



<b>Project Number:</b>	IST-026905
<b>Project Title:</b>	Multiple-Access Space-Time Coding Testbed
<b>Project Coordinator:</b>	C.F. Mecklenbräuker
<b>Deliverable Number:</b>	D2.3.2b

<b>Title of Deliverable:</b>	Report on testing of MIMO MAC algorithms
<b>Workpackage:</b>	WP-2
<b>Nature:</b>	R
<b>Dissemination level:</b>	PU
<b>Editor:</b>	Peter Lüthi
<b>Authors:</b>	see list inside
<b>Contractual Date of Deliverable:</b>	June 30, 2008
<b>Actual Date of Delivery:</b>	June 30, 2008

**Abstract:**

This deliverable reports on the testing of the MIMO *Medium Access Control* (MAC) algorithms implemented in the MASCOT real-time testbed. The report is split into three parts: The first part reports on the physical (PHY) layer characteristics of the MASCOT testbed in order to provide the base for understanding of the MAC layer characteristics. The second part summarizes the features and characteristics of the MASCOT MAC layer algorithms, and discusses the implications of different traffic scenarios on MAC scheduling algorithms and overall MAC layer performance. The third part provides measurement results of the MASCOT PHY and MAC layer.

# Contents

<b>1</b>	<b>Physical Layer Characteristics</b>	<b>6</b>
1.1	System Parameters . . . . .	6
1.2	Physical Layer Latency . . . . .	8
1.2.1	Latency in the Transmitter . . . . .	8
1.2.2	Latency in the Receiver . . . . .	8
<b>2</b>	<b>MU-MIMO MAC Characteristics</b>	<b>10</b>
2.1	Summary of MASCOT MAC . . . . .	10
2.1.1	Specification and Features . . . . .	10
2.1.2	Low-level Protocol . . . . .	11
2.1.3	Architectural Overview . . . . .	12
2.2	MAC Characteristics . . . . .	14
2.2.1	Traffic Scenarios . . . . .	14
2.2.2	Scheduling . . . . .	18
2.3	Performance Assessment . . . . .	20
2.3.1	PHY and MAC Layer Overhead . . . . .	20
2.3.2	MAC Layer Throughput Limits . . . . .	21
<b>3</b>	<b>System Verification and Basic Measurements</b>	<b>26</b>
3.1	PHY Assessment and Loopback Tests . . . . .	26
3.2	RF Imperfections . . . . .	27
3.2.1	Error Vector Magnitude . . . . .	28
3.2.2	Digital Signal Processing Implementation Loss . . . . .	29
3.2.3	Cable Channel . . . . .	29
3.2.4	Real World Wireless Channel . . . . .	30
3.3	MASCOT Testbed Measurements . . . . .	32
3.3.1	Latency Measurements . . . . .	33
3.3.2	Throughput Measurements . . . . .	33

<b>4</b>	<b>Summary and Roadmap</b>	<b>37</b>
4.1	Summary . . . . .	37
4.2	Roadmap . . . . .	38
4.2.1	Simultaneous MU-MIMO Uplink . . . . .	38
4.2.2	MMSE-SQRD ASIC Integration . . . . .	40
4.2.3	RF Imperfections . . . . .	40

# Authors

*Peter Lüthi, Markus Wenk*

Eidgenössische Technische Hochschule Zürich

tel.: +41 44 632 50 08 (Peter Lüthi)

e-Mail: {luethi,mawenk}@iis.ee.ethz.ch

# Executive Summary

The aim of Task 2.3 of the MASCOT project is the development, testing, and demonstration of multi-user (MU) MIMO algorithms on a real-time FPGA-based testbed. This deliverable describes the status of this work, the implemented setup, and the planning of the work ahead. This report focuses on the practical aspects and the testing of the MU-MIMO *Medium Access Control* (MAC) algorithm implemented in the MASCOT testbed. The MAC layer is required to successfully implement and demonstrate advanced MU-MIMO MAC functionality in the testbed. Nevertheless, some information about the physical (PHY) layer of the MASCOT testbed is given as well since it is a prerequisite for understanding the MAC layer performance aspects.

The first part of this report outlines the physical layer characteristics of the MASCOT testbed. An overview of the system parameters of the PHY layer is provided, and the various PHY layer latencies induced by communication protocol standard and practical signal processing are introduced and discussed.

The second part describes the characteristics of the implemented MU-MIMO MAC algorithms. The specifications and features of the MASCOT MAC algorithms are summarized, and an overview of the low-level MAC protocol and the software architecture is given. Different application-specific traffic scenarios are described and analyzed. The impact of these traffic scenarios on different MAC scheduling algorithms is assessed by high-level simulations and discussed afterwards. Theoretical MAC layer performance limits are introduced based on the MASCOT PHY and MAC layer characteristics.

The third part of this report lists measurement results of the MASCOT PHY and MAC layer. It describes various hardware integrity tests being carried out on the MASCOT real-time testbed, and provides insight into performance impairments caused by RF imperfections.

At the end of this report, a summary about the testing of the MU-MIMO MAC and a roadmap of intended future work are given.

# Chapter 1

## Physical Layer Characteristics

In the following, the characteristic parameters of the *Physical* (PHY) layer of the MASCOT testbed are briefly outlined and an overview of various PHY layer latencies is given. Note that most of the characteristics discussed in this chapter refer specifically to hardware. Any CPU-based *Medium Access Control* (MAC) layer processing characteristics are subject of the subsequent chapter. A broader overview of the various hardware components used within this testbed can be found in earlier project reports [4, 7].

### 1.1 System Parameters

The physical layer of the MASCOT testbed is based on IEEE 802.11a. The testbed supports up to four spatial streams. The modulation parameters are summarized in Tbl. 1.1 and the coding parameters are presented in Tbl. 1.2. The physical layer data rates for the MASCOT testbed are listed in Tbl. 1.3.

Table 1.1: Modulation parameters

Channel bandwidth	20 MHz
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
Number of subcarriers	64 in total (12 zero tones, 4 pilot tones, 48 data tones)
OFDM symbol duration	4 $\mu$ s
Guard interval	0.8 $\mu$ s
Subcarrier spacing	312.5 kHz

Table 1.2: Channel coding parameters

Number of encoders	4
Convolutional code rate	$\frac{1}{2}$
Generator polynomials	$g_0 = 133_8, g_1 = 171_8$
Punctured coding rates	$\frac{3}{4}, \frac{2}{3}$

With this parameters, the achievable raw physical layer data rates can be computed.

$$PHY \text{ data rate} = \# \text{ of data subcarriers} \cdot \text{modulation order} \cdot \# \text{ of streams} \cdot \text{code rate}$$

The physical layer data rates present an upper bound for a given modulation and coding rate. In a real system, these data rates can never be achieved due to several reasons:

- MAC layer overhead
  - Each correctly received frame needs to be acknowledged.
  - Frame errors lead to retransmissions of one or several frames.
- Latency
  - Latency introduced in the physical layer due to the necessary signal processing.

Table 1.3: Supported 4-stream data rates

Modulation	Coding rate	PHY data rate
BPSK	$\frac{1}{2}$	24 Mbit/s
BPSK	$\frac{3}{4}$	36 Mbit/s
QPSK	$\frac{1}{2}$	48 Mbit/s
QPSK	$\frac{3}{4}$	72 Mbit/s
16-QAM	$\frac{1}{2}$	96 Mbit/s
16-QAM	$\frac{3}{4}$	144 Mbit/s
64-QAM	$\frac{1}{2}$	144 Mbit/s
64-QAM	$\frac{2}{3}$	192 Mbit/s
64-QAM	$\frac{3}{4}$	216 Mbit/s

- Latency introduced in the medium access control layer due to the time necessary to react on the incoming frame.

## 1.2 Physical Layer Latency

### 1.2.1 Latency in the Transmitter

In the transmit path, latency is kept short by storing the preamble, which is the same for all frames, in time domain just before the signal is upsampled. In our hardware testbench, we measured a delay of 560 ns between the frame start signalization from the MAC layer (PowerPC) and the first data item leaving the FPGA (Fig. 1.1). Some additional delay is introduced in the RF chain until the frame is actually on the air. We can conclude that the latency introduced in the transmit path of the physical layer is about 1  $\mu$ s. Details on the transmit latency are also given in Tbl. 1.4.

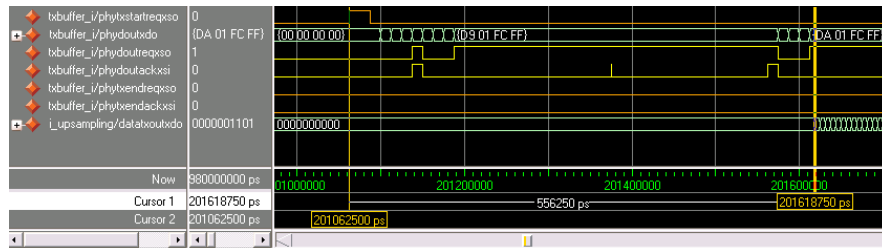


Figure 1.1: Simulation snapshot of the transmit latency in the PHY

Table 1.4: Physical layer latency in the transmit path

	Latency
MAC-PHY layer interaction delay for frame start signalling	300 ns
PHY layer digital signal processing delay	560 ns
RF processing delay	100 ns
Total PHY layer transmit latency	1 $\mu$ s

### 1.2.2 Latency in the Receiver

The design of a MIMO receiver with a small latency is a challenging task since the complexity of the signal processing necessary to separate the spatial streams is drastically increased compared to SISO systems.

There are several sources for latency in a MIMO receiver:

**OFDM symbol latency:** The transformation from time-domain into frequency domain introduces 4  $\mu\text{s}$  latency because the FFT transformation can only be applied when the entire frame has arrived.

**Preprocessing latency:** Channel estimation and preprocessing account for an initial latency at the beginning of each frame. The frame structure of packet-based MIMO-OFDM systems poses difficult challenges for the implementation of the channel estimation and the preprocessing stage of MIMO receivers, because the initial training phase for channel estimation is immediately followed by data. Since the detection of the data can only start when the preprocessing of all channel matrices is done, the data frames need to be buffered. This delay incurred by the preprocessing translates directly into detection latency. Further information on this topic can be found in [3].

