available estimation tracks the changes in load at AP1, while it stays constant for the constant load at AP2.

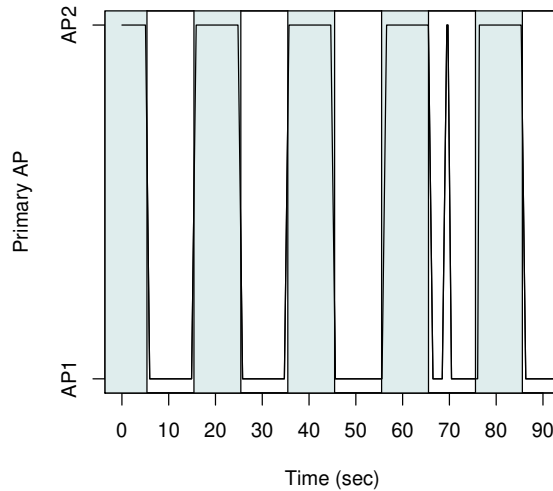


Figure 8.15: Selection of the primary AP. Shaded areas mark periods in which there is load on channel 52.

8.6.2.3 What Performance Gains are Possible under External Interference?

The measurements above showed that BEST-AP is capable of tracking changes in available bandwidth and to select the primary AP accordingly. However, the measurements give no indication what performance gains are possible under realistic channel loads. Therefore, we conducted an additional set of measurements using the same test setup, but using realistic traffic loads. As in Section 8.2, we replayed traces of the SIGCOMM 2008 conference [151]. We filtered the traces by AP and then cut them into 60 seconds pieces. The interfering stations operate one WLAN card in monitor mode, which allows to inject RAW frames. We have implemented an application that reads time-stamp, packet size and PHY rate from the traces and injects frames with approximately the same inter-frame delay into the WLAN card driver. The frames are sent with the NO-ACK flag to avoid MAC layer back-off. In addition, the application manually generates ACK frames and re-transmits 20% of the frames to account for frame losses which are not part of the traces.

In total we performed 160 tests with different traffic patterns of 60 seconds each. For each test case, we compared the performance of BEST-

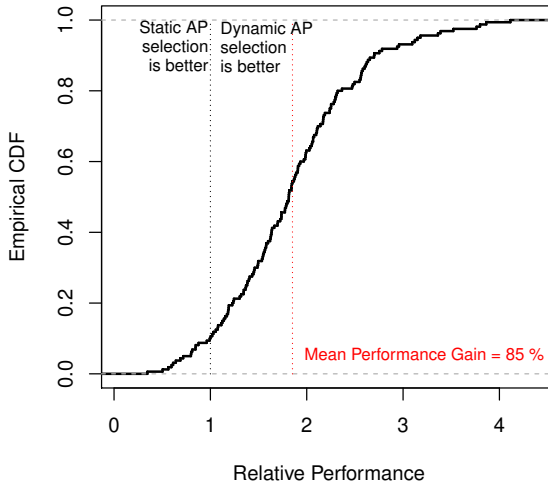


Figure 8.16: Performance under realistic external interference. Values greater than 1 mean that BEST-AP is beneficial. On average a 85% throughput increase is achieved with BEST-AP.

AP and the standard Linux mac80211 sub-system, which keeps using an AP once it is associated. We computed the relative performance as the ratio of BEST-AP’s throughput and the throughput obtained with a static association. A relative performance greater than one means that BEST-AP is better.

Figure 8.16 plots the CDF of the relative performance. BEST AP and the dynamic AP selection is beneficial in 90% of the cases with a mean improvement of 85% as compared to the standard Linux driver. When the load of one AP is high, the dynamic AP selection can lead to performance gains of up to 400%. Standard Linux may statically connect to an AP with high load, while BEST-AP dynamically selects the lightly loaded AP. In 10% of the tests standard Linux was better. In those cases, the load of both APs was about equally high and therefore switching the primary AP does not lead to any improvements, while estimating the bandwidth comes at a cost.

8.7 Conclusions

Estimating available bandwidth for wireless networks is important in order to implement services such as AP selection or Quality of Service management. However, such bandwidth estimation is difficult to achieve in WLANs since it depends on several factors such as the quality of the channel and the load created by competing traffic on that channel. In this chapter we have designed a novel bandwidth estimation mechanism for WLANs which is both accurate and responds fast to changes in available bandwidth. It combines passive measurements of available airtime with statistics of packet loss and delays using different time granularity. Using fast channel switching and handovers, the system is capable to estimate the available bandwidth to all surrounding APs. Based on our bandwidth estimation, we have designed a system which allows to better use the resources of multiple APs, even if they are on different channels. Experiments in a real testbed have shown significant performance gains compared to the standard association mechanism provided by Linux.

Currently, BEST-AP only estimates the AB for the wireless link between the station and the AP. In enterprise WLANs, the main target system of BEST-AP, APs are usually connected to the Internet with fast Ethernet links. The performance bottleneck should therefore be in the wireless link. However, in other networks, such as home networks connected via ADSL, the performance bottleneck might be in the wired backhaul. Even if BEST-AP would select an AP with an excellent wireless link, the performance could still be low. An interesting future extension of BEST-AP therefore could be to integrate bandwidth estimations of the wired backhaul. This could for example be achieved by constantly monitoring the wired backhaul with BART [76] and fusing the bandwidth estimations at the BEST-AP server.

A second important aspect not evaluated in this thesis is how BEST-AP performs if several stations optimize their throughput through dynamic AP selection. Since the current BEST-AP scheduler selects its primary AP only based on local information, the global network throughput and fairness may not be granted. One remedy for this problem would be to use the MESHMAX algorithms of Chapter 5 on the BEST-AP server to compute the primary AP for each station. This would lead to a fair and efficient distribution of resources.

Another interesting topic for future work is to use multiple WLAN cards simultaneously. Using multiple WLAN cards within one station can increase performance, but creates additional challenges. For example, the impact of Adjacent Channel Interference (ACI) is in particular severe if

several WLAN cards are used simultaneously in close proximity (see e.g. [69] and [35]). Therefore, when estimating the available bandwidth and when scheduling APs the impact of ACI needs to be considered. In the next chapter we move one step towards the use of multiple cards. We will evaluate what benefit a dedicated card for scanning provides in mobile scenarios. Scanning for new APs only requires little data transfer and hence the impact of ACI is negligible in this context.

Chapter 9

Mobile Video Streaming with BEST-AP

9.1 Introduction

In the previous chapter we have evaluated BEST-AP's ability to select the least loaded AP among several available ones in order to optimize the long-term throughput. In this chapter, we will discuss how to use BEST-AP to enable station mobility for optimizing the quality of streaming video. Streaming video for WLAN users is an interesting application for several reasons. First, streaming video gets more and more popular. In particular, live streaming video, in which pre-loading large parts of the video prior to playout is not possible, is more and more common. For example, video conferencing and IPTV applications use live streaming. Second, streaming video is a demanding application. It is both bandwidth intensive and delay critical. Thus, if a network can support live video streaming, it should also be suited for many other applications, which are typically less demanding.

Figure 9.1 illustrates how video players typically process data. Data packets are received from the network. Due to queuing effects and re-transmissions, the packets do not arrive in a constant flow, but might show variations in their packet inter-arrival time. This variation is also called *jitter*. The video decoder though requires a constant flow of packets. Therefore, typically a playout buffer is positioned between the network and the video decoder. The playout buffer does not forward incoming packets immediately to the video decoder, but stores them for a while. Thereby, it can remove the jitter in the packet inter-arrival times and enable a constant flow of packets to the video decoder. The video decoder decodes

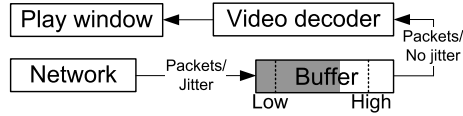


Figure 9.1: Simplified architecture of a typical video player. Video frames are coming from the network and the buffer removes jitter before the decoder decodes the frames for playback.

the incoming frames and sends them to the graphical user interface for display.

There is a tradeoff between the playout buffer size and the packet loss. If the playout buffer is too small, late packet arrivals cannot be sent to the decoder in time, and thus those packets are regarded as lost by the decoder. The result are dropped video frames and poor quality. If the buffer is too large, however, live streams and interactive video applications will experience a large end-to-end delay and therefore also poor quality. In particular with interactive applications such as video conferencing, for optimal quality the end-to-end latency preferably should not exceed 250 ms [14]. In extreme cases 500 ms delay still allow a two-way conversation. When taking into account other delays, such as coder delays and network latency, the buffer size should be even smaller.

When the video buffer level drops below a “low threshold”, the video decoder is stopped and no more new video frames are displayed to the user. Such an event is called “video freeze”. The video freeze lasts until the playout buffer is filled to the “high threshold” again. Often the low and the high threshold are set to 0% and 100% respectively.

Playing real-time video on a mobile wireless station is in particular challenging. Consider the example shown in Figure 9.2. The station is streaming a live video from a streaming server via AP1. At the same time, the station is moving away from AP1 towards AP2. At some point, the station needs to perform a handover, since it leaves the coverage area of AP1. During a handover no data can be transferred. Therefore, only the data in the video playout buffer can be used to sustain the playout during the handover.

