

Advanced Protocols for Peer-to-Peer Data Transmission in Wireless Gigabit Networks

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet
INFORMATIK

Vorgelegt

von Dipl.-Inf. Jan Friedrich

geboren am 24.02.1985 in Lutherstadt Wittenberg

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr.-Ing. Christoph Lindemann, Universität Leipzig
2. Prof. Dr. Stefan Fischer, Universität zu Lübeck

Die Verleihung des akademischen Grades erfolgt mit Bestehen
der Verteidigung am 21.08.2020 mit dem Gesamtprädikat summa cum laude.

Acknowledgments

First, I would like to thank my supervisor Prof. Dr.-Ing. Christoph Lindemann for his guidance and advice. His input was crucial to identify important scientific issues, to select the most promising approaches and to present solutions in the most comprehensible way.

I would also like to thank all my co-authors and colleagues for their input, support and the occasional distractions: Michael Petrifke, Stephan Bergemann, Sascha Gübner and Sebastian Günther.

Finally, thank you very much Jana, Birgit and Klaus.

Abstract

This thesis tackles problems on IEEE 802.11 MAC layer, network layer and application layer, to further push the performance of wireless P2P applications in a holistic way. It contributes to the better understanding and utilization of two major IEEE 802.11 MAC features, frame aggregation and block acknowledgement, to the design and implementation of opportunistic networks on off-the-shelf hardware and proposes a document exchange protocol, including document recommendation.

First, this thesis contributes a measurement study of the A-MPDU frame aggregation behavior of IEEE 802.11n in a real-world, multi-hop, indoor mesh testbed. Furthermore, this thesis presents MPDU payload adaptation (MPA) to utilize A-MPDU subframes to increase the overall throughput under bad channel conditions. MPA adapts the size of MAC protocol data units to channel conditions, to increase the throughput and lower the delay in error-prone channels. The results suggest that under erroneous conditions throughput can be maximized by limiting the MPDU size.

As second major contribution, this thesis introduces Neighborhood-aware OPPortunistic networking on Smartphones (NOPPoS). NOPPoS creates an opportunistic, pocket-switched network using current generation, off-the-shelf mobile devices. As main novel feature, NOPPoS is highly responsive to node mobility due to periodic, low-energy scans of its environment, using Bluetooth Low Energy advertisements.

The last major contribution is the Neighborhood Document Sharing (NDS) protocol. NDS enables users to discover and retrieve arbitrary documents shared by other users in their proximity, i.e. in the communication range of their IEEE 802.11 interface. However, IEEE 802.11 connections are only used on-demand during file transfers and indexing of files in the proximity of the user. Simulations show that NDS interconnects over 90 % of all devices in communication range.

Finally, NDS is extended by the content recommendation system User Preference-based Probability Spreading (UPPS), a graph-based approach. It integrates user-item scoring into a graph-based tag-aware item recommender system. UPPS utilizes novel formulas for affinity and similarity scoring, taking into account user-item preference in the mass diffusion of the recommender system. The presented results show that UPPS is a significant improvement to previous approaches.

Contents

1	Introduction	13
1.1	Motivation	13
1.2	Contributions	15
1.3	Publications	16
1.4	Thesis Outline	17
2	Background	19
2.1	IEEE 802.11	19
2.2	Frame Aggregation and Block Acknowledgment	21
2.3	Further IEEE 802.11ac Improvements	24
2.4	Wi-Fi Direct	27
2.5	Bluetooth Low Energy	28
2.6	Apple Wireless Direct Link	30
3	MAC-Layer Optimization	32
3.1	Understanding IEEE 802.11n Multi-Hop Communication	32
3.1.1	Related Work	33
3.1.2	Indoor MIMO Mesh Testbed	34
3.1.3	Enabling Multi-hop Communication in IEEE 802.11n	36
3.1.4	Experimental Setup	37
3.1.5	Measurements in Multi-hop IEEE 802.11n	38
3.1.6	Conclusion	43
3.2	MPDU Payload Adaptation	43
3.2.1	Related Work	44
3.2.2	Error-Prone Channels	46
3.2.3	Simulator Setup	47
3.2.4	Quantitative Performance Results	48

3.2.5	Conclusion	54
4	Neighborhood-Aware Opportunistic Networking on Smartphones	58
4.1	Related Work	59
4.2	The NOPPoS Protocol	61
4.2.1	Wi-Fi Opportunistic Networking	61
4.2.2	States and State Transitions	62
4.2.3	Access Point Creation	64
4.2.4	Access Point Switch	65
4.2.5	Access Point Shutdown	66
4.2.6	Slotted State Machine	67
4.3	Accurate Neighborhood Discovery	67
4.3.1	Bluetooth Low Energy	67
4.3.2	Communication Range Evaluation	68
4.4	Simulation-Based Evaluation	69
4.4.1	Detailed Short-Term Simulation with OMNeT++	69
4.4.2	Long-Term Simulation	70
4.4.3	Simulator Validation	71
4.5	Comparative Performance Study	72
4.5.1	Performance Study Scenario	72
4.5.2	Node Density Sensitivity Analysis	75
4.5.3	Mobility Sensitivity Analysis	79
4.5.4	Energy Consumption	79
4.6	Conclusion	82
5	Applications for Opportunistic Gigabit Networks	83
5.1	Neighborhood Document Sharing	83
5.1.1	Related Work	84
5.1.2	The Basic Approach	84
5.1.3	Performance Study	88
5.1.4	Off-the-Shelf Hardware Implementation	90
5.1.5	Improvements	91
5.1.6	Conclusion	91

Contents

5.2	User Preference-Based Probability Spreading	92
5.2.1	Related Work	93
5.2.2	Background	94
5.2.3	User Preference-Based Probability Spreading	96
5.2.4	Performance Evaluation	99
5.2.5	Conclusion	107
6	Conclusion and Future Work	110
6.1	Conclusion	110
6.2	Future Work	111

List of Figures

2.1	IEEE 802.11 network topologies	20
2.2	IEEE 802.11 compressed BA frame	21
2.3	IEEE 802.11 HT-immediate acknowledgement scheme	22
2.4	IEEE 802.11 A-MSDU structure	23
2.5	IEEE 802.11 A-MPDU structure	23
2.6	Spatial division multiplexing	25
2.7	Wi-Fi Direct phases	28
2.8	Bluetooth LE topology	30
2.9	AWDL activity window and channel mapping	31
3.1	Indoor MIMO mesh testbed	35
3.2	8-hop chain topology	36
3.3	MCS classes in a 8-hop communication	39
3.4	Throughput vs. number of hops	40
3.5	Single-stream throughput vs. number of hops	40
3.6	Mean aggregate size vs. number of hops	41
3.7	Mean aggregate size at each node	41
3.8	CDF of aggregate size for different path lengths	42
3.9	Throughput vs. number of hops for different maximum aggregate sizes	42
3.10	The wireless network scenario considered in the simulation model	48
3.11	The wireless network scenario considered in the fairness experiments	48
3.12	Goodput for a selection of MCS indices (600 byte MPDUs)	50
3.13	Goodput for different frame sizes under saturation	51
3.14	Goodput with increasing frame size at fixed distances	52
3.15	Utilization of different aggregation and acknowledgment window sizes	53
3.16	Goodput of different aggregation and acknowledgment window sizes	54
3.17	SRD under different BER conditions	55

List of Figures

3.18	ALD under different BER conditions	55
3.19	SRD while increasing offered load	56
3.20	SRD increase from non-saturated to saturated traffic	56
4.1	NOPPoS clique handling	62
4.2	NOPPoS state diagram	64
4.3	Utilization ratio of WLAN-Opp and NOPPoS	74
4.4	WLAN-Opp and NOPPoS state transitions in Huggle	74
4.5	WLAN-Opp and NOPPoS time per state in Huggle	75
4.6	Comparison of the average time in STA state	76
4.7	Comparison of the average AP transitions per hour	77
4.8	Comparison of the average number of empty AP per hour	77
4.9	Comparison of the average STA transitions per hour	78
4.10	Comparison of the average time in AP state	78
4.11	Comparison of the average group size	79
4.12	The number of AP transitions for various rest times	80
4.13	The time spent in STA state for various rest times	80
4.14	The time spent in AP state for various rest times	81
5.1	NDS roles	85
5.2	NDS example neighborhood with Index and User devices	86
5.3	NDS Index Utility example	88
5.4	Example illustrating the user-item graph of UPPS	97
5.5	Number of items per user in original and cleaned MovieLens dataset . .	101
5.6	Number of users per item in original and cleaned MovieLens dataset . .	101
5.7	Precision-Recall	106
5.8	NDCG@k	107
5.9	Impact of the parameter α on NDCG@10	108
5.10	Impact of the parameter α on NDCG@20	108

List of Tables

2.1	Summary of IEEE 802.11 development and features.	26
3.1	Testbed overview	36
4.1	Variables of the proposed protocol	65
4.2	OMNeT++ vs. OPSIM simulation results	72
4.3	Average group size in Huggle trace	73
4.4	Wireless energy consumption of QCA6234	81
5.1	NDS simulation based on the Huggle trace	89
5.2	NDS simulation using the random trip mobility model	90
5.3	The MovieLens dataset	100
5.4	Performance metrics for different versions of the algorithm	103
5.5	User-item association mapping	104
5.6	Performance metrics for different user preference granularity	104
5.7	Performance metrics for combinations with similarity	106

1 Introduction

1.1 Motivation

Efficient and stable wireless peer-to-peer (P2P) networks have always been the Holy Grail of wireless network research [50]. In contrast to wired networks and infrastructure-mode IEEE 802.11 networks, wireless P2P networks do not depend on the deployment of a planned infrastructure. Users in a confined area are able to communicate directly with each other on demand. Thus, wireless P2P systems differ completely from the original definition of P2P applications in wired networks [70].

With the emergence of the IEEE 802.11 standard, wireless networks have experienced a wide-spread deployment, drawing substantial interest in academia and industry. That is because IEEE 802.11 mesh networks can be built with relatively low infrastructure expenditure compared to wired broadband. Therefore, they are particularly attractive for providing fast and cost-efficient coverage for hard-to-wire areas. However, wireless P2P networks usually do not compete with IP-based wired networks or infrastructure-mode IEEE 802.11 networks, as they will most likely always achieve significantly lower goodput. For example, handling shared medium access and interference from other sources is just significantly more difficult in the wireless medium. Therefore, for practical reasons, a wired backbone for global communication will be required in the foreseeable future. However, there are use cases for special P2P applications at the edge of the network, built upon local communication, with less strict goodput and QoS requirements.

There are two kinds of wireless P2P networks: opportunistic, delay-tolerant networks [53] and mobile ad hoc networks [16]. Opportunistic networks do not require a path between two network devices to successfully exchange data. Rather, caching and mobility of the devices are utilized to forward data to the destination. The delivery is not guaranteed, and this network type is most suited for applications where the recep-

tion of a part of the data or a very delayed reception is sufficient. Good examples are information retrieval systems, where even in the wired example of global web search, not all of the possible results are displayed to the user. It is often sufficient to get at least some results.

In mobile ad hoc networks, there always has to be a path between network devices to transmit data. Here, routing is the main challenge and mobility is considered a problem, not a feature. Through mobility, paths break constantly, and new routes have to be discovered and established. This network type is most often used for sensor networks, in areas where it is a problem to deploy a wired infrastructure or for only temporarily used networks. Examples are networks on a building site, at festivals and open-air concerts.

The performance of wireless P2P networks and the utility of applications implemented in these networks can be optimized by improving all network layers, individually or in a cross-layer approach. First, there is the performance of the wireless link between two network devices: e.g. bandwidth and delay. Technological advances e.g. in the IEEE 802.11 protocol family, focus on this problem: how do the participants most efficiently communicate with each other in a fair way without sabotaging each other's transmission.

Then, there is research on how to utilize the IEEE 802.11 network links between devices to transmit data between arbitrary devices, not necessarily within communication range of each other. This is the logical network point of view, not the physical one. How to make devices known to each other, how to calculate routes between devices, when to transmit and to whom.

Finally, there is research on P2P applications. This not only includes the adaptation and evaluation of existing applications for wireless P2P networks, but also the development of completely new P2P applications. Mobile ad hoc networks and opportunistic networks create problems for applications designed with fixed and guaranteed wired connections in mind, on the one hand, but also enable completely new applications, on the other hand. Especially, the field of edge computing is of new interest. Edge computing utilizes the resources of mobile and IoT devices at the edge of the network and also require those devices to communicate directly. Just recently, Heck, Edinger, Schäfer and Becker [38] summarized the state of art in this field of science.

1.2 Contributions

This thesis tackles problems on all above mentioned fields of research to further push the performance of wireless P2P applications in a holistic way. It contributes to the better understanding and utilization of two major IEEE 802.11 MAC features, frame aggregation and block acknowledgement, to the design and implementation of opportunistic networks on off-the-shelf hardware and proposes a document sharing protocol including document recommendation.

MAC-Layer Optimization of Data Transmissions in Opportunistic Gigabit Networks

First, this thesis contributes a measurement study of the A-MPDU frame aggregation behavior of IEEE 802.11n in a real-world, multi-hop, indoor mesh testbed. The presented performance curves reveal that channel bonding nearly doubles the throughput for any fixed path length. The mean aggregate size in number of frames at each node is also doubled by channel bonding and the mean aggregate size in number of frames at each node decreases with increasing path length.

Furthermore, it contributes an approach to utilize A-MPDU subframes to increase the overall throughput under bad channel conditions. The approach, MPDU payload adaptation (MPA), adapts the size of MAC protocol data units (MPDU) to channel conditions, to increase the throughput and to lower the delay in error-prone channels. The focus is especially on the edge of the network, where even the lowest physical data rates exhibit such a high bit error rate (BER) that the probability for a successful transmission of typically sized MPDUs is very low. The results suggest that under erroneous conditions throughput can be maximized by limiting the MPDU size.

Network-Layer Optimization of Data Transmission in Opportunistic Gigabit Networks

As second major contribution, this thesis introduces and analyzes the Neighborhood-aware OPPortunistic networking on Smartphones (NOPPoS) protocol. NOPPoS creates an opportunistic, pocket switched network using current generation, off-the-shelf mobile devices and outperforms WLAN-OPP [72]. NOPPoS utilizes IEEE 802.11 ac-

cess points of mobile devices to create local, isolated networks that connect co-located mobile devices. NOPPoS assigns IEEE 802.11 station and access point roles to mobile devices based on the number of mobile devices and access points in the proximity. As main novel feature, NOPPoS is highly responsive to node mobility due to periodic, low-energy scans of its environment, using Bluetooth Low Energy advertisements. In fact, NOPPoS can determine the exact number of neighbors at any instant of time.

Applications for Opportunistic Gigabit Networks

The last major contribution is a cross-layer protocol that tightly couples an opportunistic network with a document retrieval application. The protocol, Neighborhood Document Sharing (NDS), enables users to discover and retrieve arbitrary documents shared by other users in their proximity, i.e. in the communication range of their IEEE 802.11 interface. However, IEEE 802.11 connections are only used on-demand during file transfers and indexing of files in the proximity of the user. This saves energy and limits the use of the IEEE 802.11 interface to high-throughput operations. Similar to NOPPoS, Bluetooth LE is employed to broadcast meta data to nearby devices. Simulations show that the protocol interconnects over 90 % of all devices.

Finally, NDS is extended by the content recommendation system User Preference-based Probability Spreading (UPPS), a graph-based approach. It integrates user-item scoring into a graph-based tag-aware item recommender system. Building upon the Probs [84] and PLIERS [9], [8] methods, UPPS utilizes refined formulas for affinity and similarity scoring, taking into account user-item preference in the mass diffusion of the recommender system. The presented results show that UPPS is a significant improvement to PLIERS and Probs.

1.3 Publications

This thesis is based on the following publications that were co-authored with Christoph Lindemann. His input was crucial to identify important scientific issues, to select the most promising approaches and to present solutions in the most comprehensible way. He further contributed to the completion of this thesis by guidance and advice.

Additionally, the paper [27] was co-authored with Sascha Gübner, who focused in his PhD thesis [33] on data dissemination in multi-hop wireless networks, designed the

experiments and headed the evaluation. The paper [13] was co-authored with Stephan Bergemann, who implemented one of the simulators. Michael Petrifke co-authored the paper [29] and proposed the two-step similarity score. Finally, Sebastian Günther [28] performed the experiments utilized in [28] to develop MPDU payload adaptation.

- The IEEE 802.11n frame aggregation study in Section 3.1 was published in [27].
- The IEEE 802.11ax MPDU payload adaptation approach in Section 3.2 was published in [28].
- The mobile ad hoc network protocol NOPPoS in Section 4 was published in [13].
- The graph-based recommendation algorithm in Section 5.2 was published in [29].

1.4 Thesis Outline

The remainder of this thesis is organized as follow. Chapter 2 reviews the features of IEEE 802.11 and related protocols that are crucial for understanding the proposed wireless P2P protocols and applications.

In the third chapter, the block acknowledgement feature of IEEE 802.11n, IEEE 802.11ac and IEEE 802.11ax is analyzed in the context of multi-hop wireless networks. With the help of the discovered behavior in the real world testbed and the ns-3 simulator, an approach is developed which improves the throughput on sub-optimal wireless links exhibiting a high bit-error rate. By artificially limiting the size of MAC protocol data units (MPDUs) transmitted on error-prone channels, block acknowledgements (BA) and aggregated MPDUs (A-MPDUs) are better utilized to reduce the number of retransmitted frames. ns-3 simulations show a significant increase in throughput and verify the idea.

Chapter 4 describes the protocol Neighborhood-Aware Opportunistic Networking on Smartphones which creates an opportunistic network on off-the-shelf smartphones. While the infrastructure mode of IEEE 802.11 is used to interconnect adjacent smartphones and to provide the opportunistic network for IP-based data transfer, a second radio technology, Bluetooth Low Energy, provides the means to discover peers and to transmit status information. The utilization of Bluetooth Low Energy is instrumental in minimizing energy consumption and providing each smartphone with a detailed knowledge of their proximity.

Chapter 5 introduces Neighborhood Document Sharing (NDS), a novel wireless P2P application, which significantly benefits from the improvements developed in Chapter 3 and Chapter 4. NDS utilizes two radios, similarly to NOPPoS, to discover and manually retrieve interesting documents from adjacent wireless devices. User Preference-Based Probability Spreading for Tag-Aware Content Recommendation builds on top of graph-based recommendation systems and adds on automatic selection of documents to Neighborhood Document Sharing. Finally, Chapter 6 summarizes all results.

2 Background

Wireless network technologies enable some of the most important applications of our time. Ubiquitous internet access on smartphones via cellular networks is used by billions of people at any time to access news, to communicate, to watch videos or for gaming. LTE-Advanced and the just deployed 5G networks provide up to 3 Gbps per cell. This is enough bandwidth to support thousands of concurrent video or audio streaming sessions in low quality. However, at home or at the workplace, many of the day-to-day network usage scenarios could not be offloaded to cellular 5G networks, e.g. network file access or 4K video streaming. Just a handful of users would exhaust all bandwidth available in a single cell. Nevertheless, wired Gigabit connections are also not an option as mobility is also of importance in the just mentioned setups. That is where IEEE 802.11 (Wi-Fi) fills the gap between wired networks and cellular networks.

2.1 IEEE 802.11

Modern IEEE 802.11 networks [59] provide a bandwidth that is comparable to what is available per 5G cell. The smallest network is called a basic service set (BSS) and consists of an access point (AP) and a number of clients, called stations (STA), which are connected to this access point. The IEEE 802.11 stations are the actual users of this wireless network and want to communicate, i.e. exchange data, between each other or with some network device external to this basic service set. The role of the access point is to manage access to this network (authentication and association), control network parameters (e.g. transmission frequency, i.e. the channel), power management, encryption and much more. In a BSS, each STA-to-STA transmission is tunneled through the AP. The sender first transmits its data to the AP, which then sends the data to the real destination STA.

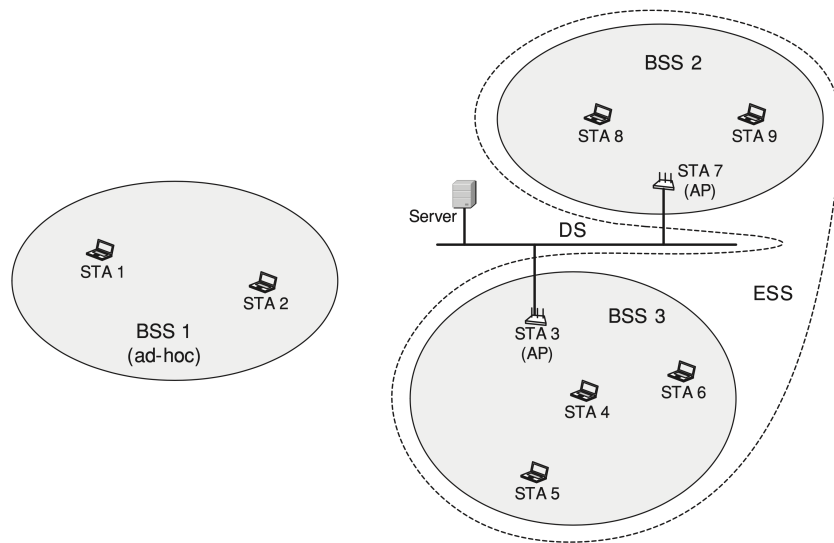


Figure 2.1: IEEE 802.11 network topologies (from [59])

An arbitrary wired network, which is called distribution system (DS) in the IEEE 802.11 standard, is used to reach network devices in other basic service sets. This network of basic service sets, connected via a DS, is called an extended service set (ESS). Finally, gateways provide access to other networks. They are also connected to the DS.

IEEE 802.11 further provides multiple different network types: e.g. ad hoc networks implemented as independent basic service set (IBSS), where stations communicate directly with each other without the help of an AP, or mesh networks, first standardized in IEEE 802.11s. While IBSS only specifies what is necessary to exchange data on layer-2 level, the mesh extension also includes authentication, routing and encryption. Unfortunately, both network types are not available on off-the-shelf smartphones running either iOS or Android. And while some Android devices can be modified to enable ad hoc network support, a P2P application that is intended to be available for everybody cannot be built on top of IBSS or mesh networks. Fortunately, there is another network protocol that enables P2P networking on off-the-shelf hardware: Wi-Fi Direct.

Octets	2	2	6	6	2	2	8 or 128	4
	Frame Control	Duration	RA	TA	BA Control	Starting Sequence Control	Block Ack Bitmap	FCS

Figure 2.2: IEEE 802.11 compressed BA frame (from [59])

2.2 Frame Aggregation and Block Acknowledgment

IEEE 802.11 utilizes layer-2 acknowledgements to notify the transmitter of MAC frames of the successful reception of data at the receiver. This is in contrast to most wired networks, e.g. IEEE 802.3 Ethernet, where the probability of transmission errors is very low, and collisions can be actively detected. In the wireless medium, especially in the unregulated ISM band (Industrial, Scientific and Medical), there is no isolation against external interference and multiple different radio protocols freely utilize the same frequencies. Moreover, there is no direct collision detection available and the bit error rate is usually much higher compared to wired networks.

Early IEEE 802.11 versions acknowledged each received frame instantly. This resulted in a high overhead because a significant time the channel was not available for data transmissions but occupied by acknowledgements. To improve the efficiency, block acknowledgements (BA) were introduced in IEEE 802.11e to acknowledge multiple MAC frames with only one transmission. Each BA frame contains a bitmap and the start sequence number, i.e. the sequence number that maps to the first bit in the bitmap (Figure 2.2). If a bit in the bitmap is set to 1 it tells the receiver of this BA that the MAC frame with the corresponding sequence number was received correctly. The transmission of the BA is delayed until a block acknowledgement request frame (BAR) is received. Now, either a BA is sent immediately (immediate BA) or whenever the device thinks it is a good time to send the BA (delayed BA). The development of the two-level frame aggregation approach, first introduced in IEEE 802.11n, redefined significant parts of the BA mechanism and mostly HT-immediate BA is relevant as of today and the default behavior.

Two-level frame aggregation was also introduced to improve MAC efficiency. In short, multiple MAC service data units (MSDU), the packets that are provided from the layer above IEEE 802.11 MAC, e.g. logical link control (LLC), are aggregated

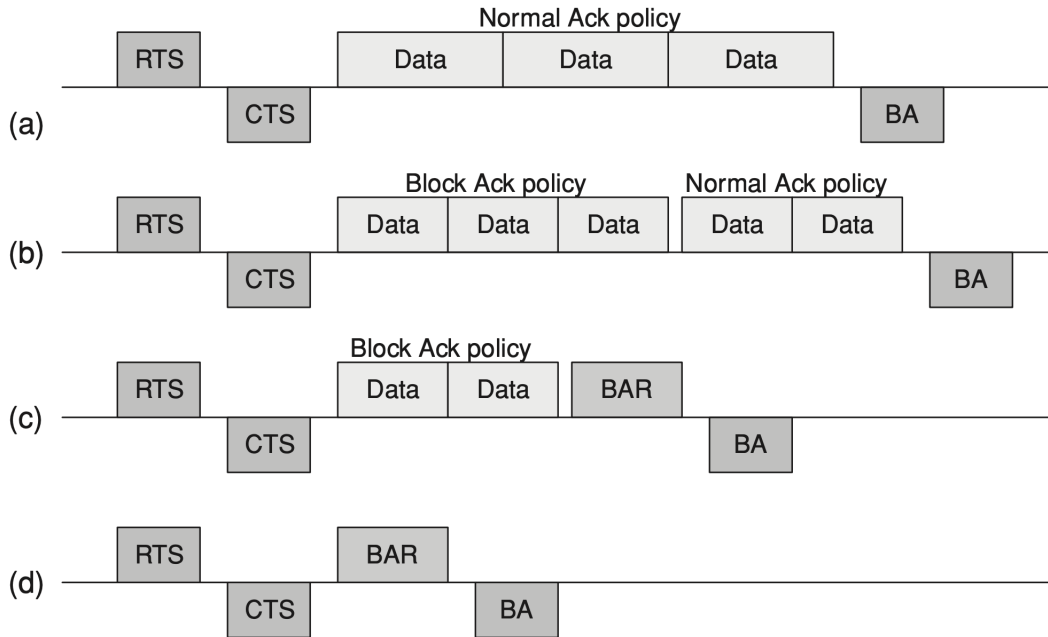


Figure 2.3: IEEE 802.11 HT-immediate acknowledgement scheme (from [59])

in two steps to form large physical protocol data units (PPDU). Thus, each data transmission contains significantly more payload than before. This aggregation is divided into two steps: first multiple MSDUs form a larger aggregated MSDU (A-MSDU; Figure 2.4). A-MSDUs constitute the payload of MAC protocol data units (MPDU). Secondly, MPDUs are aggregated to form an aggregated MPDU (A-MPDU; Figure 2.5). It is important to note, that each MPDU contains its own frame control sequence (FCS), i.e. checksum, while an A-MSDU does not.

Thus, IEEE 802.11 first aggregates each usually about 1500 bytes large MSDU (typical MTU of Ethernet networks) into larger A-MSDUs, however limits the size of each A-MSDU because of the relatively high BER of the wireless channel. Any significant transmission error that results in the loss of at least one MPDU, and thus one A-MSDU, means all the MSDUs wrapped in the A-MSDU are lost. However, the A-MSDUs in other correctly received MPDUs of the A-MPDU are not affected. In summary, MSDU aggregation increases the number of bytes per MPDU, i.e. per FCS, and MPDU aggregation increases the size of each PPDU till the maximum allocated air time in the transmission opportunity (TXOP) is reached.

Figure 2.3 illustrates how frame aggregation and block acknowledgements comple-

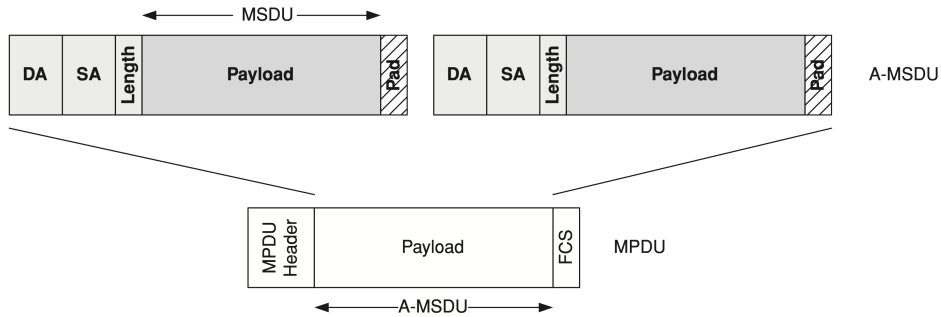


Figure 2.4: IEEE 802.11 A-MSDU structure (from [59])

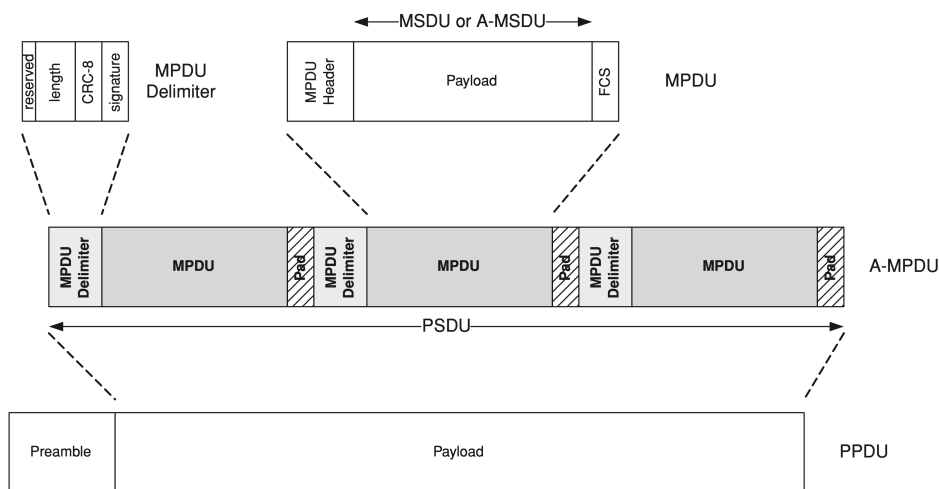


Figure 2.5: IEEE 802.11 A-MPDU structure (from [59])

ment each other. First, request-to-send (RTS) and clear-to-send (CTS) packages are exchanged to reserve the channel (TXOP) and to detect collisions early. The transmitted aggregated data frames have the Normal Ack policy bit set and the receiver immediately responds with a BA frame. Alternatively, if the Block Ack policy bit is set, the receiver delays the transmission of the block acknowledgement till a BAR is received or another data frame with the Normal Ack policy bit.