**Detection latency:** The processing and detection of each received symbol vector introduces additional latency in the receive path that strongly depends on the implementation of the used detection algorithm.

**Decoding latency:** Also the decoding stage accounts for some latency. This includes metric computation, deinterleaving, decoding, and depuncturing.

Tbl. 1.5 shows the latency figures in the receiver. The latency strongly depends on the frame length. Therefore, a lower and an upper bound for the receive latency are computed. For long frames, all the preprocessing latency can be caught up. However, for short frames, the latency is dominated by the channel preprocessing.

Table 1.5: Physical layer latency in the receive path

	Latency
Initial preprocessing latency	58 $\mu\text{s}$
OFDM symbol latency	4 $\mu\text{s}$
Detection latency	3.2 $\mu\text{s}$
Decoding latency	2.2 $\mu\text{s}$
Total PHY layer receive latency for short frames	67.4 $\mu\text{s}$
Total PHY layer receive latency for long frames	9.4 $\mu\text{s}$

# Chapter 2

## MU-MIMO MAC Characteristics

The implemented MU-MIMO MAC – also referred to as MASCOT MAC – represents a simplified MAC architecture streamlined for the needs of our real-time MIMO testbed. A detailed description about the entire hardware infrastructure including the PowerPC processor subsystem is given in [7]. More information about the MAC architecture, the software framework itself, and especially MAC scheduling aspects can be found in [6].

### 2.1 Summary of MASCOT MAC

In the following, a summary of architectural specifications and features of the MASCOT MAC is given.

#### 2.1.1 Specification and Features

The MASCOT MAC has been designed with the following criteria in mind:

- The network vicinity allows for one access point (AP) and multiple stations (STA). All devices, AP and STAs, have a unique identifier (ID) to allow for dedicated device addressing.
- The access point employs a *Point-Coordination Function* (PCF) for centralized and simplified servicing of the stations. The AP acts as master for the entire network vicinity, the STAs are configured as slaves. The AP is responsible for polling all STAs regularly according to its internal scheduling algorithm. No direct peer-to-peer traffic from one STA to another is allowed.

- An embedded PowerPC processor is used for the implementation of the MASCOT MAC layer. The CPU has a clock frequency of 240 MHz, the processor local bus and the subordinate system bus are both clocked at 80 MHz. The CPU incorporates an instruction-side and a data-side cache controller, which both communicate with their respective memory across the processor local bus.
- The MASCOT MAC employs a proprietary 32 byte MAC header and a 4 byte CRC field for each packet<sup>1</sup>. The proprietary header format allows for faster and more convenient MAC layer adaptations compared to a standard's adhering approach.
- Two different scheduling algorithms are implemented in the MASCOT MAC: A simple but fair *Round Robin* algorithm, and a more sophisticated *Budget* scheduler. By default, the budget scheduling algorithm is used, since it was designed to deliver a better aggregated system throughput, especially in asymmetric load scenarios, i.e., where one STA needs a high download rate and another STA requests only little data.
- Frame aggregation capability has been implemented in the MASCOT MAC in order to aggregate up to 10 maximum size Ethernet packets (10×1500 byte payload) to a single MIMO frame. The aggregation capability has been tested for small frames, but not yet been deployed to the entire testbed due to partial limitations in the current system configuration and implementation. Currently, the aggregation feature is disabled by default.

### 2.1.2 Low-level Protocol

The MASCOT MAC has been designed to employ a single, centralized coordinator – the AP. Only the AP is allowed to initiate a sequence of transfers. The STAs are only allowed to respond upon a specific request of the AP. The MASCOT low-level communication protocol is shown in Fig. 2.1 and consists of three different types of transfers: REQ, RSP, and optional ACK. Both REQ and RSP transfers may contain payload (DATA). The MASCOT low-level protocol is specified to allow payload in either or both transfers of a REQ-RSP sequence.

---

<sup>1</sup>In the following, the term *packet* is used as a synonym for the word *frame*, representing some kind of structured data.

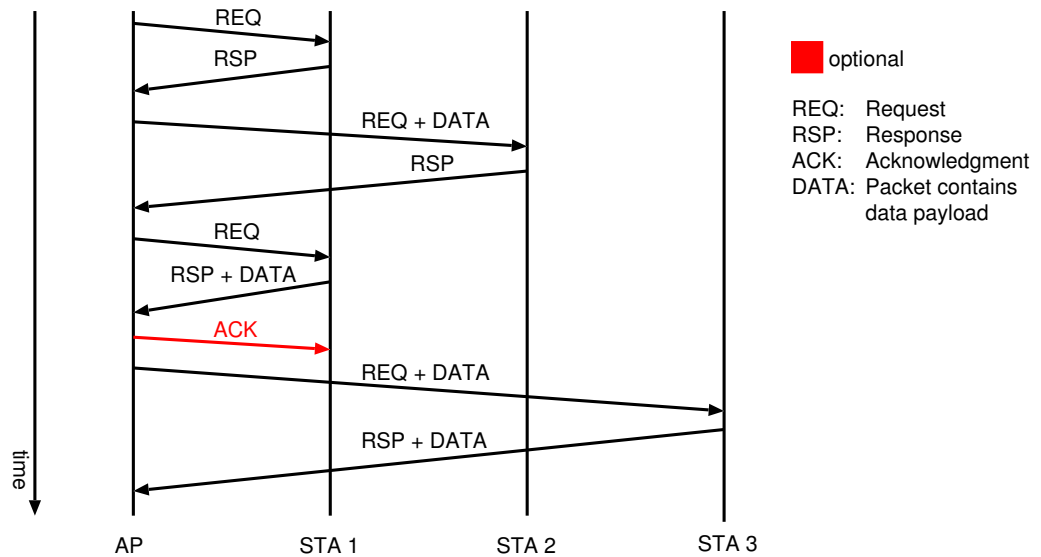


Figure 2.1: Low-level communication protocol of the MASCOT MAC.

### 2.1.3 Architectural Overview

The hardware/software layering of the MASCOT MAC is depicted in Fig. 2.2 and incorporates four major layers. The lowest layer comprises purely hardware blocks with MAC-related functionality. For the MASCOT MAC, the lowest layer consists of two low-level MAC blocks for Ethernet (ETH) and MIMO-PHY (PHY). These two blocks implement dedicated MAC functionality, which needs to be handled in hardware, e.g. transmit/receive functionality or interrupt logic. All higher layers are then purely realized in software. The next layer above the low-level MAC blocks contains also two different blocks, Ethernet Management (EM) and Baseband (BB). The EM and BB blocks are completely interrupt driven and perform all low-level protocol and queue management services for the subordinate hardware blocks. Above the EM and BB blocks, the Background (BG) layer and the interactive shell ap-

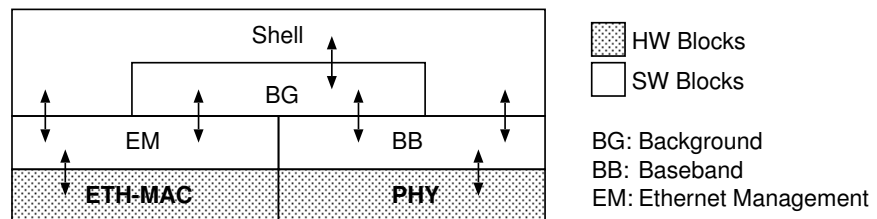


Figure 2.2: Hardware/software layering of the MASCOT MAC.

plication (Shell) are located. The BG layer controls both the EM and BB blocks and is responsible for packet routing, scheduling, and memory management. The shell application constitutes the highest layer in the software hierarchy and allows for direct user interaction with the MASCOT MAC software even during MAC operation.

The BG layer processes all packets either received from the BB or EM layer. Moreover, the BG layer is also able to accept artificially stimulated packets from the shell application, for instance a PING packet to check the network vicinity. The packet flow and handling in the BG layer is illustrated in Fig. 2.3 for both AP and STA. In the AP, the packets are first inspected for integrity, validity, and correct association. After successful inspection, the packets are individually routed to the correct destination by putting them in the corresponding MIMO transmit queue or forwarding them directly to the Ethernet link. Finally, the scheduling algorithm decides which packets are sent next across the wireless link. In the STA, the BG layer essentially has the same task as in the AP, but does not need to perform any packet routing and scheduling.

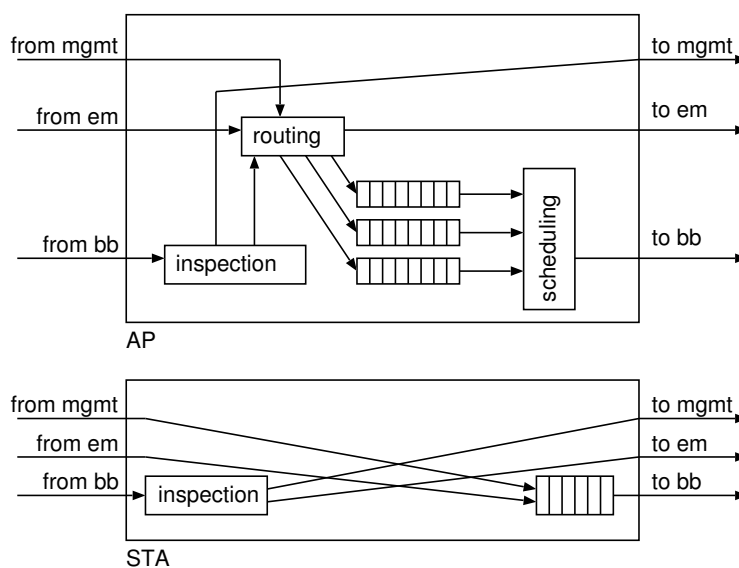


Figure 2.3: Different Background (BG) tasks for AP and STA illustrating the MASCOT MAC layer packet flow and handling.

The basic memory management data structures, for instance for queue management, are also located in the BG layer, although they are commonly used by BG, EM, and BB blocks and therefore constitute a low-level data management layer.

## 2.2 MAC Characteristics

This section provides an overview of different traffic scenarios and its implications on scheduling algorithms and MAC layer performance.

### 2.2.1 Traffic Scenarios

Efficient scheduling algorithms in terms of data rate are required only, if the traffic volume is high and the goal is to maximize the aggregated throughput of the system. In case of low data volume, the simple Round Robin scheduling algorithm performs quite well. High data rates usually imply that large Ethernet packets are transmitted, although certain network applications (e.g. a DNS server) may create high data volume with small packets. However, such applications are usually found in dedicated server environments and are not directly connected to a WLAN network.