With standard IEEE 802.11 WLANs, the video is likely to freeze in such a situation. Figure 9.3 illustrates why. In the beginning, the station is close to the AP and the video buffer level is high (1). When the station moves away from AP1, the link quality starts to deteriorate and the number of incoming frames is smaller than the number of played frames. Thus the video buffer level starts to decrease (2). When moving away even further,

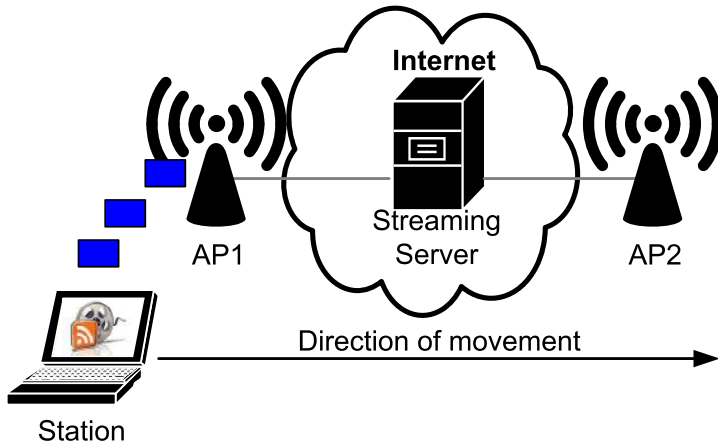


Figure 9.2: Example scenario: the station is streaming a live video and moving from AP1 towards AP2. The station needs to perform a handover when it moves too far away from AP1.

the link quality gets so bad that the connection completely breaks. Since no more video frames are received, the buffer level decreases even faster now (3). The WLAN driver of the station starts to scan for new APs (4). Meanwhile, the buffer level has reached the low threshold and the video freezes (5). After a while the WLAN driver has detected AP2 and connects to it (6). Video frames are received again and when the buffer level exceeds the high threshold, the video playback resumes (7).

The duration between (5) and (7) denotes the video freeze time. Since video freezes are annoying to the user, the video freeze time should be minimized.

9.1.1 Related Work

In literature there are two orthogonal approaches to optimize video streaming for mobile WLAN users. The first approach attempts to optimize the network itself. This includes acceleration of the scan procedure, optimization of the handovers and intelligent selection of APs. The second approach is to adapt video coding parameters to the network and to schedule the transmission of video frames under consideration of user mobility.

The related work surveyed in Chapter 8 follows the first approach. In addition, [122] and [176] investigate how to optimize the scanning procedure for new APs. The key idea behind those works is to increase the speed of AP detection by only scanning a subset of all available channels or by

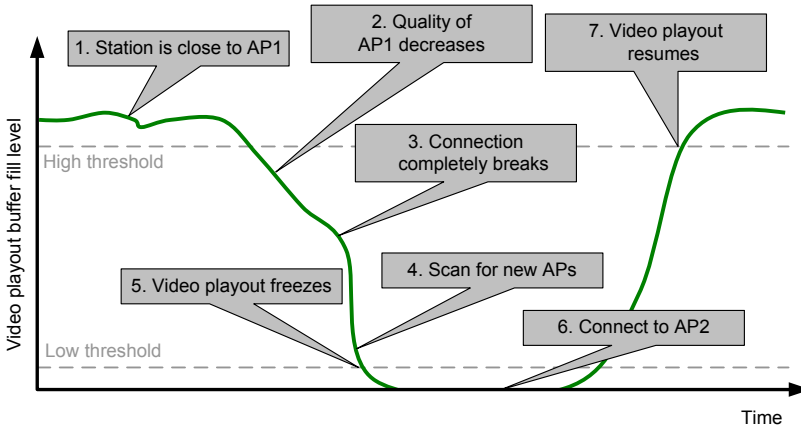


Figure 9.3: Evolution of video buffer level while the station is moving from AP1 towards AP2.

synchronizing the transmission of beacon frames. [130] proposes a method for scheduling the scanning procedure in WLANs, while considering the buffer level of the video player. Their evaluation is based on network simulations and the re-configuration of the wired backhaul network is not considered. [61] formulates and solves an optimization problem which computes the best moment for performing a handover, while taking into account the buffer fill level and the channel characteristics. The use of OpenFlow for improving streaming video has been explored in [179]. [179] requires multiple interfaces at the station and studies the concurrent use of different radio technologies (WLAN and WiMAX). [65] suggests a method to determine which WLAN AP is best for video streaming. The main idea is to stream the same video via two APs and to use the relative delay difference of the received RTP frames as indicator of congestion. The handover is performed to the AP with less congestion. [65] requires two WLAN interfaces. The two latter works do not describe when to scan for a new AP.

The second approach, to optimize the video transmission based on network characteristics, is for example suggested in [36]. The main idea behind this work is to use *Multiple Description* (MD) coding to generate two video streams and send the video via two network paths. The video player can decode each stream independently by using MD coding, but can provide higher quality when both streams are received. In the situation of a handover, the independent streams can be sent via the old and the new AP, respectively. The video coder then attempts to decode the video

in the highest possible quality for the received data. In contrast, [150] relies on a single-path transmission only and adapts the error correction parameters according to the quality of the used wireless connection.

9.1.2 Problem Statement and Contributions

In this chapter we evaluate how well BEST-AP supports mobility in general and how good the quality of streaming video in a BEST-AP WLAN with a mobile station is. We therefore follow the first approach for streaming video optimization, that is to optimize the network. The main difference between the discussed work and our approach is we only require one wireless card and that BEST-AP is based on SDN ideas.

The key contributions of this chapter are:

- **An extension of BEST-AP for mobile stations:** We discuss what changes to the initial BEST-AP system are required in order to support station mobility.
- **An evaluation of BEST-AP's ability to support real time video with mobile users:** Our testbed experiments show that BEST-AP enhances the quality of streaming video by reducing the video freeze probability. Compared to normal WLANs, BEST-AP requires smaller video playout buffers.

The remainder of the chapter is organized as follows: In Section 9.2 we describe important aspects when using BEST-AP with mobile stations. In Section 9.3 we present evaluation results. We wrap-up the chapter in Section 9.4.

9.2 Making BEST-AP Mobile

The BEST-AP architecture discussed in the previous chapter is in principle capable of handling mobile stations. By always selecting a close-by AP with low load as primary AP, the station is continuously connected to the network. The only major difference between the non-mobile and the mobile scenario is that in the mobile scenario the set of available APs changes as stations move.

Thus, in the mobile scenario, it is necessary to detect when a new AP is available (i.e. when it is in the detected state, see Section 8.4.4). Only when an AP is in the detected state, the BEST-AP scheduler can promote it to a secondary or primary AP. For this purpose, the scan module in the handover manager performs an active scanning procedure, i.e. it broadcasts

probe request frames. APs on the respective channel answer with probe response messages. The scan procedure can either be performed on the regular WLAN card, which is also used for data communication, or on a dedicated scanning card. If the regular WLAN card is used for scanning, a trade-off between the scan frequency and the achievable throughput and QoS emerges. A scan for new APs needs to be scheduled often enough to detect new APs fast enough. However, too frequent scanning would result in a reduction of throughput and QoS, as during scanning no data can be transferred. The optimal scan frequency also depends on the speed of the user. If the user does not move, only an initial scan is required as the available APs do not change later on.

A dedicated scanning card would avoid this trade-off, but increases hardware costs. Future mobile devices, such as smart phones and tablets, are presumable equipped with several flexible radios. With such devices, a dedicated scanning card would be a viable option.

A robust handover procedure is important in mobile scenarios. When the station is moving, it can happen that the scheduler would like to perform a handover to an AP, which is already out of reach, but has not yet timed out to the “Not detected” state. Therefore, in mobile scenarios the optional ACKs shown in Figure 8.8 should always be used. If no ACK is received upon a handover initiate message, the station can conclude that the AP is out of reach.

Using the PSM to temporarily buffer frames in the transmit buffer of the APs is often not useful in mobile scenarios, since the station might move away from an AP and may not handover to this AP again to request the buffered frames. Instead, in mobile networks the transmit buffer should be positioned at the OpenFlow switch. This allows to buffer downstream packets during a handover and forward them to the new AP after the handover has finished.

9.3 Evaluation

We have evaluated BEST-AP’s ability to support station mobility in the KAUMesh testbed (Section 9.3.1.1) and a WLAN testbed deployed in an office building in downtown Berlin (all other tests).

The testbed in Berlin (see Figure 9.4) consists of 8 embedded systems with a Dual Core Atom D525 1.80 GHz CPU, which act as APs. Each node is equipped with two wireless cards, of which only one is used in our experiments. The cards are based on the Atheros AR5418 IEEE 802.11abgn chip-set. To reduce interference from other WLANs deployed in the area,



Figure 9.4: Map of the testbed used for the video streaming tests. The mobile terminal moves from point A to point B and back.

the network is operated in the less occupied 5 GHz band. The APs are connected via Ethernet and a GRE tunnel to a virtual machine host, in which the OpenFlow switch, the streaming server and the control server are hosted. The OpenFlow switch uses OpenVSwitch. The KAUMesh testbed is configured as in Chapter 8.

9.3.1 Micro-Benchmarks

9.3.1.1 How well does BEST-AP Support Station Mobility?

Dynamic AP selection can also be used to support station mobility. In a normal IEEE 802.11 WLAN, stations associate to one AP (typically the one with the highest RSSI) and stay associated until the connection breaks. A break of a connection is usually detected when beacon frames are no more received and the AP does not respond to probe request frames. After the connection breaks, the station may initiate a scan for a new AP and then connect to it. Such a hard hand-off can lead to a considerable disruption and performance degradation. With BEST-AP a soft-handover is possible, in which the station is connected to two APs simultaneously and as the station moves towards one AP the role of the primary AP changes.

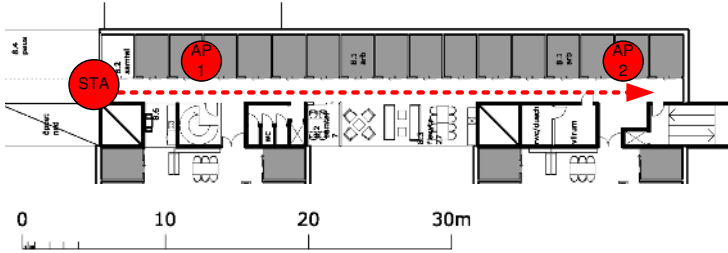


Figure 9.5: Test topology for mobility tests. The STA moves along the corridor from AP1 towards AP2.