The transmit opportunity (TXOP), mentioned above, is a detail of the channel access protocol enhanced distributed channel access (EDCA). EDCA was first introduced in IEEE 802.11e and is a mandatory part of IEEE 802.11 since IEEE 802.11n. The main idea behind EDCA is to add quality of service (QoS) features to IEEE 802.11. EDCA distinguishes four access categories (ACs) and adds heuristics, so that

a station that has higher priority frames to transmit also is more likely to successfully contend for the channel. In EDCA, the period of exclusive access to the channel is called TXOP. During this time, the station can transmit as many frames it wants and also receive acknowledgements from the receiver. In contrast to the old distributed coordination function (DCF), EDCA results in a fair access to the channel: in the end, while contending for the channel, each station can use the channel exclusively for about the same amount of time.

2.3 Further IEEE 802.11ac Improvements

The above mentioned MAC layer overhead reductions were accompanied by a number of physical layer enhancements in the last iterations of IEEE 802.11. The following new features result in an increase of the achievable physical data rate of up to 9.6 Gbps (IEEE 802.11ax, eight spatial streams, 160 MHz channel).

First, IEEE 802.11n introduced multiple-input, multiple output (MIMO), based on spatial division multiplexing (SDM). SDM is a transmission method based on spatial streams, a new utilization method of multiple antennas on the transmitter as well as on the receiver. Previously, each IEEE 802.11 communication pair transmitted exactly one radio signal. Multiple antennas were only used at the receiver for maximal-ratio combining (MRC). With MRC, two slightly different received signals at two or more receiver antennas are combined to improve the robustness of the transmission by increasing the signal-to-noise ratio. SDM however, makes use of the fact that the signals transmitted by different antennas are propagated differently in the environment (see Figure 2.6). The signals bounce off walls, furniture and other obstacles in an indoor environment. Thus, with enough space between the transmitting antennas, usually at least half the wavelength, uncorrelated spatial streams are realized. This simple approach increases the data rate by the number of spatial streams, which is $\min(\text{number transmitter antennas}, \text{number receiver antennas})$. For example, considering an AP with four transmit antennas and a STA with only two receiver antennas, two spatial streams can be realized, if the environment, i.e. the channel matrix, allows it. A special nomenclature is used to indicate the available antennas at receiver and transmitter: $\text{number transmitting antennas} \times \text{number receiving antennas}$, e.g. 2×1 . So, a 2×1 system can only make use of one spatial stream and no MRC, while a

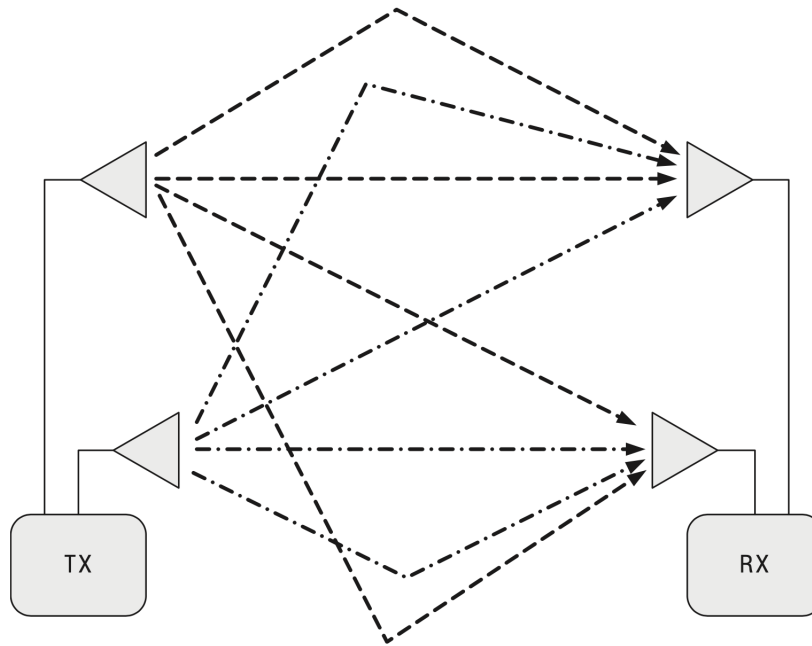


Figure 2.6: Spatial division multiplexing (from [59])

2×2 system could either use two spatial streams or one spatial stream with MRC. The choice which option is used usually depends on the channel conditions and the implementation.

Additionally to SDM and MRC, space-time block coding (STBC) was introduced in IEEE 802.11n to also increase robustness of the transmission by utilizing multiple transmission antennas while only one antenna at the receiver is available. A special coding, e.g. Alamouti coding [6] for two antennas, is used to transmit a data stream over multiple antennas and multiple time slots. In practice, in early off-the-shelf hardware deployments of IEEE 802.11n networks, an AP usually had access to two independent antennas (two independent RF chains), while STA for cost and energy reasons only utilized one antenna. In this scenario, the uplink to the AP (a 1×2 system) made use of MRC while the AP utilized STBC in the downlink to transmit to the clients (a 2×1 system).

Furthermore, the available modulation and coding scheme options in IEEE 802.11ax have increased to up to 1024-QAM with a code rate of still up to $5/6$, i.e. per 5 bits there is one redundant bit for forward error correction (FEC). This means, each OFDM

Table 2.1: Summary of IEEE 802.11 development and features.

Characteristic	IEEE 802.11n (Wi-Fi 4)	IEEE 802.11ac (Wi-Fi 5)	IEEE 802.11ax (Wi-Fi 6)
Frequency (GHz)	2.4, 5	5	2.4, 5, 6
Modulation	OFDM	OFDM	OFDMA
MIMO streams	4	8	8
MIMO	MIMO	Downlink MU-MIMO	Downlink and Uplink MU-MIMO
PHY link rate (Mbit/s)	600 (4 streams)	6933.3 (8 streams)	9607.8 (8 streams)
Maximum MPDU length (bytes)	7935	11454	11454
Maximum A-MPDU length (bytes)	65535	1048575	-
Block Ack window size	64	64	256
Bandwidth (MHz)	20, 40	20, 40, 80, 80+80, 160	20, 40, 80, 80+80, 160
OFDM subcarrier bandwidth (kHz)	312.5	312.5	78.125 (OFDMA)
OFDM symbol duration (μ s)	3.2	3.2	12.8
Guard interval duration (μ s)	0.4 or 0.8	0.4 or 0.8	0.8, 1.6 or 3.2
Highest modulation	64-QAM	256-QAM	1024-QAM

symbol can include up to $10/8 = 1.25$ times more data per OFDM symbol.

Finally, a huge impact had the increase of the channel bandwidth from 20 MHz to up to 160 MHz. This feature is called channel bonding. Channel bonding at least doubles the physical data rate by using at least two adjacent legacy IEEE 802.11a or IEEE 802.11b/g channels, respectively. Fundamentally, channel bonding increases the number of available OFDM data sub carriers from 48 in IEEE 802.11g to 468 in IEEE 802ac, when moving from 20 MHz channels to 160 MHz channels. This results in approximately tenfold maximum data rates.

IEEE 802.11ax quadruples the OFDM symbol time from 3.2 μ s to 12.8 μ s, while also quadrupling the number of sub carriers, accordingly. At first, this appears to have no impact on the available bandwidth. However, because the ratio OFDM symbol time to minimal guard interval (at least 800 ns vs. 400 ns in 802.11n and ac) was increased, the effective data rate also increased even more. Nevertheless, the main reason for the change of symbol time and number of sub carriers was to increase robustness.

In summary, the utilization of spatial division multiplexing, the increase of channel bandwidth up to 160 MHz and up to 1024-QAM modulation have been instrumental in increasing the physical data rate drastically since IEEE 802.11a/g. Additionally, the just recently finalized IEEE 802.11ax standard focuses on high efficiency. However, the new features like OFDMA and upstream multi-user MIMO are of no relevance of this thesis and will not be discussed here.

2.4 Wi-Fi Direct

Wi-Fi Direct [17] builds on IEEE 802.11 BSS networks, consisting of one access point and multiple stations, and on high-level protocols like the dynamic host configuration protocol (DHCP) and Wi-Fi Protected Setup (WPS). It creates a local and fully configured IP network containing all participating stations, so that any IP-based application can be used in a Wi-Fi Direct network. Common use cases for Wi-Fi Direct are file sharing, network printers or screen sharing.

Devices that want to communicate with each other join the same Wi-Fi Direct Group (Figure 2.7), an IEEE 802.11 BSS not connected to a DS. Firstly, they need to discover each other. Each Wi-Fi Direct device sends probe requests on each of the three Social channels 1, 6 and 11 and then listens on its chosen Listen channel (either 1, 6 or 11). This discovery phase lasts till at least one device was able to receive a probe request and answered with a probe response. Afterwards, they decide who should act as AP. The AP role is called Group Owner (GO). It is assigned based on a numerical value called GO Intent that is exchanged between the devices. Finally, the non-GO devices establish an encrypted connection with the chosen GO via WPS and get IP addresses assigned via DHCP. In the end, a temporary on-demand P2P network is created that allows all participating stations, also the GO, to run any IP based network application and to freely exchange data.

Wi-Fi Direct also includes optimizations that reduce the group formation time by numerous seconds (persistent group formation). But unfortunately, Wi-Fi Direct cannot be used to form a mobile multi-hop ad hoc network. However, as it will be discussed in a later chapter, it can be utilized to build opportunistic networks. And while Wi-Fi Direct is not available on all operating systems (e.g. iOS), any device supporting IEEE 802.11 can still connect to any Wi-Fi Direct GO by treating the GO just as any other infrastructure AP.

Wi-Fi Direct not only provides a solution to interconnect devices by means of IEEE 802.11 connections, but also makes service discovery straight forward by making use of the generic advertisement service (GAS) of IEEE 802.11u. Typical service discovery protocols like Universal Plug and Play (UPnP) or Bonjour can be wrapped into MAC layer frames and be exchanged between still not authenticated stations. Thus, any Wi-Fi Direct device is able to identify any other device that provides a special service, e.g. a printer, projector or any other P2P application. IEEE 802.11 itself includes

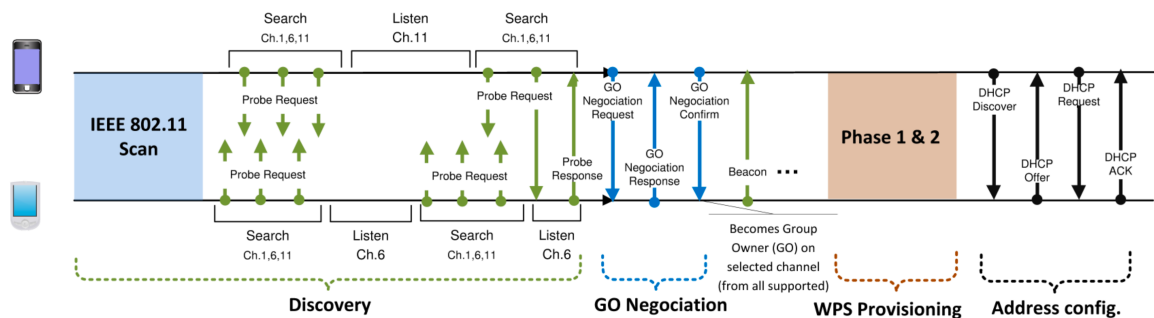


Figure 2.7: Wi-Fi Direct phases (from [17])

only the basic service set identifier (BSSID) that is usually the MAC address of the AP, and the user-configurable service set identifier (SSID). Both are not well suited to discern available services.

2.5 Bluetooth Low Energy

While IEEE 802.11 is a good solution for general purpose and high data rate applications like file transfer, web browsing and video streaming, and thus a good replacement for wired Ethernet connections, its main flaw is a high power consumption. This makes it inadequate for small internet of things (IoT) devices like fitness trackers, heart rate monitors or smart sensors e.g. for temperature or humidity. While smart home devices mainly make use of ZigBee [25], a protocol based on IEEE 802.15.4, smartphones and most smart devices connected to a smartphone, at the moment, utilize either Core Bluetooth or Bluetooth Low Energy (Bluetooth LE) [40].

Bluetooth LE, despite its name, is not the next generation of the old Bluetooth protocol, but a new protocol stack that was included in Bluetooth 4.0. However, Bluetooth LE also utilizes the 2.4 GHz ISM band. However, the old protocol stack, called Core Bluetooth in the remainder of this thesis, will be further developed independently from Bluetooth LE and focuses on personal area networks (PAN). It is nowadays mostly used for headphones, speakers, keyboards and similar devices.

Bluetooth LE distinguishes two device roles: peripheral and central. Smartphones, smart home hubs and similar devices with a fast CPU, a large battery or even connected to an outlet, implement central role and are responsible for all expensive tasks

like providing a gateway to the internet or storing and post processing the data received from the peripherals. Peripherals, on the other hand, are expected to be accessories, e.g. heart rate trackers, step counters or temperature sensors. They provide read and write access to generic values, called GATT characteristics, via the Generic Attribute Profile (GATT) protocol. Furthermore, GATT characteristics are grouped into what is called a GATT service. For example, a heart rate tracker provides the heart rate GATT service that includes characteristics for the current heart rate and the location of the heart rate sensor (e.g. wrist or waist).

GATT is based on the client / server model, where Bluetooth LE peripherals implement the server side to provide access to their GATT services and characteristics. BLE centrals, e.g. smartphones, connect to the heart rate monitor and send a request to read the current value stored in the characteristic, or even register to be notified about future updates of the value stored in this characteristic. If the Bluetooth LE peripheral is a watch that implements the appropriate GATT time service, a connected smartphone writes the current time periodically to the GATT characteristic on the watch.

Bluetooth LE peripherals are responsible for sending advertisements on the three advertising channels, to tell scanning centrals about their existence and about their GATT services. GATT services and characteristics are identified by UUIDs. Depending on the number of services supported by a Bluetooth LE peripheral, active service discovery has to be used by a Bluetooth LE central to get the list of all GATT services and characteristics supported by a specific Bluetooth LE peripheral.

Finally, Bluetooth LE was explicitly designed for use cases other than what is supported by Core Bluetooth. The payload of each Bluetooth LE packet is more or less restricted to only 27 bytes. The transmission of larger payloads is supported (approx. 500 bytes) by setting the More Data (MD) bit, however the data is transmitted in consecutive BLE packets. Furthermore, there is no block acknowledgement. Each sent packet has to be actively acknowledged to allow the sender to transmit the next packet. This results in a much lower data rate compared to Core Bluetooth, much less than 2 Mbps.

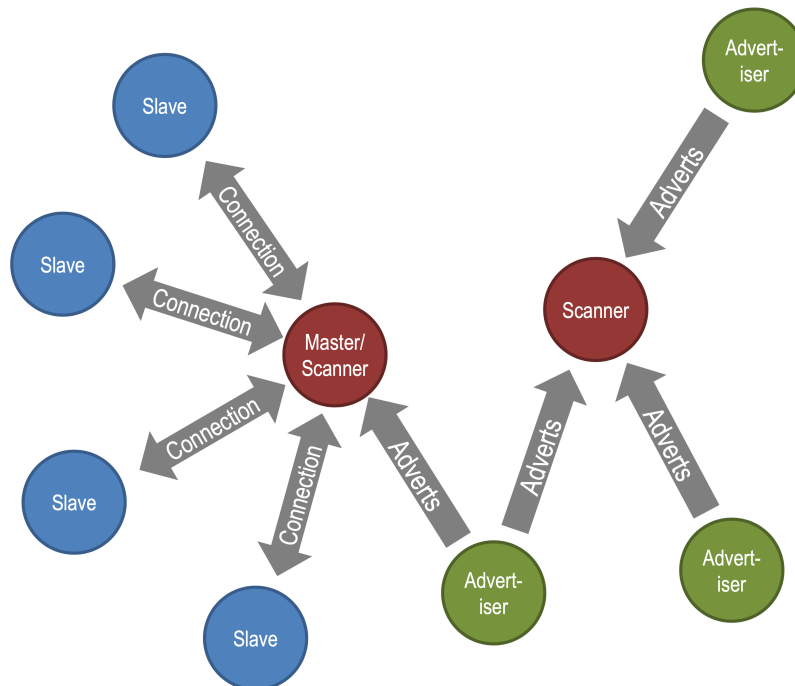


Figure 2.8: Bluetooth LE topology (from [40])

2.6 Apple Wireless Direct Link

Despite the Wi-Fi Alliance’s and Bluetooth SIG’s efforts to create a universal protocol for one-hop P2P connections, Apple deployed its own version of an IEEE 802.11 based ad hoc protocol years ago [71] on billions of devices. It is called Apple Wireless Direct Link (AWDL). Similar to Wi-Fi Direct, AWDL contains an algorithm to select a master device. The device with the highest master metric, which is transmitted in all AWDL action frames, becomes master. However, this master does not adapt the role of an IEEE 802.11 AP, it is only responsible to synchronize the activity of all devices in the proximity by synchronizing the clocks and activity periods of all devices.

AWDL uses time slots, called Availability Windows (AW), to reduce the overall power consumption and to support infrastructure Wi-Fi and P2P connections simultaneously. Devices are usually only actively participating in the AWDL protocol every fourth AW, so the AWs of all devices have to be synchronized. However, devices can decide to utilize more AWs, called Extension Windows (EW), e.g. during the transmission or reception of data. Moreover, each AWDL device can also tune to a different channel per AW, so all other AWDL devices have to be told about the channels the

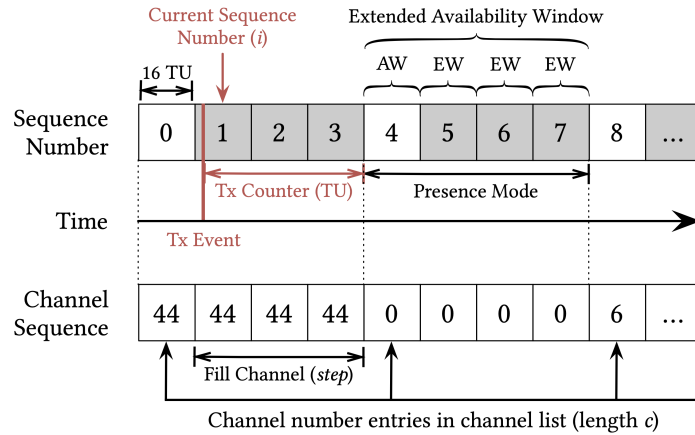


Figure 2.9: AWDL activity window and channel mapping (from [71])

device will be available at in the next AWs. Finally, data is transmitted directly from sender to receiver, without any intermediate device, during AWs in normal IEEE 802.11 data frames.

To broadcast the just discussed synchronization information, each AWDL device sends two vendor-specific action frames (AF): master indication frames (MIF) and periodic synchronization frames (PSF). Both contain the master election and AW synchronization information, while MIFs additionally contain service discovery information and device capability information. Just like Wi-Fi Direct, AWDL wraps Bonjour service discovery packages into Wi-Fi frames.

3 MAC-Layer Optimization

3.1 Understanding IEEE 802.11n Multi-Hop Communication

As discussed in 2.3, IEEE 802.11n [4] first introduced a MIMO-based physical layer, providing higher data rates of up to 600 Mbit/s, higher range and interference tolerance. These features made IEEE 802.11n a promising technology for building carrier grade wireless mesh networks.

This section presents a comprehensive measurement study of the multi-hop behavior of the IEEE 802.11n A-MPDU protocol in an indoor mesh testbed. Opposed to previous work [55], [58], and [68], an 802.11n wireless testbed in ad hoc mode (IBSS) is used, rather than the infrastructure mode. Hence, multi-hop communication is investigated. When the following experiments were conducted, Linux kernel and network system software enhancements were required to enable multi-hop communication in 802.11n.

The presented measurement study quantitatively describes characteristics of IEEE 802.11n on multi-hop paths like throughput, aggregate size and utilized MIMO features. It is thought to still be valid also for IEEE 802.11ax when not utilizing MU-MIMO. The throughput behavior and its dependence on path length, maximum aggregate size and channel bonding option is analyzed. Also, the standard Linux rate adaptation algorithm, a crucial element of the IEEE 802.11n efficiency, and its performance in multi-hop scenarios, is investigated. Furthermore, the analysis reveals details on the aggregate size in a multi-hop flow and its dependence on path length, node position within a flow and channel bonding option. The main findings are as follows:

- channel bonding nearly doubles the throughput for any fixed path length

- the mean aggregate size in number of frames at each node is also doubled by channel bonding
- mean aggregate size in number of frames at each node decreases with increasing path length
- limiting the aggregate size severely impacts throughput performance for both single-hop communication and multi-hop path
- the advantage of spatial division multiplexing fades away with increasing path length
- throughput degrades as the path length is increased, like in previous IEEE 802.11 amendments

The author of this thesis built the measurement infrastructure, analyzed and fixed the Linux kernel used in the IBSS experiments and conducted the experiments as presented in [27] together with his co-authors. The measurement infrastructure builds on the diploma thesis of the author [26]. The results also have been briefly discussed in [33].

3.1.1 Related Work

LaCurts et al. [46] analyzed traces gathered from 110 different wireless mesh networks deployed by Meraki using both 802.11b/g and 802.11n devices. They studied accuracy of SNR-based bit rate adaptation, the impact of opportunistic routing and the prevalence of hidden terminals. Opposed to this work, the following experiments focus on the impact of frame aggregation, spatial division multiplexing and space-time block coding on network performance. Furthermore, an indoor mesh testbed with little interference from 802.11a/b/g background traffic is utilized. Kim et al. [44] proposed a modification of the IEEE 802.11 MAC to allow aggregation of unicast and broadcast frames and evaluated it using a wireless node prototype. Opposed to [44], the following experiments analyze frame aggregation in the existing IEEE 802.11n standard using commodity hardware.

Halperin et al. [35] showed that wireless packet delivery can be accurately predicted using 802.11n channel state information measurements as input to an OFDM receiver

model. Khattab et al. [43] experimentally showed that 802.11n medium access worsens flow starvation as compared to 802.11a/b/g and designed an asynchronous MIMO MAC protocol that tackles the problem. Pefkianakis et al. [55] studied MIMO based rate adaptation in 802.11n wireless networks in a real testbed in infrastructure mode and proposed a MIMO aware rate adaptation scheme. Opposed to [35], [43] and [55], the following experiments consider multi-hop communication under IEEE 802.11n instead of 1-hop communication in infrastructure mode.

Pelechrinis et al. [56], [57], [58] conducted experimental studies on the behavior of MIMO links in different topologies. They mainly focused on throughput in isolation and with competing 802.11g-links [56], impact of the different 802.11n specific features on the peak performance [57], and packet delivery ratio under different physical data rates [58]. Shrivastava et al. [68] studied the impact of channel bonding and interference of 802.11g on 802.11n-links in a real testbed deployment. Opposed to [56], [57], [58] and [68], the following experiments focus on frame aggregation in a multi-hop mesh network instead of a 1-hop infrastructure mode WLAN.

Koivunen et al. [45] presented sample results from a measurement campaign of multi-link MIMO channels at 5.3 GHz in an indoor office environment. Piazza et al. [61] demonstrated a new reconfigurable antenna array for MIMO communication systems that improves link capacity in closely spaced antenna arrays. Opposed to [45] and [61], the following experiments focus on MAC mechanisms in IEEE 802.11n rather than on physical layer issues.

[30] characterized the effective throughput for multi-hop paths in IEEE 802.11n wireless mesh networks as a function of bit error rate, aggregation level, and path length. Li et al. [47] proposed an analytical model assuming saturated traffic. They derived the effective throughput and optimal frame and fragment sizes for single-hop links. Papathanasiou et al. [54] investigated through simulations the efficiency of multicast beamforming optimization over IEEE 802.11n. Opposed to [30], [47] and [54], the findings presented in this thesis are derived from measurements in a real IEEE 802.11n indoor mesh testbed.

3.1.2 Indoor MIMO Mesh Testbed

The Atheros chipset AR 9223, which was employed in all devices participating in the following experiments, utilizes two receive and two transmit antennas. Thus, all IBSS

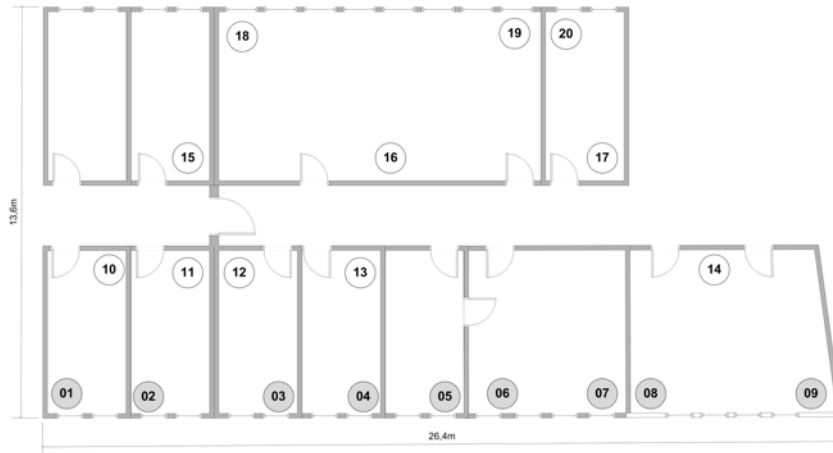


Figure 3.1: Indoor MIMO mesh testbed

transmissions could utilize a 2×2 system with up to two spatial streams. According to the notation in [55], in the following sections the usage of one stream is denoted single-stream mode and the usage of two streams double-stream mode.

The maximum number of frames to be aggregated by the utilized hardware was limited to 32 frames. In the following sections this maximum allowed number of frames to be aggregated is denoted as maximum aggregate size.

The deployed indoor MIMO Mesh Testbed comprised 20 wireless mesh nodes located in 10 rooms in a department building covering roughly 250 m². The rooms were separated by 15 cm thick light-gypsum walls, except for a solid firewall between nodes 02 and 03. An overview of the testbed with the node locations is depicted in Figure 3.1. Note, that the doors were mainly closed during experiments. Each node consisted of a Siemens ESPRIMO P2510 PC with an Intel Celeron 3.2 GHz processor, 512 MB RAM, 80 GB HDD and a D-Link DWA-547 wireless PCI network interface card (NIC). This NIC was equipped with three 5 dBi omnidirectional antennas and an AR 9223 Atheros chipset, able to support 802.11n-based MIMO communication in the 2.4 GHz band. Each node ran openSUSE 11.2 as operating system with a modified kernel based on Linux 2.6.34. The IEEE 802.11n NIC was supported by the ath9k Linux driver.

To allow remote management of the nodes, each node also utilized a Gigabit Ethernet NIC. Hence, wireless experiments could be managed from a remote computer and traces could be copied and evaluated through the wired network. Table 3.1 shows a detailed description of hardware and software components of the testbed.



Figure 3.2: 8-hop chain topology

Table 3.1: Testbed overview

Component	Description
PC	Siemens ESPRIMO P2510 Celeron 3.2 GHz, 512 MB RAM, 80 GB HDD
Wireless Card	D-Link DWA-547 PCI NIC equipped with 3 antennas
Chipset	Atheros AR 9223, operating at 2.4 GHz
Operating System	openSUSE 11.2 with kernel version 2.6.34

3.1.3 Enabling Multi-hop Communication in IEEE 802.11n

To enable the MIMO testbed to create a wireless mesh network in ad hoc mode allowing multi-hop communication, the at that time stable Linux kernel had to be modified in several ways.

Firstly, to let each wireless node know its neighbors' 802.11n capabilities, the periodically transmitted IBSS beacons had to be extended to carry extra information. Issues concerning the joining of nodes to an ad hoc network had to be resolved, so that finally each node communicates with the offered high data rates. The NICs had to be configured in Linux as normal IBSS interface and as monitoring interface, simultaneously, to allow capturing management and erroneous frames. Furthermore, the ath9k device driver was modified to set a limit for the number of frames to be aggregated, at run time. In addition, an extensive trace module was implemented, to log the MAC sequence numbers of each frame transmitted in each A-MPDU. The resulting trace files also maintained the information which transmitted MAC frame was received with errors and had to be retransmitted. This trace was later used to map frames on the receiver side with the appropriate frames on the transmitter. During the development of the trace module, particular care was taken to ensure the additional CPU and IO overhead, caused by the trace module, had no effect on the IEEE 802.11n operation. This characteristic of the trace module was validated in numerous experiments.

Additionally, the behavior of the A-MPDU aggregation mechanism in Linux 2.6.34

was changed to always send A-MPDU frames, also when only one MPDU was available for transmission. Note, that this is allowed by the IEEE 802.11n standard and the normal behavior of IEEE 802.11ac and IEEE 802.11ax. Moreover, experiments proved that this changed behavior increases throughput.

3.1.4 Experimental Setup

The 8-hop chain topology that was utilized in the following experiments is depicted by the shaded nodes of the indoor MIMO mesh testbed depicted in Figure 3.1. The nodes were positioned to let the antennas face into the building to enrich the multipath scattering, crucial for spatial division multiplexing. All nodes ran in ad hoc mode and static IPv4 routes had been configured, to make sure data is transmitted according to the topology in Figure 3.2. Each experiment lasted 60 seconds. 10 independent replicates of each experiment were conducted. The results are presented with a 95% confidence interval. The width of the confidence intervals is depicted as bars in the plots. The bandwidth measurement tool iperf [2] created saturated UDP traffic at the sender with a payload size of 1460 bytes. UDP traffic was selected to limit the influence of the TCP exponential backoff mechanism that may degrade throughput on multi-hop paths.

Special care was taken to minimize the impact of IEEE 802.11 transmissions of devices in the vicinity, that are not under control of the experiment. Therefore, initially, a one-week long-term experiment had been conducted, measuring the throughput to identify time slots with the least external interference. During the working hours between 8am and 8pm, the measured throughput was influenced by external interference, especially due to students who access the web wirelessly through their IEEE 802.11 equipped laptops. Hence, experiments were conducted at night or during the weekend at which time little interference due to 802.11a/b/g background traffic occurred.

Furthermore, the Linux rate adaption algorithm Minstrel HT was utilized to choose the most appropriate MCS class for the topology and channel conditions. Minstrel HT is the default rate adaptation algorithm for 802.11n in Linux and an advancement of the widely used SampleRate algorithm [14] by Bicket.