In a typical WLAN network, it can be expected that for a scenario with high data volumes the majority of Ethernet packets have sizes near the maximum Ethernet packet size of 1518 bytes. In the following, a selected set of high-rate data streams are briefly discussed:

**TCP Stream** The TCP protocol [10, 1] provides a reliable, byte-stream-oriented connection between two endpoints. The *slow* startup behavior of TCP avoids initial buffer overflow at network nodes between the endpoints. The rate of the TCP stream is steadily increased during transmission towards the maximum. The *congestion avoidance* feature treats lost packets as an indication of congestion and automatically throttles the bandwidth. The TCP protocol is widely used for data-centric connection types with strict ordering and error handling such as FTP [11] file transfers. These transfers profit from high network bandwidth and do not impose low latency requirements.

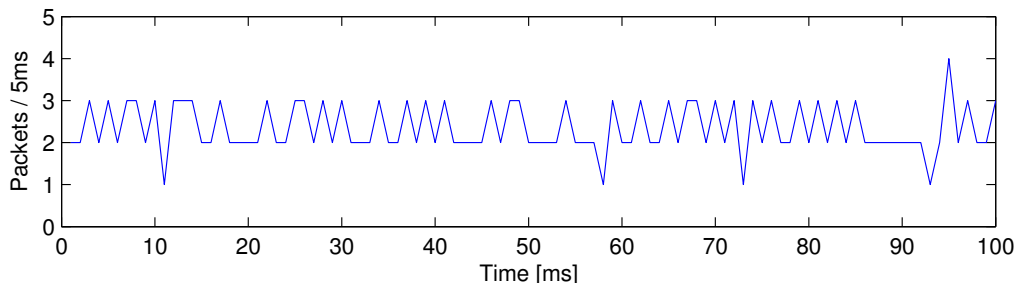


Figure 2.4: Burst behavior of a TCP data stream of 3 Mbit/s.

A typical TCP data stream employing maximum payload size of 1500 byte is shown in Fig. 2.4. As can be seen, there is a more or less constant data flow, and no explicit packet burst behavior is visible. A TCP stream also implies a constant flow of ACK packets in the opposite direction. In the stream under consideration, one ACK packet was generated for about every two data packets. An ACK packet contains 66 bytes.

An important aspect to consider is the TCP *window size*. Although TCP streams do not ask primarily for low latency, the latency shouldn't be too high either. In TCP, data is transmitted up to a dedicated *window size*, before waiting for an acknowledgment. If the *round trip delay* exceeds a certain value, the instantaneous throughput is automatically throttled:

$$\text{Throughput} \leq \frac{\text{window size}}{\text{round trip delay}}$$

The original TCP specification limits the TCP window size to 65 kByte, or roughly 45 maximum-size Ethernet packets. However, modern TCP stack implementations allow for tuning the TCP window size up to 1 GByte. Nevertheless, it is important to keep the TCP packet delay low in order to avoid throughput degradation by the automatic throttling mechanism.

**IP-TV Video Stream** The ETH Zurich broadcasts various TV channels within the university network. IP multicast is used for the technology of delivery. Audio and video data are encoded using MPEG-2 compression and are then multiplexed into one MPEG-2 RTP [13] stream transmitted as UDP [9]. RTP/UDP provides no congestion avoidance like TCP does, the data rate to be used has to be determined by other means. The IP-TV streams within the ETH Zurich network have an average data rate of 3Mbit/s and cannot be altered by the user.

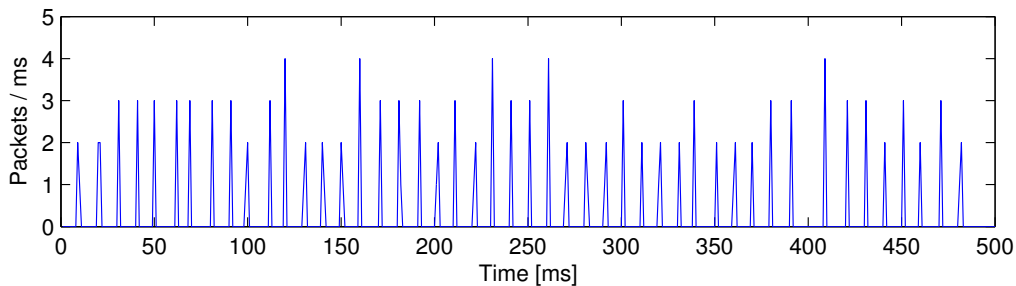


Figure 2.5: Burst behavior of a IP-TV video stream of 3Mbit/s.

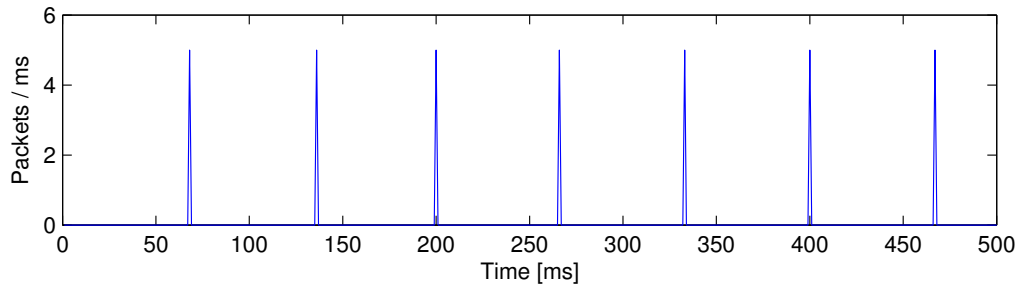


Figure 2.6: Burst behavior of a H.264 video stream of 2 Mbit/s.

The packet flow of a IP-TV stream of 3 Mbit/s is shown in Fig. 2.5. In contrast to TCP, this stream incorporates a dedicated packet burst behavior. The characteristic peaks occur at quite short inter-packet delays of 15-20  $\mu$ s and 110  $\mu$ s, and two peaks at significant larger inter-burst delays, 10 ms and 17 ms, respectively. There are additional peaks at inter-packet delays between 20  $\mu$ s and 350  $\mu$ s, but those are not shown in the figure.

**RTP Video Stream** This type of stream is closely related to IP-TV video streams. Since IP-TV streams multiplex the audio/video streams into a single RTP/UDP stream, it is interesting to analyze a multimedia stream where this is not the case. Typical applications like video conferencing use a dedicated RTP/UDP connection for each type of multimedia content (audio, video, data) independently.

A 2 Mbit/s video stream between two video conferencing endpoints has been captured and is depicted in Fig. 2.6. The video is encoded using the H.264 codec. Unlike IP-TV, the packet size is not constant in the video stream, since modern video codecs like H.264 apply good motion compensation. If there is little motion in the video, the encoded video frames contain almost no data. Video frames that do not fit within one Ethernet packet have to be fragmented. Fragmentation is usually done at a dedicated boundary such that the impact of lost packets for the entire video frame is minimized. This is the reason why the packet size varies and the encoded video frame is not simply put into as few Ethernet packets as possible.

**Comparison of Packet Delay** The packet delay for several types of network traffic is depicted in in Fig. 2.7. Different types of network traffic exhibit quite different burst characteristics. With respect to multimedia (RTP) streams, there are many packet bursts with extremely short packet delays of

several microseconds within the burst, but rather long inter-burst delays of multiple milliseconds, as shown in Fig. 2.8.

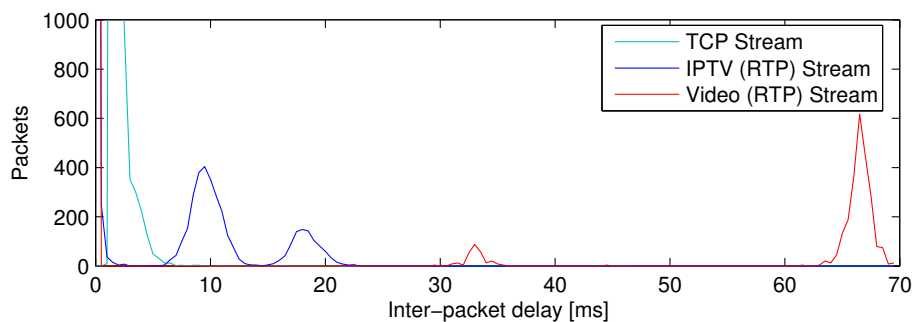


Figure 2.7: Inter-packet delay characteristics of selected streams. Packet peaks are cut at 1000 packets in order to visualize small peaks.

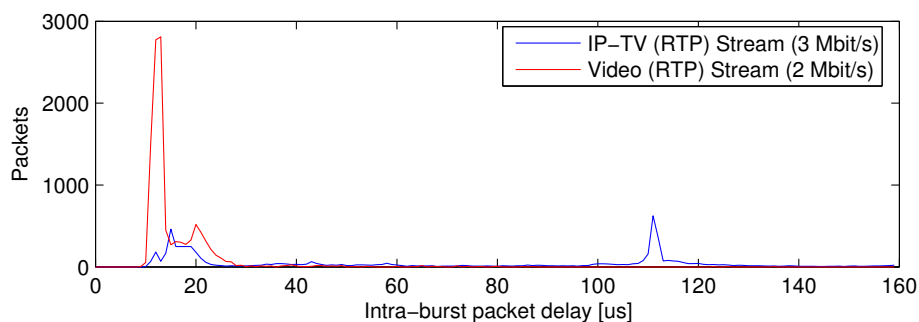


Figure 2.8: Intra-burst packet delay characteristics of multimedia streams. In a burst, packets arrive with extremely short delays of about 10-20  $\mu$ s.

### 2.2.2 Scheduling

In the MASCOT testbed, the AP has to schedule all packet exchanges with the associated STAs. In order to maximize the aggregated throughput, a packet scheduling algorithm needs to best adapt to different network and traffic scenarios under various circumstances. The focus of this section resides on the high-level packet scheduling algorithm implemented in the BG layer of the MASCOT MAC. Low-level scheduling aspects such as packet retransmission or rate adaptation are not discussed here. Moreover, the scheduling algorithms under consideration for our testbed operate at *best effort* and do not implement any *Quality of Service* (QoS) mechanisms. Packets are only processed based on their source and destination address and not based on their type.