We evaluated the ability of BEST-AP to support such type of mobility in the test setup shown in Figure 9.5. The aim of the experiment is to investigate what happens when the STA moves from AP1 to AP2 along the corridor of the Computer Science department facilities at Karlstad University. To have reproducible, automated and unbiased experiments, we recorded the signal strength during one walk along the corridor at the APs and the station. For the actual experiments, we placed the station in the middle between the APs and over time changed the transmission power at the APs and the station so that the reception strength matches the previously obtained trace.

We repeated this “simulated” walk 30 times with the default Linux wireless subsystem and the dynamic AP switching with BEST-AP. Figure 9.6 shows that the TCP throughput with BEST-AP is ca. 12 Mbit/s on average, while the standard Linux wireless sub-system only provides 4 Mbit/s. As exemplified in Figure 9.7, BEST-AP constantly monitors the available bandwidth and switches the primary AP as soon as the bandwidth estimation of the secondary AP exceeds the estimation of the currently primary AP. Thereby, the TCP throughput remains constantly high. In contrast, the default Linux wireless system stays connected with AP1 all the time. Even though AP2 would provide better quality, the station does not switch to this AP, as the connection does not break. This is because beacon and probe frames are usually sent at a low PHY rate, which is robust to errors and therefore has a large coverage range. Newer Linux kernels allow to configure Connection Quality Monitoring (CQM) and a hand-off hysteresis. With this approach a station can handover to a new AP, if the RSSI difference exceeds a user-configurable threshold (e.g. 20 dB). However, in our experiments CQM and the hand-off hysteresis did not seem to work properly and therefore did not have any impact on the results.

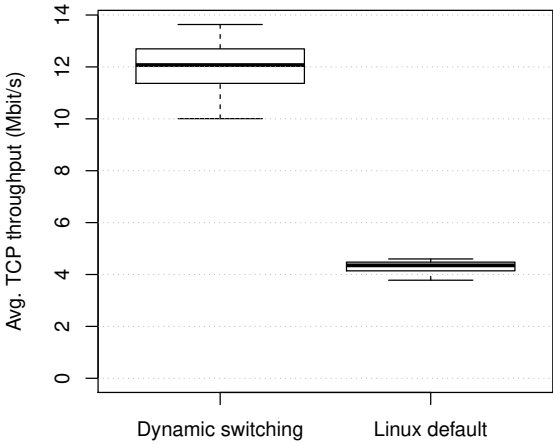


Figure 9.6: TCP throughput with station mobility. With dynamic switching the primary AP is chosen according to the available bandwidth estimate. Linux default uses the RSSI to select the best AP.

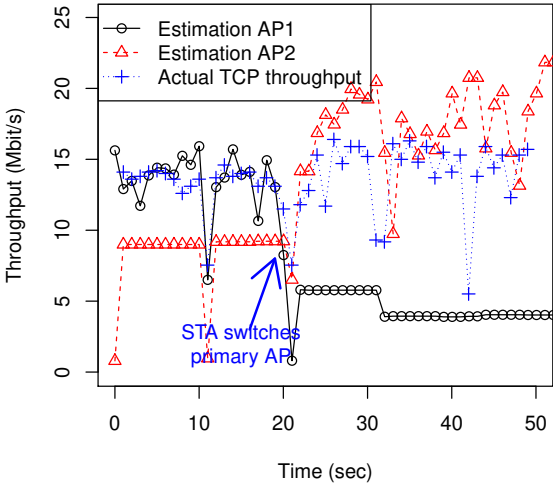


Figure 9.7: TCP throughput during one test run. After 20 seconds AP2 is better and hence selected as primary AP.

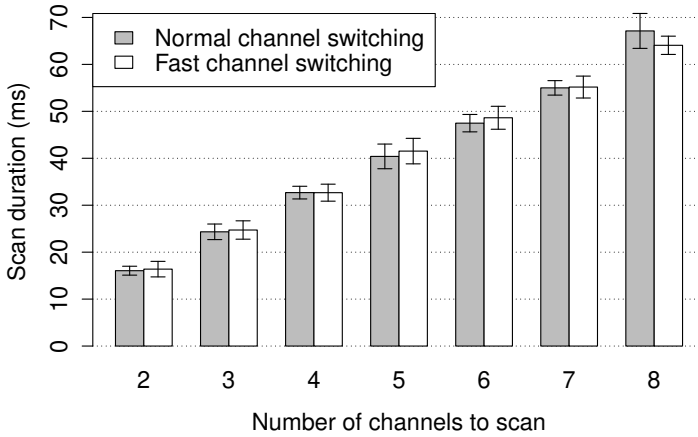


Figure 9.8: Duration of a scan for new APs. Each channel takes approximately 8 ms to be scanned.

9.3.1.2 How Long does it Take to Scan for new APs?

Even with BEST-AP, the mobile station needs to scan for new APs occasionally in order to detect new target APs for a handover. While scanning, the station needs to process a list of channels and send out probe request messages on each channel. After sending a probe request message, the station needs to remain on the channel for a while to wait for probe response messages. We remain on the channel for 5 ms, which turned out to be a good compromise between the probability of missing a probe response and the duration of the scan. In our experiments, switching from one channel to another channel requires approximately 1.2 ms if we empty the transmission WLAN queue prior to a switch and recalibrate the card after the switch (“Normal channel switching”). If we just tune the synthesizer to the new frequency (“Fast channel switching”), the channel switch takes approximately 1 ms. Figure 9.8 plots the duration for scanning 2-8 channels. The scan duration increases linearly with the number of channels to scan. Each extra channel adds about 8 ms (1.2 ms for switching, 5 ms for waiting and listening and 1.8 ms for processing the frames) to the total scan duration.

9.3.1.3 How Long does a Handover Take?

As discussed in Section 8.4.3, a handover consists of a series of messages to be exchanged. We call the time between the transmission of the handover initiate message and the reception of the ACK message “ACK duration”.

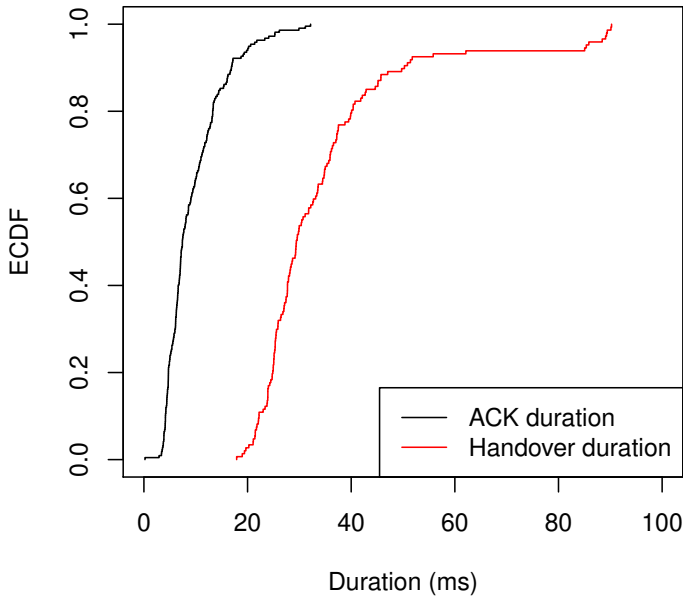


Figure 9.9: ECDF of handover and ACK duration. For a small fraction of tests signaling messages got lost and thus the handover duration increases by approximately 40 ms.

The total handover duration spans the period from the transmission of the handover initiate to the transmission of the handover complete message. The handover duration is very important for streaming video applications. The handover should be faster than video buffer size, since otherwise the video playout would freeze.

We measured how long our optimized handover takes during 100 handovers and plot the ECDF in Figure 9.9. The average ACK and handover durations are 9.4 ms and 34.7 ms respectively. For a small fraction, the handover duration exceeds 50 ms. In those cases one of the control messages (e.g. the handover initiate) got lost and needed to be retransmitted after a timeout. In contrast, [120] reports handover durations of 6-9 seconds for IEEE 802.11i secured WLANs. In Chapter 6 we have seen handover times of 2 seconds for non-optimized handovers in unsecured mesh networks. BEST-APs handover durations are much lower in comparison to those non-optimized handovers.

9.3.2 Video Streaming

For our experiments we streamed a pre-recorded MPEG-2 video using VLC [12] and HTTP-streaming to the client laptop. The video player on the client laptop is custom made and based on the GStreamer framework [5].

A person was walking with the laptop 5 times from point A to B and back (see Figure 9.4), while streaming the video. No artificial background load was generated in our tests, but other co-located WLANs used the same channels as our experiments. One walk from point A to B took approximately 90 seconds. The video player has a playout buffer of 400 ms and logs the buffer fill level every 20 ms to a file. If within 400 ms no new video frames are received, the video freezes until the buffer is filled up again. In other words, the low threshold is 0% and the high threshold is 100%. Since video freezes impair the perceived quality, they should be avoided.

In the following we compare 1.) Linux with `wpa_supplicant` 2.) Linux with `wpa_supplicant` and optimized scanning 3.) BEST-AP *with* a dedicated scanning card and RSSI-based AP selection and 4.) BEST-AP *without* a dedicated scanning card and RSSI-based AP selection and 5.) BEST-AP with AP selection based on the available bandwidth estimation. Options 3.) and 4.) are an intermediary step from options 1.)/2.) to the full BEST-AP system. Options 3.)/4.) benefit from the fast handover procedure of BEST-AP, but do not make an intelligent estimation of the available bandwidth.

The optimized scanning procedure in 2.) only scans on the four channels which are actually used by our network, while the normal scanning procedure of 1.) scans all channels. With BEST-AP and without a dedicated scanning card we scan for a new AP every 2 seconds. With a dedicated scanning card in theory a continuous scan for new APs would be possible. To keep the probe traffic low, we initiate a new scan only every 500 ms. To account for the higher dynamicity, we reduce the times on the primary and secondary AP compared to the previous chapter. In this chapter, the primary APs are used for 2 seconds and secondary APs for 100 ms.