3.1.5 Measurements in Multi-hop IEEE 802.11n

The first experiment measured the achievable throughput on a multi-hop chain with a varied path length from 1 to 8 hops. An example for an 8-hop chain is depicted in Figure 3.2. The rate adaptation algorithm was allowed to choose from all supported IEEE 802.11n MCS classes (0 to 15 in this case), allowing to utilize spatial division multiplexing on links with adequate link quality. The experiment was repeated with both activated and deactivated channel bonding option. The measured throughput results are depicted in Figure 3.4.

To evaluate the rate adaptation algorithm, Figure 3.3 plots the probability mass function of the chosen MCS classes of each transmitter on the 8-hop chain. All rates, both with single-stream and double-stream communication, were utilized. This is a clue that the higher double-stream rates cannot be employed on every link. Furthermore, the rate adaptation algorithm tends more towards single-stream rates when channel bonding is activated. In this case, it chooses MCS 5 most frequently, while MCS 13 is mostly used when channel bonding is not activated. This is due to the higher interference sensitivity with activated channel bonding. So, as a trade-off, more robust single-stream rates are chosen.

In the next experiment, the rate adaptation algorithm was only allowed to use single-stream rates. Figure 3.4 shows that activated channel bonding increases throughput by nearly 100 % , from 140 Mbit/s to 75 Mbit/s. This result agrees with corresponding results of earlier work [55], [57], and [68]. In fact, the quantitative results of Figure 3.4 are in between the smallest and largest values of corresponding 1-hop throughput results reported in [55], [57], and [68]. Furthermore, the throughput degrades with increased path length, just like in 802.11a/b/g networks. Moreover, active channel bonding also increases throughput significantly for larger path lengths, increasing the throughput on an 8-hop chain by about 80 % , from 4 Mbit/s to 7.3 Mbit/s.

Figure 3.5 shows that limiting the number of spatial streams severely degrades throughput by about 60 % for 1-hop flows, both for activated and deactivated channel bonding. This effect flattens with increasing path length, leading to nearly the same throughput at 8 hops. On the one hand, more links with lower quality are involved in the multi-hop communication, generating a throughput bottleneck, on the other hand, increased medium contention and higher collision probability are limiting factors on longer paths and exceed the effect of higher rate choices. One can expect that this

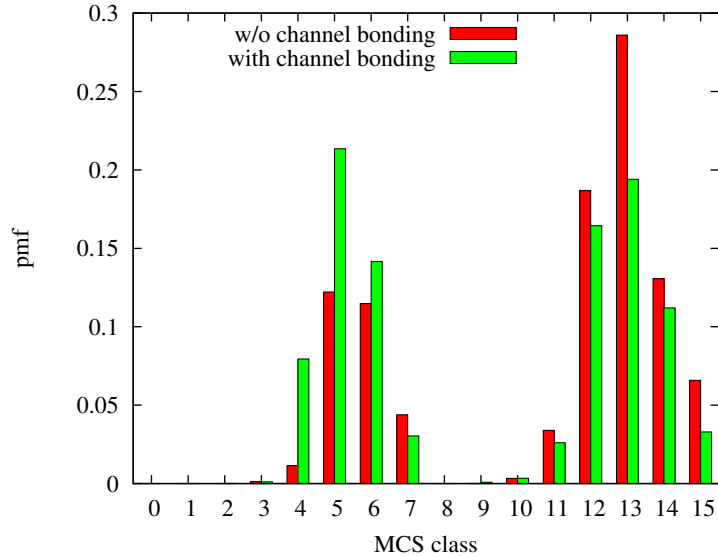


Figure 3.3: Fraction of utilized MCS classes in a multi-hop communication with 8 hops with and w/o channel bonding

also has an effect for competing flows, as the contention is comparable.

To get more insights on frame aggregation under IEEE 802.11n, Figure 3.6 plots the mean number of aggregated frames for varying path lengths. The mean aggregate size with activated channel bonding is nearly twice the size as without this option, about 29 frames compared to 16 frames per aggregate, respectively. Additionally, the aggregate size decreases with higher path lengths and nearly halves on an 8-hop path both for activated and deactivated channel bonding. Both effects might be a result of the increased physical data rate provided with channel bonding on the one hand and the decreased possible throughput on longer multi-hop paths, on the other hand.

However, the mean aggregate size does not decrease as fast as the throughput on longer paths because a transmitter still can aggregate enough frames when waiting for a transmission opportunity. This is again evidence that increased medium contention is the limiting factor on longer multi-hop paths.

Figure 3.7 takes a detailed look on the mean aggregate size at each node for a fixed 8-hop chain topology. Inside the flow, the aggregate sizes differ, leading to higher aggregate sizes near the source and lower ones near the destination. This could be the result of a decreasing queue saturation level on the transmitters along the chain, so the last transmitter has fewer frames to aggregate.

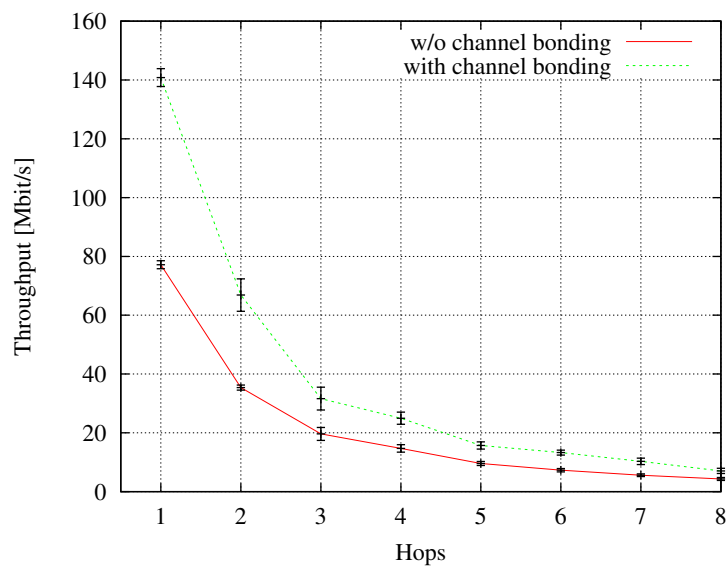


Figure 3.4: Throughput vs. number of hops with and w/o channel bonding without restrictions

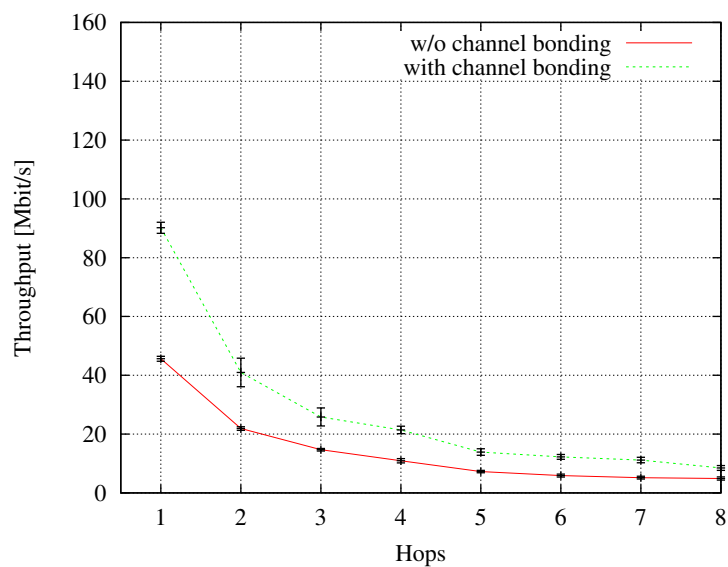


Figure 3.5: Throughput vs. number of hops with and w/o channel bonding restricted to single-stream mode

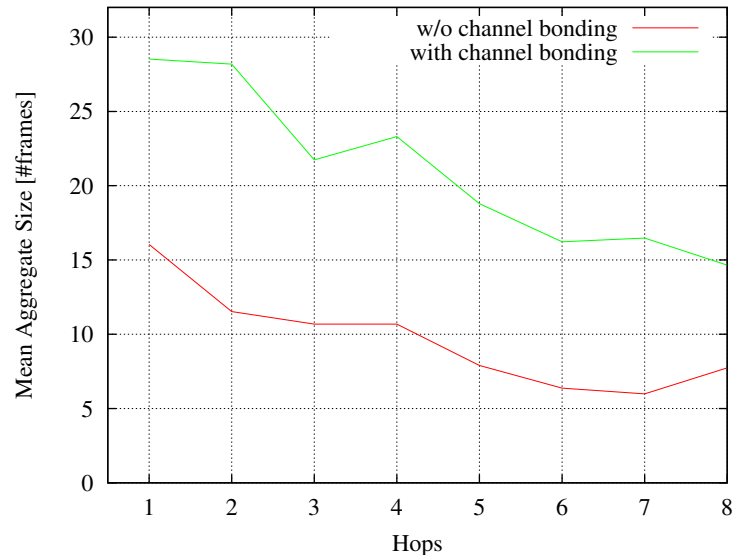


Figure 3.6: Mean aggregate size vs. number of hops with and w/o channel bonding

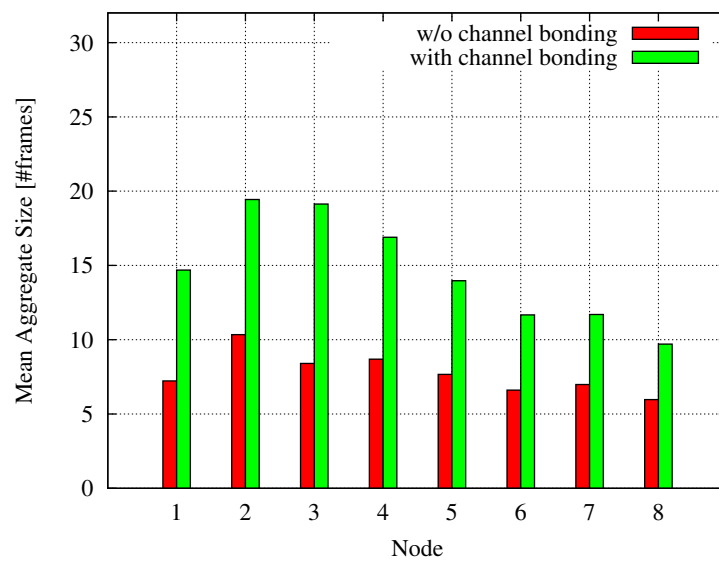


Figure 3.7: Mean aggregate size at each node for a multi-hop communication with 8 hops

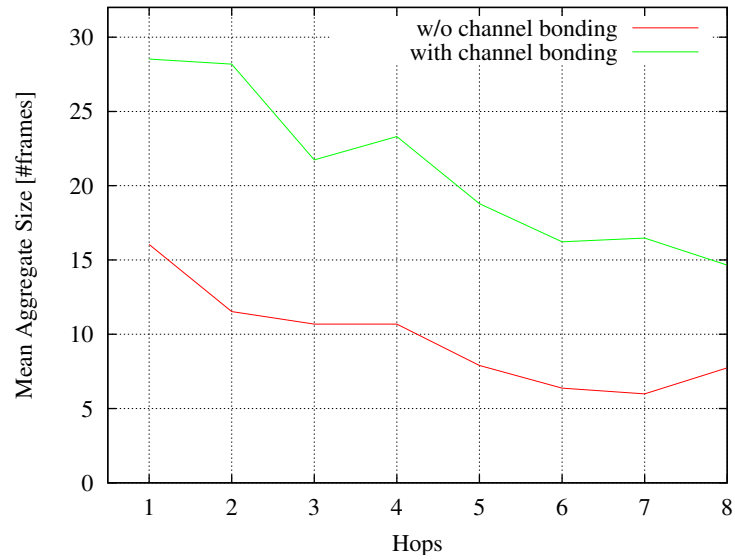


Figure 3.8: Cumulative distribution function of aggregate size for different path lengths and w/o channel bonding

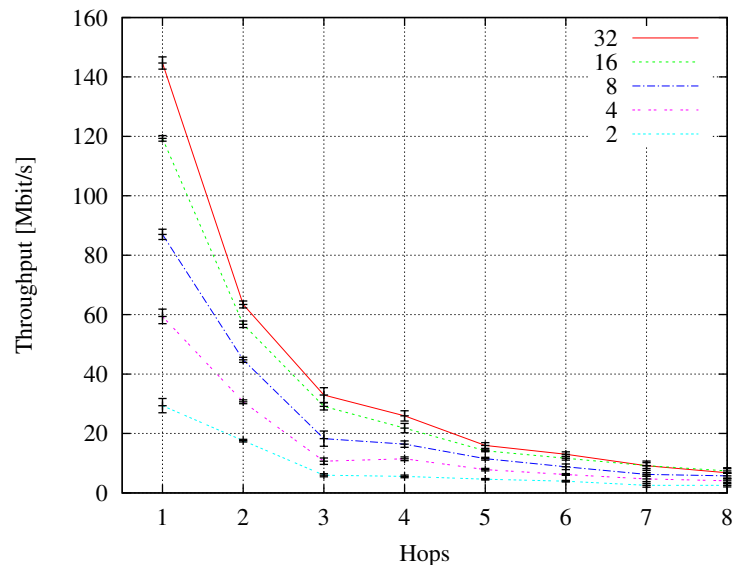


Figure 3.9: Throughput vs. number of hops for different maximum aggregate sizes

Figure 3.8 plots the cumulative distribution function of the aggregation size for selected path lengths with channel bonding deactivated. While for a 1-hop flow half of all frames are transmitted in aggregates greater than 20 frames, on an 8-hop chain half of all frames are transmitted in aggregates greater than only 7 frames. Thus, the increased path length leads to much smaller aggregates and also broadens the spectrum of used aggregate sizes.

The last experiment analyzed the influence of the maximum aggregate size on the throughput by gradually reducing the maximum allowed number of frames per aggregate. Figure 3.9 shows that an inappropriate choice of the maximum aggregate size can potentially quarter throughput. This effect slightly vanishes with longer path lengths. For path lengths greater than 4 hops, with a maximum aggregate size of 32 frames nearly the same throughput is achieved as with an aggregate size of 16 frames.

3.1.6 Conclusion

A measurement study of the multi-hop behavior of IEEE 802.11n in a real-world indoor mesh testbed for quantitatively investigating characteristics of IEEE 802.11n on multi-hop paths was presented. In particular, it revealed details on the multi-hop behavior of the aggregation level.

The presented performance curves reveal that channel bonding nearly doubles the throughput for any fixed path length. The mean aggregate size in number of frames at each node is also doubled by channel bonding and the mean aggregate size in number of frames at each node decreases with increasing path length.

3.2 MPDU Payload Adaptation

In contrast to the experiments in Section 3.1 that focused on the fundamental properties of A-MPDU aggregation in multi-hop scenarios, the approach presented in this section utilizes A-MPDU subframes to increase the overall throughput under bad channel conditions. The approach, MPDU payload adaptation (MPA), adapts the size of MAC protocol data units (MPDU) to channel conditions, to increase the throughput and lower the delay in error-prone channels. The focus is especially on the edge of the network, where even the lowest physical data rates exhibit such a high bit error rate (BER) that the probability for a successful transmission of typically sized MPDUs is

very low.

IEEE 802.11ac and IEEE 802.11ax still fundamentally utilize the same A-MPDU aggregation that was introduced with IEEE 802.11n. Most importantly, limits like the maximal MSDU size and the maximal number of subframes per A-MPDU have increased. Most other added optimizations, especially in IEEE 802.11ax, focus on a more efficient utilization of the higher data rates in a crowded environment with many stations. Various papers have covered the IEEE 802.11ax amendment. Bellalta [11] provides a look at the usage scenarios for IEEE 802.11ax and discusses the different new features and concepts. Deng et al. [22] extensively explain how the new technologies work together to achieve a better utilization and reduced error rates. Afaqui, Garcia-Villegas and Lopez-Aguilera [5] also give a brief overview of new features, provide a comparison to previous amendments and discuss key challenges. [5], [11] and [22] show that most of the enhancements provided with IEEE 802.11ax are focused around a more effective utilization of the available bandwidth and a more robust IEEE 802.11 based wireless network to improve user experience and reliability.

MPDU payload adaptation was collaboratively developed and published by the author of this thesis and his co-authors in [28].

3.2.1 Related Work

Within the last decade, among the various new IEEE 802.11 amendments, the introduction of IEEE 802.11n was a milestone, introducing PHY features like MIMO and channel bonding, as well as MAC features like frame aggregation and adding mandatory support for block acknowledgments. IEEE 802.11ac and IEEE 802.11ax have improved these mechanisms further, but their major contributions are improved physical layer data rates.

Lin and Wong [48] studied A-MSDU and A-MPDU frame aggregation in IEEE 802.11n and proposed an analytical model to find the optimal A-MSDU size for small payloads. In contrast, in this discussion of MPA, MSDU aggregation is deliberately disabled to have full control of the MPDU size.

Seytnazarov, Choi and Kim [65] studied the block acknowledgment window operation of IEEE 802.11n/ac using a Markov Chain model. As [65], this discussion of MPA focuses on the influence of error rates on aggregation and acknowledgment mechanisms. Opposed to that, MPA is evaluated using IEEE 802.11ax ns-3 simulations.

Additional simulations show how MPA performs with competing traffic flows.

Bellalta and Kosek-Szott [12] studied MU UL/DL transmissions in an ideal channel with varying traffic flows and network setups using an analytical model. Opposed to that, the evaluation of MPA focuses on a single transmission and on the influence of the error rate on throughput and delay using ns-3. Additionally, non-saturated traffic flows are considered and the performance of higher aggregation limits under erroneous conditions investigated.

In [41], Inamullah and Raman show the limitations of the compressed block acknowledgment scheme and explain its implications for throughput. They explain the hindering effects for aggregation and suggest a modified scheme, for which they calculate the possible throughput gain. The following analysis of MPA will show similar characteristics of the compressed block acknowledgment scheme. In the following, also the interaction of other factors (e.g. frame error rates, frame size and delay) will be discussed.

Wang and Wei [77] have studied the performance of several MAC layer enhancements added by the 802.11n amendment. They explained improvements and limitations of the two different frame aggregation mechanisms as well as block acknowledgment and demonstrated the performance using a ns-2 simulation. They show, that in general bigger aggregate sizes benefit the throughput, but their measurements also indicate more complex correlations. MPA builds on those ideas.

Aggregation mechanisms have been the subject of many research articles, studying the effectiveness of different aggregation and acknowledgement mechanisms and proposing new schemes. Shrivastava, Rayanchu, Yoon and Banerjee [68] outlined the magnitude of improvement one could expect from some of the new IEEE 802.11n features. They showed that aggregation can improve fairness in the presence of legacy IEEE 802.11g networks. Skordoulis, Ni, Chen, Stephens, Liu and Jamalipour [69] studied the effectiveness of A-MSDU and A-MPDU aggregation using simulations. Visoottiviseth, Piroonsith and Siwamogsatham [76] conducted a performance study using commercially available hardware, observing different usage patterns of aggregation and acknowledgment schemes depending on the implementations. While most of this work has focused on the aggregation and acknowledgment mechanisms, the interactions of multiple factors are often not considered. Opposed to that, the following MPA analysis studies the influence of different factors on an erroneous channel and explores optimization potential.

Zhang et al. [81] chose a completely different approach by developing a whole new acknowledgement scheme that utilizes a dedicated feedback channel to provide immediate reception confirmations (“micro ACKs”) to the sender and therefore allows retransmission within the same interval. Both their analytical model as well as their implementation show significant performance and efficiency improvements. Opposed to [81], the MPA analysis provides insight into the existing mechanisms and shows how to use those mechanisms more efficiently.

3.2.2 Error-Prone Channels

The main idea behind A-MPDU aggregation is to reduce the overhead of IEEE 802.11 channel access. It enables the transmission of several MPDUs within one physical protocol data unit (PPDU). Each MPDU retains its own frame check sequence (FCS). Thus, in case of transmission errors, only the affected MPDUs have to be retransmitted and not all data of the PPDU is lost.

MPDU payload adaptation (MPA) is based on the idea that the A-MPDU aggregation scheme can also be utilized to improve network performance at the edge, where the signal-to-noise ratio (SNR) is very low and, thus, the bit error rate (BER) very high. Usually, rate adaptation counteracts low SNR and reduces e.g. the number of spatial streams or the MCS index. However, at best, it is only able to select the lowest MCS index. Adapting the MPDU size creates another dimension for rate adaptation to cope with channel conditions.

With good channel conditions, the utilization of very large MPDUs is favored to improve throughput and to reduce the overhead. However, by limiting and actively adapting the MPDU size based on channel conditions, e.g. by reducing the MPDU size to 100 bytes and utilizing A-MPDU frame aggregation with up to 256 MPDUs in IEEE 802.11ax, this approach effectively forces the integration of FCS fields per 100 byte chunks of the user payload. Thus, the channel is only able to destroy small MPDUs, requiring a smaller fraction of the user payload to be retransmitted in comparison to the usual MPDU sizes.

MPA especially aims at improving the transfer of large data (e.g. video streaming) which typically utilizes the maximum allowed payload size per MPDU.

3.2.3 Simulator Setup

The following simulations have been conducted using the widely used ns-3 discrete-event driven network simulator [39]. However, the original source code was slightly modified to obtain additional log output and to also be able to specify the A-MPDU and A-MSDU limits for each simulation.

The primary network setup, as depicted in Figure 3.10, consists of an IEEE 802.11ax station (STA node) and an IEEE 802.11ax access point (AP node). The nodes always communicate with IEEE 802.11ax 160 MHz wide channels in the 5 GHz band. In the first experiments, these nodes have exclusive access to the channel. There is no interference from any other device. The acknowledgments frames (ACK) are sent at basic rates, since HT, VHT and HE ACK are not supported by ns-3 at the time of writing this thesis. The $0.8 \mu s$ guard interval is used and RTS / CTS transmission is disabled. The nodes make use of the default compressed block acknowledgments (BACK) of IEEE 802.11ax and the default A-MPDU aggregation, i.e. respond with BACK frames immediately after reception of a data frame.

The frame size of the transmitted MPDUs is controlled at application level in the experiments using an UDP application. It creates a saturated UDP data transmission between STA and AP. UDP is usually preferred to TCP in throughput experiments to reduce the effect of TCP congestion avoidance. The payload of each UDP datagram is varied from 100 bytes to 2250 bytes and A-MSDU aggregation is intentionally disabled in ns-3. Thus, the size of each UDP datagram also deterministically determines the size of each MPDU.

The utilized simulator scenario utilizes the YansWifiPhy and YansWifiChannel modules for the physical model. Path loss for these modules is calculated using the Log Distance Propagation Loss model, which introduces packet errors due to fading and therefore creates an erroneous channel. The frame error rate (FER) estimation of this model is based on distance between nodes and the modulation and coding scheme (MCS). Both STA and AP transmit data frames with IEEE 802.11ax MCS index 7 (64-QAM, 5/6 coding) to achieve significant bit error rates even at a distance as low as 11 m. All devices transmit control frames with IEEE 802.11ax MCS index 0 (BPSK, 1/2 coding).

No mobility is used in the experiments. All STA and AP nodes remain at fixed positions at all time. This leads to no changes in distance between the nodes during

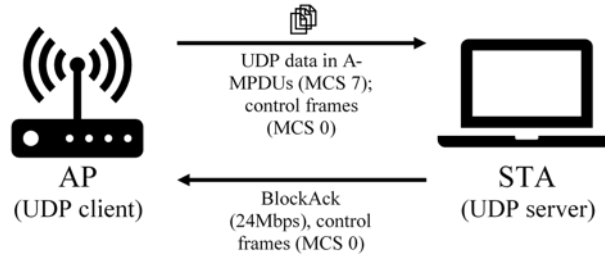


Figure 3.10: The wireless network scenario considered in the simulation model

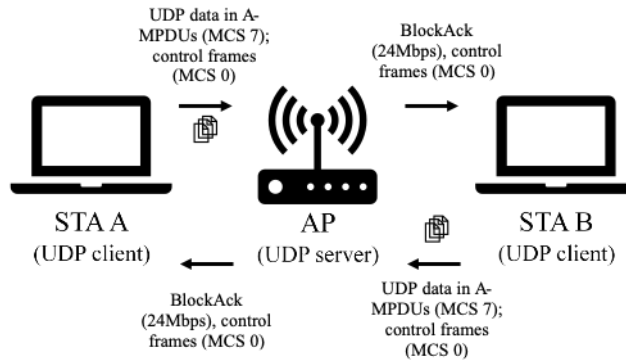


Figure 3.11: The wireless network scenario considered in the fairness experiments

an experiment.

The fairness studies employ a slightly different simulation setup with an additional STA node (Figure 3.11). The roles are reversed, so that the AP node receives UDP packets from both STA nodes. All nodes use the same 160 MHz channel.

The following section describes a number of experiments with different parameter combinations, according to the scheme outlined below. Each of these combinations was run multiple times with the same seed but different run numbers (independent replicates). Data exchange started one second after simulation start, to avoid initialization effects. The measurements were taken across a span of 5 seconds of simulation time. The figures plot the results with a confidence interval of 95 %.

3.2.4 Quantitative Performance Results

The following performance measures are used to evaluate the performance of the experiments:

Goodput is the payload per time unit received by ns-3 at application layer.

Application layer delay (ALD) is the time span on application layer between creation of an UDP packet and reception of this packet at the receiver.

Sender receive delay (SRD) is the time span on MAC layer between the first transmission of an MPDU and its successful reception at the receiver.

Frame traffic describes the combined size of data frames per time unit that are sent across the wireless medium in order to deliver MPDUs. This includes all MPDUs and their retransmissions.

MPDU Payload Adaptation

Wireless network adapters supporting IEEE 802.11ax choose the optimal transmission parameters to fully utilize the available channel conditions and to produce the largest possible goodput. Usually, the most important parameter that is varied is the MCS index used to transmit data frames. Figure 3.12 shows the achievable goodput for a selection of MCS indices at up to 50 m when using 600 byte MPDUs and a saturated UDP traffic.

Each graph exhibits a sudden drop in goodput. The increasing distance steadily decreases the signal-to-noise-ratio (SNR). With a decreasing SNR, bit errors occur significantly more frequently, till the SNR drops below a certain threshold and the probability of bit errors is too high to transmit data at all. At this point, usual rate adaptation switches to the next lower MCS index. The now achievable goodput is, however, lower than the maximal goodput of the previous MCS index. Also, the differences in maximal goodput between adjacent MCS indexes are significantly larger when the modulation scheme is changed (e.g. MCS 4 with 16-QAM vs. MCS 5 with 64-QAM). In these cases, an instant change to the next lower MSC index is not necessarily required at the instant the goodput drops.

Essentially, a very similar figure can be plotted when keeping the MCS index at a fixed value and increasing the transmission distance, therefore, increasing the BER (Figure 3.13). In this simulation of a saturated UDP traffic from AP to STA, transmitting data frames with MCS 7, one spatial stream, an A-MPDU size of 64 sub-frames, and no bit errors at all, an MPDU size of 1500 bytes results in the highest goodput. With increasing BER, the goodput drops significantly, till a smaller MPDU size provides a higher goodput. Starting with a BER of about 0.4×10^{-4} , 1000 byte MPDUs

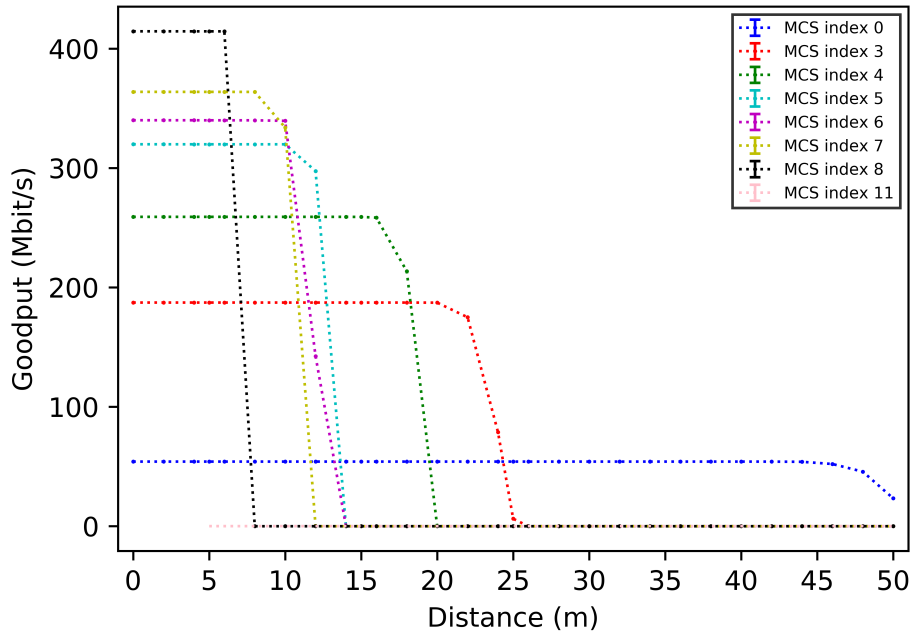


Figure 3.12: Goodput for a selection of MCS indices (600 byte MPDUs)

would be the best choice. And after a BER of about 1×10^{-4} , AP would optimally transmit 500 byte MPDUs. Just like with the MCS rate adaptation, with increasing BER, MPDU sizes have to be steadily reduced to optimize goodput.

Additionally, at each start of a transmission in a channel with unknown properties, it is not clear which MPDU size provides the best performance. All MPDU sizes have to be probed to measure the actual achievable goodput per MPDU size. In the 4×10^{-4} example in Figure 3.12 200 byte MPDUs result in the highest goodput.

The effect of the MPDU size can be explained as follows: MAC efficiency, and thus also goodput, usually increase with an increasing MPDU size in a stable channel with a fixed, low BER (Figure 3.13). However, an increase in MPDU size also results in an increased frame error rate (FER). There is a moment, when the negative impact of the retransmissions caused by the high frame error rate outweighs the benefits of an increased MAC efficiency.

Moreover, as can be seen in Figure 3.13, the negative impact of increases in the BER on goodput is significantly higher when using larger MPDUs compared to smaller MPDUs.