The quality metrics for scheduling algorithms appropriate for our system can be derived from various criteria such as *aggregated throughput*, *latency*, *fairness*, *starvation*, *implementation complexity*, and *scalability* in terms of number of associated STAs. For our testbed, the primary quality metric is the aggregated data throughput. Latency and fairness are not of primary concern, but are considered whenever end-user applications show a remarkable degradation. Implementation complexity is also taken into account because of limited processing and memory resources in our testbed.

The following two scheduling algorithms have been implemented and assessed in simulation. Practical assessment in terms of direct performance comparison has not been carried out in the testbed to date.

- **Round Robin Scheduler** As the name implies, this scheduling algorithm serves each associated STA consecutively in turn. Moreover, the algorithm scales well with increasing number of clients to serve, and is commonly known as prototype for fairness-based scheduling. Each STA gets the same number of opportunities to exchange data packets, and starvation does not occur.
- **Budget Scheduler** This algorithm serves each associated STA consecutively in turn. Each time a STA is served,  $n$  data exchanges are scheduled,  $n \in [0, \dots, N_{max}]$ . The basic idea of this algorithm is that for each scheduling round, all STA have the same channel access *budget* and each data exchange has a *cost*. The algorithm will schedule as many packets as possible, until either the budget is completely used, or there are no more packets left to be transmitted. If the total cost exceeds the budget, the available budget is reduced (for all STAs) accordingly in the next scheduling round. The cost function considers the modulation

and coding rate, i.e., essentially the *bits per symbol* property. The cost function basically reflects the *frame duration* for a given packet size.

The performance of both scheduling algorithms for a packet stream with arbitrarily high data rate is visualized in Fig. 2.9. The generic simulation setup consists of one AP and three STAs, employing 64-QAM modulation with coding rate 5/6 (available in simulation only). The data shown is the throughput based on the packet transfers going from the AP to STA1. As can be seen, the Budget scheduling algorithm achieves approximately a throughput of 180 Mbit/s, what is a performance improvement of 66% compared to the Round Robin scheduler. The theoretical throughput limit in this scenario is 188 Mbit/s.

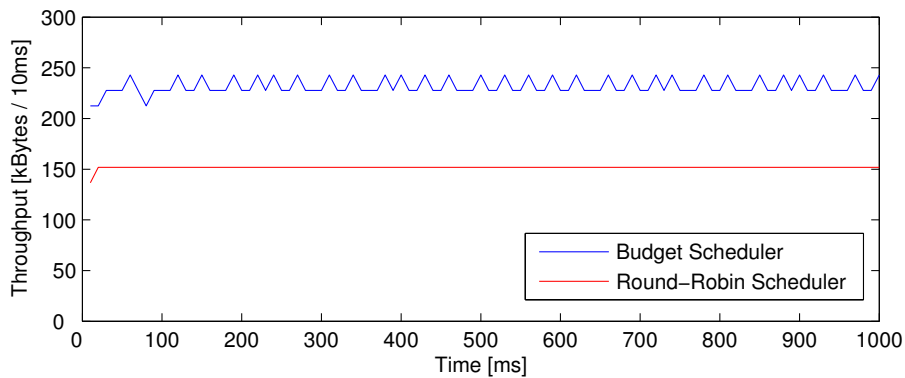


Figure 2.9: Throughput comparison between *Budget* and *Round Robin* scheduler. These results are derived from a dedicated high-level simulation environment to assess the characteristics and performance of both scheduling algorithms.

**Conclusion** A good scheduling algorithm is able to quickly adapt itself to different burst characteristics in order to best meet its quality metric. In our case, the primary quality metric is high aggregated throughput. The proposed Budget scheduling algorithm promises to maximize data throughput and performs significantly better than the Round Robin scheduler. However, packet streams exhibiting excessive burst characteristics might suffer from an increased latency when employing the Budget scheduling algorithm.

## 2.3 Performance Assessment

Network-oriented communication systems usually employ a layered protocol architecture in order to ensure reliable data transfers across a multitude of different physical media. The different protocol layers are for instance responsible for error handling, flow control, congestion control, or encryption. The MASCOT testbed needs to handle only two layers – the PHY and MAC layers – in order to allow for seamless integration into an entire network infrastructure. Since every protocol layer has its individual protocol overhead affecting the end-user throughput, it is essential to know which throughput can be expected at maximum at a certain protocol layer. In the following, an overview of the protocol overheads in the MASCOT testbed and the corresponding maximum throughput figures are given.

### 2.3.1 PHY and MAC Layer Overhead

The individual components contributing to the total protocol overhead for the PHY and MAC layer are listed in Tbl. 2.1 and Tbl. 2.2, respectively. For the subsequent throughput figures, the underlying SIFS time has been assumed to be 16  $\mu$ s, although the SIFS time in the MASCOT testbed varies with the frame length. The legacy training involves short and long training symbols and amounts to 24  $\mu$ s. To sum up, the total PHY layer overhead is equivalent to 60  $\mu$ s, and the total MAC layer overhead comprises 36 byte.

Table 2.1: PHY layer overhead of MASCOT testbed

SIFS	16 $\mu$ s
Legacy Training	24 $\mu$ s
Legacy SIG	4 $\mu$ s
HT Training	16 $\mu$ s
Total PHY Layer Overhead	60 $\mu$ s

Table 2.2: MAC layer overhead of MASCOT testbed

MAC Header	32 byte
CRC	4 byte
Total MAC Layer Overhead	36 byte

### 2.3.2 MAC Layer Throughput Limits

Based on the characteristic system parameters of the MASCOT testbed summarized in Tbl. 2.3 and the inherent PHY and MAC layer overhead listed in Tbl. 2.1 and Tbl. 2.2, we can derive throughput limits for dedicated scenarios. Taking all throughput-detrimental effects of PHY and MAC layer into account leads to a figure we call *raw MAC layer throughput*. But what does *raw* stand for? The term *raw* indicates that the MAC layer throughput calculation is based on a setup implying perfect channel state information, no interference and no outage, hence no errors and no retransmissions occur. Still, all protocol overhead is present, and the MAC layer throughput is calculated based on the MAC payload only.

Two example scenarios A and B are introduced in order to illustrate the effect of inherent protocol overhead and different payload sizes on the raw MAC layer throughput. The settings of the two scenarios are given in Tbl. 2.4.

Table 2.3: System parameters of MASCOT testbed.

Spatial multiplexing mode	4
Numbers of subcarriers	48
Guard interval	0.8 $\mu$ s
Symbol time	4.0 $\mu$ s

Table 2.4: Example scenarios for MAC throughput calculation.

Scenario	A	B
Modulation	BPSK	64-QAM
Coding rate	1/2	3/4
Coded bits per subcarrier per stream	1	6
Aggregated coded bits per channel use (4x4 MIMO)	96	864

If the size of the MAC payload is increased, the raw MAC layer throughput increases as well. But it heavily depends on the MAC payload size, whether the raw MAC layer throughput is significantly lower than the characteristic PHY layer throughput figure. For scenarios A and B, the corresponding throughput numbers are listed in Tbl. 2.5 and Tbl. 2.6 for different sizes of MAC payload.

**Scenario A** employs BPSK modulation and rate 1/2 coding. The corresponding throughput numbers are shown in Tbl. 2.5. For small MAC payload sizes, the inherent protocol overhead dominates and hence degrades the raw MAC layer throughput. At payload sizes beyond 1500 bytes, the modulation scheme starts to limit the MAC layer throughput, since the theoretical PHY layer throughput limit of 24 Mbit/s is approached.

Table 2.5: Scenario A: Raw MAC layer throughput for BPSK r1/2

PHY overhead [ $\mu\text{s}$ ]	60					
MAC overhead [bytes]	36					
MAC payload [bytes]	32	100	1500	4096	16384	65536
Total size of MAC part [bytes]	68	136	1536	4132	16420	65572
Length of MAC part [symb]	6	12	128	345	1369	5465
Duration of MAC part [ $\mu\text{s}$ ]	24	48	512	1380	5476	21860
Total frame duration [ $\mu\text{s}$ ]	84	108	572	1440	5536	21920
Frames per second [fps]	11904.8	9259.3	1748.3	694.4	180.6	45.6
PHY throughput <sup>a</sup> [Mbit/s]	24					
MAC throughput <sup>b</sup> [Mbit/s]	3.0	7.4	21.0	22.8	23.7	23.9

<sup>a</sup>characteristic PHY layer throughput excluding any protocol overhead

<sup>b</sup>based on MAC payload and including PHY and MAC layer overhead

**Scenario B** employs 64-QAM modulation and rate 3/4 coding. Tbl. 2.6 lists the corresponding throughput numbers. Again, for small MAC payload sizes, the raw MAC layer throughput is significantly degraded, compared to the theoretical PHY layer throughput limit. High MAC layer throughput numbers can only be achieved by using large MAC payload sizes. When seeking to attain raw MAC layer throughput figures near the corresponding theoretical PHY layer throughput limit, packet aggregation is a premise.

A sweep of different MAC payload sizes for various modulation and coding schemes supported by the MASCOT testbed is depicted in Fig. 2.10. The plot is based on the system parameters given in Tbl. 2.3 and the PHY and MAC layer overhead listed in Tbl. 2.1 and Tbl. 2.2. The average MIMO frame transmission time for characteristic MAC payload sizes is visualized in Fig. 2.11. The vertical dotted line in the plots represents the PHY protocol overhead, and illustrates the dominant effect of the overhead for small payload sizes.