With BEST-AP and bandwidth estimation, a secondary AP is promoted to primary AP as soon as it has a higher bandwidth estimate than the current primary AP. With RSSI-based AP selection, the primary AP is always the one with the currently highest RSSI (no hysteresis is used).

9.3.2.1 How Smooth is the Video Playout?

Table 9.1 shows that with the standard Linux system 28.40% and 12.52 % of the time the video is in freeze mode. With our BEST-AP, the video is

Mode	Freeze time fraction	Number of freeze events	Maximum freeze time (seconds)	Mean freeze time (seconds)
Best-AP/BW-Estimation/No scan card	3.31%	9	7.83	2.72
BEST-AP/RSSI/No scan card	0.73%	15	2.24	0.46
BEST-AP/RSSI/Scan card	3.25%	11	6.24	2.24
Linux/Optimized scanning	12.48%	55	8.43	2.01
Linux/Standard scanning	28.30%	79	32.06	2.75

Table 9.1: Video freeze events

frozen only 3.31%, 0.73% and 3.25% of the time respectively. BEST-AP's bandwidth estimation does not contribute to a more stable video playout. In contrast, the freeze time fraction with the bandwidth estimation is slightly higher than with just the RSSI-based AP selection. This is because the bandwidth estimation requires handovers to secondary APs, while with the RSSI-based estimation there is only a primary AP and no secondary APs, which are probed periodically. As a consequence, fewer handovers are performed with the RSSI-based estimation. Since each handover can fail and result in a video freeze, the RSSI-based solution is better here. Of course, in presence of significant background traffic, this picture may change. Since the RSSI-based AP selection does not consider the load on the AP, a handover to a fully loaded AP or channel could be triggered. The bandwidth estimation in BEST-AP would avoid such a decision.

Interestingly, with the dedicated scan card, the freeze time is slightly *higher* than without. This can mainly be attributed to two factors: first, with a dedicated scanning card updates on the APs quality are obtained more frequently and hence handovers happen more often. Such behavior could be avoided with a more intelligent handover scheduler. Second, in our experiments with the dedicated scanning card two relatively long freeze events occurred as result of failed handovers, which have large impact on

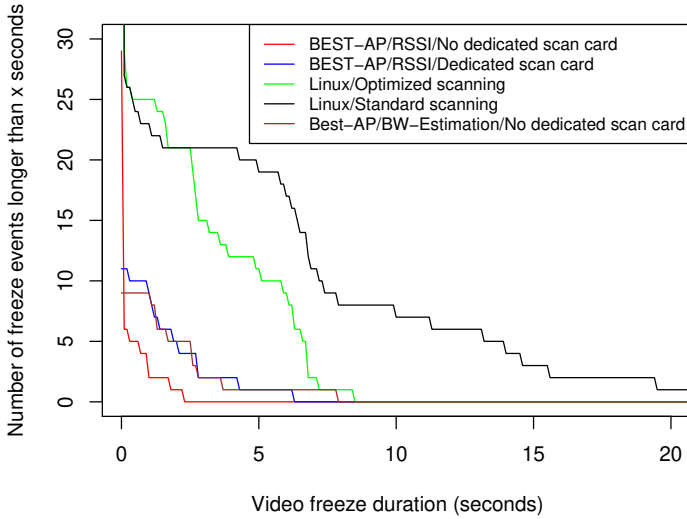


Figure 9.10: Distribution of freeze event durations. Standard systems have many and long freeze events while BEST-AP has few and short freeze events.

the average video freeze time. This phenomenon can also be observed from Figure 9.10, which shows the distribution of the freeze event durations. The figure further shows that the standard Linux implementation leads to many long freeze events, which have severe impact on the user perception. In particular if the channel scan is not optimized, freeze events last ten seconds or more. With BEST-AP however, the number of freeze events is small and the duration is usually short and thus not very disturbing to the viewer.

9.3.2.2 Is a Dedicated Scanning Card Necessary?

The total number of freeze events is lower if a dedicated scanning card is in operation, while the total length of freeze events is higher. With our optimized scanning procedure the benefits in terms of the total number of freeze events are marginal and also the subjective quality differences experienced during the measurements were low. On the one hand, a dedicated scanning card might decrease the system complexity, as the question when to schedule a new scan is easier to be answered then. On the other hand, a dedicated scanning card increases hardware cost and

energy consumption. Considering all those factors, a dedicated scanning card is only useful, if it allows to be integrated into the system with low effort, for example through software defined radios.

9.3.2.3 What is a Good Playout Buffer Size?

The playout buffer adds delay, which should be kept small, in particular for real-time applications such as video conferencing. Decreasing the playout buffer size would increase the number of freeze events, while increasing the buffer size would offer more protection against freeze events. Figure 9.10 shows, that in order to offer protection against all freeze events, the buffer size should be approximately 3-5 seconds large. Such large buffers are suitable for on-demand video, but the large delay would render real-time applications unusable. However, we remark, that even with the 400 ms buffer used in our experiments, the number of noticeable freeze events is relatively low and the perceived quality is good.

9.4 Conclusions

In this chapter we have evaluated BEST-AP with mobile video streaming. The evaluation has shown that BEST-AP, due to its optimized scanning procedure and fast handovers, has fewer video freezes than the standard Linux system. Moreover, if a freeze event occurs, it is shorter with BEST-AP than with the reference system. Altogether, BEST-AP therefore allows smaller video playout buffers and increases the quality of real-time video applications.

BEST-AP's architecture also allows to incorporate the video buffer level in the decision when to perform a handover. An interesting future extension of the work presented in this chapter hence would be to schedule the handovers according to an optimal control policy computed with [61]. A more extensive evaluation with background traffic and multiple users would provide insight in the benefit of using the bandwidth estimation instead of RSSI-based AP selection.

Conclusions

10.1 Achievements and Contributions

The fundamental question studied in this thesis is at what moment a STA should perform a handover to which AP and how this handover can be performed efficiently. The aim of this thesis is to provide a comprehensive view on this question. To this end, the thesis describes a journey from a MAC layer and channel model to the optimization of a specific application in a real network. With regard to the research questions formulated in Chapter 1, the thesis contributes to the state-of-the-art as follows:

1. *If a station is in the coverage area of several APs, when should the station use a given AP?*

We have answered this question from different perspectives. In Chapters 4 and 5 we have discussed mathematical optimization models that allow a central controller to compute which station should use which AP. It is relatively easy for a station to determine which is the best AP from its perspective, but such a decision is not optimal from a global perspective, both in terms of network efficiency and fairness. In addition, Chapter 4 showed that optimizing the network operation only based on the present network state is sub-optimal, in particular when each re-configuration of the network has a cost. Chapters 8 and 9 tackled the question from a more practical standpoint. The proposed method and system enables a station to efficiently find the best available AP using estimations of the available bandwidth, which are obtained in a non-intrusive way. The evaluated AP selection mechanism does not consider global objectives, but the system architecture is in principle suited to also include such

considerations. Moreover, it is possible to incorporate application requirements in the handover decision.

2. *How can the current, closed architecture of WLANs and wireless mesh networks be opened up and evolved to allow the easy deployment of new applications to enhance mobility support and resource distribution fairness?*

Chapters 6, 7 and 8 outlined new architectures, which use the Open-Flow protocol and ideas from the SDN framework to enhance the mobility support and resource distribution in WLANs and WMNs. The architectures can be deployed in today's networks with small modifications in the APs and stations. The architectures of Chapters 6 and 7 can be deployed in existing networks, in parallel to legacy protocols and architectures. This allows a step-wise migration from the legacy architecture to the proposed one, instead of a disruptive deployment of a completely new system.

3. *How to estimate the quality of a link between an AP and a station in a fast way and how to enable seamless handovers between APs?*

In Chapter 8 we have developed a new method to estimate the available bandwidth of a wireless link. The method achieves the accuracy of probe-based bandwidth estimation tools, but does not require any artificial probe traffic. Instead, the proposed method uses normal data traffic to obtain channel statistics. By using simultaneous connections to multiple APs and an SDN-enabled wired backhaul, stations can perform rapid handovers. In Chapter 9 we have demonstrated that the proposed system is capable of providing real-time video for mobile WLAN users in a higher quality than current systems.

10.2 Future Directions

Research is a process and not a finished product. As such, the results presented in this thesis should only be seen as a snapshot of this process and not as final comprehensive answers. Each question answered in this thesis has created new questions. Some of those questions may be minor and insignificant, but others may be worth to be investigated in the future.

In the last sections of Chapters 3-9 we have already highlighted limitations and open questions related to the respective chapter. In the following we take a step back and consider the thesis as a whole when asking for open issues and future work.

For the sake of simplicity, the models in this thesis approximate the achievable throughput as a linear function. This makes modeling easy, as a wide-range of standard tools exist for this class of systems. However, reality is not linear. A potential future work should therefore analyze what other, better approximations are possible (for example by convex functions) and study optimization problems inside such a more realistic framework. Centralized optimization is not always the best choice, in particular for large and dynamic systems. For this reason, an interesting question is how to distribute some of the proposed optimization algorithms and what scalability improvements could be gained.

Testbed evaluation is a first step to evaluate an idea under realistic conditions. As most other testbeds, also the testbeds used in this thesis are relatively small. Therefore scalability properties cannot be studied adequately. Thus a logical next step is to perform experiments in a larger testbed, potentially including real users. This would allow to understand how scalable the proposed ideas are and how the proposed optimizations behave with realistic workloads. Using the community testbeds developed in the CONFINE project [107] would be interesting in this context.