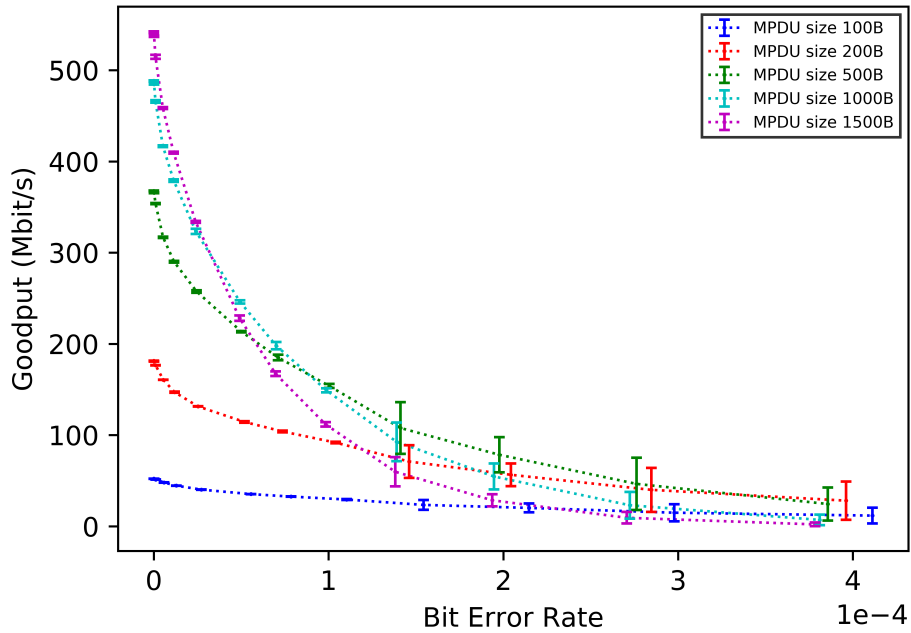


Figure 3.13: Goodput for different frame sizes under saturation

Figure 3.14 shows the goodput of a saturated channel where the MPDU size is varied between 100 Bytes and 2250 Bytes. For low BER scenarios, the goodput improves as the MAC overhead steadily decreases. For bit error rates higher than 3.3×10^{-5} , the goodput decreases above a certain MPDU size. As frame error rate depends on both frame size and the underlying BER, larger frames will also show decreasing goodput at some point under even smaller BERs.

A-MPDU Limits

Looking back at Figure 3.13, the rapid degradation of goodput with increasing BER is surprising. However, this behavior can be explained by the format of the compressed BACK (Figure 2.2) and the maximal number of subframes per A-MPDU in IEEE 802.11ax (Table 2.1). Each A-MPDU can contain at most 256 MPDUs with a MAC sequence number in the window $[s_0, s_0 + 255]$. s_0 is the start sequence number of the BACK bitmap. The worst-case scenario can be constructed when first transmitting an A-MPDU with 256 MPDUs, and only the first MPDU is transmitted with errors. Thus, the next A-MPDU can only contain exactly one MPDU.

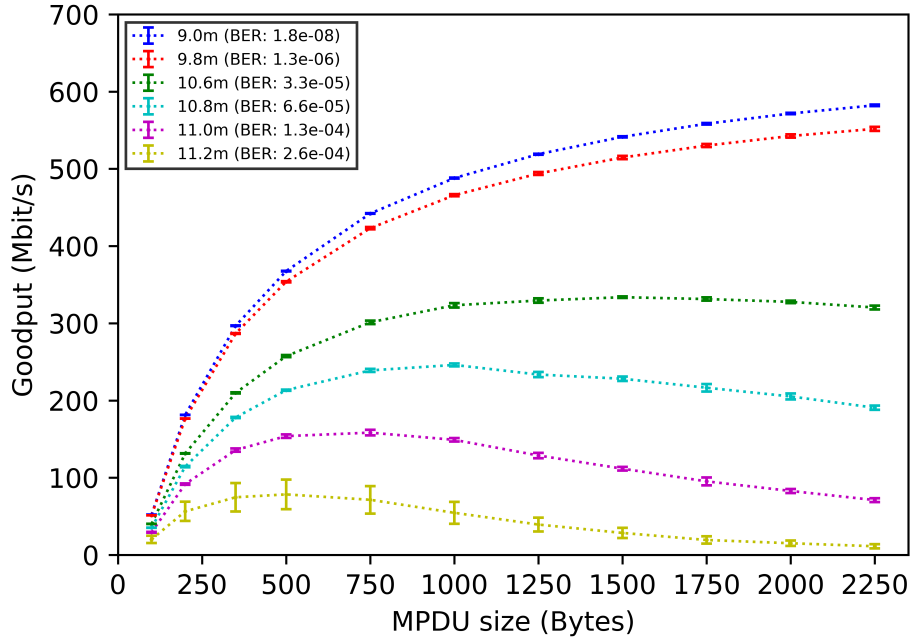


Figure 3.14: Goodput with increasing frame size at fixed distances

For the next experiment, ns-3 was further modified to support different aggregation/acknowledgment window sizes. Experiments were conducted with different BERs and saturated traffic flows to study the goodput and effectiveness of increased A-MPDU limits. Figure 3.15 shows the ratio of observed number of MPDUs per A-MPDU compared to the respective A-MPDU limit. Increasing the A-MPDU limit decreases utilization, as retransmissions can block off slots that are not available to carry payloads. This has been documented in [8, 11] extensively. Unfortunately, this problem cannot be fixed when using compressed BA, as it is limited by the memory available in the network interface. Larger limits required more memory and most other BA strategies require completely different network interface hardware.

Figure 3.16 shows that increasing these limits still improves the resulting goodput. This is also supported by Figure 3.13, which shows a decrease in overall utilization of the BA window, however, the absolute number of MPDUs per A-MPDU still increases when increasing the A-MPDU limit. However, the goodput drops under erroneous conditions more rapidly than the utilization ratio. When a frame has to be retransmitted, it not only occupies additional channel time, but it may also hinder the advancement

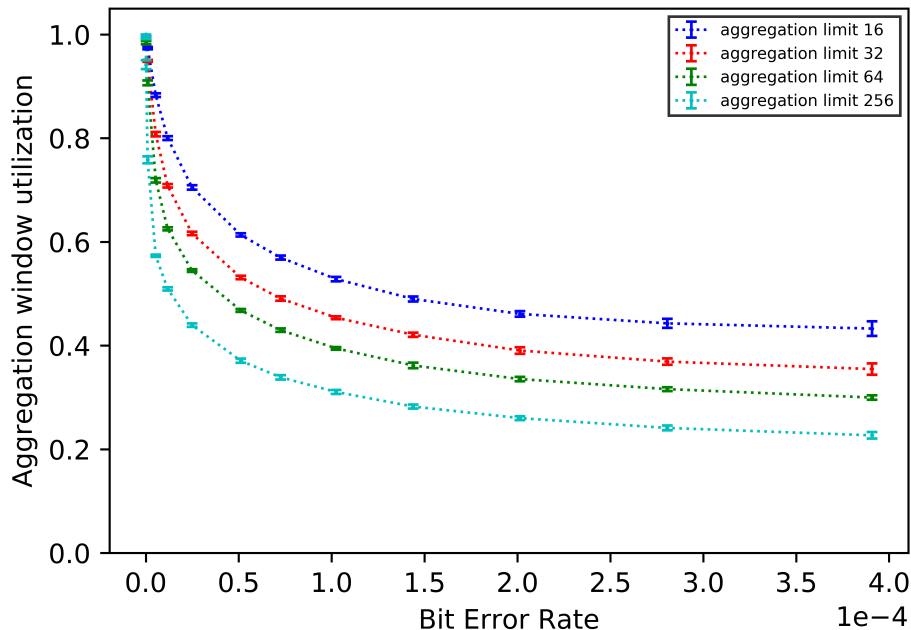


Figure 3.15: Utilization of different aggregation and acknowledgment window sizes

of the BA window. This leads to a decreased effectiveness for subsequent transmissions, as fewer MPDUs can be aggregated, and many of them are retransmissions. As discussed before, the impact of this hindering depends on the relative position of the first erroneous frame within the A-MPDU, with frames at the beginning causing higher degradation.

Optimizing Delay

Two delay performance measures were collected to study the delay characteristics and the influence of the IEEE 802.11ax MAC mechanisms on non-saturated UDP traffic: the application layer delay (ALD) and the sender receive delay (SRD) on MAC layer.

Figure 3.17 shows that SRD under low BER conditions only shows linear growth that can be attributed to longer transmission times for the larger frames. However, there is exponential growth for the highest BER scenario. This is, as discussed in the Section 3.2.4, a result of the increased successful transmission time caused by the increased number of retransmissions, as well as less efficiently utilized A-MPDUs, causing more transmissions and more channel access operations.

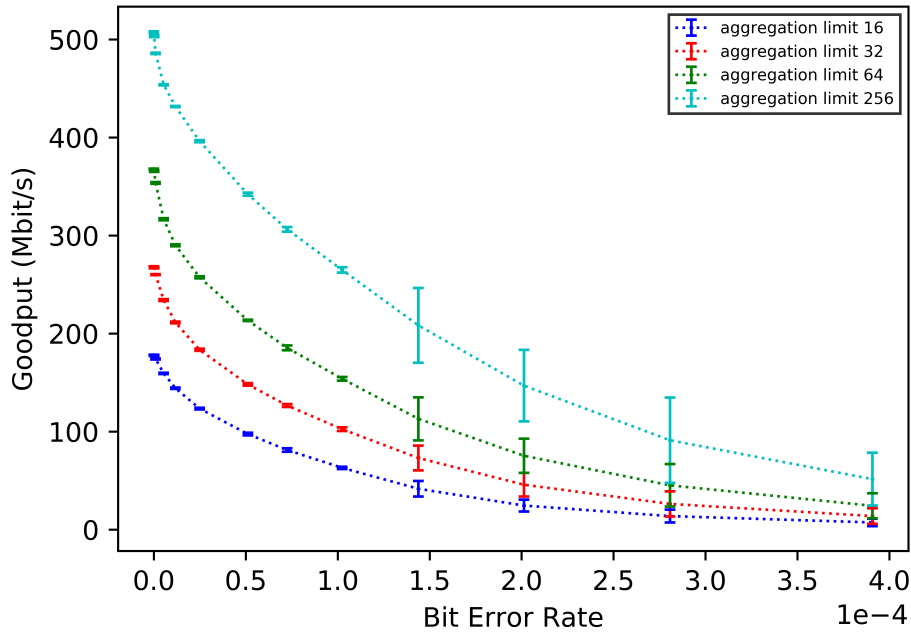


Figure 3.16: Goodput of different aggregation and acknowledgment window sizes

The ALD plots shown in Figure 3.18 have a similar characteristic as the SRD plots in Figure 3.17. However, all ALD measurements are roughly about 100 times the SRD measurements, with some ALD measurements peaking up into the 3-digit-millisecond range. This is most likely caused by the queuing delay. Apparently, each MPDU has to wait for approximately 100 MPDUs to be transmitted beforehand.

Figure 3.19 compares the development of SRD with increasing load, employing a fixed MPDU size. The delay rises steadily before stalling when saturation is reached and packets are dropped. Higher bit error rates exhibit higher MAC layer delay. In Figure 3.20, SDR is shown while the offered load transitions from unsaturated to saturated conditions, for different MPDU sizes, while keeping the BER fixed. Here, the MAC layer delay also increases with increasing MPDU sizes.

3.2.5 Conclusion

This section proposed MPDU payload adaptation (MPA) based on channel conditions in error-prone environments. The widely known ns-3 discrete-event simulator was utilized to conduct experiments with varying traffic flows and environment conditions.

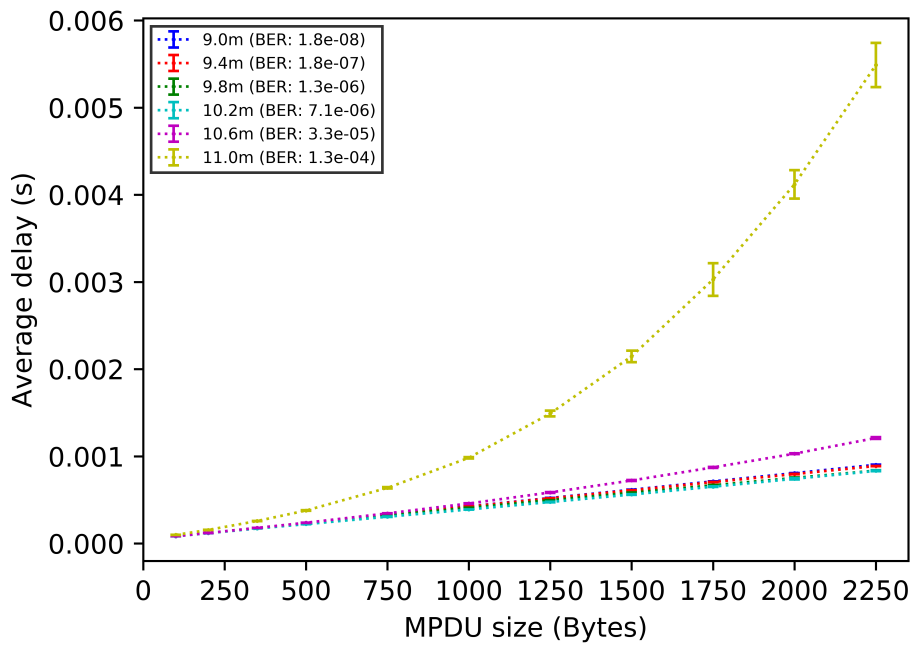


Figure 3.17: SRD under different BER conditions

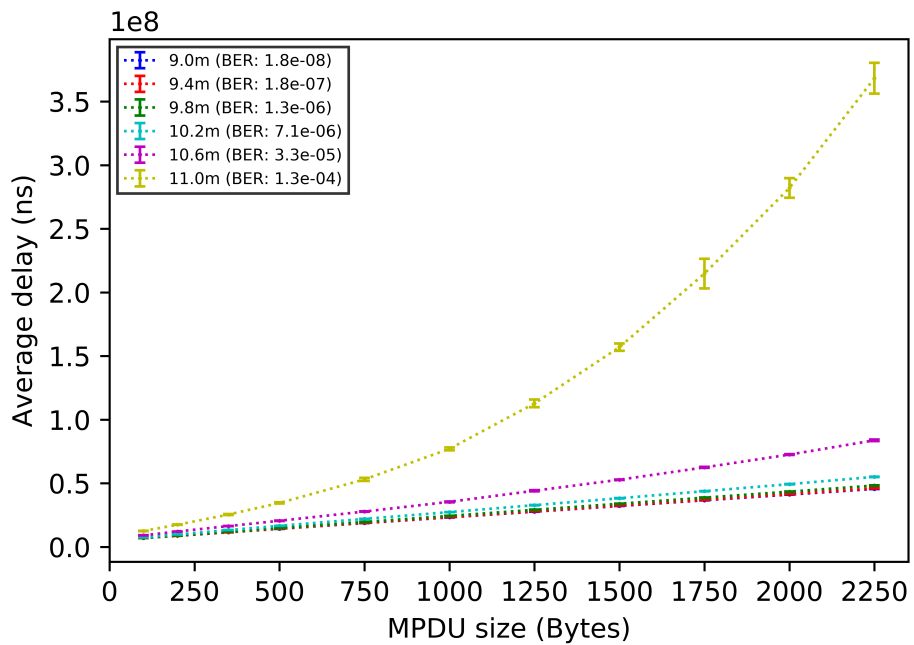


Figure 3.18: ALD under different BER conditions

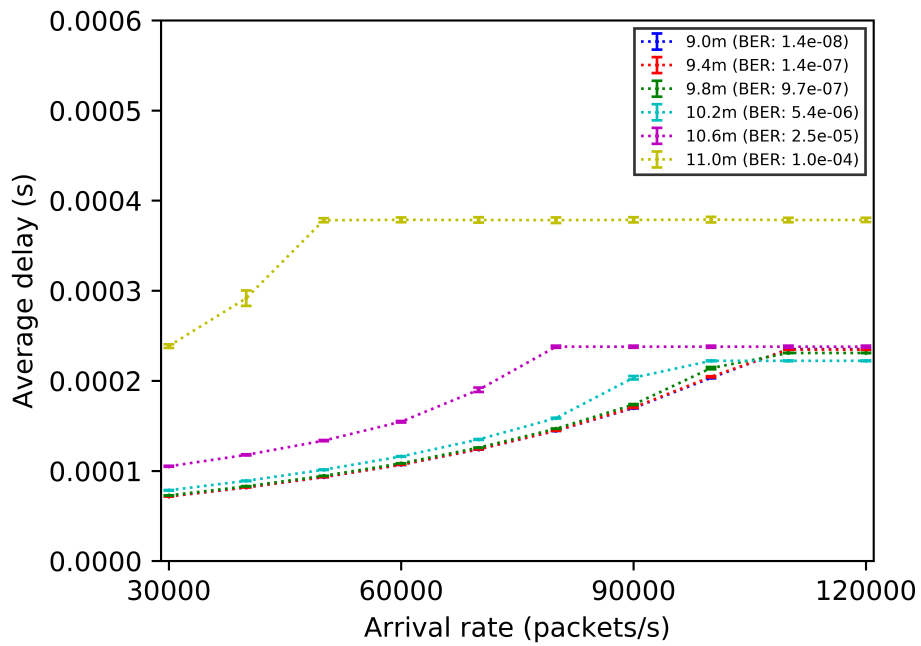


Figure 3.19: SRD while increasing offered load

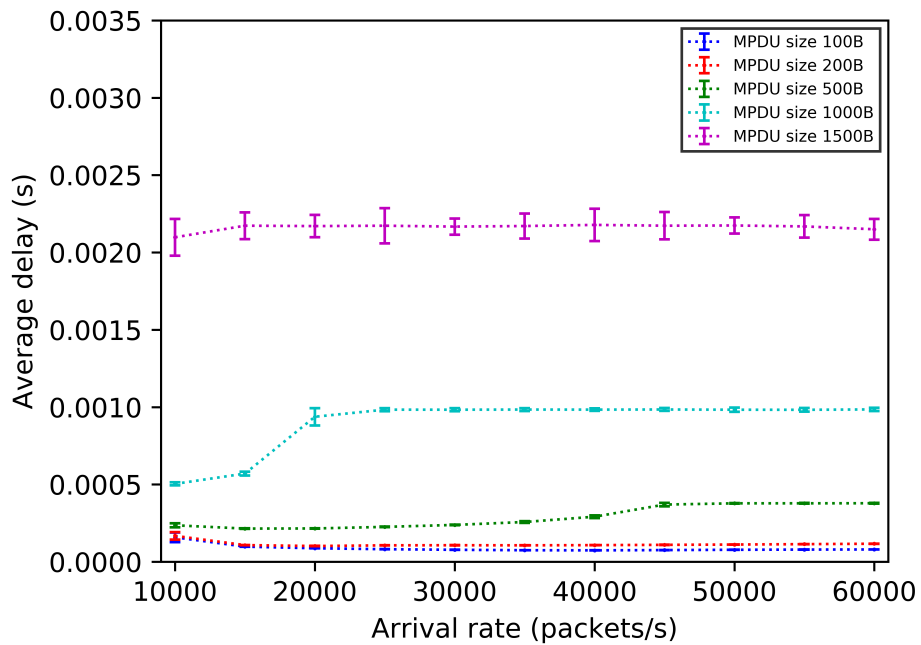


Figure 3.20: SRD increase from non-saturated to saturated traffic

It was observed that peak throughput is not always achieved by minimizing overhead or maximizing aggregation. A careful selection of MPDU sizes is crucial. The results suggest that under erroneous conditions throughput can be maximized by limiting the MPDU size. Moreover, smaller MPDUs also reduce delay.

Furthermore, the investigations revealed that retransmission wait times are the most important reason for delay spikes. These wait times are increased when using compressed block acknowledgements in error-prone channels, as a result of the low utilization of the acknowledgment window. Consequently, the low utilization has a major negative impact on the goodput.

4 Neighborhood-Aware Opportunistic Networking on Smartphones

In recent years, opportunistic networking [20] has emerged as a new mechanism for wireless communication. Opportunistic networking takes advantage of human mobility for content dissemination by establishing device-to-device communications on-the-fly, when people with modern mobile devices (e.g. tablets or laptops) meet. Thus, opportunistic networks constitute an evolution of mobile ad hoc networks (MANET). MANETs require end-to-end communication paths for data delivery, while opportunistic networking solely relies on occasional encounters of people and their devices with common mobility patterns, interests, etc. for data dissemination. The Wi-Fi Alliance introduced Wi-Fi Direct [78] as a technology for device-to-device (D2D) communication over IEEE 802.11, which can implement opportunistic networking. However, as noted by several authors [19], [72], [80], Wi-Fi Direct is not well suited for this form of D2D communication and various research problems have to be solved, before the protocol can be put into practice.

This section introduces and analyzes the Neighborhood-aware OPPortunistic networking on Smartphones protocol, denoted as NOPPoS. NOPPoS assigns IEEE 802.11 station and access point roles to mobile devices based on the number of mobile devices and access points in the proximity. As main novel feature, NOPPoS is highly responsive to node mobility due to periodic, low-energy scans of its environment. In fact, NOPPoS can determine the exact number of neighbors at any instant of time. Therefore, NOPPoS can assign roles to mobile devices in a more efficient way than previous approaches (e.g. [72], [78]) could do.

NOPPoS builds on top of WLAN-Opp [72], [73] and introduces a technique for determining the accurate number of current neighbors based for all scenarios. This is opposed to [72] that approximates the number of neighbors based on the stations in the previous access point. By taking the accurate number of current neighbors into

account for determining the switching probabilities between roles, the likelihood for opening multiple hotspots simultaneously in a neighborhood is substantially lower. This holds in particular for mass events like pop concerts or sports events, where a large number of devices reside in a limited area.

NOPPoS was collaboratively developed and published by the author of this thesis and his co-authors in [13].

4.1 Related Work

In recent years, Wi-Fi Direct has been introduced by the Wi-Fi Alliance as a technology especially tailored for D2D communication [78]. Conti, Delmastro, Minutiello, and Paris [19] showed how to implement opportunistic networking with Wi-Fi Direct. Zhang, Wang, and Tan [80] improved the Wi-Fi Direct group formation protocol by taking into consideration the Received Signal Strength Indicator (RSSI) values for negotiating the group owner. Nonetheless, as stated in the recent paper [72], Wi-Fi Direct still suffers from a cumbersome manual pairing process and considerable energy consumption.

Trifunovic, Kurant, Hummel, and Legendre introduced WLAN-Opp [72] for implementing opportunistic networking on modern mobile devices running on off-the-shelf Android systems. In WLAN-Opp, a mobile device always takes one of the three roles: (1) IDLE: scanning for networks, (2) STA: being associated as a station to an access point, and (3) AP: access point mode. Mobile devices switch between these roles based on probabilities determined by the number of their current neighbors. WLAN-Opp determines this number of current neighbors in most scenarios based on an approximation of the number of devices observed in the last access point. If this number is not known, WLAN-Opp assumes the number of current neighbors to be 2. In [73], Trifunovic, Picu, Hossmann and Hummel reanalyzed and improved the power fairness of WLAN-Opp. NOPPoS builds on the role model and state machine of WLAN-Opp [72], [73], but introduces refined equations for the state transitions due to accurately discovering the number of neighbors. Therefore, NOPPoS avoids Wi-Fi scans nearly completely and finally reduces overall energy consumption significantly (e.g. more than 80 % reduction during device discovery).

Schäfer et al. [63] utilize Wi-Fi Direct connections to neighbors for task scheduling

on edge devices. They analyze the benefit of communicating tasks to neighbors and utilizing the computational resources of neighbors. In contrast to NOPPoS, they do not focus on creating a general-purpose opportunistic network. Battery level and other resources are also a crucial part of their approach. However, this internal and external context is utilized to decide if it is worth to connect to a neighbor. NOPPoS utilizes these characteristics to decide which device is most apt to manage the Wi-Fi Direct group.

The survey paper [62] proposes a taxonomy for different approaches for neighbor discovery for opportunistic networking, including mobility agnostic, mobility aware, colocation and probabilistic. NOPPoS utilizes multiple radio technologies and falls in their taxonomy into colocation and probabilistic.

The utilization of multiple radio technologies for neighbor discovery has been examined in various publications in the past. Shih, Bahl and Sinclair [67] proposed to use a low-power radio for discovery and high-power Wi-Fi for data transmission. They showed that this approach improves the battery life compared to Wi-Fi only systems by 115 %. Another approach for reducing energy consumption lies in utilizing ZigBee as technology for neighbor discovery and Wi-Fi for data transmission [42], [60]. In both previous works, the author observed that ZigBee performed slightly better than Bluetooth in terms of energy consumption, but Bluetooth is more common on recent mobile devices.

Han and Srinivasan proposed to employ Bluetooth radio technology for neighborhood discovery [36]. Ananthanarayanan and Stoica [7] showed how to utilize Bluetooth for RSSI-based movement detection so that neighborhood scans are only triggered after significant movements. Bakht, Carlson, Loeb, and Kravets [10] utilized Bluetooth technology for optimizing power consumption leveraging on knowledge about clustering nodes. Opposed to [7], [10], [36], NOPPoS does not need to take into account the history of previously discovered nodes, is not only triggered by user movements, and does not require clustering of nodes.

Mawad and Fischer [52] proposed a hybrid approach to opportunistic networks, where different regions of directly connected mobile devices are interconnected with infrastructure connections. This approach is currently not available with NOPPoS but a future extension of NOPPoS to interconnect multiple regions is possible.

4.2 The NOPPoS Protocol

4.2.1 Wi-Fi Opportunistic Networking

The purpose of NOPPoS is to create an opportunistic, pocket-switched network using current generation, off-the-shelf mobile devices. For this purpose, NOPPoS utilizes the Wi-Fi access point of mobile devices to create local isolated networks that connect co-located mobile devices. Driven by the mobility of users, mobile devices will eventually switch to other access points and, therefore, forward data to other mobile devices. In this way, NOPPoS enables message delivery according to the store-carry-forward paradigm employed, for example, in [24]. NOPPoS does not specify the store-carry-forward message protocol but provides the TCP/IP network and efficient peer discovery so any message forwarding protocol can be implemented on off-the-shelf mobile devices.

The crucial component of NOPPoS is a finite state machine, whose state transitions tell mobile devices when to enable their own access point, when to connect to the access point of a neighbor or even when to change their current access point. Its transitions depend on parameters of the surrounding (e.g. number of neighbors, number of open access points). In addition, the transitions are not evaluated continuously, but at discrete instances of time. Typical Wi-Fi operations take time (e.g. Wi-Fi scan, opening access points, connecting to access points), and the authors of [72] have already discussed sufficient slot times (time between state transition evaluations).

The state transitions of NOPPoS are designed to (1) avoid overlapping access points, (2) maximize the number of neighbors connected to the same access point, (3) connect the members of cliques to different access points and to (4) minimize energy consumption by minimizing Wi-Fi scans and access points. Some of these policies are contradictory, so NOPPoS aims to balance them heuristically. Moreover, NOPPoS mobile devices are only able to discover access points and other NOPPoS devices in their neighborhood. A global view of the location of all NOPPoS devices or a group formation phase as in Wi-Fi Direct is not possible using the standard IEEE 802.11 access point operation that is built into off-the-shelf mobile devices. This limitation is most problematic in the initial phase of NOPPoS, when there are no access points to interconnect neighboring devices. Each NOPPoS device has an independent probability to enable an access point. This probability is inversely proportional to the number

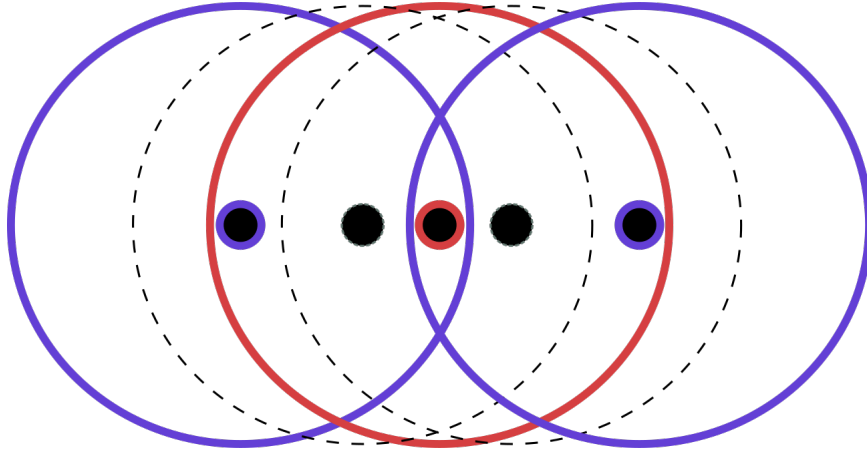


Figure 4.1: In this scenario with 5 nodes (dark circles in the center, surrounded by the radial communication range), NOPPoS most likely creates the two blue access points. Therefore, NOPPoS moves the nodes in the clique in the center to different access points

of neighbors (Figure 4.1). This behavior is in line with design goals (1), (3) and (4), but violates (2). However, caused by subsequent movement of smartphones and the NOPPoS policy to switch access points, if there is another access point with more connected nodes in the vicinity, NOPPoS also obeys (2) in the long run behavior.

Finally, in contrast to WLAN-Opp, NOPPoS assumes a second, low-power radio technology to continuously determine the accurate number of neighbors. This radio technology should require significantly less energy than Wi-Fi, operate on another frequency band to not interfere with the Wi-Fi operation and exhibit approximately the same communication range. NOPPoS only requires this technology to advertise the presence and state of mobile devices. Real data transmission is not required. Considering the technologies available on modern mobile devices, Bluetooth LE is the obvious choice.

4.2.2 States and State Transitions

NOPPoS is driven by a state machine (Figure 4.2), which runs on each node of the system. A node is either idle (IDLE state), provides an access point (AP state), or takes the role of a station connecting to an access point in the proximity (STA state) in NOPPoS. State changes take place according to transition probabilities, which take into account the number of nodes currently residing in the radio range; i.e. the neigh-

bors. As an entirely distributed approach, NOPPoS only requires the transmission of Wi-Fi and Bluetooth LE beacons among neighbors.

Opposed to [72], NOPPoS utilizes the accurate number of current neighbors. In IDLE state, a NOPPoS node continuously scans for Bluetooth LE beacons to determine the number of access points and the number of other IDLE nodes. If there exists at least one access point (`networksAvailable()` returns true), it connects to a randomly chosen access point and switches to STA state (`connect()`). The node remains in this state until another, likely more prominent access point becomes available in its radio range (`shouldSwitchNetwork()` returns true), or the access point, to which the node is currently connected, is closed.