Table 2.6: Scenario B: Raw MAC layer throughput for 64-QAM r3/4

PHY overhead [ $\mu\text{s}$ ]	60					
MAC overhead [bytes]	36					
MAC payload [bytes]	32	100	1500	4096	16384	65536
Total size of MAC part [bytes]	68	136	1536	4132	16420	65572
Length of MAC part [symb]	1	2	15	39	153	608
Duration of MAC part [ $\mu\text{s}$ ]	4	8	60	156	612	2432
Total frame duration [ $\mu\text{s}$ ]	64	68	120	216	672	2492
Frames per second [fps]	15625.0	14705.9	8333.3	4629.6	1488.1	401.3
PHY throughput <sup>a</sup> [Mbit/s]	216					
MAC throughput <sup>b</sup> [Mbit/s]	4.0	11.8	100.0	151.7	195.0	210.4

<sup>a</sup>characteristic PHY layer throughput excluding any protocol overhead

<sup>b</sup>based on MAC payload and including PHY and MAC layer overhead

**Conclusion** High MAC layer throughput can only be achieved by applying packet aggregation. Especially for high-order modulation and coding schemes, the important aspect of the MAC payload size becomes apparent. The MASCOT testbed does currently not operate with frame aggregation, although aggregation support for up to 10 maximum Ethernet size packets has been conceptually foreseen and partially implemented.

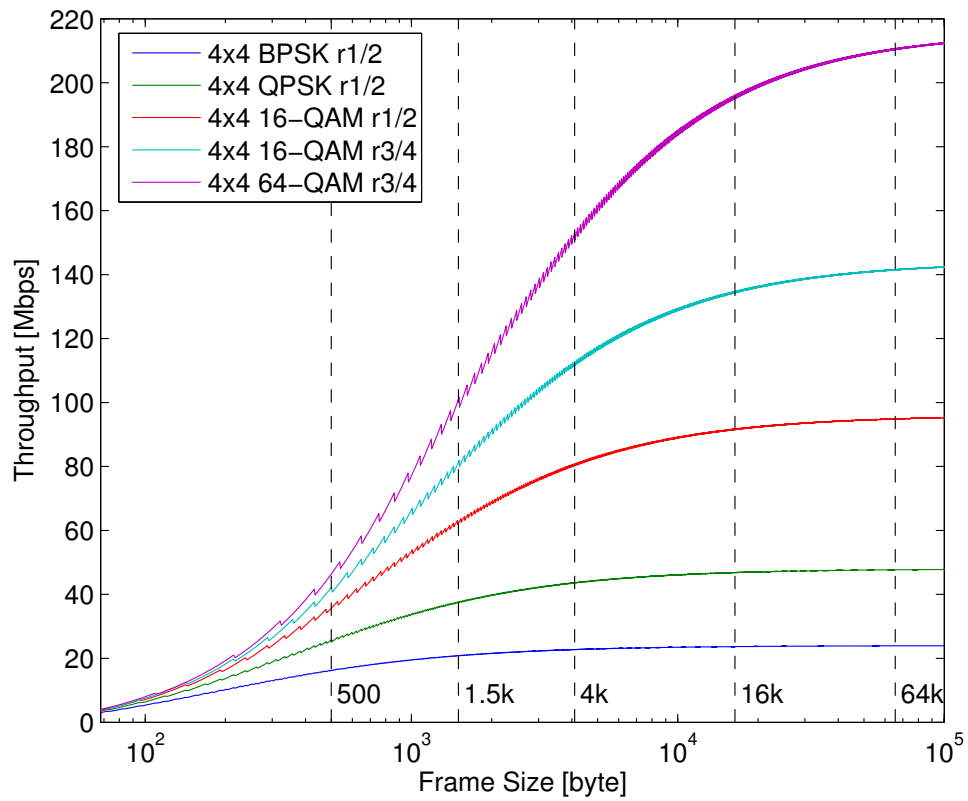


Figure 2.10: Raw MAC layer throughput for different modulation and coding schemes, based on the MASCOT testbed employing 48 subcarriers, normal guard interval, and an average interframe spacing of 16  $\mu$ s.

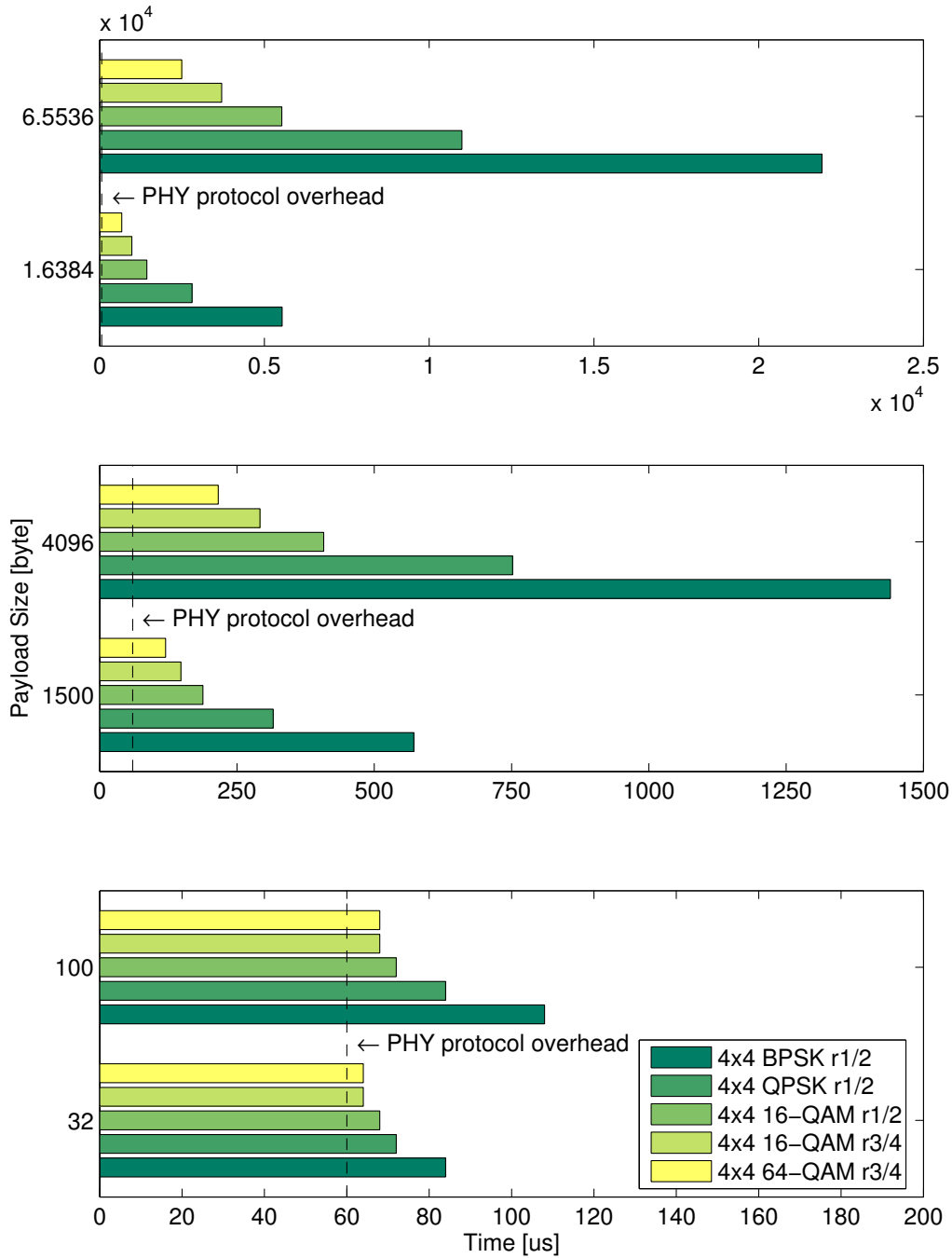


Figure 2.11: Average frame transmission time for different modulation and coding schemes. These results are based on the MASCOT testbed employing 48 subcarriers, normal guard interval, and an average interframe spacing of 16 μs. The resulting total PHY layer overhead including interframe spacing amounts to 60 μs.

# Chapter 3

## System Verification and Basic Measurements

In order to evaluate the performance of a system, it is necessary to know the parameters and the theoretical limits of the system. In this chapter, starting from the system parameters, we give insight into the system performance and highlight several aspects that influence the system behavior.

### 3.1 PHY Assessment and Loopback Tests

The implementation of loopback modes in the MIMO physical layer proved to be very useful debugging means. In total, there are four different loopback modes available. An overview of the loopback modes is presented in Fig. 3.1.

**MAC loopback:** In MAC loopback test mode, the data in the transmit buffer are directly written into the receive buffer.

**PHY loopback:** In PHY loopback test mode, all the signal processing on the large FPGA board (VAMP board) is carried out. At the end of the transmit chain, there is a buffer which stores the transmitted frame. Once the entire frame is in that buffer, it is fed back and processed by the receive chain. Finally, it is written into the receive buffer and can be checked for correctness.

**BAT loopback:** In BAT loopback test mode, all the physical layer signal processing is stimulated. The transmitted frame is first processed on the large FPGA board (VAMP board), passes the interface to the smaller FPGA board (BAT board), gets upsampled and is afterwards stored in a buffer, just before the frame would leave the digital world.

Again, once the entire frame is in the buffer, it is sent back into the receive chain and can finally be checked in the receive buffer.

**DAC-ADC loopback:** The DAC-ADC loopback mode stimulates all involved boards and interfaces. The transmitted frame is processed by the entire physical layer including the upsampling and the interface to the DAC. The used DAC/ADC device (AD9861) provides a loopback mode where the incoming data are sent back. The same loopback buffer already used for the BAT loopback is used to store the frame coming back from the DAC/ADC. As in the other loopback modes, the frame is taken out of the buffer and put onto the receive chain, once the entire frame has passed the transmit chain.

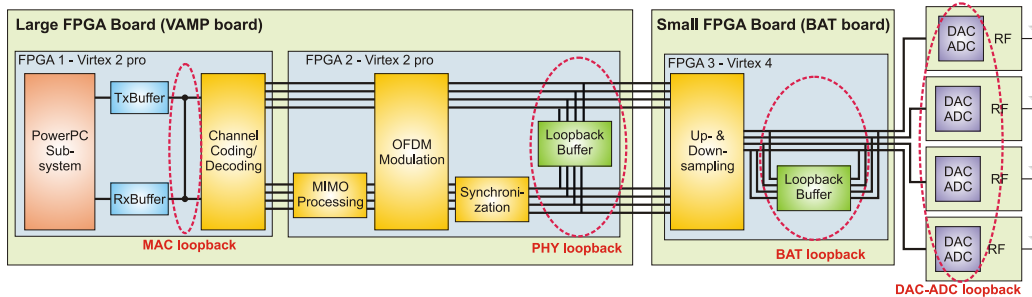


Figure 3.1: Four different loopback tests can be carried out to verify the proper operation of the physical and medium access control layers.