The system architectures presented in this thesis are based on the idea of SDNs. More specifically, we described different SDN control layers. The SDN application layer and its interface to the control layer (the so-called north-bound API) is a largely unexplored research field. An interesting open issue is therefore to design a WLAN mobility management application layer for SDNs and an appropriate north-bound API.

10.3 Final Remarks

In our journey from a MAC layer model to the optimization of a specific application in a real network we have developed a number of new algorithms and systems. In terms of important metrics, such as throughput, our proposals show better performance than the state-of-the-art. To make the improvements more comprehensible to non-scientist or engineers, several demonstrators have been developed. Those demonstrators are not part of this thesis, but short video clips presenting them can be found at [8].

Notational Conventions

Mathematical notation:

- Sets are denoted with capital letters, e.g. X .
- Elements of sets are, whenever possible, denoted with the respective small letter, e.g. x .
- Vectors are written in bold font, e.g. \mathbf{v} .
- A link between nodes i and j is denoted as (i, j) .
- Quantities related to a link, e.g. the flow f on a link (i, j) , is denoted with the name of the link in the subscript, i.e. $f_{(i,j)}$.
- The expression $(i, j) \in E \forall (i \in V)$ selects all edges in E , which start at node i and end at any other node.

The meaning of the respective symbols is summarized in Tables 3.1, 4.1, 5.1 and 8.1. Unless stated otherwise, for all plots the error bars show the standard deviation.

Bibliography

- [1] Chaska Wireless. <http://www.chaskamn.com/internet-solutions/>. Last accessed: 30.10.2012.
- [2] Cisco Prime Network Control System Series Appliances. <http://www.cisco.com/en/US/products/ps11686/index.html>. Last accessed: 30.10.2012.
- [3] Download Software - Cisco Systems. <http://www.cisco.com/cisco/software/>. Last accessed: 30.10.2012.
- [4] Floodlight OpenFlow Controller. <http://floodlight.openflowhub.org/>. Last accessed: 30.10.2012.
- [5] GStreamer: open source multimedia framework. <http://gstreamer.freedesktop.org/>. Last accessed: 30.10.2012.
- [6] IBM ILOG CPLEX. <http://www.ibm.com/software/integration/>. Last accessed: 30.10.2012.
- [7] iperf. <http://www.noc.ucf.edu/Tools/Iperf/>. Last accessed: 30.10.2012.
- [8] KAUMesh Youtube channel. <http://www.youtube.com/kaumesh>. Last accessed: 30.10.2012.
- [9] Linux Wireless. <http://linuxwireless.org/en/developers/>. Last accessed: 30.10.2012.
- [10] MadWifi project website. <http://madwifi-project.org/>. Last accessed: 30.10.2012.

- [11] Multi-Generator (MGEN). <http://cs.itd.nrl.navy.mil/work/mgen/>. Last accessed: 30.10.2012.
- [12] Official page for VLC media player, the Open Source video framework! <http://www.videolan.org/vlc/index.html>. Last accessed: 30.10.2012.
- [13] OpenWrt Website. <https://openwrt.org/>. Last accessed: 30.10.2012.
- [14] TIA TSB-116-A Telecommunications - IP Telephony Equipment - Voice Quality Recommendations for IP Telephony. <http://standardsdocuments.tiaonline.org/tia-tsb-116-a.htm>. Last accessed: 30.10.2012.
- [15] IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-1997*, pages i–445, 1997.
- [16] Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band. *IEEE Std 802.11a-1999*, page i, 1999.
- [17] Supplement to IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band. *IEEE Std 802.11b-1999*, pages i–90, 2000.
- [18] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Spectrum and Transmit Power Management Extensions in the 5 GHz Band in Europe. *IEEE Std 802.11h-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003))*, pages 1–59, 2003.
- [19] IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan

- Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, pages i –67, 2003.
- [20] ISO/IEC International Standard - Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements. *ISO/IEC 8802-11, Second edition: 2005/Amendment 6 2006: IEEE STD 802.11i-2004 (Amendment to IEEE Std 802.11-1999)*, pages c1 –178, 23 2004.
- [21] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS) Transition. *IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008)*, pages 1 –126, 15 2008.
- [22] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs. *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, pages 1 –244, 12 2008.
- [23] Draft STANDARD for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Wireless Network Management. *IEEE Unapproved Draft Std P802.11v/D7.0*, July 2009.
- [24] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE*

- Std 802.11y-2008, and IEEE Std 802.11w-2009*), pages 1 –565, 29 2009.
- [25] IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover. *IEEE Std 802.21-2008*, pages c1 –301, 21 2009.
 - [26] IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking. *IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11w-2009, IEEE 802.11n-2009, IEEE 802.11p-2010, IEEE 802.11z-2010, IEEE 802.11v-2011, and IEEE 802.11u-2011)*, pages 1 –372, 10 2011.
 - [27] IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1 –2793, 29 2012.
 - [28] Software-Defined Networking: The New Norm for Networks. Technical report, Open Networking Foundation, April 2012.
 - [29] Murad Abusubaih and Adam Wolisz. An optimal station association policy for multi-rate ieee 802.11 wireless lans. In *Proceedings of the ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM) 2007*, pages 117–123. ACM, October 2007.
 - [30] P.A.K. Acharya, A. Sharma, E.M. Belding, K.C. Almeroth, and K. Papagiannaki. Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach. In *Proceedings of the IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON) 2008*, pages 1–9, June 2008.
 - [31] N Ahmed and S Keshav. Smarta: A self-managing architecture for thin access points. In *Proceedings of ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT) 2006*, December 2006.

- [32] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., 1993.
- [33] Mansoor Alicherry, Randeep Bhatia, and Li (Erran) Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2005*, pages 58–72. ACM, August 2005.
- [34] Yair Amir, Claudiu Danilov, Michael Hilsdale, Raluca Musăloiu-Elefteri, and Nilo Rivera. Fast handoff for seamless wireless mesh networks. In *Proceedings of the International Conference on Mobile systems, Applications and Services (MobiSys) 2006*, pages 83–95. ACM, June 2006.
- [35] Angelakis, V. and Papadakis, S. and Siris, V.A. and Traganitis, A. Adjacent channel interference in 802.11 a is harmful: Testbed validation of a simple quantification model. *IEEE Communications Magazine*, 49(3):160–166, 2011.
- [36] J.G. Apostolopoulos and M.D. Trott. Path diversity for enhanced media streaming. *IEEE Communications Magazine*, 42(8):80–87, 2004.
- [37] G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas. Dynamic Cross-Layer Association in 802.11-Based Mesh Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2007*, pages 2090–2098. IEEE, May 2007.
- [38] P. Bahl, M.T. Hajiaghayi, K. Jain, S.V. Mirrokni, L. Qiu, and A. Saberi. Cell Breathing in Wireless LANs: Algorithms and Evaluation. *IEEE Transactions on Mobile Computing*, 6(2):164–178, February 2007.
- [39] Nicola Baldo, Federico Maguolo, Marco Miozzo, Michele Rossi, and Michele Zorzi. ns2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2. In *Proceedings of the International Conference on Performance Evaluation Methodologies and Tools (ValueTools) 2007*, pages 16:1–16:8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), October 2007.

- [40] A. Banchs, N. Bayer, David Chieng, A. de la Oliva, B. Gloss, M. Kretschme, S. Murphyk, M. Natkaniec, and F. Zdarsky. CAR-MEN: Delivering carrier grade services over wireless mesh networks. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2008*, pages 1–6, September 2008.
- [41] N. Bayer, D. Sivchenko, H.J. Einsiedler, A. Roos, A. Uzun, S. Gondor, and A. Kupper. Energy optimisation in heterogeneous multi-RAT networks. In *International Conference on Intelligence in Next Generation Networks (ICIN) 2011*, pages 139 – 144, October 2011.
- [42] Y. Bejerano, Seung-Jae Han, and Li Li. Fairness and Load Balancing in Wireless LANs Using Association Control. *IEEE/ACM Transactions on Networking*, 15(3):560 –573, June 2007.
- [43] Y. Bejerano, Seung-Jae Han, and Li Li. Fairness and Load Balancing in Wireless LANs Using Association Control. *IEEE/ACM Transactions on Networking*, 15(3):560 –573, June 2007.
- [44] M. Berkelaar, J. Dirks, K. Eikland, and P. Notebaert. lp-solve: A mixed integer linear programming (MILP) solver. <http://sourceforge.net/projects/lpsolve>. Last accessed: 30.10.2012.
- [45] Dimitri Bertsekas and Robert Gallager. *Data networks (2nd ed.)*. Prentice-Hall, Inc., 1992.
- [46] G Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.
- [47] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello. Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2007*, pages 1181–1189. IEEE, May 2007.
- [48] Mats Björkman and Bob Melander. Impact of the Ethernet Capture Effect on Bandwidth Measurements. In *Proceedings of the IFIP-TC6 / European Commission International Conference on Broadband Communications, High Performance Networking, and Performance of Communication Networks*. Springer-Verlag, May 2000.