After a NOPPoS node has switched to another more prominent access point, it stays in STA state. In case the access point, to which a node is connected, is closed, the node returns to IDLE state. Note that as long as a node resides in STA state it does not open an access point itself. This is only possible in IDLE state, if no other known access point is already available in the radio range (`shouldOpenAP()` returns true and `networkAvailable()` returns false).

Whenever there is no access point available in its neighborhood, a node eventually opens an access point itself and switches to AP state. Now, other IDLE nodes, as well as STA nodes that want to switch to another AP, discover this access point and decide whether or not to connect based on their current number of neighbors. The likelihood that a node may close its access point voluntarily (`shouldCloseAP()` returns true) only depends on the number of connected nodes and the time that passed since opening the access point. Figure 4.2 depicts the state transition diagram of the protocol for NOPPoS.

Opposed to [72], NOPPoS introduces deterministic state transitions into the state diagram of Figure 4.2, based on the accurate number of neighbors. Using these state transitions, NOPPoS is able to decide whether or not a node should open an access point or whether to switch to another more prominent access point. As [72], NOPPoS still partially relies on probabilities for switching between states. For example, nodes switch to AP state based on a probability parameterized on the number of IDLE neighbors.

Table 4.1 summarizes the input variables of the proposed protocol.

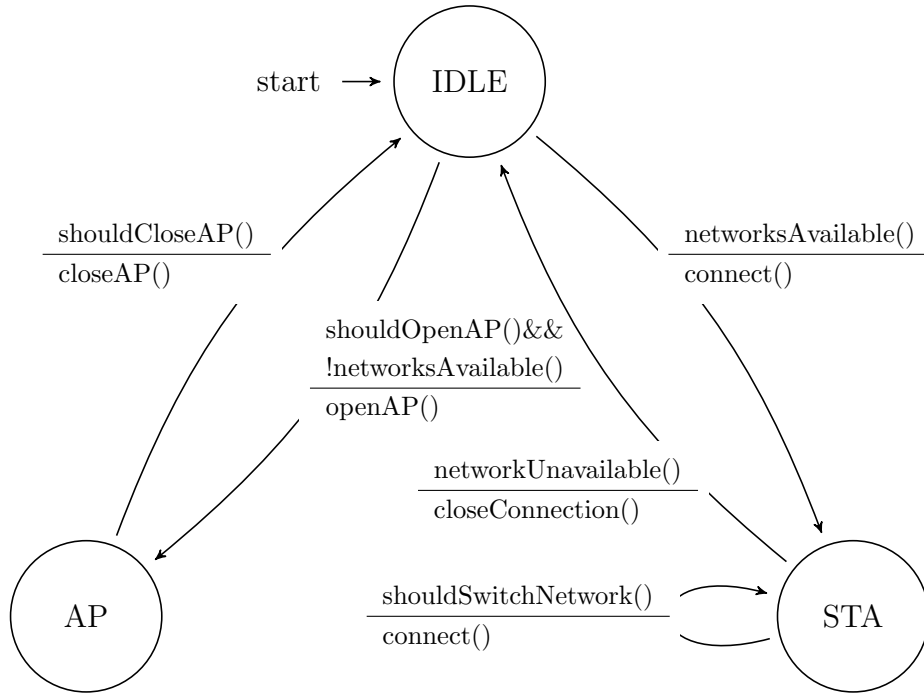


Figure 4.2: State transition diagram of the protocol for opportunistic networking running on each mobile device

4.2.3 Access Point Creation

For each node in state IDLE in NOPPoS, a node switches to AP state and opens an access point with probability $p_{on}^{AP}(N_{acc}^{IDLE}, N_{AP}, t_{off}^{AP})$. Those IDLE neighbors are the most likely candidates to connect to an opened access point since all STA and AP nodes are already occupied. Moreover, only the IDLE nodes in the neighborhood may switch to AP state themselves at the same time.

These nodes have to fulfill the condition in (1), i.e. $t_{off}^{AP} > t_{off,min}^{AP} \wedge N_{acc}^{IDLE} > 0 \wedge N_{AP} = 0$. Till there has elapsed enough time since the last AP state on the device, no other transition to AP state is possible. Opposed to [72], NOPPoS also takes into account the number of observed access points, N_{AP} , and the accurate number of neighbors that are in IDLE state, N_{acc}^{IDLE} , as input values. If there are no IDLE neighbors observed in the neighborhood of the node, an AP state transition will not occur.

Finally, if there is already at least one node in AP state available in the neighborhood, it connects to this access point rather than opening an access point itself and

Table 4.1: Variables of the proposed protocol

Variables	Description
t_{off}^{AP}	Time since last AP state
t_{on}^{AP}	Time in AP state
N_c	Current neighbors in same access point
N_{acc}	Exact number of neighbors
N_{acc}^{IDLE}	Exact number of neighbors in IDLE state
N_{AP}	Exact number of neighbors in AP state

thus changes to STA mode instead. Equation 4.1 states the mathematical expression for the probability for opening an access point:

$$p_{on}^{AP}(N_{acc}^{IDLE}, N_{AP}, t_{off}^{AP}) = \begin{cases} \frac{1}{N_{acc}^{IDLE}} & t_{off}^{AP} > t_{off,min}^{AP} \wedge N_{acc}^{IDLE} > 0 \wedge N_{AP} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

4.2.4 Access Point Switch

For each node in state STA in NOPPoS, a node may voluntarily disconnect from its current access point and connect to a randomly chosen other access point with probability $p_{switch}^{STA}(N_{acc}, N_c, N_{AP})$. This happens when the node determines that there is another access point in its neighborhood whose number of connected nodes in state STA likely exceeds the number of nodes connected to the current access point.

The main objective of nodes in STA state for connecting to another node in state AP (i.e. another access point) lies in encouraging bigger node clusters, to speed up spread of information and to decrease the likelihood of interfering transmissions from other access points. NOPPoS leverages the fact that each node does not only know the exact number of nodes in AP state in its neighborhood, but also the number of nodes residing in its radio range, which are connected to another access point. This is opposed to [72], where the access point switch solely depends on the number of nodes connected to the current access point, N_c . However, the exact number of nodes connected to a specific other access point cannot be determined. NOPPoS assumes that the nodes are evenly split up among these other access points. Therefore, according to (2), a NOPPoS node in state STA only switches to another access point, if it is likely that this access point

contains more STA nodes than the current access point. This likelihood is determined by averaging the number of other nodes in state STA over the number of other nodes in state AP, i.e. $\frac{N_{acc}-N_c}{N_{AP}-1} > N_c$ in Equation 4.2, which a node observes for considering $p_{switch}^{STA}(N_{acc}, N_c, N_{AP})$.

The number of times a node might switch to another access point increases with the number of observed access points. However, the procedure described above decreases the number of access points available (access points bleed out and close themselves). Thus, after a short time, neighboring nodes should be connected to a small number of non-overlapping access points.

To summarize, Equation 4.2 states the mathematical expression for the condition to switch to another access point:

$$p_{switch}^{STA}(N_{acc}, N_c, N_{AP}) = \begin{cases} 1 & \frac{N_{acc}-N_c}{N_{AP}-1} > N_c \wedge N_{AP} > 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

4.2.5 Access Point Shutdown

A node in state AP in NOPPoS closes its access point and switches to state IDLE with probability $p_{off}^{AP}(t_{on}^{AP}, N_c)$. This happens either if no STA node connected after the time $t_{on,min}^{AP}$ passed or the time elapsed since entering AP state exceeds the maximum time $t_{on,max}^{AP}$. The last condition ensures fairness among NOPPoS nodes in terms of energy consumption: the AP state is the most expensive state in terms of energy consumption and, ideally, each NOPPoS node should act as an access point for an equal amount of time.

Note that the condition $t_{on}^{AP} > t_{on,min}^{AP} \wedge N_c = 0$ in Equation 4.3 is influenced by $p_{switch}^{STA}(N_{acc}, N_c, N_{AP})$ of Equation 4.2. This is due to Equation 4.2 indicating that nodes in state STA are likely to switch to another node in state AP in order to build a larger group of connected nodes and thus eventually leaving the node in AP state with no more communication partners. The objective for limiting the time a node resides in state AP lies in providing both fairness for energy consumption among the nodes in an area and speeding up spread of information.

Equation 4.3 states the mathematical expression for the condition for closing an access point:

$$p_{off}^{AP}(t_{on}^{AP}, N_c) = \begin{cases} 1 & t_{on}^{AP} > t_{on,max}^{AP} \vee (t_{on}^{AP} > t_{on,min}^{AP} \wedge N_c = 0) \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

Opposed to [72], Equation 4.3 yields that NOPPoS is considerably less likely to shut down an ongoing data transmission. In fact, this may only happen when the time limit for opening the access point has elapsed, i.e. $t_{on}^{AP} > t_{on,max}^{AP}$. Furthermore, defining p_{off}^{AP} in a probabilistic way creates undesirable scenarios, like two co-located AP nodes simultaneously switching to IDLE state, or the creation of small independent groups, which hamper the global spread of information.

4.2.6 Slotted State Machine

NOPPoS does not evaluate the above state machine continuously, but waits a specific amount of time, the slot time, between two state transition evaluations. Different NOPPoS nodes are not synchronized and slot times are not fixed. This is ensured by selecting the slot time per state uniformly at random from t_{min}^{state} to t_{max}^{state} . This mechanism is used because actual transitions between states take time: e.g. it takes time to open an access point, to scan and to connect to an access point. It has to be prevented that an access point is opened on one device and closed already before it is even scanned and connected to by a neighbor device. That is why [72] also proposed a different slot time for each role so that the probabilities to switch roles are only evaluated after a certain minimum amount of time has elapsed. They also state that it is beneficial to implement the slot time of the IDLE state to be half the slot time chosen in AP and STA state.

4.3 Accurate Neighborhood Discovery

4.3.1 Bluetooth Low Energy

NOPPoS utilizes Bluetooth LE to implement real time continuous tracking of the accurate number of neighbors in the near vicinity. Being especially designed for low power consumption, it provides a power-efficient discovery process. Furthermore, NOPPoS only passively scans the neighborhood for the number of devices. There is no need to connect and actually exchange data. Only the presence of a device in the near

vicinity is relevant. If a device leaves the neighborhood, no more beacons are received. Thus, NOPPoS knows the number of devices just decreased. If a new device enters the scanning area, its beacons are received and NOPPoS learns that the number of neighbors increased.

NOPPoS creates access points only in the 5 GHz band, as this setting is available in state-of-the-art mobile devices (e.g. LG Nexus 5X) and will most likely be the standard for mobile devices, soon. Therefore, neighborhood discovery using Bluetooth LE in the 2.4 GHz ISM band does not affect Wi-Fi communication at all. Additionally, NOPPoS assumes a near equal Wi-Fi and Bluetooth LE beacon range. Fortunately, to minimize energy consumption in Wi-Fi AP mode, mobile devices use lower transmission power than dedicated stationary access points. Bluetooth LE typically utilizes only up to 2.5 mW TX power, and requires less RX power [32] at the receiver to successfully process incoming messages. Moreover, using the 5 GHz band also reduces the Wi-Fi transmission range.

The current NOPPoS state machine state is included in the Bluetooth LE beacons. To accomplish this, NOPPoS leverages the fact that Bluetooth LE advertisements may contain application specific information. This information is used in two different ways by NOPPoS:

1. The state of neighbors is utilized for the state transition calculations.
2. Wi-Fi scans are reduced to a bare minimum. They are only required when NOPPoS decides to switch to the STA mode or to change the access point, to gather MAC address and other protocol information. Simultaneous Wi-Fi and Bluetooth LE scans would contradict the idea of reducing the overall power consumption.

Finally, NOPPoS needs to uniquely identify neighbors, but it cannot rely on Bluetooth MAC addresses. Modern mobile operating systems change them frequently for security reasons. Thus, NOPPoS also adds a unique device identifier to the beacons.

4.3.2 Communication Range Evaluation

One of NOPPoS' most crucial assumptions is the near equal range of Bluetooth LE and Wi-Fi beacons, as Bluetooth LE beacons are used to indicate the Wi-Fi communication

range of neighboring devices. This first appears counter-intuitive, as Bluetooth LE is supposed to require much less energy than Wi-Fi. However, the Bluetooth LE receivers exhibit a higher sensitivity, i.e. they are able to successfully process signals with a much smaller received signal power than Wi-Fi receivers. Typically, Bluetooth LE receivers support signals with up to -105 dBm, while Wi-Fi supports only up to -85 dBm. Also, Wi-Fi transmitters in mobile devices use significantly less transmission power than their router counterparts, whereas Bluetooth LE transmitters typically only support a maximum TX power of 4 dBm by design.

The Wi-Fi and Bluetooth LE range of Google Nexus 5X, a Samsung S7 and a Blackberry Priv Android mobile device were evaluated during the development of NOPPoS, to confirm the near equal transmission range of both technologies. These phones are produced by different manufactures, employ different Wi-Fi and Bluetooth LE chips and cover a significant market share of Android mobile devices. Experimental results with these phones should be representative for a significant number of other Android devices.

First, a Nexus 5X was utilized to transmit Wi-Fi and Bluetooth LE beacons, simultaneously. The embedded Android Wi-Fi access point created the Wi-Fi beacons, while Bluetooth LE beacon transmission was done according to the AltBeacon [1] specification, at the maximum power setting. Then, the Samsung S7 and the Blackberry Priv scanned for Wi-Fi and Bluetooth LE beacons, while their distance to the Nexus 5X was increased. At a distance of about 70 m in an open office environment, both beacons (Wi-Fi and Bluetooth LE) were barely visible and most transmissions were dropped at the receiver. This experiment was also repeated with the Blackberry Priv as transmitter with similar results.

4.4 Simulation-Based Evaluation

4.4.1 Detailed Short-Term Simulation with OMNeT++

The quantitative evaluation of opportunistic networking approaches requires a large number of nodes, appropriate mobility and a longer period of time. Network simulators enable researchers to meet these requirements in a controlled but limited environment. To compare WLAN-Opp and NOPPoS in a most realistic simulation, both protocols have been developed as OMNeT++ [75] modules.

In compliance with NOPPoS assumptions, an equal transmission range of Wi-Fi and Bluetooth LE beacons is also assumed in the simulation environment. This assumption is crucial, as at the time of writing this thesis, Bluetooth LE is not modeled in OMNeT++. Fortunately, it allows to model Bluetooth LE beacon transmission in the network simulator package INET, which is an extension to the discrete event simulator OMNeT++, using Wi-Fi access point beacon transmissions.

Each NOPPoS node builds on the INET WirelessHost compound module with two wireless radios. The first radio uses the IEEE 802.11 AP Management implementation and it is used for the AP state of our state machine. It accurately simulates IEEE 802.11 operations such as authentication, association and beaconing. As described above, its beacons also model NOPPoS Bluetooth LE neighbor discovery. The second interface utilizes the standard IEEE 802.11 STA implementation with a special NOPPoS agent implementation. Finally, two new modules implement the NOPPoS and WLAN-Opp state machines, respectively. They also control the AP and STA interfaces and are added as submodule to the WirelessHost.

As stated in [72], different transitions take different amounts of times to complete. Switching on access point mode (e.g. opening a Wi-Fi hotspot) on Android mobile devices takes approximately 4.5 s and a Wi-Fi scan takes 5 s. This would make 5 s the minimum interval for scanning in IDLE state. Also, a synchronized operation of two or more devices is to be avoided to guarantee fairness. To prevent synchronization, randomness is introduced by defining the timing interval for IDLE state as [5.0, 7.5] s. As stated above, the IDLE state should be slotted at half the slot time chosen in AP and STA state. This leads to a slot time for AP and STA state of [10, 15] s. Furthermore, as access point creation is not instantaneous in reality. Thus, in the following OMNeT++ simulations, the access point is visible in the simulator only after 5 s of simulation time in AP state have passed.

4.4.2 Long-Term Simulation

The drawback of the OMNeT++ INET simulations is the computation time required to run a complete simulation over a long period of time with a significant number of nodes. This is due to the overly accurate simulation of every aspect of the IEEE 802.11 standard. Thus, a stripped down, special-purpose WLAN-Opp and NOPPoS simulator, OPSIM, was developed and validated with runs of up to five hours against

the OMNeT++ implementation. This simulator only implements the state machine and its transitions, as introduced in Figure 4.2. Since the WLAN-Opp and NOPPoS state machines are designed to use the timings of real-world Wi-Fi hardware, the whole IEEE 802.11 standard implementation, including beacon transmission and authentication, is assumed to no longer be of relevance for the simulation. Instead, OPSIM simulations focus on node movement and transmission ranges to evaluate contact opportunities between nodes. Therefore, OPSIM can much more efficiently simulate the crucial performance parameters of the two systems for a large number of nodes and over the appropriate amount of time.

Also, all OPSIM simulations assume a fixed and equal communication range for both Bluetooth LE and Wi-Fi beacons, as the physical layer of the radio channel is not simulated in OPSIM.

4.4.3 Simulator Validation

As just discussed, two simulators are now available to compare WLAN-Opp and NOPPoS: the OMNeT++ INET based simulator that implements most aspects of IEEE 802.11, and the second one, OPSIM, that focuses mainly on the distances of the nodes and the probabilistic slot times. OPSIM is used to evaluate NOPPoS in large scale and crowded environments, like in city or campus areas, as these provide the optimal use cases for opportunistic networks. However, as OPSIM ignores most parts of the IEEE 802.11 and Bluetooth LE protocol, the OMNeT++ based simulator is first utilized to validate that the results of the simplified model of OPSIM still reflect real world scenarios.

The test setting is: 10 nodes in an 100 x 100 m area, random trip mobility model, waiting period chosen uniformly at random from [1, 5] minutes with a speed also chosen uniformly at random from [1, 2] m/s and 5 h simulation time. The simulations are repeated with random node positions five times each. Table 4.2 provides the results. Evidently, OPSIM accurately reproduces the results of OMNeT++: the differences in the state times are very low and the number of IDLE-to-STA and IDLE-to-AP transitions in the state machine show the same qualitative results. The partially significant differences in the number of state transitions have to be attributed to missing IEEE 802.11 beacon transmission and AP authorization / association in OPSIM.

Table 4.2: OMNeT++ vs. OPSIM simulation results

	WLAN-Opp		NOPPoS	
	OMNeT++	OPSIM	OMNeT++	OPSIM
Number of STA states / hour	12.85 \pm 0.78	13.74 \pm 0.46	5.94 \pm 0.17	5.56 \pm 0.34
Number of AP states / hour	4.30 \pm 0.50	4.77 \pm 0.24	1.79 \pm 0.11	1.53 \pm 0.08
Time spent in STA state (%)	74.62 \pm 0.77	74.38 \pm 0.59	79.07 \pm 0.42	78.89 \pm 0.56
Time spent in AP state (%)	18.78 \pm 0.44	18.53 \pm 0.71	16.93 \pm 0.32	16.70 \pm 0.26

4.5 Comparative Performance Study

4.5.1 Performance Study Scenario

The following evaluations focus on the movement of students on a campus. The Huggle project [64] provides the tracked contacts of students from the Cambridge University. The dataset involves 36 nodes and lasts about 11 days containing about 10.641 contact events.

Additionally, the simulations utilize a special class of the random trip mobility model for large-scale, perfect simulations [15]. New trips at transitions are sampled according to their steady-state distribution in the mobility model. In particular, the simulations utilize random waypoint mobility with non-zero pauses. Node speeds in m/s are uniformly distributed in [1, 2]. Node pause times in minutes are uniformly distributed in [1, 60]. The nodes move in a 400 x 400 m region over the course of eight hours.

10 independent simulation runs for each experiment are conducted. The results show the 95 % confidence interval.

Performance Evaluation on Huggle Trace

The first experiment evaluates the real utility of NOPPoS for pocket switched networks. The optimal network would allow neighboring nodes to communicate with each other, as long as they are in transmission range. Therefore, the utilization ratio, as defined in [72] and Equation 4.4, is utilized to compare WLAN-Opp and NOPPoS in three different scenarios: the Huggle trace, the random trip model with 250 nodes / km² and the random trip model with 750 nodes / km² (Figure 4.3).

Table 4.3: Average group size in Huggle trace

System	NOPPoS	WLAN-Opp
Average Group size	16.6 ± 0.7	2.2 ± 0.1
Relative Improvement	754 %	100 %

$$r = \frac{t_{com}}{t_{contact}} \quad (4.4)$$

Evidently, NOPPoS significantly outperforms WLAN-Opp in all three scenarios, with the largest performance improvement in the Huggle trace (33 % better than WLAN-Opp).

Table 4.3 compares the average group sizes. NOPPoS groups are significantly larger than groups created by WLAN-Opp. This is due to the fact that in real world scenarios there are significantly more cliques and NOPPoS can much more accurately determine whether or not to switch to another node in AP state to form bigger clusters. The utilization ratio mainly depends on the efficiency of the state transitions in the state machine and times spent in communication states (i.e. AP state and STA state). A node is able to communicate, when (1) the node is in STA state (at least the access point is a viable communication partner) or (2) the node is in AP state and at least one other node is connected.

While the time spent in STA state is nearly the same in both systems (Figure 4.5), the number of STA transitions (Figure 4.4) is significantly lower for NOPPoS. NOPPoS nodes stay longer in the same access point while WLAN-Opp nodes switch more frequently. This effect is due to the more accurate way to decide when to switch to another access point of NOPPoS. WLAN-Opp nodes do not have any knowledge about their neighborhood in terms of the number of STA or IDLE neighbors. They can only estimate if it would be feasible to switch to another access point based on the number of connected nodes to the same access point. NOPPoS nodes on the other hand can accurately decide whether or not it is feasible and worthwhile to switch to another access point and therefore to risk loss of the current communication, which might not be desirable for systems that exchange large chunks of data.

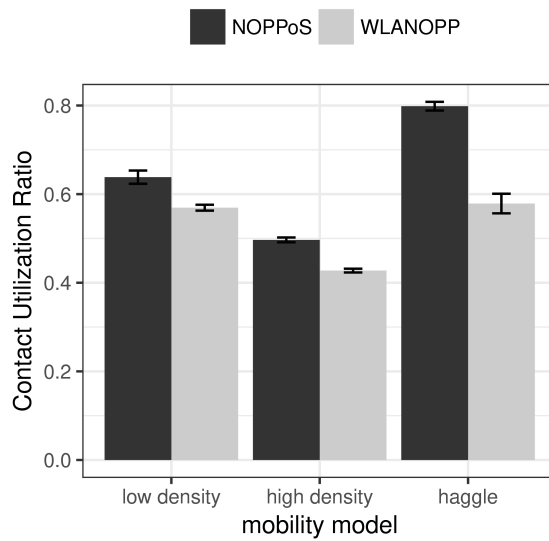


Figure 4.3: Comparison of the utilization ratio of WLAN-Opp and NOPPoS in the random trip model with 250 nodes / km² (low density), 750 nodes / km² (high density) and the Haggel trace

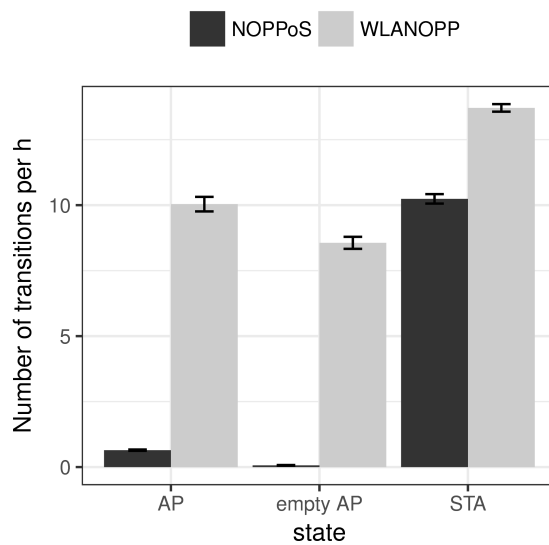


Figure 4.4: Comparison of the number of state transitions per hour of WLAN-Opp and NOPPoS on the Haggel trace

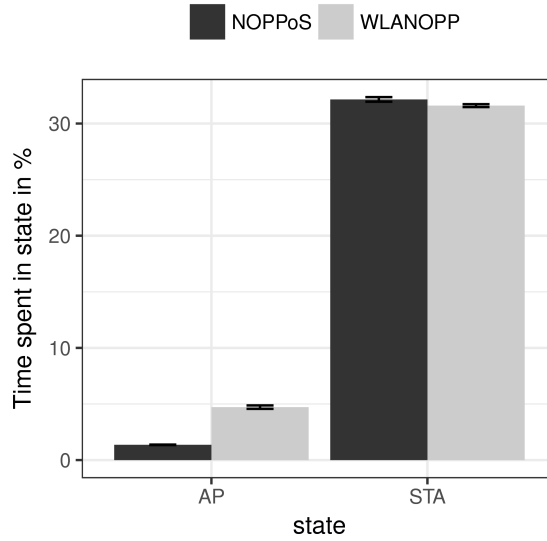


Figure 4.5: Comparison of the time spent in relevant states of WLAN-Opp and NOPPoS on the Huggle trace

4.5.2 Node Density Sensitivity Analysis

The next experiments focus on evaluating how time spent in the states STA and AP, number of state transitions and group size are influenced by the node density in the opportunistic networks NOPPoS and WLAN-Opp.

Figure 4.7 and Figure 4.8 evidently show that most of the access points opened by WLAN-Opp remain empty, with a decrease in the ratio of empty access points with increased node density. This whole observation can be explained with the fact, that WLAN-Opp still opens access points when there are no neighbors around (since it assumes 2 neighbors if the last number of neighbors was 0 and cannot update its backoff interval as fast as the environment changes). Naturally, this happens more often in the sparse node scenarios. With increasing node density, also the probability that there are AP candidates to connect to increases. However, WLAN-Opp closes empty access points just as fast as NOPPoS does, which explains the relatively small effect on the time spent in AP state (Figure 10) (1:10 AP state transitions vs. only 1.25:2.0 AP time for the lowest node density).

Figure 4.6 shows, that the time spent in STA state is nearly identical for both algorithms throughout all node density scenarios with a very small lead for NOPPoS in higher density scenarios. However, WLAN-Opp switches more frequently, resulting in

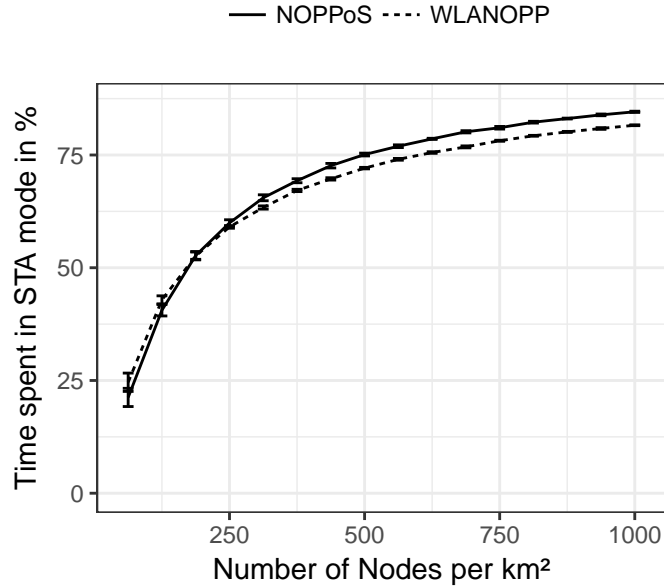


Figure 4.6: Comparison of the average time in STA state

a higher rate of STA transitions (Figure 4.9) increasing with node density. Figure 4.10 shows a significantly shorter time in AP state for NOPPoS, which potentially results in a shorter overall communication time for NOPPoS. However, this time also includes the times no other node is connected to the access point. Fortunately, NOPPoS opens significantly fewer access points that remain unconnected to (empty APs) as can be observed in Figure 4.8. Many of the AP states in WLAN-Opp cannot be used for communication and thus waste energy. Subtracting the empty AP time from the time spent in AP state (Figure 4.10) results in similar effective communication times.

In terms of average group size (Figure 4.11), NOPPoS only opens an access point when there are IDLE nodes in close proximity. This leads to overall significantly larger groups throughout all scenarios. Larger groups result in more diverse communication opportunities and thus enhance the opportunistic network performance.

Overall, NOPPoS offers the same amount of time to communicate among nodes but does not break up potentially ongoing communications as fast, and does not open as much empty access points. Therefore, nodes do not spend as much time in AP state and create larger groups than WLAN-Opp.