The DAC-ADC loopback mode can also be used to measure the implementation loss incurred by the digital signal processing, as will be presented in Sec. 3.2.2.

## 3.2 RF Imperfections

The requirements for high-throughput MIMO-OFDM systems pose many challenges to the implementation of the RF front-end. Even the most advanced RF front-ends include imperfections such as phase noise, I/Q imbalance, nonlinearities and the like. In this section, we try to measure and characterize the RF imperfections introduced in our testbed. A deeper treatment of RF imperfections in MIMO OFDM systems can be found in [12]. First, in order to get a baseline, the implementation loss introduced by the digital signal processing is characterized. Afterwards, measurements with a cable channel were carried out to get insight into the RF impairments. Measurements over real-world channels conclude this section on RF impairments.

### 3.2.1 Error Vector Magnitude

The error vector magnitude (EVM) is a measure used to quantify the performance of a digital radio transmitter or receiver. It is computed from the received constellation point and the ideal constellation point as illustrated in Fig. 3.2.

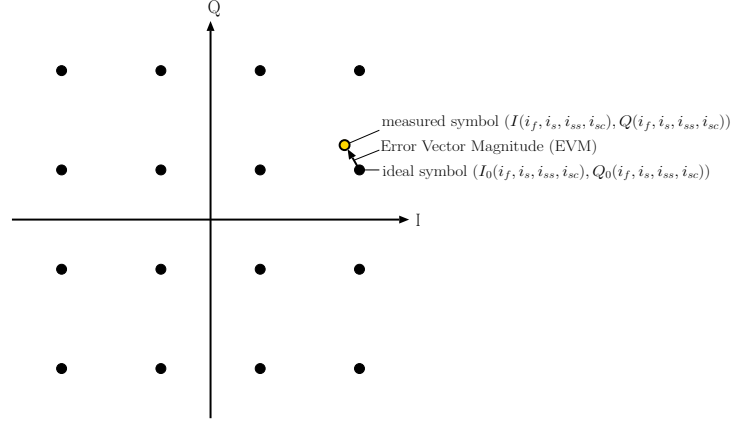


Figure 3.2: Constellation error - error vector magnitude (EVM)

In [2], the allowed relative constellation error is given for each data rate. According to (20-89) in [2], the average RMS of all errors in a packet is computed as

$$EVM_{RMS} = \frac{1}{N_f} \sum_{i_f=1}^{N_f} \left( \frac{1}{N_{SYM} N_{SS} N_{ST} P_0} \sum_{i_s=1}^{N_{SYM}} \sum_{i_{ss}=1}^{N_{SS}} \sum_{i_{sc}=1}^{N_{ST}} \left( (I(i_f, i_s, i_{ss}, i_{sc}) - I_0(i_f, i_s, i_{ss}, i_{sc}))^2 + (Q(i_f, i_s, i_{ss}, i_{sc}) - Q_0(i_f, i_s, i_{ss}, i_{sc}))^2 \right) \right)^{0.5} \quad (3.1)$$

where

$N_f$  is the number of frames for the measurement;

$(I_0(i_f, i_s, i_{ss}, i_{sc}), Q_0(i_f, i_s, i_{ss}, i_{sc}))$  denotes the ideal symbol point of the  $i_f$ th frame,  $i_s$ th OFDM symbol of the frame,  $i_{ss}$ th spatial stream,  $i_{sc}$ th subcarrier of the OFDM symbol in the complex plane;

$(I(i_f, i_s, i_{ss}, i_{sc}), Q(i_f, i_s, i_{ss}, i_{sc}))$  denotes the observed point of the  $i_f$ th frame,  $i_s$ th OFDM symbol of the frame,  $i_{ss}$ th spatial stream,  $i_{sc}$ th subcarrier of the OFDM symbol in the complex plane;

$P_0$  is the average power of the constellation.

The RMS value of the error vector magnitude is usually given in percent (%) or in decibel (dB).

$$EVM_{RMS}[dB] = 20 \log_{10}(EVM_{rms})$$

### 3.2.2 Digital Signal Processing Implementation Loss

Quantization and filtering in the up- and downsampling blocks introduce an implementation loss in the digital signal processing. The DAC-ADC loopback mode was used to quantify this implementation loss for the MASCOT testbed. The scenario looked as follows: The tested terminal was configured in DAC-ADC loopback mode. A frame was transmitted and received back over the DAC-ADC interface. The received frame was recorded after the MMSE detector. This procedure has been repeated for different modulation schemes.

Fig. 3.3 shows the constellation point diagram for different modulation schemes. The  $EVM_{RMS}$  was computed for all data subcarriers according to (3.1) with  $N_f = 1$ ,  $N_{SYM} = 25$ ,  $N_{SS} = 1$  and  $N_{ST} = 48$ . The measured EVM is the basis for all further measurements. It characterizes the maximum achievable performance. With a perfect wireless link ('ideal' RF chain, perfect channel state information, no noise) this performance can be achieved.

### 3.2.3 Cable Channel

To get insight into the imperfections introduced by our RF chain, we connected two terminals with an RF cable and attenuators and transmitted frames over this 'cable channel'.

Fig. 3.4 shows the constellation diagrams for this scenario. The constellation points become clearly larger due to RF impairments. The  $EVM_{RMS}$  reduces from about -34 dB to about -18 dB. We assume, that the following two reasons are mostly responsible for this performance degradation:

**I/Q imbalance:** Mutual differences in the I- and Q-path of direct-conversion transceiver architectures cause I/Q imbalance. I/Q imbalance results in a limited image rejection.

**Phase noise:** Imperfections in the radio frequency oscillator, which results from thermal noise, results in random deviation of the frequency. A detailed analysis of phase noise can be found in [12].

Further investigations on both topics will be carried out.

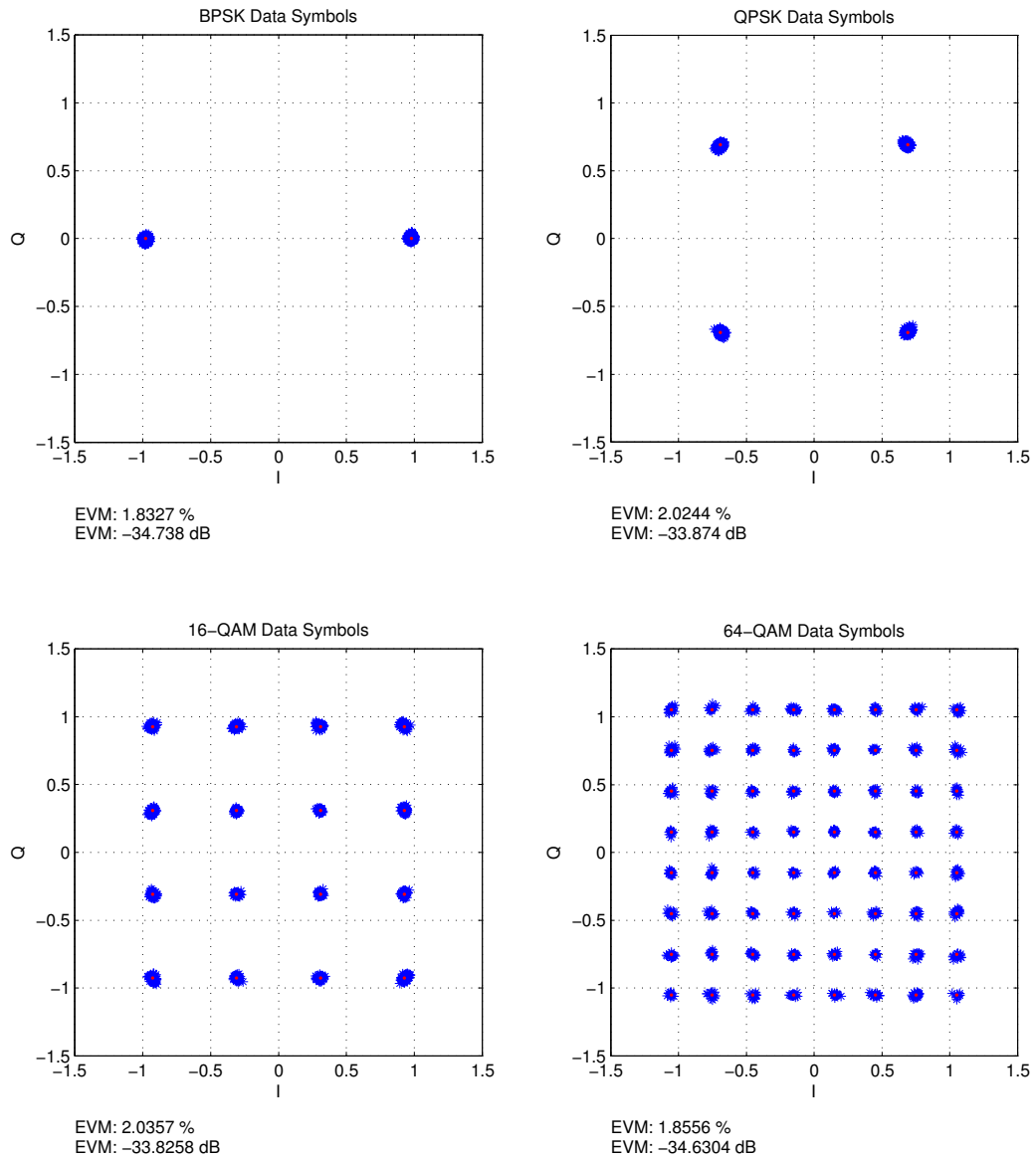


Figure 3.3: Constellation diagrams for different modulation schemes in digital loopback mode – for characterization of the implementation loss in the digital domain.