- [49] T. Bonald, A. Ibrahim, and J. Roberts. The impact of association on the capacity of WLANs. In *Proceedings of the 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT) 2009*, pages 1 –10, June 2009.
- [50] O. Braham and G. Pujolle. Virtual wireless network urbanization. In *Proceedings of Network of the Future Conference (NOF) 2011*, pages 31 –34, November 2011.
- [51] Zheng Cai. *Maestro: Achieving Scalability and Coordination in Centralized Network Control Plane*. Phd thesis, Rice University, August 2011.
- [52] P. Calhoun, R. Suri, N. Cam Winget, M. Williams, S. Hares, B. O’Hara, and S. Kelly. Lightweight Access Point Protocol. RFC 5412 (Historic), February 2010.
- [53] M.E.M. Campista, P.M. Esposito, I.M. Moraes, L.H.M. 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*, 22(1):6 –12, Janury-February 2008.
- [54] M.M. Carvalho and J.J. Garcia-Luna-Aceves. Delay analysis of IEEE 802.11 in single-hop networks. In *Proceedings of IEEE International Conference on Network Protocols (ICNP) 2003*, pages 146 – 155, November 2003.
- [55] M. Casado, M.J. Freedman, J. Pettit, Jianying Luo, N. Gude, N. McKeown, and S. Shenker. Rethinking Enterprise Network Control. *IEEE/ACM Transactions on Networking*, 17(4):1270 –1283, August 2009.
- [56] R. Chandra, P. Bahl, and P. Bahl. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2004*. IEEE, March 2004.
- [57] Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra, and Paramvir Bahl. A case for adapting channel width in wireless networks. *SIGCOMM Comput. Commun. Rev.*, 38(4):135–146, August 2008.
- [58] P. Chatzimisios, A.C. Boucouvalas, and V. Vitsas. Performance analysis of IEEE 802.11 DCF in presence of transmission errors. In

- Proceedings of IEEE International Conference on Communications 2004*, volume 7, pages 3854 – 3858. IEEE, June 2004.
- [59] D.S. Chen, R.G. Batson, and Y. Dang. *Applied Integer Programming: Modeling and Solution*. John Wiley & Sons, 2010.
- [60] Gene Cheung, Jeongkeun Lee, Sung-Ju Lee, and P. Sharma. On the Complexity of System Throughput Derivation for Static 802.11 Networks. *IEEE Communications Letters*, 14(10):906 –908, October 2010.
- [61] Lawrence Chow, Bradley Collins, Nick Bambos, Nico Bayer, Hans Einsiedler, Christoph Peylo, Peter Dely, and Andreas Kasser. Playout-Buffer Aware Hand-Off Control for Wireless Video Streaming. In *Proceedings of IEEE Global Communications Conference (GLOBECOM) 2012*. IEEE, December 2012.
- [62] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.
- [63] The OpenFlow Consortium. OpenFlow Website. <http://www.openflowswitch.org/wp/downloads/>. Last accessed: 30.10.2012.
- [64] Jorge Crichigno, Min-You Wu, and Wei Shu. Protocols and architectures for channel assignment in wireless mesh networks. *Ad Hoc Networks*, 6(7):1051 – 1077, 2008.
- [65] G. Cunningham, S. Murphy, L. Murphy, and P. Perry. Seamless handover of streamed video over UDP between wireless LANs. In *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC) 2005*, pages 284–289. IEEE, January 2005.
- [66] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin. On the Linear Behaviour of the Throughput of IEEE 802.11 DCF in Non-Saturated Conditions. *IEEE Communications Letters*, 11(11):856 –858, November 2007.
- [67] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin. Unsaturated Throughput Analysis of IEEE 802.11 in Presence of Non Ideal Transmission Channel and Capture Effects. *IEEE Transactions on Wireless Communications*, 7(4):1276 –1286, April 2008.
- [68] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing.

- In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2003*, pages 134–146. ACM, September 2003.
- [69] P. Dely, M. Castro, S. Soukhakian, A. Moldsvor, and A. Kassler. Practical considerations for channel assignment in wireless mesh networks. In *Proceedings of IEEE Global Communications Conference (GLOBECOM) Workshops 2010*, pages 763–767, December 2010.
- [70] Peter Dely and Andreas Kassler. KAUMesh Demo. In *Proceedings of 9th Scandinavian Workshop on Wireless Ad-hoc and Sensor Networks*, May 2009.
- [71] Q. Dong, S. Banerjee, and B. Liu. Throughput Optimization and Fair Bandwidth Allocation in Multi-Hop Wireless LANs. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2006*, pages 1–12. IEEE, April 2006.
- [72] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810 (Proposed Standard), March 2010.
- [73] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2004*, pages 114–128, New York, NY, USA, 2004. ACM.
- [74] K. Duffy and A.J. Ganesh. Modeling the Impact of Buffering on 802.11. *IEEE Communications Letters*, 11(2):219–221, February 2007.
- [75] Ken Duffy, David Malone, and D.J. Leith. Modeling the 802.11 distributed coordination function in non-saturated conditions. *IEEE Communications Letters*, 9(8):715–717, 2005.
- [76] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman. Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering. In *IEEE/IFIP Network Operations and Management Symposium (NOMS) 2006*, pages 73–84, April 2006.
- [77] Huifang Feng, Yantai Shu, Shuyi Wang, and Maode Ma. SVM-Based Models for Predicting WLAN Traffic. In *Proceedings of IEEE*

- International Conference on Communications (ICC) 2006*, volume 2, pages 597–602. IEEE, June 2006.
- [78] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.
- [79] Ying Ge, T. Kunz, and L. Lamont. Quality of service routing in ad-hoc networks using OLSR. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, page 9 pp., January 2003.
- [80] A. Ghosh, R. Jana, V. Ramaswami, J. Rowland, and N.K. Shankaranarayanan. Modeling and characterization of large-scale Wi-Fi traffic in public hot-spots. In *Proceedings of 2011*, pages 2921–2929, April 2011.
- [81] Domenico Giustiniano, Eduard Goma, Alberto Lopez Toledo, Ian Dangerfield, Julian Morillo, and Pablo Rodriguez. Fair WLAN backhaul aggregation. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2010*, pages 269–280. ACM, September 2010.
- [82] S. Govindan, H. Cheng, ZH. Yao, WH. Zhou, and L. Yang. Objectives for Control and Provisioning of Wireless Access Points (CAPWAP). RFC 4564 (Informational), July 2006.
- [83] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. NOX: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38:105–110, July 2008.
- [84] D. Gupta, D. Wu, P. Mohapatra, and Chen-Nee Chuah. Experimental Comparison of Bandwidth Estimation Tools for Wireless Mesh Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2009*, pages 2891–2895, April 2009.
- [85] Z.J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Conference Record of IEEE International Conference on Universal Personal Communications Record (1997)*, volume 2, pages 562–566 vol.2, October 1997.

- [86] M.N. Halgamuge, K. Ramamohanarao, H.L. Vu, and M. Zukerman. Evaluation of handoff algorithms using a call quality measure with signal based penalties. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2006*, volume 1, pages 30 –35, April 2006.
- [87] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *SIGCOMM Comput. Commun. Rev.*, 41(4), August 2010.
- [88] Tsuyoshi Hamaguchi, Takuya Komata, Takahiro Nagai, and Hiroshi Shigeno. A Framework of Better Deployment for WLAN Access Point Using Virtualization Technique. In *Proceedings of IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA) 2010*, pages 968–973, April 2010.
- [89] Shaddi Hasan, Yahel Ben David, Robert Colin Scott, Eric Brewer, and Scott Shenker. Enabling Rural Connectivity with SDN. Technical Report UCB/EECS-2012-201, University of California at Berkeley, October 2012.
- [90] Brandon Heller, Rob Sherwood, and Nick McKeown. The Controller Placement Problem. In *Proceedings of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN) 2012*. ACM, August 2012.
- [91] TC Hu. The maximum capacity route problem. *Operations Research*, 9(6):898–900, 1961.
- [92] Jane-Hwa Huang, Li-Chun Wang, and Chung-Ju Chang. Coverage and capacity of a wireless mesh network. In *Proceedings of International Conference on Wireless Networks Communications and Mobile Computing 2005*, volume 1, pages 458 – 463, June 2005.
- [93] Te-Yuan Huang, Kok-Kiong Yap, Ben Dodson, Monica S. Lam, and Nick McKeown. PhoneNet: a phone-to-phone network for group communication within an administrative domain. In *Proceedings of the ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld) 2010*, pages 27–32. ACM, August 2010.
- [94] M. Jain and C. Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput.

- IEEE/ACM Transactions on Networking*, 11(4):537 – 549, August 2003.
- [95] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, 1991.
- [96] D. Johnson, N. Ntlatlapa, and C. Aichele. Simple pragmatic approach to mesh routing using BATMAN. In *Proceedings of IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries (WCIDT) 2008*, October 2008.
- [97] Jangeun Jun and M.L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, 10(5):8 – 14, October 2003.
- [98] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabo. FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI) 2008*, April 2008.
- [99] JongWon Kim, Huazhi Gong, and Kitae Nahm. Access Point Selection Tradeoff for IEEE 802.11 Wireless Mesh Network. In *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC) 2007*, pages 818–822. IEEE, January 2007.
- [100] Seongkwan Kim, Sunghyun Choi, Se kyu Park, J. Lee, and Sungmann Kim. An Empirical Measurements-based Analysis of Public WLAN Handoff Operations. In *Proceedings of International Conference on Communication System Software and Middleware (Comsware) 2006*, pages 1 –6, 0-0 2006.
- [101] Sung Won Kim, Byung-Seo Kim, and Yuguang Fang. Downlink and uplink resource allocation in IEEE 802.11 wireless LANs. *IEEE Transactions on Vehicular Technology*, 54(1):320 – 327, January 2005.
- [102] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2005*, pages 73–87. ACM, August 2005.