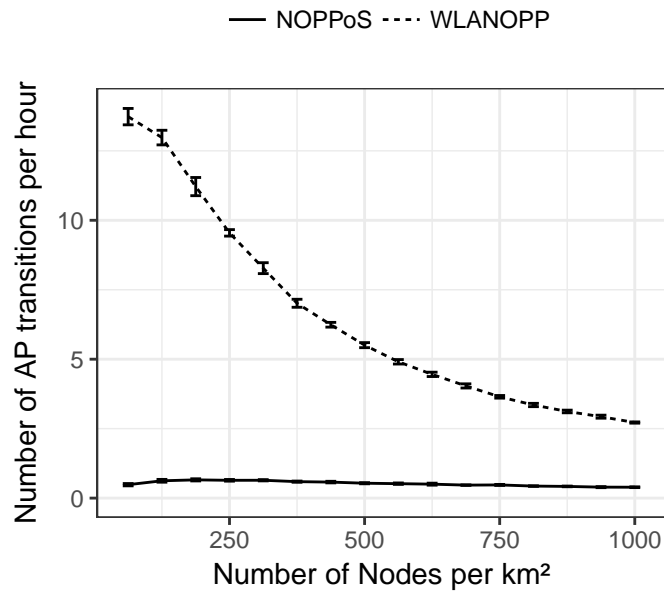


Figure 4.7: Comparison of the average AP transitions per hour

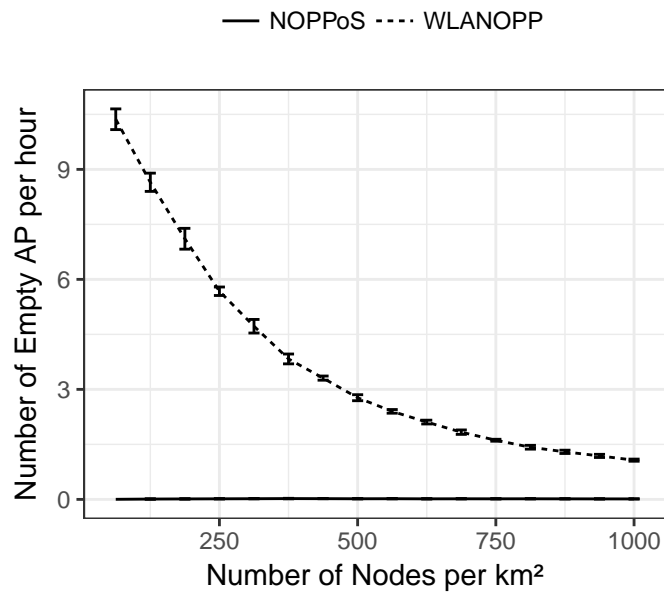


Figure 4.8: Comparison of the average number of empty AP per hour

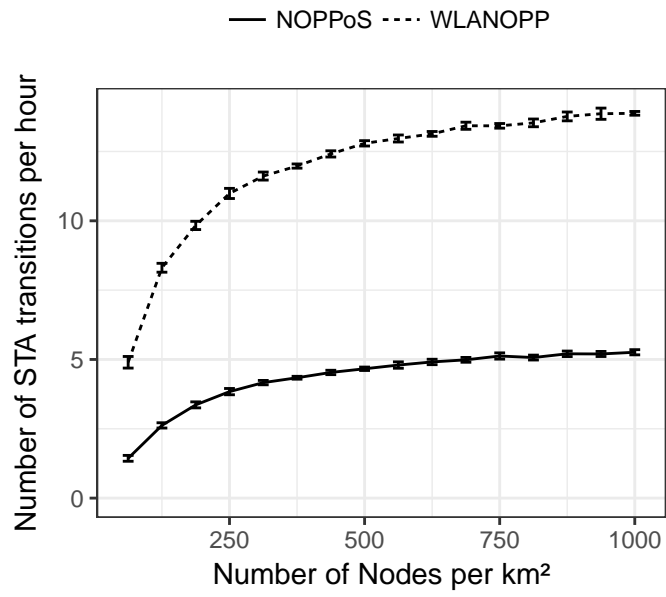


Figure 4.9: Comparison of the average STA transitions per hour

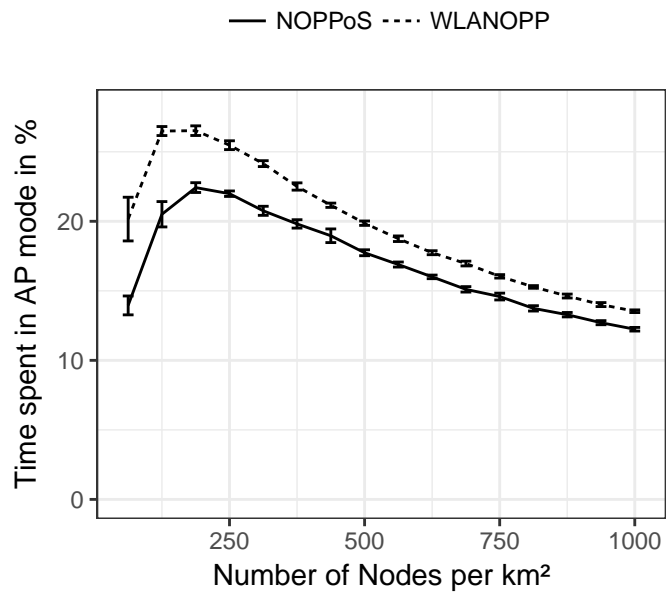


Figure 4.10: Comparison of the average time in AP state

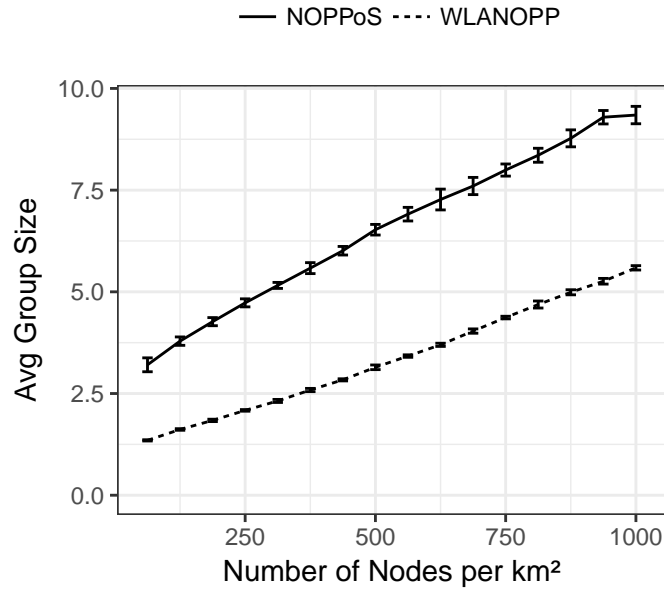


Figure 4.11: Comparison of the average group size

4.5.3 Mobility Sensitivity Analysis

To verify that the above results also hold for different mobility scenarios, the next experiments vary the rest time of the random trip mobility model. Three different rest time scenarios are considered: short (1-10 minutes), medium (1-60 minutes), and long (60-180 minutes). The node density is fixed at 500 nodes / km².

Figure 4.12 shows, that the number of opened access points increases with decreased rest time. This is due to the increased mobility, which is associated with an increased churn rate. Nevertheless, the effect of different rest times on the time spent in STA and AP states is negligible (Figures 4.13, 4.14). The communication time remains the same for all scenarios.

4.5.4 Energy Consumption

Finally, NOPPoS radically reduces power consumption when compared to WLAN-Opp. Considering neighbor discovery, NOPPoS effectively replaces Wi-Fi scanning, utilized by WLAN-Opp in IDLE and STA state, with Bluetooth LE inquiring and inquiry scanning (beacon transmission and reception). The following values are calculated based on the specifications of an LG Nexus 5X. It incorporates the QCA6234 integrated dual-band Wi-Fi and Bluetooth 4.0 combined chip [3]. The relevant power

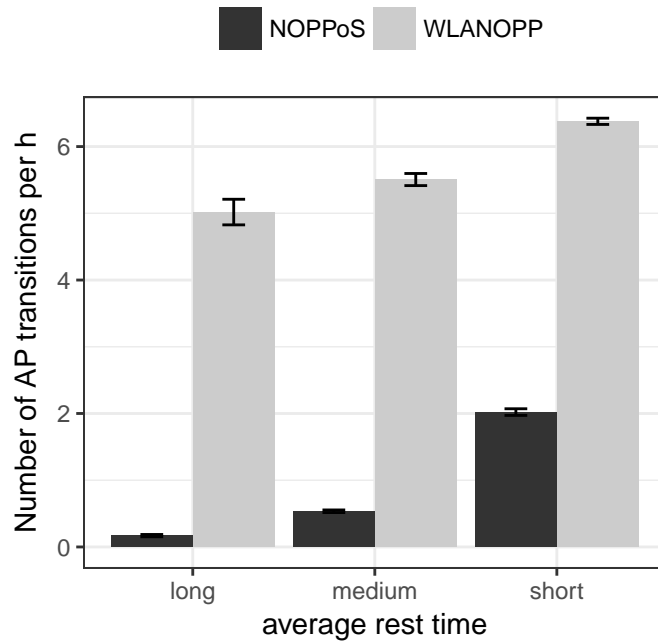


Figure 4.12: The number of AP transitions for various rest times of the random trip mobility model

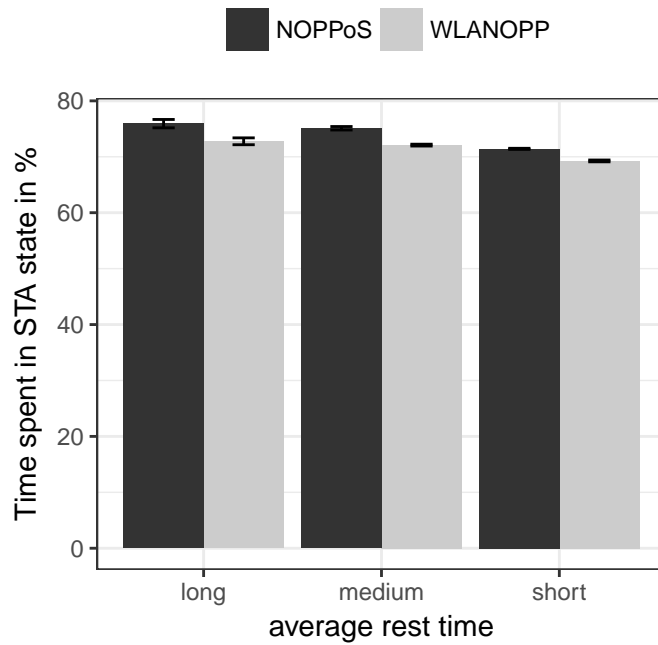


Figure 4.13: The time spent in STA state for various rest times of random trip mobility model

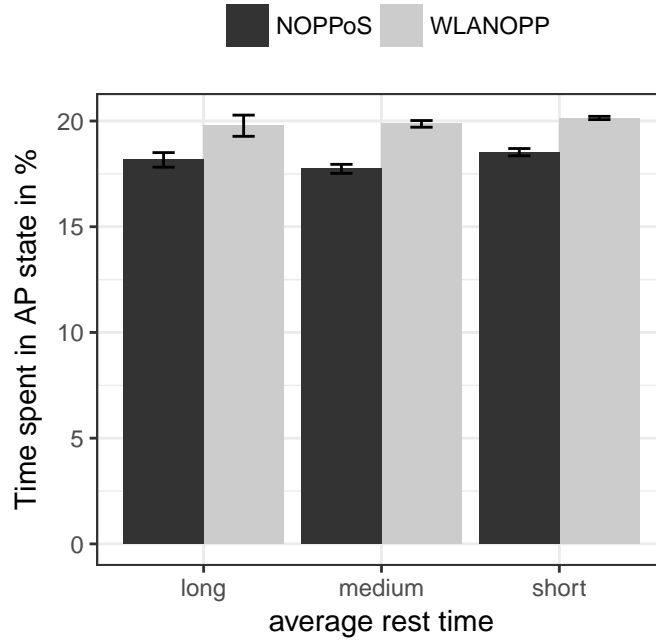


Figure 4.14: The time spent in AP state for various rest times of the random trip mobility model

Table 4.4: Wireless energy consumption of QCA6234

System	Continuous RX (mW)	Continuous TX (mW)
Wi-Fi 2.4 GHz 54 Mbps	227.7	824.9
Wi-Fi 5 GHz 54 Mbps	247.5	989.9
Bluetooth Inquiry	0.873	39.6

consumption specifications are depicted in Table 4.4.

The continuous combined power consumption of Bluetooth results in 40.5 mW. A typical Wi-Fi scan on Android mobile devices takes 4,5 s of the 5 s scan interval leading to a RX duty cycling of 90 % or 222.75 mW at 5 GHz. This means NOPPoS reduces power consumption while in discovery phase by 87 %. This is achieved by only activating Wi-Fi scanning once the Bluetooth LE discovery phase yields possible neighbors in AP mode. Avoiding to open empty access points more efficiently also helps reducing the power consumption in AP mode by over 20 % depending on the node density (Figure 4.10).

4.6 Conclusion

This section introduced NOPPoS, a neighborhood-aware opportunistic networking approach on smartphones. As major contribution, NOPPoS is governed by refined equations for the state transitions at each node. These equations are based on knowing the exact number of other nodes in the radio range of each node. It was shown that the number of other nodes in the neighborhood can be accurately determined by periodic, low-energy scans. Therefore, NOPPoS is both highly responsive to node mobility and energy-efficient.

NOPPoS was evaluated utilizing the Huggle mobility trace. The presented quantitative results evidently show that NOPPoS outperforms the approach WLAN-Opp. NOPPoS creates larger groups than WLAN-Opp, nodes spend less time in AP state. Thus, energy consumption is further reduced and the contact utilization ratio is increased by up to 33%.

5 Applications for Opportunistic Gigabit Networks

This chapter builds on ideas of the NOPPoS protocol from Chapter 4. First, parts of NOPPoS are refactored and tightly coupled with a cross-layer optimized document sharing application to create Neighborhood Document Sharing (NDS). Afterwards, a recommendation system is described that can be utilized to retrieve interesting documents automatically using NDS.

5.1 Neighborhood Document Sharing

The first proposed new application for IEEE 802.11 based P2P networks is a proximity-based document transfer protocol. It is called Neighborhood Document Sharing, short NDS. In contrast to the opportunistic networking scheme in the previous chapter, a cross-layer solution is developed that tightly couples the network and the application. It enables users to discover and retrieve arbitrary documents shared by other users in their proximity, i.e. in the communication range of their IEEE 802.11 interface. However, IEEE 802.11 connections are only used on-demand during file transfers and indexing of documents in the proximity of the user. This saves energy and minimizes the use of the IEEE 802.11 interface for only high-throughput operations.

In contrast to widely available solutions like Airdrop, documents are not pushed from sender to receiver. The user has a coarse overview of all documents available in his proximity, independent of the users that are sharing the documents. Essentially, only the user that is retrieving documents is interacting with the application and downloads interesting documents.

Just like in NOPPoS, Bluetooth LE is employed additionally to the Wi-Fi interface to discover other NDS devices. It is used to broadcast the device status (e.g. NDS

role and device id) to neighbors and to communicate the WPA2 pre-shared key and SSID to join the network. This is done by creating specially crafted Bluetooth LE advertisements with proprietary GATT services.

5.1.1 Related Work

Additional to the related work of NOPPoS, NDS touches another field of research. NDS can be seen as an implementation of the Information-Centric Networking (ICN) paradigm where devices are only interested in retrieving selected content. All requests on network layer only focus on content. NDS describes the components to implement an ICN on off-the-shelf smartphones.

Hail, Amadeo, Molinaro and Fischer [34] proposed caching and content forwarding strategies that could be easily implemented in NDS to extend NDS to a complete ICN system.

Lindemann and Waldhorst [49] proposed in 2004 Passive Distributed Indexing (PDI). PDI caches the broadcasted queries of all interconnected devices in an IEEE 802.11 network. However, NDS aims to minimize the IEEE 802.11 connections between mobile devices and utilizes special device roles to manage the index. Broadcasts are not required.

5.1.2 The Basic Approach

NDS differentiates two main device roles: User and Index (Figure 5.1). Index devices are responsible for creating an index of all documents available on User devices in the proximity. To do this, an Index device creates a Wi-Fi access point and is available to User devices to drop the list of their documents and to retrieve the list of documents of other users. User devices looking for available documents only have to connect to all Index devices in their neighborhood. A connection to all NDS devices is unnecessary. Also, a permanent connection to another NDS device or some kind of IEEE 802.11 access point is not required. There are two other intermediate roles, Download and Index Negotiation. They are only used for very brief moments. The Download role is used during document downloads and Index Negotiation to minimize adjacent IEEE 802.11 access points.

All NDS devices transmit Bluetooth LE advertisements periodically, containing NDS

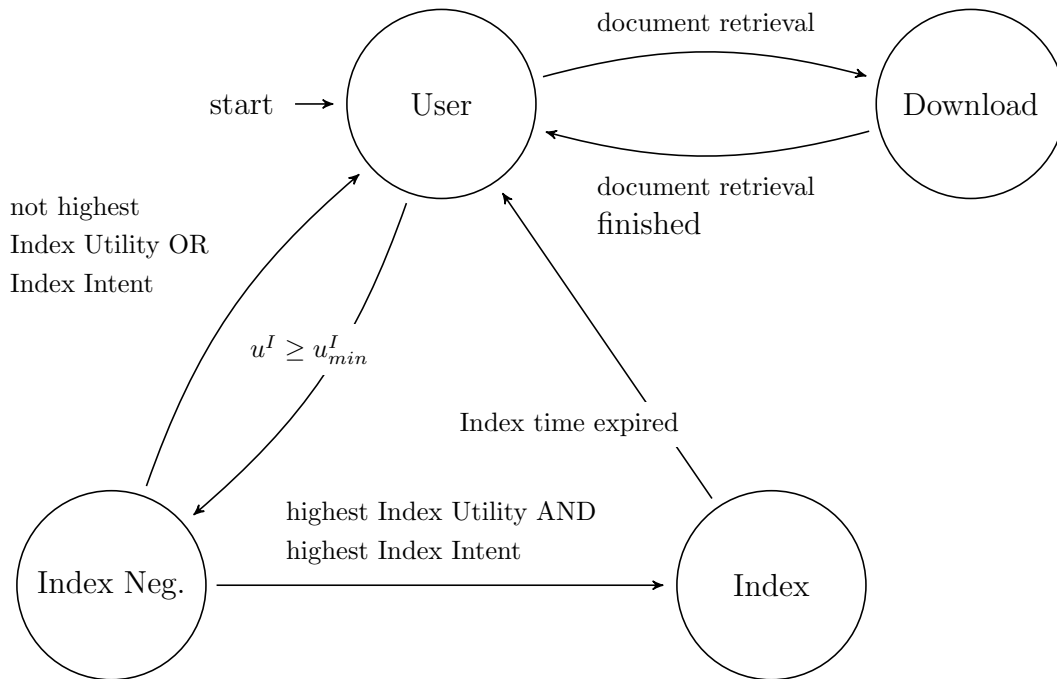


Figure 5.1: NDS roles

meta data: at least the unique NDS device ID and the current NDS role. NDS Users and Indexers have additional role-specific fields, which will be discussed shortly. These advertisements allow any NDS device to determine all other available NDS devices in the proximity. Just like NOPPoS, NDS has the basic assumption that the transmission range of Bluetooth LE is nearly identical to the range of IEEE 802.11. Thus, each NDS User merges the document information available at NDS Indexes and the location information provided by the Bluetooth LE advertisements to determine the documents available at any time.

NDS also aims to make multiple NDS Indexes available to each NDS User. This tries to solve the issue that the transmission range of an NDS User might differ significantly from the range of the NDS Indexes (see Figure 5.2). For example, an NDS device located 50 m away from the only NDS Index is able to transmit data via Wi-Fi with NDS Users the NDS Index has never seen. This is also in contrast to NOPPoS, where all devices can only communicate with other devices connected to the same access point. Even if NOPPoS devices stand next to each other, they are not able to exchange information if they are part of different NOPPoS groups.

The Index role is significantly more expensive in regard to energy consumption, in

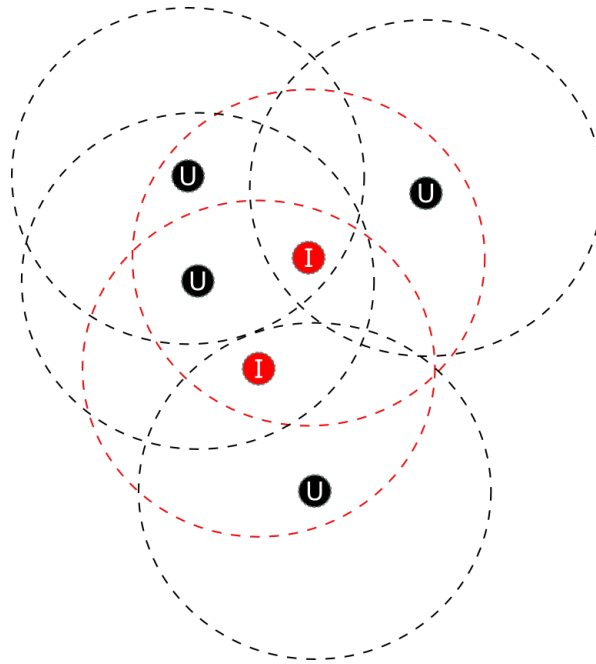


Figure 5.2: NDS example neighborhood with Index and User devices

comparison to the User role. Therefore, the Index role is dropped regularly to force other NDS devices to become one of the next Indexes. This is similar to NOPPoS. However, the algorithm to determine when to switch to the Index role is very different. For example, an Index is not always required. As long as no new devices enter the neighborhood and the devices do not alter their documents, the state all NDS devices received from the previous Index remains valid. As soon, as a new device is discovered or a device changes its document collection, e.g. by adding a new document, an index update on all devices in the neighborhood is required. The following approach aims to equally share the burden of being an NDS Index device between all devices.

Each NDS User decides independently whether an additional Index is required or not. To enable this assessment, each NDS device also broadcasts the number of Indexes it is able to access and calculates a numerical value, the Index Utility u^I , periodically. NDS Index devices count themselves as accessible Index. The Index Utility u^I provides a heuristic to assess the benefit of creating an Index at an NDS device. If the assessed utility is above or equal a certain threshold u_{min}^I , Index creation is triggered. The calculation of the Index Utility is based on the notion that any Index has a utility whenever it makes previously unknown devices known to each other (see Figure 5.3).

Moreover, whenever there is at least one neighbor that does not have access to at least one Index, the Index Utility is also supposed to be equal or higher to the threshold u_{min}^I to help this device and to start the Index creation procedure immediately. Equation 5.1 provides the basic Index Utility function that will be used in the remainder of this section. It does not completely capture the notion of newly connected NDS devices but will provide the baseline for future improvements. Also, it enables a straightforward implementation utilizing the Android Bluetooth LE library, as the transmitted NDS device status has to fit in the Bluetooth LE advertisements.

$$u^I = \sum_{k=0}^m N_k \frac{1}{4^k} \quad (5.1)$$

In Equation 5.1, N_k is the number of neighbors that currently have access to k Indexes. Thus, each neighbor increases the utility (because he gets access to a new Index). However, neighbors with a low number of Indexes increase the Index Utility significantly more than neighbors with a high number of Indexes. m is the number of potential Indexes in the neighborhood. This equation requires to set $u_{min}^i = 1$, so $u^I \geq u_{min}^i$ to start the Index creation process, whenever there is one neighbor with no access to any Index, as discussed above.

Before finally switching to Index role, the User switches to Index Negotiation role to tell the neighbors of its intent to create an additional Index, broadcasts its Index Utility and starts a timer. As soon as this timer runs out, it looks at all other received Index Utility values. If its own value is highest, it moves on to Index role. Otherwise, it switches back to User role. An additional value that is also transmitted in Index Negotiation role is the Index Intent. Just like the GO Intent in Wi-Fi Direct or the master metric in AWDL, the Index Intent considers battery level, computational capacities and acts as tie breaker. When an NDS device with the highest Index Utility detects, there is another device with the same Index Utility, but higher Index Intent, it also falls back to User role.

NDS devices postpone the transmission of Bluetooth LE advertisements when switching to User role for some seconds, to first properly assess the neighborhood. This avoids unnecessarily triggering other NDS devices to starting Index creation.

To finally retrieve documents, an NDS User device switches to Download role, creates its own IEEE 802.11 access point and makes the list of files, it is interested in, available in its Bluetooth LE advertisements. Any NDS User in transmission range

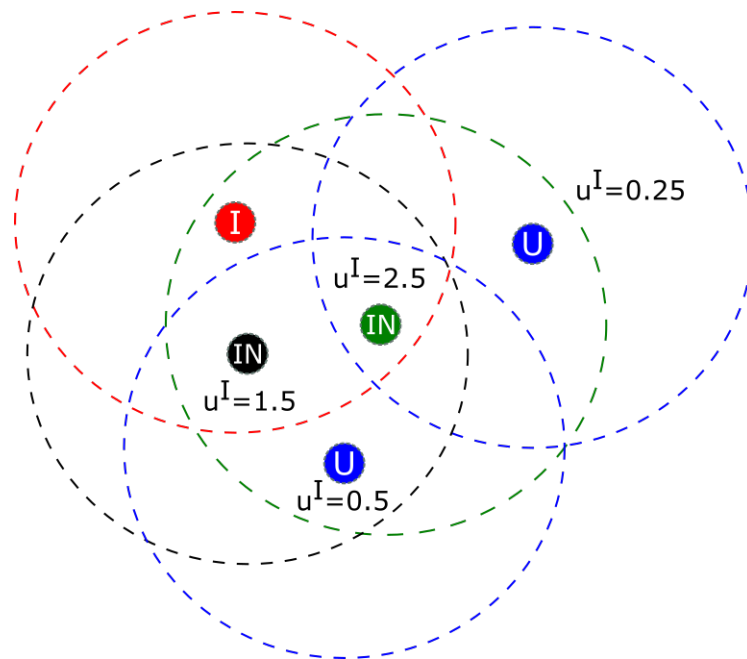


Figure 5.3: NDS Index Utility example. The blue devices are new. The red device is the current only Index. The green device has an Index Utility above the threshold and the highest Index Utility of all non-Index devices, so will soon switch from Index Negotiation role to Index role.

receives this document request. In theory, each requested document should be available at least from one NDS User in the neighborhood. These NDS Users will connect to the IEEE 802.11 access point of the requesting NDS Download device to provide the documents. Multiple NDS Users potentially connect simultaneously. Thus, the requesting NDS Download device has to manage multiple Wi-Fi peers and might be able to use parallel downloads from different devices. This connection is not required to be instantaneous. NDS Users providing the requested file might be currently fulfilling the document request of another NDS user or decide another available NDS User is more equipped to handle the request. Nevertheless, the requesting NDS User just leaves the requested document in the Bluetooth LE advertisements till it is provided or the user decides the document is not required anymore.

5.1.3 Performance Study

The discrete event simulator OPSIM, which was initially developed to evaluate NOPoS (Section 4), is now extended to the NDS protocol. It implements the three NDS

Table 5.1: NDS simulation based on the Huggle campus dataset: utilization ratio and number of concurrent Index devices.

Index time limit	Mean utilization ratio	Mean # concurrent Index devices
5 min	0.928 ± 0.0088	2.76 ± 0.0919
10 min	0.958 ± 0.013	3.21 ± 0.148
20 min	0.971 ± 0.009	4.09 ± 0.315

roles User, Index Negotiation and Index. The Download role is currently not implemented as this role is not required for the following evaluations. Index Negotiation role lasts at most 10 seconds. When this time has passed, the NDS device is either in Index role or in User role. The document collection of each NDS device is assumed to be static.

The following evaluation focuses again on the movement of students on a campus. Therefore, first, NDS is evaluated on the 36 nodes dataset from the Huggle project [64]. The movements and contacts of the first 30 nodes of this dataset are utilized during the first 10 hours. The performance of NDS is evaluated with the utilization ratio as introduced in Section 4.5.1. However, while in WLAN-OPP and NOPPoS, devices have to be connected to the same Wi-Fi access point to communicate with each other, NDS requires both devices to have access to at least one shared Index. This increases the utilization ratio significantly (Table 5.1). All results are presented with the 95 % confidence interval. The simulations are repeated five times each.

Keeping devices longer in Index role increases the utilization ratio, however, even more significantly increases the number of Index devices. The latter is a result of device movement: the utility of some Index devices diminishes with time. 10 minutes Indexes appear to optimally balance power consumption and the utility of the protocol.

The second experiment analyzes how the number of Indexes develops with an increasing number of NDS devices. The test setting is: 20, 40 and 80 devices in an 400 x 400 m area, 10 minute Indexes, random trip mobility model, waiting period chosen uniformly at random from [1, 60] minutes with a speed also chosen uniformly at random from [1, 2] m/s and 10 h simulation time. The simulations are repeated with random node positions five times each. Table 5.2 has the results.

The utilization ratio is nearly identical for each number of devices and also comparable to the utilization ratio achieved with the Huggle trace when using 10-minute

Table 5.2: NDS simulation using the random trip mobility model. The utilization ratio and number of concurrent Index devices with increasing number of NDS devices and 10 minute Indexes.

Number of devices	Mean utilization ratio	Mean # concurrent Index devices
20	0.963 ± 0.0041	8.55 ± 0.199
40	0.962 ± 0.0030	16.96 ± 0.223
80	0.961 ± 0.0030	26.63 ± 0.360

Indexes (see Table 5.1). However, the number of concurrent Index devices is significantly increased. This is due to the significantly larger area and the lower density of devices.

5.1.4 Off-the-Shelf Hardware Implementation

Android supports the creation of normal IEEE 802.11 access points as well as Wi-Fi Direct groups. The latter have the added benefit to also provide service discovery features and to restrict routing so that connected devices do not route all their app traffic through the access point. Only the pre-shared key required for WPA2 encryption is provided by the OS and no app can provide its own pre-shared key. Fortunately, the key selected by the OS is communicated to the app, so other NDS devices only have to retrieve the key e.g. through Bluetooth LE. As Bluetooth LE is already the core of the NDS device discovery protocol, it is straightforward to implement a GATT service to retrieve the pre-shared key and also the SSID of the access point created by an NDS Index or Download device.

Android also has support for Bluetooth LE central and peripheral roles. While especially the peripheral role was only supported by high-end devices in the past, most modern Android devices now support both roles. Peripheral role is required to create the Bluetooth LE advertisements used to transmit NDS status and also to implement the GATT services that provide crucial parts of NDS.

iOS does not allow to create IEEE 802.11 access points by apps, however, has stable support for Bluetooth LE, even for the peripheral role. Therefore, the current NDS implementation focuses on Android and requires NDS Index devices to run the Android OS.

5.1.5 Improvements

Up until now, the Index Utility does not really capture the notion of additionally interconnected devices. It is just a coarse heuristic. To improve the Index Utility calculation, the device status transmitted by each device has to include the device IDs of all accessible NDS devices in the neighborhood. Unfortunately, this is too much data to include in Bluetooth LE advertisements. NDS devices could implement an additional Bluetooth LE GATT service, to enable other NDS devices to actively query these Index device IDs. However, this most likely increases energy consumption, delay and congestion in the 2.4 GHz band, significantly. Moreover, it is not clear whether this more accurate calculation would result in a significant better Index device selection and Index quality.

Also, caching has not been discussed up until now. Caching is crucial to retrieve files that are not available via one-hop connections between two adjacent NDS devices. Like in any other opportunistic document sharing application, NDS devices can at least remember the meta data of each document they have ever encountered and submit this information to NDS Indexes. So, over time, every NDS device accumulates a list of all documents available over multiple hops. NDS devices also store the information, which document is available in the direct neighborhood and which requires additional steps. Finally, each device transmits its interest on not currently available documents to its neighbors. If another NDS device happens to encounter this document somewhere else, it can decide to add this document to its cache, based on the available cache size and the number of interested NDS devices.

5.1.6 Conclusion

This section introduced the Neighborhood Document Sharing (NDS) protocol to provide an opportunistic and energy efficient document Index and retrieval technology for off-the-shelf smartphones. It builds on the ideas of NOPPoS, utilizes smartphones temporarily to host a document index of devices in the neighborhood and utilizes direct device-to-device connections to retrieve documents.

Evaluations in the network simulator OPSIM show that NDS makes the document corpus of all devices known to most neighbors. Android implementation details have been presented. The next step is to completely implement NDS for Android and to evaluate the protocol on real devices.