### 3.2.4 Real World Wireless Channel

Fig. 3.5 shows the constellation points for different modulation schemes over a real world channel. The propagation channel corresponds to scenario 4 presented in Sec. 3.3.2. It can be observed, that the performance degradation from the cable channel to the (rich scattering) wireless channel is only

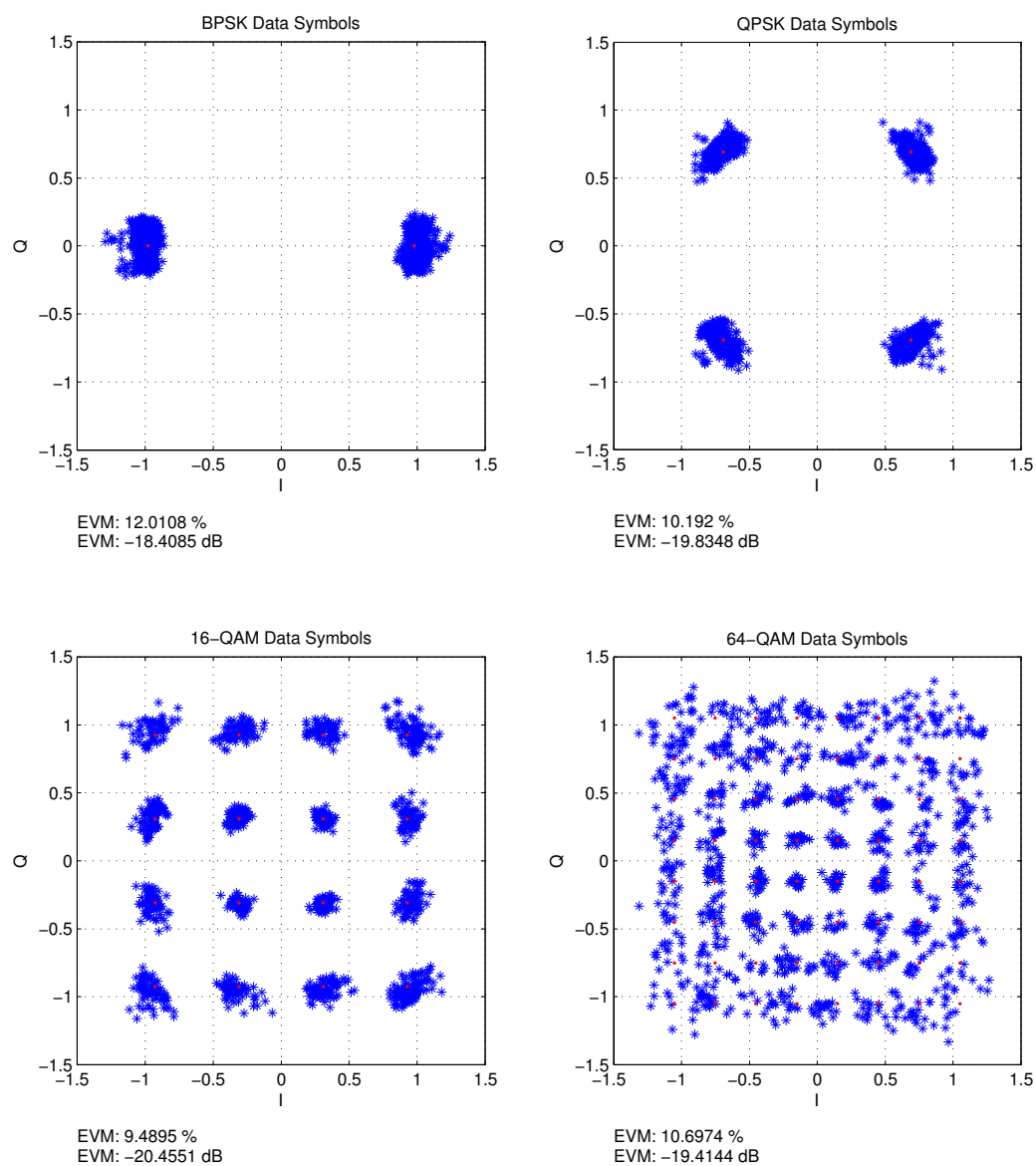


Figure 3.4: Constellation diagrams for different modulation schemes over the cable channel – for characterization of impairments caused by RF imperfections.

minor, which means that the testbed performance is mostly limited by the RF performance at the moment.

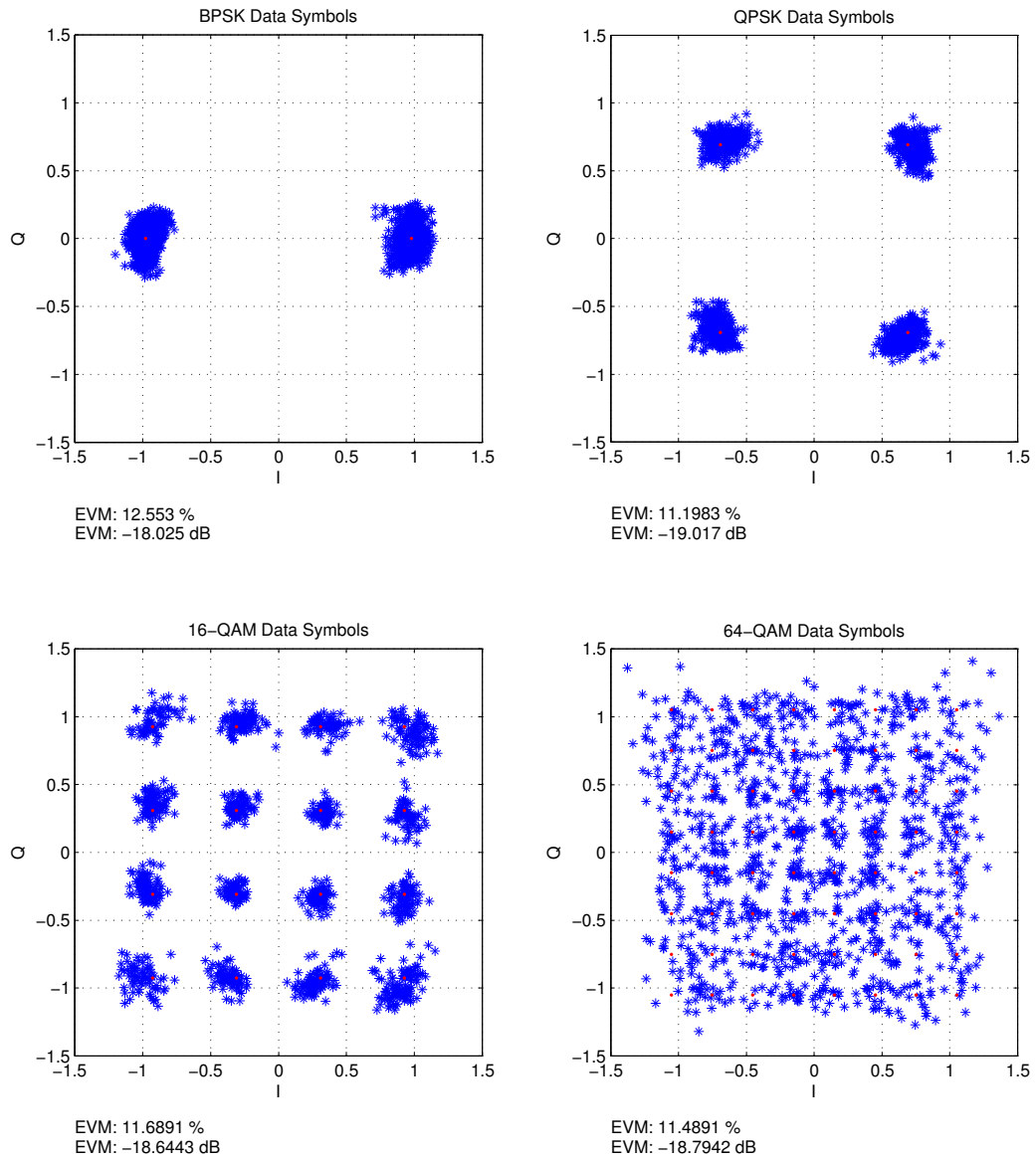


Figure 3.5: Constellation diagrams for different modulation schemes over a real world wireless channel (corresponds to scenario 4 in Sec. 3.3.2.)

### 3.3 MASCOT Testbed Measurements

This section presents measurements on latency and throughput with the current testbed setup.

### 3.3.1 Latency Measurements

Fig. 3.6 shows the measured idle time of the medium, what corresponds to the interval between the end of a received frame being measured on the air and the start of the transmitted frame on the air. As discussed in Sec. 1.2, there is an initial latency in the receive path of the physical layer due to channel estimation and preprocessing. This initial latency can be caught up during reception of the subsequent part of the frame. The measurement results show that it takes about 80 OFDM symbols to catch up the initial training and preprocessing latency. Once, this latency is caught up, the minimum inter-frame delay is almost constant at  $15\ \mu\text{s}$ .

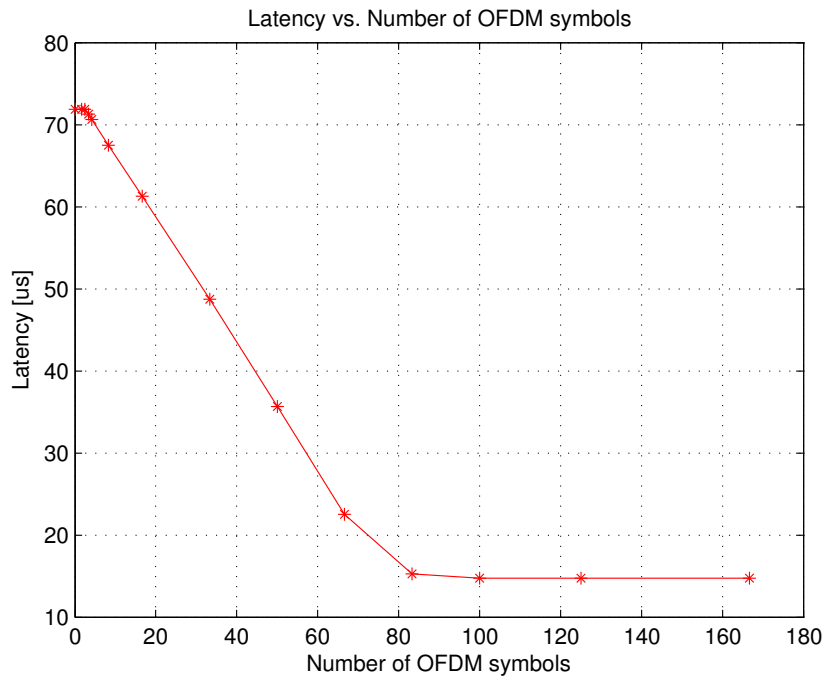


Figure 3.6: Minimum inter-frame delay caused by PHY transmit and receive latencies measured for different numbers of OFDM symbols.