- [103] SV Krishnamurthy, Michalis Faloutsos, and V Mhatre. MDG: measurement-driven guidelines for 802.11 WLAN design. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2007*. ACM, September 2007.
- [104] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. Algorithmic aspects of capacity in wireless networks. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS) 2005*, pages 133–144. ACM, June 2005.
- [105] K. Lakshminarayanan, V. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement (IMC) 2004*. ACM, October 2004.
- [106] Nagios LCC. Nagios Documentation. <http://www.nagios.org/documentation/>. Last accessed: 30.10.2012.
- [107] Leandro Navarro, Pau Escrich Garcia, Axel Neumann. Community-Lab: Overview and Invitation to the Research Community. In *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P) 2012*. IEEE, September 2012.
- [108] Douglas J. Leith, Qizhi Cao, and Vijay G. Subramanian. Realising max-min fairness in 802.11e mesh networks. In *Proceedings of IEEE International Conference on Wireless Pervasive Computing (ISWPC) 2010*, pages 424–429. IEEE Press, May 2010.
- [109] M. Li, M. Claypool, and R. Kinicki. WBest: a Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks. In *Proceedings of IEEE Conference on Local Computer Networks (LCN) 2008*, October 2008.
- [110] Wei Li, Yong Cui, Shengling Wang, and Xiuzhen Cheng. Approximate Optimization for Proportional Fair AP Association in Multi-rate WLANs. In Gopal Pandurangan, V. Anil Kumar, Gu Ming, Yunhao Liu, and Yingshu Li, editors, *Wireless Algorithms, Systems, and Applications*, volume 6221 of *Lecture Notes in Computer Science*, pages 36–46. Springer Berlin / Heidelberg, 2010.
- [111] Zhenyu Li, Zhongzhao Zhang, and Lei Wang. A novel QoS routing scheme for MPLS traffic engineering. In *Proceedings of International Conference on Communication Technology Proceedings (ICCT) 2003*, volume 1, pages 474 – 477, April 2003.

- [112] Li Erran Li AND Z. Morley Mao AND Jennifer Rexford. Toward Software-Defined Cellular Networks. In *Proceedings of the European Workshop on Software Defined Networking*, October 2012.
- [113] Chi-Chun Lo and Ming-Hua Lin. QoS provisioning in handoff algorithms for wireless LAN. In *Proceedings of International Zurich Seminar on Broadband Communication*, pages 9 –16, February 1998.
- [114] M. Luglio, C. Roseti, G. Savone, and F. Zampognaro. TCP performance on a railway satellite channel. In *Proceedings of International Workshop on Satellite and Space Communications (IWSSC) 2009*, pages 434 –438, September 2009.
- [115] Lin Luo, D. Raychaudhuri, Hang Liu, Mingquan Wu, and Dekai Li. Improving End-to-End Performance of Wireless Mesh Networks through Smart Association. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2008*, pages 2087 –2092, April 2008.
- [116] Lin Luo, Dipankar Raychaudhuri, H Liu, and M Wu. Joint Association, Routing and Bandwidth Allocation for Wireless Mesh Networks. In *Proceedings of IEEE Global Communications Conference (GLOBECOM) 2008*. IEEE, December 2008.
- [117] Juho Määttä and Timo Bräysy. A novel approach to fair routing in wireless mesh networks. *EURASIP Journal on Wireless Communication Networks*, 2009:5:1–5:13, January 2009.
- [118] David Malone, Ken Duffy, and Doug Leith. Modeling the 802.11 Distributed Coordination Function in Nonsaturated Heterogeneous Conditions. *IEEE/ACM Transactions on Networking*, 15(1):159–172, February 2007.
- [119] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 2004.
- [120] Ivan Martinovic, Frank A. Zdarsky, Adam Bachorek, and Jens B. Schmitt. Measurement and Analysis of Handover Latencies in IEEE 802.11i Secured Networks. In *Proceedings of the European Wireless Conference (EW) 2007*, April 2007.
- [121] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan

- Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38:69–74, March 2008.
- [122] A. Mishra, M. Shin, and W.A. Arbaush. Context caching using neighbor graphs for fast handoffs in a wireless network. In *Proceedings of IEEE 2004*, volume 1, pages xxv+2866, March 2004.
- [123] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *SIGCOMM Comput. Commun. Rev.*, 33:93–102, April 2003.
- [124] H. Mittelman. Benchmarks for optimization software. <http://plato.la.asu.edu/bench.html>. Last accessed: 30.10.2012.
- [125] G.E. Moore et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.
- [126] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A.W. Moore, A. Koliouisis, and J. Sventek. Control and understanding: Owning your home network. In *Proceedings of International Conference on COMMunication Systems and NETWORKS (COMSNETS) 2012*, pages 1–10, January 2012.
- [127] Rohan Murty, Jitendra Padhye, Ranveer Chandra, and A. Designing high performance enterprise wi-fi networks. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation 2008*, April 2008.
- [128] D. Nace and M. Pioro. Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial. *Communications Surveys Tutorials, IEEE*, 10(4):5–17, quarter 2008.
- [129] T. Nadeau and P. Pan. Software Driven Networks Problem Statement (Internet-Draft), 2011.
- [130] Jae-Wook Nah, Sung-Man Chun, Song Wang, and Jong-Tae Park. Adaptive handover method with application-awareness for multimedia streaming service in wireless LAN. In *Proceedings of International Conference on Information Networkin (ICOIN) 2009*, pages 1–7, January 2009.
- [131] Wlodzimierz Ogryczak, Michal Pioro, and Artur Tomaszewski. Telecommunications network design and max-min optimization problems. *Journal of Telecommunications and Information Technology*, 4:43–, 2005.

- [132] M. Peinhardt and V. Kaibel. On the Bottleneck Shortest Path Problem. Technical Report ZIB-Report 06-22, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2006.
- [133] C. Perkins. IP Mobility Support for IPv4. RFC 5944, November 2010.
- [134] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.
- [135] Ben Pfaff, Justin Pettit, Teemu Koponen, Keith Amidon, Martin Casado, and Scott Shenker. Extending networking into the virtualization layer. In *Proceedings of ACM Workshop on Hot Topics in Networks (HotNets) 2009*. ACM, October 2009.
- [136] Michal Pioro, Mateusz Zotkiewicz, Barbara Staehle, Dirk Staehle, and Di Yuan. On Max-Min Fair Flow Optimization in Wireless Mesh Networks. *Ad Hoc Networks, Special Issue on Models and Algorithms for Wireless Mesh Networks*, 2011.
- [137] Lili Qiu, Yin Zhang, Feng Wang, Mi Kyung Han, and Ratul Mahajan. A general model of wireless interference. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2007*, pages 171–182. ACM, September 2007.
- [138] A. Qureshi and J. Gutttag. Horde: Separating Network Striping Policy from Mechanism. In *Proceedings of the International conference on Mobile systems, applications and services (MobiSys) 2005*. ACM, June 2005.
- [139] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
- [140] Bozidar Radunovic and Le BouDecember Jean-Yves. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 15(5):1073–1083, 2007.
- [141] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2006*, pages 1 –12. IEEE, April 2006.

- [142] A. Raniwala, D. Pradipta, and S. Sharma. End-to-End Flow Fairness Over IEEE 802.11-Based Wireless Mesh Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2007*, pages 2361 –2365. IEEE, May 2007.
- [143] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [144] A.R. Rebai, M.F. Rebai, H.M. Alnuweiri, and S. Hanafi. An enhanced heuristic technique for AP selection in 802.11 handoff procedure. In *Proceedings of IEEE 17th International Conference on Telecommunications (ICT) 2010*, pages 576 –580, April 2010.
- [145] David Reed. *A balanced introduction to computer science*. Pearson Prentice Hall, 2008.
- [146] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Proceedings of Passive and Active Measurement (PAM) 2003*, April 2003.
- [147] A. Riedl. Optimized routing adaptation in IP networks utilizing OSPF and MPLS. In *Proceedings of IEEE International Conference on Communications (ICC) 2003*, volume 3, pages 1754 – 1758. IEEE, May 2003.
- [148] S. Robitzsch, L. Murphy, and J. Fitzpatrick. An analysis of the Received Signal Strength accuracy in 802.11a networks using Atheros chipsets: A solution towards self configuration. In *Proceedings of IEEE Global Communications Conference (GLOBECOM) Workshops 2012*, pages 1429 –1434. IEEE, December 2011.
- [149] T. Salonidis, M. Garetto, A. Saha, and E. Knightly. Identifying High Throughput Paths in 802.11 Mesh Networks: a Model-based Approach. In *Proceedings of IEEE International Conference on Network Protocols (ICNP) 2007*, pages 21–30, October 2007.
- [150] A. Schorr, A. Kassler, and G. Petrovic. Adaptive media streaming in heterogeneous wireless networks. In *IEEE Workshop on Multimedia Signal Processing 2004*, pages 506–509. IEEE, September 2004.
- [151] Aaron Schulman, Dave Levin, and Neil Spring. CRAWDAD data set umd/sigcomm2008 (v. 2009-03-02). Downloaded from <http://crawdad.cs.dartmouth.edu/umd/sigcomm2008>, March 2009.

- [152] Mahboobeh Sedighizad, Babak Seyfe, and Keivan Navaie. MR-BART: Multi-Rate Available Bandwidth Estimation in Real-Time. *Journal of Network and Computer Applications*, 35(2):731 – 742, 2012.
- [153] C.E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.
- [154] Irfan Sheriff, Prashanth Aravinda Kumar Acharya, and Elizabeth M. Belding. Measurement-driven admission control on wireless backhaul networks. *Comput. Commun.*, 31(7):1354–1371, 2008.
- [155] Rob Sherwood, Michael Chan, Adam Covington, Glen Gibb, Mario Flajslik, Nikhil Handigol, Te-Yuan Huang, Peyman Kazemian, Masayoshi Kobayashi, Jad Naous, Srinivasan Seetharaman, David Underhill, Tatsuya Yabe, Kok-Kiong Yap, Yiannis Yiakoumis, Hongyi Zeng, Guido Appenzeller, Ramesh Johari, Nick McKeown, and Guru Parulkar. Carving research slices out of your production networks with OpenFlow. *SIGCOMM Comput. Commun. Rev.*, 40:129–130, January 2010.
- [156] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. CENTAUR: realizing the full potential of centralized wlans through a hybrid data path. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2009*, pages 297–308. ACM, September 2009.
- [157] V.A. Siris, G. Stamatakis, and E. Tragos. A Simple End-to-End Throughput Model for 802.11 Multi-Radio Multi-Rate Wireless Mesh Networks. *IEEE Communications Letters*, 15(6):635–637, 2011.
- [158] Vasilios A. Siris. Optimization of Dense WLANs: Models and Algorithms. Technical Report Technical Report 390, Foundation for Research and Technology - Hellas (FORTH), May 2007.
- [159] Libo Song, D. Kotz, Ravi Jain, and Xiaoning He. Evaluating Next-Cell Predictors with Extensive Wi-Fi Mobility Data. *IEEE Transactions on Mobile Computing*, 5(12):1633 –1649, December 2006.
- [160] S. Speicher. OLSR-FastSync: Fast Post-Handoff Route Discovery in Wireless Mesh Networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall) 2006*, pages 1 –5, September 2006.