5.2 User Preference-Based Probability Spreading

The document sharing approach NDS, as presented in the previous section, relies on user interaction to select the interesting documents for download from other users. However, this approach can easily be extended by an automatic retrieval based on content recommendation. Numerous popular applications allow their users to label their content with freely assigned terms denoted as tags or hashtags. Examples are the MovieLens DB (movie tracks) [37], the highly popular social networks Facebook (text, pictures, and videos), Instagram (photos and short videos), last.fm (audio tracks), Twitter (short text messages), and YouTube (videos). This phenomenon of labeling content with freely chosen tags from an uncontrolled vocabulary is denoted as a folksonomy.

Folksonomies automatically build relationships between content and the tags assigned to this content, as well as between each user and the tags associated with content this user has consumed or owns. The latter constitutes highly personalized information, and, hence, give rise for collecting user profiles. Since folksonomies can also quickly adapt to changes in users' interests, recommender systems can effectively be built upon a folksonomy.

Several approaches have been proposed for tag-aware recommender systems in the literature. In the survey [83], such systems are divided into graph-based, tensor-based and topic-based systems. Graph-based systems inspect the tripartite graphs between users, items and tags while tensor-based systems use tensors as a different representation. Topic-based systems employ the relations between the tags and, therefore, are able to produce more comprehensible recommendations.

User Preference-based Probability Spreading for content recommendation (UPPS) also constitutes a graph-based approach. It integrates user-item scoring into a graph-based tag-aware item recommender system. Building upon the ProBS [84] and PLIERS [9], [8] methods, UPPS utilizes refined formulas for affinity and similarity scoring, taking into account user-item preference in the mass diffusion of the recommender system.

The approach in this chapter was collaboratively developed and published by the author of this theses and his co-authors in [29]. Michael Petrifke proposed the two-step similarity score.

5.2.1 Related Work

Most recommendation system approaches utilize the user-item matrix, especially all collaborative filtering algorithms. State of the art research results in collaborative filtering were presented in a recent survey by Shi, Larson and Hanjalic [66]. In the survey [83], tag-aware recommender systems were divided into graph-based, tensor-based and topic-based systems.

Tso-Sutter, Marinho, and Schmidt-Thieme demonstrated a first approach to build a tag-aware recommender system [74]. They proposed a generic method that reduces the ternary correlations to two-dimensional correlations. This approach enabled them to apply standard collaborative filtering algorithms. Opposed to [74], UPPS weights the user and item relationship. Additionally, UPPS does not require the transformation step of [74], and, hence, introduces no information loss that might lower the recommendation quality.

Zhou, Kuscsik, Liu, Medo, Wakeling, and Zhang introduced ProbS and HeatS as two graph-based approaches [84]. Both methods calculate an item score for the items of all the users that have items in common with the currently inspected user. While ProbS highly promotes popular items, HeatS recommends items with low popularity. Furthermore, they presented a hybrid approach that combines ProbS and HeatS for improving recommendation results. Arnaboldi, Campana, Delmastro, and Pagani presented PLIERS, which builds upon the mass diffusion process [9]. PLIERS favors items with a similar popularity over items owned by the user. In [8], Arnaboldi and his co-workers showed how to employ PLIERS for content-dissemination in mobile, distributed networks. Lü and Liu introduce Preferential Distribution (PD) that builds a biased mass diffusion process, which redistributes more mass to items with low popularity [51].

ProbS, HeatS, PLIERS and PD only consider features that are directly related to the graph structure, such as the number of users connected to an item. For example, a video, which is watched in full length or is highly rated by a user, has the same preference for being included in the recommended item set as a video, which is not watched completely or is rated low. Opposed to [9], [8], [51], [84], UPPS considers a score to assess the quality of the user-item relationship and utilizes this score to modify the mass diffusion on the user-item and item-tag graphs.

Wu and Zhang proposed a scoring function similar to TF-IDF using the tags as-

signed to items [79]. They incorporate weighting into a graph-based recommendation algorithm. Opposed to [79], UPPS incorporates features into a scoring function, which can handle the user-item relationship based on user preferences.

Gemmell, Schimoler, Ramezani, Christiansen, and Mobasher proposed a weighted hybrid recommender by combining graph-based recommendation with item-based collaborative filtering [31]. They chose an adapted variant of the well-known PageRank algorithm, denoted by FolkRank, as graph-based recommendation system. Opposed to [31], UPPS is computationally inexpensive and, therefore, very well suited for mobile devices.

Zhang, Zhou, and Zhang first proposed a diffusion approach on user-item-tag tripartite graphs by splitting the graph into two bipartite graphs [82]. They applied mass diffusion to both bipartite graphs and used a linear combination to aggregate the results. Opposed to [82], UPPS integrates a user item preference into the mass diffusion on both graphs. Additionally, UPPS introduces a two-step similarity score that finds items that are related to the items, but not directly connected to tags of the target user.

5.2.2 Background

A typical recommender system has access to the items that are associated with each user (e.g. watched or ranked videos) and the tags that are attached to each item (i.e. folksonomy). It describes relationships between elements of the following sets:

$$\text{Users } U = \{u_1, u_2, \dots, u_n\}$$

$$\text{Items } I = \{i_1, i_2, \dots, i_m\}$$

$$\text{Tags } T = \{t_1, t_2, \dots, t_r\}$$

The relationships can be modeled using adjacency matrices. The corresponding adjacency matrix for the user-item relationship is denoted by A . If a user u_i collects an item i_j , then $a_{ij} = 1$ and 0, otherwise. Similarly, the item-tag relationship is represented by the matrix A' where the entry $a'_{jk} = 1$ if an item i_j has assigned the tag t_k and 0, otherwise.

Representations of folksonomies either use two bipartite graphs or one tripartite graph. Item recommendation methods that work on these graphs are called graph-

based methods. One example for graph-based methods is Probability Spreading (ProbS) [84]. Given a target user u_t , ProbS produces a score for each item utilizing mass diffusion. That is, the score of an item i_j is calculated as:

$$f_{j,t}^{probs} = \sum_{l=1}^n \sum_{s=1}^m \frac{a_{lj}a_{ls}a_{ts}}{k_I(u_l)k_U(i_s)}, j = 1, 2, \dots, m \quad (5.2)$$

$k_I(u_l)$ represents the number of items of user u_l . $k_U(i_s)$ is the number of users related to item i_s . The basic principle of mass diffusion is that each item of the user has an initial resource assigned to it. Subsequently, this resource is redistributed equally to all users that are connected to this item. In the next step, the users distribute the received resources again equally to all items they are connected to. The resources received at each graph vertex are summed up. The resulting sum of resources at each item is utilized as an item score for the recommendation.

ProbS favors popular items since there exist more users that distribute the resource to those items. However, PLIERS [9], building upon ProbS, favors items with similar popularity over items the users already own. To achieve this, PLIERS utilizes an additional weight. For target user u_t , PLIERS calculates the score of an item i_j as:

$$f_{j,t}^{pliers} = \sum_{l=1}^n \sum_{s=1}^m \frac{a_{lj}a_{ls}a_{ts}}{k_I(u_l)k_U(i_s)} \frac{|U_s \cap U_j|}{k_U(i_j)}, j = 1, 2, \dots, m \quad (5.3)$$

U_s and U_j are the sets of the users connected to item i_s and i_j , respectively. PLIERS measures the ratio of the overlap between those two sets to the total number of users connected to i_j . The portion of the resource that was distributed originally from item i_s to i_j is weighted with the overlapping ratio. Therefore, in PLIERS, items with a low number of connected users shared with the target item contribute less to the resulting item score than items that have many users in common with the target item. Equation 5.2 can be extended to the item-tag graph by switching the entries of matrix A to those of A' . A modification of PLIERS to the item-tag graph has also been introduced in [9].

Subsequently, the scores calculated on the user-item graph are referred to as affinity score and scores calculated on the item-tag graph as similarity score. Finally, these single scores are linearly combined to compute the overall recommendation score for each item.

5.2.3 User Preference-Based Probability Spreading

ProbS and PLIERS are based on the idea that a user is in some way connected to an item. This connection might be derived from the fact that a user watched a video, downloaded an item or just added an item to a list. In many application scenarios, more information can be derived from the connection of the user to an item, to model the preference of the user to this item. An often-available feature is a manual user rating, e.g. a like or a dislike, which is assigned to an item by the user. Further evidence of the user preference can be found implicitly by evaluating how the user interacts with the item, e.g. watching a video multiple times or ending the playback of a video after a short time.

This idea can be incorporated into the resource redistribution in the mass diffusion process. The mass diffusion process of ProbS and PLIERS [9], [84] assigns the same initial resource to all items of a user and redistributes the resources equally. In contrast, User Preference-based Probability Spreading, denoted as UPPS, utilizes resources reflecting the user preferences. If the user prefers an item over another item, it should have a higher impact on the recommendation score.

Affinity Scoring

The UPPS affinity score requires three scoring functions: S_1 , S_2 , and S_3 . The function S_1 assigns the initial resources and reflects the user-item preferences. The function S_2 determines the distribution of the resources from the items to the other users in the second step of the mass diffusion. The function S_3 distributes resources from those users to the target items as third step of the mass diffusion process. The whole process in UPPS is described by the following equation for target user u_t :

$$f_{j,t}^{aff} = \sum_{l=1}^n \sum_{s=1}^m \frac{S_3(u_l, i_j) \cdot S_2(u_l, i_s) \cdot S_1(u_t, i_s)}{r(u_l) \cdot r(i_s)}, j = 1, 2, \dots, m \quad (5.4)$$

$r(i_s) = \sum_{l=1}^n S_2(u_l, i_s)$ is the sum of all S_2 associations between users and the item i_s and $r(u_l) = \sum_{s=1}^m S_3(u_l, i_s)$ is the sum of all S_3 associations between user u_l and his items. The normalization by $r(i_s)$ and $r(u_l)$ guarantees that the sum of all $f_{j,t}^{aff}$ equals the sum of the distributed resources S_1 . In contrast to S_2 and S_3 , a normalization is not necessary for S_1 because it would not change the order of the final results.

At this moment, the scoring functions are not further defined. They depend on

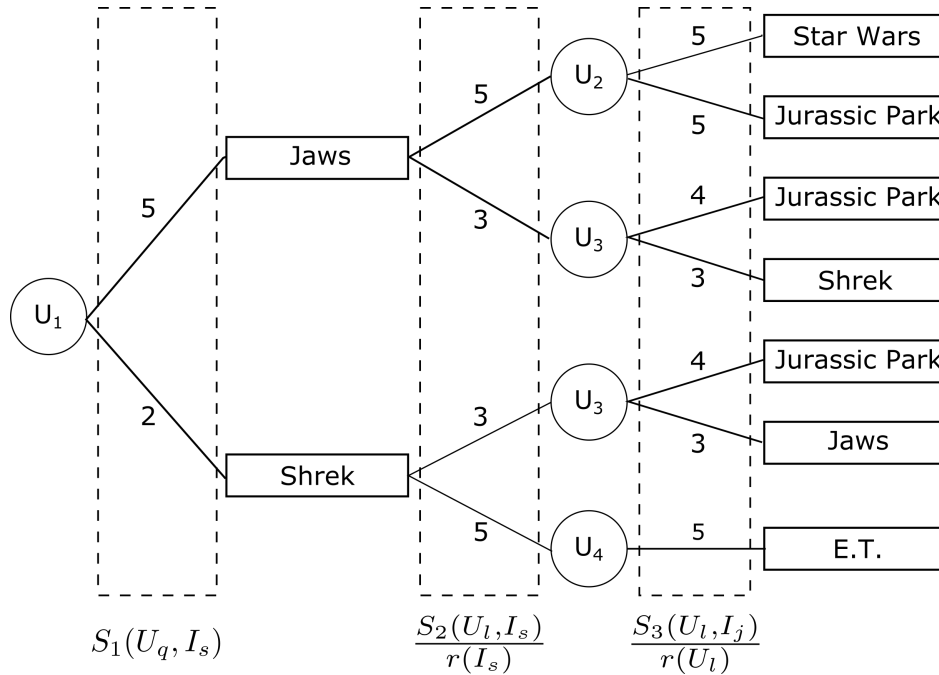


Figure 5.4: Example illustrating the user-item graph of UPPS

the use case and the available features. Each scoring function might linearly combine the values of the different available features to estimate the user-item preference. A machine learning algorithm might be appropriate, if multiple features are available. A reasonable assumption is that the score has to be 0, if no edge between user and item exists in the graph. The formula is identical to ProbS, if the scoring functions are reduced to the appropriate entries of the adjacency matrix, i.e.: $S_1(u_t, i_s) = a_{ts}$.

Figure 5.4 shows an example of a user-item graph. The weights of the edges are based on the results of the scoring functions. For example, user U_1 prefers “Jaws” to “Shrek”. Therefore, initially the algorithm distributes a mass of 5 to the item node “Jaws”, while “Shrek” only receives a mass of 2. Both items would receive the same resource in ProbS and PLIERS. In the next step of the mass diffusion, U_2 receives a higher portion of the mass of “Jaws” than U_3 in contrast to the equally distribution of the other two algorithms. The same applies to the third step where mass is distributed from U_2 and U_3 to their items. The candidate items in the lower part of the example would receive less mass from the distribution started on “Shrek” because of the low initially mass which lowers consequentially their contribution to the overall score.

Similarity Scoring

Additionally, UPPS includes the association between users and items in the calculation of the similarity that is derived from the item-tag graph. Therefore, items with a higher association to a user have more impact on the generation of candidates based on the item-tag graph. The scoring is calculated for the target user u_t with the following equation:

$$f_{j,t}^{sim} = \sum_{z=1}^r \sum_{s=1}^m \frac{a'_{jz} \cdot a'_{sz} \cdot S_1(u_t, i_s)}{k_I(t_z)k_T(i_s)}, j = 1, 2, \dots, m \quad (5.5)$$

$k_T(i_s)$ is the number of tags of item i_s and $k_I(t_z)$ is the number of items with tag t_z . Equation 5.5 calculates the distribution of a resource for a path: item \rightarrow tag \rightarrow item.

Items of the target user receive as resource the value of $S_1(u_t, i_s)$. This resource is then evenly distributed to all tags t_z of the item, which is expressed by the term $a'_{sz}/k_T(i_s)$. Afterwards, it is evenly distributed from the tag t_z to all items i_s with this tag, which might include the candidate item i_j . This is expressed by the term $a'_{jz}/k_I(t_z)$. Finally, the received distributed resources are summed up on the candidate items. Note that if $S_1(u_t, i_s) = a_{ts}$, the calculation in Equation 5.5 is identical to the similarity calculation of ProbS.

Two-step Similarity Scoring

The just introduced similarity score can be further extended to a two-step similarity score, which considers paths of the form: item \rightarrow tag \rightarrow item \rightarrow tag \rightarrow item. The two-step score includes items that are connected in the item-tag graph to the items of the user by an intermediate item.

The assumption is that all items that are found with the similarity score of Section 5.2.3 are still in the candidate item list, since it is possible to use the original item as the intermediate item. The following formula calculates $f_{j,t}^{sim*}$ for all items $j = 1, \dots, m$ and the target user u_t :

$$f_{j,t}^{sim*} = \sum_{z_2=1}^r \sum_{l_2=1}^m \sum_{z_1=1}^r \sum_{l_1=1}^m \frac{a'_{jz_2} \cdot a'_{l_2z_2} \cdot a'_{l_2z_1} \cdot a'_{l_1z_1} \cdot S_1(u_t, i_{l_1})}{k_I(t_{z_2}) \cdot k_T(i_{l_2}) \cdot k_I(t_{z_1}) \cdot k_T(i_{l_1})} \quad (5.6)$$

Included is the user-item association between the target user u_t and his items, which is denoted by $S_1(u_t, i_{l_1})$. The formula is derived from the normal similarity

by repeating the distribution process. This results in candidate items that are not forced to have tags in common with the items of the user but are still related to those items. Social tagging systems are not restricted to a preset well-defined set of tags. Therefore, users might assign different tags to an item that actually mean the same or are highly related to each other.

UPPS Scoring

The final recommendation score for the candidate items i_j is the linear combination of the above introduced scorings:

$$f_{j,t}^{UPPS} = \alpha \cdot f_{j,t}^{aff} + (1 - \alpha) \cdot f_{j,t}^{sim}, j = 1, 2, \dots, m \text{ with } \alpha \in [0, 1] \quad (5.7)$$

The same formula applies to $f_{j,t}^{sim*}$ instead of $f_{j,t}^{sim}$. In Equation 5.7, the values of $f_{j,t}^{aff}$ and $f_{j,t}^{sim}$ are normalized, so that the values of these functions lie both in the range $[0, 1]$ in order to simplify the procedure for estimating the optimal value for the weight parameter α . The normalization can easily be done by dividing the scores by $\max_j f_{j,t}^{aff}$ and $\max_j f_{j,t}^{sim}$, respectively, or any other function that maps the origin values into the range of $[0, 1]$.

Sorting the items in descending order by the corresponding score in the scoring vector $f_{j,t}^{UPPS}$ yields a list of item recommendations. In fact, a set of n recommendations is given by reducing the list to the top- n results. Of course, the results of UPPS allow the subsequent further filtering of recommendations using a diversification and other algorithms. Moreover, UPPS can also be employed as one component of a hybrid recommendation system.

5.2.4 Performance Evaluation

Dataset and Data Cleaning

The well-known MovieLens ml-20m [37] dataset is the basis for the following evaluation. This dataset contains relations between movies, users and tags that were collected between 1995 and 2015. The included movie ratings were assigned by users. The ratings contain values in the range of $[0.5, 5]$ with a step size of 0.5. In the following evaluations, these ratings will be the only feature to derive the user-item preference as discussed in Section 5.2.3. The dataset does not contain additional features that

Table 5.3: The MovieLens dataset

Dataset	Users	Individual Tags	Items	User-Item Relations	Tag-Item Relations	Mean Items per User	Mean Users per Item
ml-20m	138,493	35,086	26,744	20,000,263	195,735	144.4	747.8
ml-20m w/o outliers	102,467	30,878	21,955	4,745,842	188,901	46.3	284.9

would help in further refining the user-item preference, e.g. the number of playbacks in a video streaming site.

Like done in previous work [9], the dataset is cleaned by an initial outlier analysis: users collecting more than 300 items and items that were collected by more than 10,000 users are removed. Furthermore, the tags are normalized by the Porter stemmer, which replaces the original tags with their base form reduction. Therefore, different inflections of the tags are reduced to the same stemmed term. Additionally, the 200 most popular English words are removed from the set of tags and all terms are converted to lower case.

Table 5.3 summarizes information about the size of the original dataset as well as about the cleaned dataset. The cleaned dataset still contains information about more than 100,000 users, 30,000 distinct tags and 20,000 items, which is in the same scale as the original MovieLens dataset. Furthermore, Table 5.3 indicates that the number of user-item relationships is reduced to about a fourth of the relationships in the original dataset. However, the number of tag-item relations is only slightly reduced. Figure 5.5 compares the number of items per user in both datasets. The number of users per items is further illustrated in the boxplots of Figure 5.6. The median of the users per item remains nearly unchanged, while the mean is significantly reduced.

Evaluation Methodology

First, 80 % of the items of the currently analyzed user are randomly removed. The remaining items are used to produce the recommendations. Only users in the graph that are related to at least 100 videos are considered. As a consequence, more than 20 seed items for evaluating UPPS versus Probs and PLIERS are available. In the end, the chosen methodology evaluates whether or not the algorithms can predict the previously removed items. This is done for 500 randomly chosen users. All figures include the 90 % confidence interval.

The rating feature contained in the MovieLens dataset is utilized for determining how relevant an item is to the target user. In fact, the ten-level relevance score of

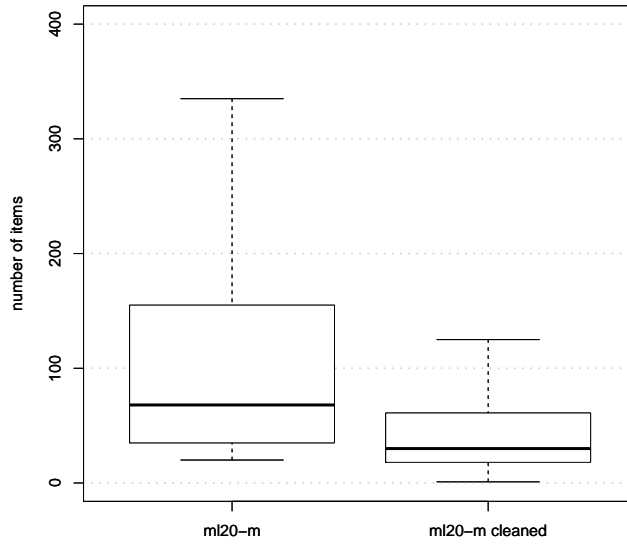


Figure 5.5: Number of items per user in original and cleaned MovieLens dataset

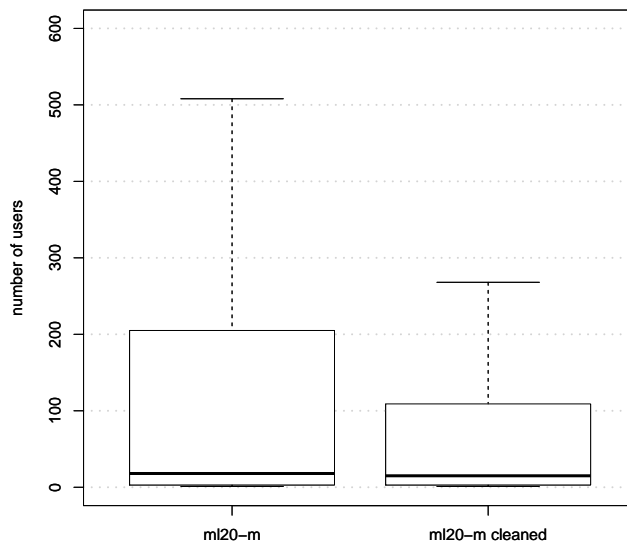


Figure 5.6: Number of users per item in original and cleaned MovieLens dataset

MovieLens also enables the utilization of Normalized Discounted Cumulative Gain (NDCG) as metric for evaluating the recommendation quality up to position 50. Low positions are noteworthy to be considered because UPPS may well be part of a hybrid recommendation system, which applies pruning and reordering (e.g. based on diversification) of the results of UPPS in a second step.

Precision up to position k , $P@k$ ($k = 10, 20, 50$), is also presented. Its calculation is based on the relevance score of MovieLens. Items with a rating higher than 2.5 are considered to be relevant; items with scores lower or equal to 2.5 as not relevant. Note that opposed to Precision, NDCG penalizes when a relevant item gets a low position in the ranking by logarithmically weighting the positions.

The final calculated metric is Novelty [82]. Novelty analyzes, if the results are unexpected or surprising for the user. A high novelty score means the results are popular items and thus were expected to be generated.

Comparison of Different Versions of UPPS Affinity Scoring

The first experiment focuses on the affinity score. In the end, it has the highest impact on the quality of the recommender system. It is based on the notion that a target user might like items that are recommended by a significant number of users that enjoyed the items of the target user. In contrast, the similarity score is based on the item-tag association. Here, each user might see something completely different in an item (e.g. drama vs comedy) but still enjoys it. Another focus of this experiment is, how to best integrate the user-item association in the mass diffusion process of the affinity score of UPPS (e.g. $UPPS_{\text{basic}}$ versus $UPPS_a$). The UPPS results are not only compared to ProbS and PLIERS, but also to their versions restricted to the affinity score ($ProbS_a$ and $PLIERS_a$).

As discussed in Section 5.2.3, UPPS is able to utilize the user-item association in the three scoring functions S_1 , S_2 and S_3 of Equation 5.4 independently. All $UPPS_{\text{basic}}$ variants set $S_2(u_l, i_s) = a_{l_s}$ and $S_3(u_l, i_j) = a_{l_j}$. Thus, the user-item association is only used in the first step of the mass diffusion process $S_1(u_t, i_s)$, which is equally distributed to all users u_l connected to item i_s and from there to all items i_j of that user. In contrast, the $UPPS_a$ variants utilize the user-item association $p(u, i)$ scores of MovieLens also in S_2 and S_3 .

Additionally, some of the analyzed UPPS variants use different models to utilize

Table 5.4: Performance metrics for different versions of the algorithm

	NDCG@10	NDCG@20	NDCG@5	P@10	P@20	P@50
ProbS _a	0.3808	0.3521	0.3133	0.4216	0.3730	0.3034
PLIERS _a	0.2349	0.2296	0.2207	0.2694	0.2562	0.2271
UPPS _{basic} w/ Exp	0.3864	0.3575	0.3180	0.4234	0.3764	0.3073
UPPS _{basic} w/o Exp	0.3864	0.3565	0.3179	0.4250	0.3753	0.3074
UPPS _a w/Exp	0.3914	0.3593	0.3211	0.4232	0.3734	0.3086
UPPS _a w/o Exp	0.3917	0.3622	0.3225	0.4252	0.3795	0.3108

the user-item association $p(u, i)$ of MovieLens in S_1, S_2, S_3 (only S_1 for the UPPS_{basic} variants). The so far described scoring functions use the linear rating value directly as score. This is also true for the "w/o Exp" variants, where $S_1(u_t, i_s) = p(u_t, i_s)$, $S_2(u_l, i_s) = p(u_l, i_s)$ and $S_3(u_l, i_j) = p(u_l, i_j)$. However, the exponential "w/ Exp" UPPS versions set $S_1(u_t, i_s) = 1.5^{p(u_t, i_s)}$, $S_2(u_l, i_s) = 1.5^{p(u_l, i_s)}$ and $S_3(u_l, i_j) = 1.5^{p(u_l, i_j)}$. Therefore, user items i_s with a higher score contribute a significantly higher mass to the mass diffusion process. Additionally, the whole mass diffusion process is significantly more biased to links with a high user-item association. Table 5.4 reports the results.

UPPS without the exponential scoring function, UPPS_a w/o Exp, yields the overall best performance. UPPS_a w/o Exp has about 3 % performance gain on the NDCG@10 metric compared to ProbS_a and 60 % compared to PLIERS_a. The basic version of UPPS, UPPS_{basic} still performs slightly better than the baseline algorithms. PLIERS aims to recommend items that have a similar popularity to the items of the user. This could be the cause of the poor performance of PLIERS in terms of NDCG and Precision. It is possible that the items of the target users in MovieLens differ greatly in their popularity.

Additionally, it can be seen that the usage of the exponential scoring function produces slightly worse results than the direct usage of the rating value. This is also true for other variations of UPPS that only include the user preference in the second or third step of the mass diffusion. Those perform similar to UPPS_{basic} and, therefore, are not included in Table 5.4. The following experiments only consider UPPS_a w/o Exp that does not use the exponential scoring function.

Table 5.5: User-item association mapping

Rating	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
5-level	1		2		3		4		5	
3-level	1			2			3			
2-level	1				2					

Table 5.6: Performance metrics for different user preference granularity

	NDCG@10	NDCG@20	NDCG@50	P@10	P@20	P@50
UPPS _a 5-Level	0.3929	0.3641	0.3211	0.4302	0.3852	0.3098
UPPS _a 3-Level	0.3941	0.3654	0.3220	0.4342	0.3879	0.3110
UPPS _a 2-Level	0.3905	0.3619	0.3199	0.4318	0.3832	0.3092

Sensitivity to User Preference Granularity

The second experiment evaluates the impact of the granularity of the user-item associations on the performance of the UPPS affinity score. Table 5.6 has the results. The MovieLens dataset provides 10 different score levels (0.5 to 5.0 with 0.5 step size). However, other real-world systems might only differentiate 5 levels (e.g. 1 to 5) or even less. Thus, it is interesting to see, whether any non-binary user-item association can be utilized in UPPS to increase the quality of recommended items. The user-item associations of MovieLens are artificially mapped to 5-level, 3-level and 2-level granularity counterparts as described in Table 5.5.

Other systems that provide the same granularity, but with other underlying scores, e.g. 0.1 to 1.0 in 0.1 steps, can be easily mapped to the identical scoring functions S_n utilized in Section 5.2.4 (e.g. $S_1(u_t, i_s) = 1.5^{5 \cdot p(u_t, i_s)}$ for the just given example).

Table 5.6 presents UPPS_a performance when utilizing the low-level ratings of Table 5.5. The performance differences are only marginal. Most surprisingly, the level-3 ratings perform best. This can be attributed to a side-effect of the level rating reduction: the difference of initial resources assigned to high and low ranked items is significantly reduced. This is also in line with our findings, where UPPS_a w/o Exp performed better than UPPS_a w/ Exp.

Similarity Scoring

This experiment evaluates the impact of the similarity on the quality of the recommendation algorithms and applies the same methodology as in the first experiment. Again, ProbS and PLIERS are the baseline algorithms. Table 5.7 has the results.

The algorithms with only an "s" at the end of their name are restricted to the similarity score by setting α to zero. The similarity calculations differ between ProbS, PLIERS and UPPS. PLIERS includes a component to favor items with probability similar to the calculation of the affinity. UPPS uses the similarity of Equation 5.5 that includes the user preference in the first step of the mass diffusion. Additionally, the results include an UPPS variant that uses the two-step similarity calculation of Equation 5.6. Those results are denoted by "s*" in the name. Whenever a combination of affinity and similarity is used, the parameter α is optimized in the linear combination to yield the best possible results for every algorithm. Those versions are indicated by "a+s".

The UPPS algorithms that include user preference values into the similarity score produce slightly better results compared to the baselines, while ProbS_s and PLIERS_s are similar to each other. It is noticeable that the similarity produces poor results on its own compared to the affinity score. The performance of the algorithms with a combination of affinity and similarity is mostly dominated by the affinity.

However, the two-step similarity score performs much better, e.g. it increases the NDCG@10 from 0.0489 to 0.1653. The combination of affinity and two-step similarity yields the best results of this comparison. On the NDCG@10 a 30 % gain to the already good results of UPPS with normal similarity is measurable. It has to be considered that the two-step similarity increases the computational complexity, which increases with the number of items and tags in the dataset.

Figure 5.7 shows the precision-recall curve. It measures the precision at 5 % levels of the recall. The precision is determined for each user at each recall level and then averaged over all users. Only UPPS_{a+s*} and UPPS_{a+s} are considered and compared to the best versions of PLIERS and ProbS. It can be seen that the precision is higher up to recall level 0.35. Additionally, Figure 5.8 evaluates the NDCG at all positions between 1 and 50. The NDCG value is averaged for all users. It outperforms the other algorithms on all positions.

Additionally, Table 5.7 includes Novelty, denoted as Nov, for recommendation lists

Table 5.7: Performance metrics for combinations with similarity

	NDCG@10	NDCG@20	NDCG@50	P@10	P@20	P@50	Nov@10	Nov@20	Nov@50	α
ProbSs	0.0380	0.0525	0.0648	0.0548	0.0713	0.0775	6.5068	15.1638	34.2277	0
ProbSa+s	0.3866	0.3581	0.3164	0.4272	0.3798	0.3055	32.7384	46.0988	72.8788	0.7
PLIERSs	0.0356	0.0487	0.0614	0.0508	0.0657	0.0735	6.7532	15.8924	35.0338	0
PLIERSa+s	0.2365	0.2308	0.2224	0.2704	0.2570	0.2288	6.3760	9.3536	16.4603	0.8
UPPSs	0.0489	0.0605	0.0708	0.0668	0.0780	0.0820	7.1036	16.6604	36.0759	0
UPPSa+s	0.3988	0.3678	0.3258	0.4330	0.3841	0.3127	33.9908	48.7548	74.4421	0.7
UPPSs*	0.1653	0.1661	0.1654	0.2066	0.1918	0.1727	260.5828	255.2074	291.2202	0
UPPSa+s*	0.5188	0.4616	0.3908	0.5418	0.4620	0.3585	51.1792	77.5380	111.7226	0.5

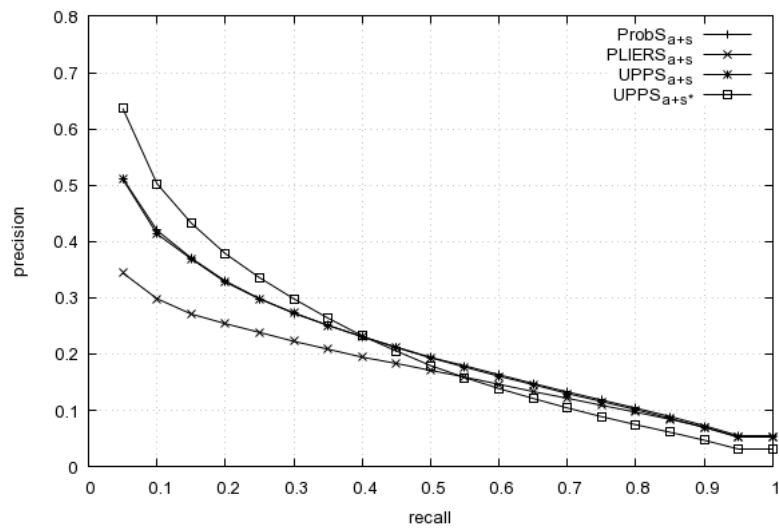


Figure 5.7: Precision-Recall

of length 10, 20 and 50. It can be seen that UPPS performs similar to ProbS, which is an indicator, that the general characteristics of the recommendation lists is not changed in comparison to ProbS. In contrast, PLIERS seems to recommend items with a lower popularity that are more novel and surprising.

Parameter Sensitivity Study

This final experiment analyzes the impact of the weighting factor α in Equation 5.7. This weighting controls the relation between affinity score and similarity score. NDCG at positions 10 and 20 is presented to compare the α weights. Only the affinity score is used when $\alpha = 1$ and only the similarity score when $\alpha = 0$. In Figure 5.9 and 5.10, 80 % of the items of the user are removed like in the other evaluations. UPPS_{a+s} produces

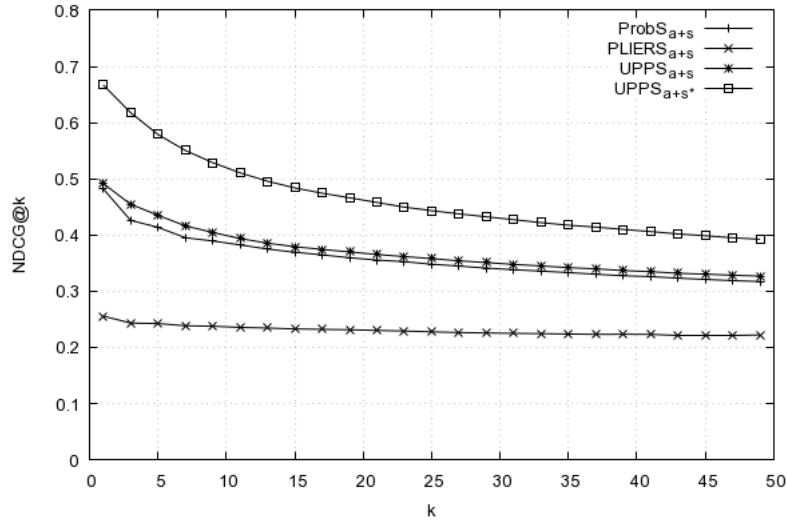


Figure 5.8: NDCG@k

the best results for an α value of 0.7, while $UPPS_{a+s^*}$ yields the best performance for an α value of 0.5. For same values of α , $UPPS_{a+s^*}$ always performs better than ProbS and PLIERS.

5.2.5 Conclusion

This section introduced User Preference-based Probability Spreading for content recommendation, UPPS. The rationale behind UPPS lies in the integration of user-item preference into mass diffusion for recommender systems. Therefore, UPPS can not only utilize obvious user-item preferences such as a like or a dislike, but can also take into account implicit signals, e.g. watching a video, downloading a file or adding an item to a list for the user's preferences. It was shown how to integrate user-item preference scoring into both the affinity and the similarity scores derived from the user-item graph and item-tag graph, respectively. In addition to the similarity score, two-step similarity scoring was introduced, which recommends items of the item-tag graph connected to items of the target user via an intermediate item.

For a comparative performance study of UPPS, the MovieLens ml-20m dataset was utilized. The presented results show that UPPS achieves an improvement in the NDCG@10 measure by more than 30 % over ProbS and more than 100 % over PLIERS. Similar quantitative results are observable for NDCG@k measures ($k=1, \dots, 50$).

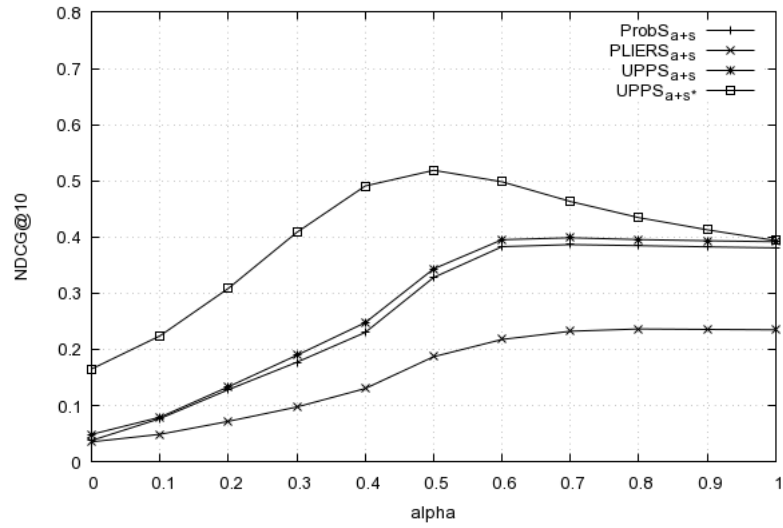


Figure 5.9: Impact of the parameter α on NDCG@10

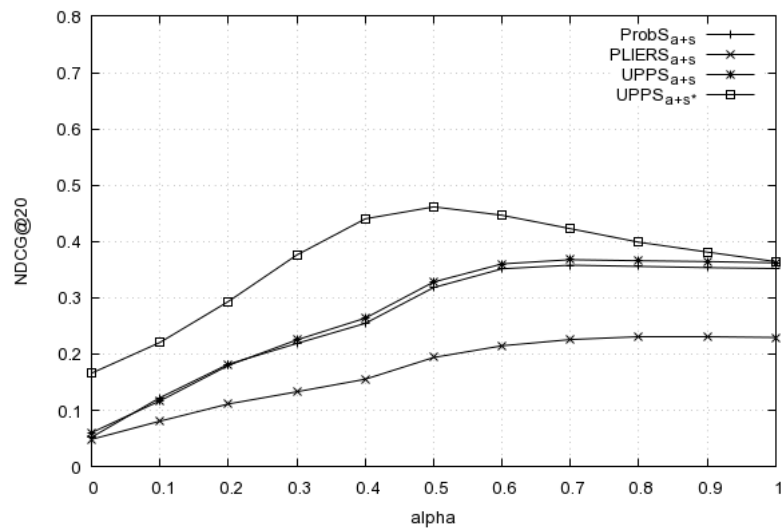


Figure 5.10: Impact of the parameter α on NDCG@20

Moreover, UPPS achieves an improvement of 25 % in P@10 over ProbS and more than 100 % over PLIERS. Plotting the precision versus the recall level, UPPS clearly outperformed ProbS and PLIERS in precision up to a recall of 0.35.

6 Conclusion and Future Work

6.1 Conclusion

This thesis first contributed a measurement study of the A-MPDU frame aggregation behavior of IEEE 802.11n in a real-world, multi-hop network with off-the-shelf hardware [27]. This study utilized a special-purpose, indoor mesh testbed. The presented performance curves reveal that channel bonding nearly doubles the throughput for any fixed path length. The mean aggregate size, in number of frames per A-MPDU at each node in the multi-hop chain, is also doubled by channel bonding. Also, the mean aggregate size, in number of frames at each node, decreases with increasing path length.

Secondly, this thesis contributed an approach to utilize A-MPDU subframes to increase the overall throughput under bad channel conditions [28]. The introduced approach, MPDU payload adaptation (MPA), adapts the size of MAC protocol data units to channel conditions, to increase the throughput and to lower the delay in error-prone channels. The focus was especially on the edge of the network, where even the lowest physical data rates exhibit such a high bit error rate that the probability for a successful transmission of typically sized MPDUs is very low. The results suggest that under erroneous conditions, throughput can be maximized by limiting the MPDU size.

Thirdly, this thesis introduced Neighborhood-Aware Opportunistic Networking on Smartphones (NOPPoS). NOPPoS creates an opportunistic, pocket switched network, using current generation, off-the-shelf mobile devices [13]. NOPPoS utilizes IEEE 802.11 access points of mobile devices to create local, isolated networks that connect co-located mobile devices. NOPPoS assigns IEEE 802.11 STA and AP roles to mobile devices based on the number of mobile devices and access points in the proximity. As main novel feature, NOPPoS is highly responsive to node mobility due to periodic,

low-energy scans of its environment, using Bluetooth Low Energy advertisements. In fact, NOPPoS outperforms WLAN-OPP [72], because it determines the exact number of neighbors at any instant of time.

Additionally, this thesis introduced a cross-layer protocol that tightly couples an opportunistic network with a document retrieval application. The protocol, Neighborhood Document Sharing (NDS), enables users to discover and retrieve arbitrary documents shared by other users in their proximity, i.e. in the communication range of their IEEE 802.11 interface. IEEE 802.11 connections are utilized on-demand during the file transfer operation and the Indexing operation in the proximity of the user. This saves energy and minimizes the use of the IEEE 802.11 interface for only high-throughput data transfers. Similarly to NOPPoS, Bluetooth LE energy is employed to broadcast device and service discovery information to nearby devices. Simulations show that the protocol interconnects over 90 % of all devices.

Finally, NDS was extended by the content recommendation system User Preference-based Probability Spreading (UPPS), a graph-based approach [29]. It integrates user-item scoring into a graph-based tag-aware item recommender system. Building upon ProbS [84] and PLIERS [9], UPPS utilizes refined formulas for affinity and similarity scoring, taking into account user-item preference in the mass diffusion of the recommender system. The presented results show that UPPS is a significant improvement to PLIERS and ProbS.

6.2 Future Work

IEEE 802.11ax just recently introduced novel protocols to increase the efficiency in the scenario of multiple, simultaneously transmitting devices [21]. First, there is OFDMA that replaces the traditional OFDM modulation. Instead of utilizing all subcarriers for one transmission, groups of adjacent subcarriers are assigned to different devices. In contrast to OFDM in IEEE 802.11ac and before, an AP only has to contend for the medium once to transmit simultaneously to different stations not only utilizing different spatial streams but also different frequencies. This helps to reduce delay in low-bandwidth applications. Additionally, up-link MU-MIMO extends the existing down-link MU-MIMO functionality. Previously, access points were able to transmit simultaneously to different stations by making use of spatial streams / spatial division

multiplexing. Transmitting to n stations requires at least n antennas at the access point and the transmitted signals are coded in a way that only the specific receivers are able to decode the signals transmitted. With the up-link extension, access points are now also able to schedule simultaneous transmissions from stations to the access point.

While OFDMA and up-link and down-link MU-MIMO have only been specified in the context of access points, these IEEE 802.11 enhancements have a huge potential for wireless P2P applications. Till recently, only IEEE 802.11 broadcast transmissions were available to transmit simultaneously to multiple stations. However these transmissions are limited to basic physical data rates (e.g. MCS 0) and do not support acknowledgements on MAC level. This resulted in unicast transmissions being more efficient for real applications, even when transmitting the same data to multiple recipients. OFDMA and MU-MIMO, while still essentially using unicast transmissions, could change this and enable new, high-performance wireless P2P applications. Therefore, the author of this thesis will focus in future work on the development of an integration of OFDMA and MU-MIMO into IBSS networks.

Finally, TCP is still the most utilized transport layer protocol in most applications. However, novel protocols, e.g. QUIC [18], build on top of UDP, and effectively move the implementation of the transport protocol to the application layer. This enables application developers to utilize algorithms that are not built into operation systems or into network hardware (e.g. routers, switches, firewalls). This is especially interesting for opportunistic networks and wireless IEEE 802.11 networks. It is well known that the general-purpose TCP transport layer protocol is not well suited for IEEE 802.11 networks [23]. Moreover, IEEE 802.11 itself implements algorithms also found in TCP: e.g. acknowledgements, packet buffering and reordering. Therefore, not only TCP but also IEEE 802.11 potentially adds unnecessary delay for applications like video conferencing and online gaming. In future work, it will be examined whether it is possible and beneficial, to also add a mode similar to UDP to the IEEE 802.11 MAC layer.

Bibliography

- [1] “Android beacon library.” [Online]. Available: <https://altbeacon.github.io/android-beacon-library/>
- [2] “iperf, the TCP/UDP bandwidth measurement tool.” [Online]. Available: <https://sourceforge.net/projects/iperf2/>
- [3] *QCA6234 Integrated Dual-Band 2x2 802.11n + Bluetooth® 4.0*. [Online]. Available: https://developer.qualcomm.com/qfile/28871/lm80-p0598-12-qca6234_datasheet.pdf
- [4] “IEEE 802.11n: Standard for wireless lan medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput,” 2009.
- [5] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, “IEEE 802.11ax: Challenges and requirements for future high efficiency WiFi,” *IEEE Wireless Communications*, vol. 24, no. 3, pp. 130–137, 2016.
- [6] S. Alamouti, “A simple transmit diversity technique for wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.
- [7] G. Ananthanarayanan and I. Stoica, “Blue-Fi: enhancing Wi-Fi performance using bluetooth signals,” in *Proc. of the 7th International Conference on Mobile Systems, Applications, and Services*. ACM, 2009, pp. 249–262.
- [8] V. Arnaboldi, M. Campana, F. Delmastro, and E. Pagani, “Tag-based recommender system for context-aware content dissemination in opportunistic networks,” *University of Pisa Tech Report*, 2015.

- [9] V. Arnaboldi, M. G. Campana, F. Delmastro, and E. Pagani, “A personalized recommender system for pervasive social networks,” *Pervasive and Mobile Computing*, vol. 36, pp. 3–24, 2017.
- [10] M. Bakht, J. Carlson, A. Loeb, and R. Kravets, “United we find: Enabling mobile devices to cooperate for efficient neighbor discovery,” in *Proc. of the Twelfth Workshop on Mobile Computing Systems & Applications*, 2012, pp. 1–6.
- [11] B. Bellalta, “IEEE 802.11ax: High-efficiency WLANs,” *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.
- [12] B. Bellalta and K. Kosek-Szott, “AP-initiated multi-user transmissions in IEEE 802.11ax WLANs,” *Ad Hoc Networks*, vol. 85, pp. 145–159, 2019.
- [13] S. Bergemann, J. Friedrich, and C. Lindemann, “Neighborhood-aware opportunistic networking on smartphones,” in *Proc. of the 14th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2017, pp. 126–134.
- [14] J. C. Bicket, “Bit-rate selection in wireless networks,” Ph.D. dissertation, Massachusetts Institute of Technology, 2005.
- [15] J.-Y. L. Boudec and M. Vojnovic, “The random trip model: Stability, stationary regime, and perfect simulation,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1153–1166, 2006.
- [16] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, “Routing protocols in ad hoc networks: A survey,” *Computer Networks*, vol. 55, no. 13, pp. 3032–3080, 2011.
- [17] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device-to-device communications with Wi-Fi Direct: overview and experimentation,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, 2013.
- [18] G. Carlucci, L. D. Cicco, and S. Mascolo, “HTTP over UDP: an experimental investigation of QUIC,” in *Proc. of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 609–614.
- [19] M. Conti, F. Delmastro, G. Minutiello, and R. Paris, “Experimenting opportunistic networks with WiFi Direct,” in *Wireless Days*. IEEE, 2013, pp. 1–6.

- [20] M. Conti and M. Kumar, “Opportunities in opportunistic computing,” *Computer*, vol. 43, no. 1, pp. 42–50, 2010.
- [21] D.-J. Deng, K.-C. Chen, and R.-S. Cheng, “IEEE 802.11ax: Next generation wireless local area networks,” in *Proc. of the 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. IEEE, 2014, pp. 77–82.
- [22] D.-J. Deng, Y.-P. Lin, X. Yang, J. Zhu, Y.-B. Li, J. Luo, and K.-C. Chen, “IEEE 802.11ax: highly efficient WLANs for intelligent information infrastructure,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 52–59, 2017.
- [23] S. M. ElRakabawy, A. Klemm, and C. Lindemann, “TCP with adaptive pacing for multihop wireless networks,” in *Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 288–299.
- [24] P. T. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulié, “Epidemic information dissemination in distributed systems,” *IEEE Computer*, vol. 37, no. 5, pp. 60–67, 2004.
- [25] S. Farahani, *ZigBee wireless networks and transceivers*. Newnes, 2011.
- [26] J. Friedrich, “Design and implementation of a measurement infrastructure for the quantitative analysis of IEEE 802.11n wireless multi-hop networks,” Diplomarbeit, Leipzig University, 2011.
- [27] J. Friedrich, S. Frohn, S. Gübner, and C. Lindemann, “Understanding IEEE 802.11n multi-hop communication in wireless networks,” in *Proc. of the 2011 International Symposium of Modeling and Optimization of Mobile, Ad Hoc, and Wireless Networks*. IEEE, 2011, pp. 321–326.
- [28] J. Friedrich, S. Günther, and C. Lindemann, “Performance analysis of compressed block acknowledgment in IEEE 802.11ax,” in *Proc. of the 17th ACM International Symposium on Mobility Management and Wireless Access*, 2019, pp. 103–110.
- [29] J. Friedrich, C. Lindemann, and M. Petrifke, “User preference-based probability spreading for tag-aware content recommendation,” in *Proc. of the 19th International Conference on High Performance Computing and Communications*. IEEE, 2017, pp. 380–387.

- [30] S. Frohn, S. Gübner, and C. Lindemann, “Analyzing the effective throughput in multi-hop IEEE 802.11n networks,” *Computer Communications*, vol. 34, no. 16, pp. 1912–1921, 2011.
- [31] J. Gemmell, T. Schimoler, M. Ramezani, L. Christiansen, and B. Mobasher, “Improving folkrank with item-based collaborative filtering,” *Recommender Systems & the Social Web*, 2009.
- [32] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology,” *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.
- [33] S. Gübner, “Effective data dissemination over multi-hop gigabit wireless networks,” Ph.D. dissertation, Leipzig University, 2015.
- [34] M. A. M. Hail, M. Amadeo, A. Molinaro, and S. Fischer, “On the performance of caching and forwarding in information-centric networking for the IoT,” in *Proc. of the 13th International Conference on Wired and Wireless Internet Communications*, 2015, pp. 313–326.
- [35] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Predictable 802.11 packet delivery from wireless channel measurements,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 159–170, 2011.
- [36] B. Han and A. Srinivasan, “eDiscovery: Energy efficient device discovery for mobile opportunistic communications,” in *Proc. of the 20th IEEE International Conference on Network Protocols*. IEEE, 2012, pp. 1–10.
- [37] F. M. Harper and J. A. Konstan, “The MovieLens datasets,” *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.
- [38] M. Heck, J. Edinger, D. Schäfer, and C. Becker, “IoT applications in fog and edge computing: Where are we and where are we going?” in *Proc. of the 27th International Conference on Computer Communication and Networks*. IEEE, 2018.
- [39] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, “Network simulations with the ns-3 simulator,” *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.

- [40] R. Heydon and N. Hunn, *Bluetooth Low Energy: The Developer's Handbook*, 2012.
- [41] M. Inamullah and B. Raman, "Frame aggregation in 802.11ac: Need for modified block ack," in *Proc. of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 708–710.
- [42] T. Jin, G. Noubir, and B. Sheng, "WiZi-Cloud: Application-transparent dual ZigBee-WiFi radios for low power internet access," in *Proc. of 2011 IEEE INFOCOM*. IEEE, 2011, pp. 1593–1601.
- [43] A. K. F. Khattab, A. Sabharwal, and E. W. Knightly, "Fair randomized antenna allocation in asynchronous MIMO multi-hop networks," in *Proc. of 17th International Conference on Computer Communications and Networks*. IEEE, 2008, pp. 111–118.
- [44] W. Kim, H. K. Wright, and S. Nettles, "Improving the performance of multi-hop wireless networks using frame aggregation and broadcast for TCP ACKs," in *Proc. of the 2008 ACM CoNEXT Conference*, 2008, pp. 1–12.
- [45] J. Koivunen, P. Almers, V.-M. Kolmonen, J. Salmi, A. Richter, F. Tufvesson, P. Suvikunnas, A. Molisch, and P. Vainikainen, "Dynamic multi-link indoor MIMO measurements at 5.3 GHz," in *Proc. of the 2nd European Conference on Antennas and Propagation*, 2007, pp. 1–6.
- [46] K. LaCurts and H. Balakrishnan, "Measurement and analysis of real-world 802.11 mesh networks," in *Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010, pp. 123–136.
- [47] T. Li, Q. Ni, D. Malone, D. J. Leith, Y. Xiao, and T. Turletti, "Aggregation with fragment retransmission for very high-speed WLANs," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 591–604, 2009.
- [48] Y. Lin and V. W. Wong, "Frame aggregation and optimal frame size adaptation for IEEE 802.11n WLANs," in *Proc. of IEEE Globecom 2006*. IEEE, 2006, pp. 1–6.
- [49] C. Lindemann and O. P. Waldhorst, "Exploiting epidemic data dissemination for consistent lookup operations in mobile applications," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 44–56, 2004.

- [50] —, “Peer-to-Peer-Systeme für drahtlose Multihop-Netze,” *Informatik-Spektrum*, vol. 29, no. 3, pp. 222–226, 2006.
- [51] L. Lü and W. Liu, “Information filtering via preferential diffusion,” *Physical Review E*, vol. 83, no. 6, 2011.
- [52] Y. Mawad and S. Fischer, “HIDTN: Hybrid DTN and infrastructure networks for reliable and efficient data dissemination,” in *Proc. of the 28th International Telecommunication Networks and Applications Conference*. IEEE, 2018, pp. 1–8.
- [53] V. F. Mota, F. D. Cunha, D. F. Macedo, J. M. Nogueira, and A. A. Loureiro, “Protocols, mobility models and tools in opportunistic networks: A survey,” *Computer Communications*, vol. 48, pp. 5–19, 2014.
- [54] C. Papathanasiou and L. Tassiulas, “Multicast transmission over IEEE 802.11n WLAN,” in *Proc. of the 2008 IEEE International Conference on Communications*, 2008, pp. 4943–4947.
- [55] I. Pefkianakis, Y. Hu, S. H. Y. Wong, H. Yang, and S. Lu, “MIMO rate adaptation in 802.11n wireless networks,” in *Proc. of the 16th Annual International Conference on Mobile Computing and Networking*. ACM, 2010, pp. 257–268.
- [56] K. Pelechrinis, I. Broustis, T. Salonidis, S. V. Krishnamurthy, and P. Mohapatra, “Design and deployment considerations for high performance MIMO testbeds,” in *Proc. of the 4th Annual International Conference on Wireless Internet*, 2008, pp. 1–9.
- [57] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, “Analyzing 802.11n performance gains,” *Proc. ACM MobiCom (poster session)*, 2009.
- [58] —, “Experimental characterization of 802.11n link quality at high rates,” in *Proc. of the 5th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*. ACM, 2010, pp. 39–46.
- [59] E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac*. Cambridge university press, 2013.

- [60] T. Pering, V. Raghunathan, and R. Want, “Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup,” in *Proc. of the 8th International Conference on VLSI Design*. IEEE, 2005, pp. 774–779.
- [61] D. Piazza, N. J. Kirsch, A. Forenza, R. W. Heath, and K. R. Dandekar, “Design and evaluation of a reconfigurable antenna array for mimo systems,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 3, pp. 869–881, 2008.
- [62] R. Pozza, M. Nati, S. Georgoulas, K. Moessner, and A. Gluhak, “Neighbor discovery for opportunistic networking in internet of things scenarios: A survey,” *IEEE access*, vol. 3, pp. 1101–1131, 2015.
- [63] D. Schäfer, J. Edinger, J. Eckrich, M. Breitbach, and C. Becker, “Hybrid task scheduling for mobile devices in edge and cloud environments,” in *Proc. of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 2018, pp. 669–674.
- [64] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD dataset cambridge/haggle (v. 2009-05-29),” *CRAWDAD wireless network data archive*, 2009.
- [65] S. Seytnazarov, J.-G. Choi, and Y.-T. Kim, “Enhanced mathematical modeling of aggregation-enabled WLANs with compressed BlockACK,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1260–1273, 2019.
- [66] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix,” *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45, 2014.
- [67] E. Shih, P. Bahl, and M. J. Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices,” in *Proc. of the 8th Annual International Conference on Mobile Computing and Networking*. ACM, 2002, pp. 160–171.
- [68] V. Shrivastava, S. K. Rayanchu, J. Yoonj, and S. Banerjee, “802.11n under the microscope,” in *Proc. of the 8th ACM SIGCOMM Conference on Internet Measurement*. ACM, 2008, pp. 105–110.

- [69] D. Skordoulis, Q. Ni, H.-H. Chen, A. Stephens, C. Liu, and A. Jamalipour, “IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs,” *IEEE Wireless Communications*, vol. 15, no. 1, pp. 40–47, 2008.
- [70] R. Steinmetz and K. Wehrle, *Peer-to-peer systems and applications*. Springer, 2005.
- [71] M. Stute, D. Kreitschmann, and M. Hollick, “One billion apples’ secret sauce: Recipe for the Apple wireless direct link ad hoc protocol,” in *Proc. of the 24th Annual International Conference on Mobile Computing and Networking*, 2018.
- [72] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre, “WLAN-Opp: Ad-hoc-less opportunistic networking on smartphones,” *Ad Hoc Networks*, vol. 25, pp. 346–358, 2015.
- [73] S. Trifunovic, A. Picu, T. Hossmann, and K. A. Hummel, “Adaptive role switching for fair and efficient battery usage in device-to-device communication,” *Mobile Computing and Communications Review*, vol. 18, no. 1, pp. 25–36, 2014.
- [74] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme, “Tag-aware recommender systems by fusion of collaborative filtering algorithms,” in *Proc. of the 2008 ACM Symposium on Applied computing*, 2008, pp. 1995–1999.
- [75] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proc. of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 2008, p. 60.
- [76] V. Visoottiviseth, T. Piroonsith, and S. Siwamogsatham, “An empirical study on achievable throughputs of IEEE 802.11n devices,” in *Proc. of the 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. IEEE, 2009.
- [77] C.-Y. Wang and H.-Y. Wei, “IEEE 802.11n MAC enhancement and performance evaluation,” *Mobile Networks and Applications*, vol. 14, no. 6, pp. 760–771, 2009.
- [78] Wi-Fi-Alliance, “Wi-Fi Peer-to-Peer (P2P) technical specification,” 2011.
- [79] P. Wu and Z.-K. Zhang, “Enhancing personalized recommendations on weighted social tagging networks,” *Physics Procedia*, vol. 3, no. 5, pp. 1877–1885, 2010.

- [80] H. Zhang, Y. Wang, and C. C. Tan, “WD2: An improved WiFi-Direct group formation protocol,” in *Proceedings of the 9th ACM MobiCom Workshop on Challenged Networks*. ACM, 2014, pp. 55–60.
- [81] J. Zhang, H. Shen, K. Tan, R. Chandra, Y. Zhang, and Q. Zhang, “Frame retransmissions considered harmful: improving spectrum efficiency using micro-acks,” in *Proc. of the 18th Annual International Conference on Mobile Computing and Networking*, 2012, pp. 89–100.
- [82] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, “Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs,” *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 179–186, 2010.
- [83] —, “Tag-aware recommender systems: A state-of-the-art survey,” *Journal of Computer Science and Technology*, vol. 26, no. 5, pp. 767–777, 2011.
- [84] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, “Solving the apparent diversity-accuracy dilemma of recommender systems,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.

Bibliographische Angaben

Friedrich, Jan: Advanced Protocols for Peer-to-Peer Data Transmission in Wireless Gigabit Networks. Dissertation, Universität Leipzig, 2020.

Wissenschaftlicher Werdegang

Persönliche Angaben

Jan Friedrich

geboren am 24.02.1985 in Lutherstadt Wittenberg

Schulbildung und Bundeswehr

1995 - 2004 Paul-Gerhardt-Gymnasium Gräfenhainichen

Abschluss: Abitur

2004 - 2005 Grundwehrdienst

Studium und Beruf

2005 - 2011 Studium der Informatik

Universität Leipzig

Abschluss: Diplom

seit 2011 Wissenschaftlicher Mitarbeiter

Lehrstuhl für Rechnernetze und Verteilte Systeme

Universität Leipzig

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den 09.04.2020

Jan Friedrich