- [161] D. Staehle, B. Staehle, and R. Pries. Max-Min Fair Throughput in Multi-Gateway Multi-Rate Mesh Networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Spring) 2010*, pages 1–5, May 2010.
- [162] A.P. Subramanian, M.M. Buddhikot, and S. Miller. Interference aware routing in multi-radio wireless mesh networks. In *IEEE Workshop on Wireless Mesh Networks (WiMesh) 2006*, pages 55–63, September 2006.
- [163] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao. Towards Programmable Enterprise WLANs with Odin. In *Proceedings of the ACM SIGCOMM workshop on Hot topics in Software Defined Networks (HotSDN) 2012*, pages 49–54, August 2012.
- [164] Jian Tang, R. Hincapie, Guoliang Xue, Weiyi Zhang, and R. Bustamante. Fair Bandwidth Allocation in Wireless Mesh Networks With Cognitive Radios. *IEEE Transactions on Vehicular Technology*, 59(3):1487–1496, March 2010.
- [165] The OpenFlow Consortium. OpenFlow Switch Specification 1.1, 2012.
- [166] N. Thompson, G. He, and H. Luo. Flow Scheduling for end-Host Multihoming. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2006*. IEEE, April 2006.
- [167] A. Tønnesen, A. Hafslund, and Ø. Kure. The UniK-OLSR plugin library. In *The OLSR Interop and Workshop*, August 2004.
- [168] Amin Tootoonchian and Yashar Ganjali. HyperFlow: a distributed control plane for OpenFlow. In *Proceedings of the Internet Network Management Conference on Research on Enterprise Networking (INM/WREN) 2010*, pages 3–3. USENIX Association, April 2010.
- [169] E.Z. Tragos, R. Bruno, E. Ancillotti, K. Grochla, and V.A. Siris. Automatically configured, optimised and QoS aware wireless mesh networks. In *Proceedings of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC) 2010*, pages 2081–2086. IEEE, September 2010.
- [170] R. Van Nee. Breaking the Gigabit-per-second barrier with 802.11AC. *Wireless Communications, IEEE*, 18(2):4, April 2011.

- [171] Timo Vanhatupa, Marko Hännikäinen, and Timo D. Hämäläinen. Performance model for IEEE 802.11s wireless mesh network deployment design. *J. Parallel Distrib. Comput.*, 68:291–305, March 2008.
- [172] Arunchandar Vasan, R. Ramjee, and T. Woo. Echos - enhanced capacity 802.11 hotspots. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2005*, pages 1562–1572. IEEE, March 2005.
- [173] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley. Facilitating access point selection in IEEE 802.11 wireless networks. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement (IMC) 2005*, pages 26–26. USENIX Association, October 2005.
- [174] Steven J. Vaughan-Nichols. Gigabit Wi-Fi Is on Its Way. *Computer*, 43(11):11–14, November 2010.
- [175] B.H. Walke, S. Mangold, and L. Berlemann. *IEEE 802 wireless systems*. John Wiley and Sons West Sussex, England, 2006.
- [176] Haitao Wu, Kun Tan, Yongguang Zhang, and Qian Zhang. Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2007*, pages 749–757. IEEE, May 2007.
- [177] Yang Xiao. Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs. *IEEE Transactions on Wireless Communications*, 4(4):1506–1515, July 2005.
- [178] Fei Xin and Abbas Jamalipour. TCP throughput and fairness performance in presence of delay spikes in wireless networks. *International Journal of Communication Systems*, 18(4):395–407, 2005.
- [179] Kok-Kiong Yap, Te-Yuan Huang, Masayoshi Kobayashi, Michael Chan, Rob Sherwood, Guru Parulkar, and Nick McKeow. Demo: Lossless Handover with n-casting between WiFi-WiMAX on Open-Roads. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom) 2009*. ACM, September 2009.
- [180] Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown. Open-Roads: empowering research in mobile networks. *SIGCOMM Comput. Commun. Rev.*, 40:125–126, January 2010.

- [181] Yiannis Yiakoumis, Kok-Kiong Yap, Sachin Katti, Guru Parulkar, and Nick McKeown. Slicing home networks. In *Proceedings of the ACM SIGCOMM workshop on Home networks (HomeNets) 2011*, pages 1–6, August 2011.
- [182] Jun Yin, Xiaodong Wang, and D.P. Agrawal. Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2004*, volume 3, pages 1654–1659, March 2004.
- [183] Hongqiang Zhai, Xiang Chen, and Yuguang Fang. How well can the IEEE 802.11 wireless LAN support quality of service? *IEEE Transactions on Wireless Communications*, 4(6):3084–3094, November 2005.
- [184] Yu Zheng, Kejie Lu, Dapeng Wu, and Yuguang Fang. Performance Analysis of IEEE 802.11 DCF in Imperfect Channels. *IEEE Transactions on Vehicular Technology*, 55(5):1648 –1656, September 2006.
- [185] Chenxi Zhu and M.S. Corson. QoS routing for mobile ad hoc networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM) 2002*, volume 2, pages 958 – 967 vol.2, June 2002.
- [186] Yanfeng Zhu, Qian Ma, C. Bisdikian, and Chun Ying. User-Centric Management of Wireless LANs. *IEEE Transactions on Network and Service Management*, 8(3):165 –175, September 2011.

Abbreviations and Acronyms

AB	Available Bandwidth
AC	Access Category
ACK	Acknowledgment
AIFS	Arbitration Interframe Space
AODV	Adhoc On-Demand Distance Vector Routing Protocol
AOF	Aggregate Objective Function
AP	Access Point
ASIC	Application-Specific Integrated Circuit
BIP	Binary Integer Program
BPSK	Binary Phase Shift Keying
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
CAPEX	Capital Expenditures
CCA	Channel Clear Assessment
CCK	Complementary Code Keying
CFP	Contention Free Period
CP	Contention Period

CSMA/CA	Carrier Sensing Multiple Access Collision Avoidance
CTS	Clear to Send
CW	Contention Window
DCF	Distributed Coordination Function
DFS	Dynamic Frequency Selection
DIFS	Distributed Coordination Function Interframe Space
DLC	Data Link Control
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
DTIM	Delivery Traffic Indication Map
EDCA	Enhanced Distributed Channel Access
ESS	Extended Service Set
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FHSS	Frequency Hopping Spread Spectrum
ForCES	Forwarding and Control Element Separation
HCCA	HCF Controlled Access
HCF	Hybrid Coordination Function
IBSS	Independent Basic Service Set
ICI	Inter Carrier Interference
IEEE	Institute of Electrical and Electronics Engineers
IP	Integer Program
IR	Infrared
ISM	Industrial, Scientific and Medical
LLC	Logical Link Control

LP	Linear Program
MAC	Medium Access Control
MAP	Mesh Access Point
MCS	Modulation and Coding Scheme
MCS	Monitoring and Control Server
MD	Multiple Description
MIHF	Media Independent Handover Function
MILP	Mixed Integer Linear Program
MIMO	Multiple Input Multiple Output
MLR	Multiple Linear Regression
MP	Mesh Point
MPP	Mesh Portal
MPR	Multi-Point Relay
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
NOC	Network Operations Center
OCS	OpenFlow Control Server
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing Protocol
ONF	Open Networking Foundation
OPEX	Operational Expenditures
PC	Point Coordinator
PCF	Point Coordination Function
PHY	Physical Layer
PIFS	Point Coordination Function Interframe Space

PLCP	Physical Layer Convergence Protocol
PMD	Physical Medium Dependent Sublayer
PMK	Pairwise Master Key
PoA	Point of Attachment
PTK	Pairwise Transient Key
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RREP	Route REPLY
RREQ	Route REQuest
RTS	Request to Send
SAP	Service Access Point
SDN	Software Defined Networking
SIFS	Short Interframe Space
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
SSID	Service Set Identifier
STA	Station
TC	Topology Control
TPC	Transmit Power Control
TXOP	Transmission Opportunity
U-NII	Unlicensed National Information Infrastructure
WMAN	Wireless Metropolitan Area Network
WMN	Wireless Mesh Network
WPA2	Wi-Fi Protected Access II
WPAN	Wireless Personal Area Network



Architectures and Algorithms for Future Wireless Local Area Networks

Future Wireless Local Area Networks (WLANs) with high carrier frequencies and wide channels need a dense deployment of Access Points (APs) to provide good performance. In densely deployed WLANs associations of stations and handovers need to be managed more intelligently than today.

This dissertation studies when and how a station should perform a handover and to which AP from a theoretical and a practical perspective. We formulate and solve optimization problems that allow to compute the optimal AP for each station in normal WLANs and WLANs connected via a wireless mesh backhaul. Moreover, we propose to use software defined networks and the OpenFlow protocol to optimize station associations, handovers and traffic rates. Furthermore, we develop new mechanisms to estimate the quality of a link between a station and an AP. Those mechanisms allow optimization algorithms to make better decisions about when to initiate a handover. Since handovers in today's WLANs are slow and may disturb real-time applications such as video streaming, a faster procedure is developed in this thesis.

Evaluation results from wireless testbeds and network simulations show that our architectures and algorithms significantly increase the performance of WLANs, while they are backward compatible at the same time.

ISBN 978-91-7063-464-2

ISSN 1403-8099

DISSERTATION | Karlstad University Studies | 2012:53