### 3.3.2 Throughput Measurements

Throughput measurements with the MASCOT testbed were carried out with the software tool Iperf. Iperf is a commonly used network throughput assessment tool that can create TCP and UDP data streams and measure the throughput [8]. Measurements were carried out for four different scenarios:

**Scenario 1:** Cable channel: In this scenario, two testbed terminals are connected with each other by RF cables and attenuators. In principle, in such a scenario no frames should be lost.

**Scenario 2:** Very short distance line-of-sight propagation: In this scenario two terminals are placed very close to each other. The distance between the two terminals is 1 meter and there is line-of-sight (LOS) propagation

**Scenario 3:** Short distance line-of-sight propagation: In this scenario, the two terminals are placed about 5 meters apart from each other. Again, there is LOS propagation.

**Scenario 4:** Short distance non-line-of-sight propagation: In this scenario, the two terminals are again placed about 5 meters apart from each other. However, now we have a non-line-of-sight (NLOS) propagation condition as there is a wall in between the two terminals.

Tbl. 3.1 lists the average measured throughput over 100 seconds for each scenario and modulation.

Table 3.1: Measured TCP throughput for different modulations and scenarios

Modulation	BPSK	BPSK	QPSK	QPSK	16-QAM	16-QAM	64-QAM	64-QAM	64-QAM
Rate	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{4}$
Cable channel [Mbit/s]	8.9	11.9	21.0	25.9	30.4	23.5	18.8	4.8	1.1
Real channel, 1 m distance, LOS [Mbit/s]	13.9	NA	19.9	NA	3.3	NA	NA	NA	NA
Real channel, 5 m distance, LOS [Mbit/s]	14.1	17.6	20.2	14.2	24.3	0.79	1.7	NA	NA
Real channel, 5 m distance, NLOS [Mbit/s]	14.1	17.8	20.4	23.3	24.5	0.46	1.7	NA	NA

The measurements reveal the following observations:

- In scenario 1, the performance should actually be the best for all modulation schemes. However, in the case of BPSK, better throughput figures are achieved for the other scenarios. Further investigations showed that with the cable connection some kind of coupling occurs between the two stations. This results in a larger probability of a wrong synchronization, which locks the AGC and a frame arriving during this time will be over-amplified and therefore wrong. The reason why this effect is very pronounced in BPSK mode, is the fact that Ethernet frames have a maximum size of 1518 bits. This means that only in the BPSK case all the latency introduced by the preprocessing can be caught up.
- In scenario 2, where we have a very short line-of-sight propagation path (1 m), the channel has not enough degrees of freedom to allow other rates than rate  $\frac{1}{2}$  and low modulation orders.
- With the current setup, the highest sustained TCP throughput is achieved with 16-QAM modulation and rate  $\frac{1}{2}$ .

# Chapter 4

## Summary and Roadmap

### 4.1 Summary

Various aspects of the MASCOT real-time testbed contributing to the overall MAC layer performance have been pointed out in this report.

First, the characteristic parameters of the MASCOT PHY layer have been introduced, along with different latencies in the transmit and receive path of the MIMO PHY layer.

Second, the specifications and features of the MASCOT MAC layer were shown, and the MASCOT MAC protocol and software architecture have been presented. Several application-specific traffic scenarios and their impact on different MAC layer scheduling algorithms have been analyzed and discussed. The theoretical performance assessment of the MASCOT MAC identified the PHY and MAC layer overhead and showed the corresponding implications on the theoretical MAC layer throughput limit.

The third chapter contains MASCOT testbed measurement results for both, PHY and MAC layer. The measurement results using the digital loop-back functionality of the MASCOT testbed showed that all modulation and coding schemes are working correctly and with sufficient quality. The implementation loss in the digital domain of the system has been kept adequately low to successfully operate all modulation and coding schemes. However, the subsequent performance assessment across an orthogonal cable channel and real world wireless channels revealed still existing performance impairments even after spending an increased effort on mitigating RF imperfections. Nevertheless, the reliable operation of 16-QAM modulation schemes across real world wireless channels has been achieved.

## 4.2 Roadmap

The remaining tasks for WP 2.3 of the MASCOT project are as follows:

- Implementation of a simultaneous collision-based multi-user MIMO uplink in the MASCOT real-time testbed.
- Integration of the SQRD-ASIC developed within the MASCOT project in the MASCOT real-time testbed. This provides the base for real-world implementation and assessment of advanced MIMO detection methods.
- Further investigations and improvements in the RF chain to mitigate RF imperfections and to allow for high-rate MIMO transmissions.

### 4.2.1 Simultaneous MU-MIMO Uplink

In a simultaneous MU-MIMO scenario Fig. 4.1, all users transmit data simultaneously and in the same frequency band to a common base station. Each user is transmitting data from one or several antennas and possibly at different rates.

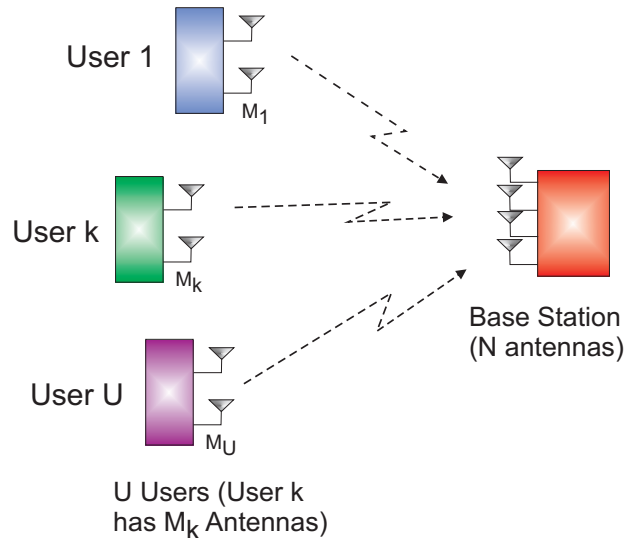


Figure 4.1: Simultaneous MU-MIMO uplink scenario.

## Basic Setup

The basic setup is outlined in Fig. 4.2. At least 2 users are communicating simultaneously with a base station. In a first phase, all users are connected to the same clock source in order to simplify the synchronization task in the base station.

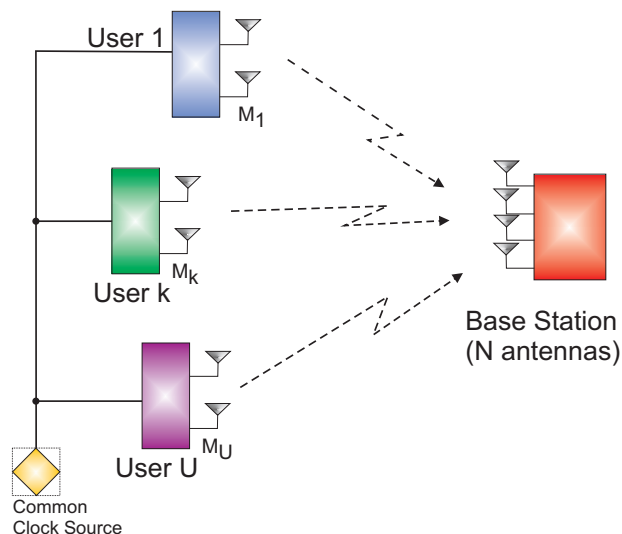


Figure 4.2: Simultaneous MU-MIMO uplink scenario with a common clock source.

The implementation of a simultaneous MU-MIMO uplink scenario involves the following tasks:

- Support for a variable number of streams per user.
- Synchronization mechanism for simultaneous frame transmission from different users (frame start).
- Coordination of MIMO training amongst all users, i.e., which user transmits which training sequence.
- MAC layer modifications in order to support simultaneous MU-MIMO uplink.
- PHY layer modifications in order to support instantaneous signalling of different coding rates per stream and from different users (signal field).

The goals of the basic setup are:

- Proof of concept for simultaneous MU-MIMO uplink.
- Stating of potential gain of simultaneous MU-MIMO uplink compared to a time-division-duplex-based uplink.
- Assess the practical aspects and requirements for such a scenario. Provide insight into practical issues and the potential protocol overhead.

### 4.2.2 MMSE-SQRD ASIC Integration

In case FPGA technology can not satisfy the demands for clock speed or throughput of certain algorithm implementations, the ETHZ-IIS has the ability for VLSI circuit integration in ASIC technology. For instance, selected MIMO-related algorithms or parts of the MIMO physical layer can be integrated as ASIC, and then be incorporated in the MASCOT testbed.

The integration of the MMSE-SQRD ASIC [5] into the MASCOT testbed will demonstrate the feasibility of this approach. Moreover, it will also provide the base for implementation and real-world assessment of advanced MIMO detection methods (e.g., sphere decoding) in the testbed. First, the detection performance of the physical layer can potentially be improved by applying advanced MIMO detection methods to the real-time testbed. Second, practical aspects of advanced MIMO detection methods can also be assessed in the context of MU-MIMO communication schemes.

### 4.2.3 RF Imperfections

Although we already put a lot of effort into improving the RF signal processing in the receiver and transmitter, there are still some open issues. As discussed in Sec. 3.2.2, the error vector magnitude of the received constellation points is still too high to reliably transmit 64-QAM symbols, even across a cable. The reason for the high EVM is not yet completely understood. Preliminary assumptions for causes of these RF performance impairments are:

- Suboptimal calibration or misconfiguration of parameters in the transceiver chip
- I/Q imbalance
- Non-linearities in the RF
- Quantization noise

As time permits, further investigations will be carried out to mitigate these technical issues.

# Bibliography

- [1] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), April 1999. Updated by RFC 3390. [14](#)
- [2] ANSI/IEEE 802.11n-2007/D2.0. *Draft STANDARD for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput*, 2007 edition. Reference number: IEEE P802.11n/D2.00. [28](#)
- [3] A. Burg, S. Haene, D. Perels, P. Luethi, N. Felber, and W. Fichtner. Algorithm and VLSI Architecture for Linear MMSE Detection in MIMO-OFDM Systems. May 2006. [9](#)
- [4] A. Burg, P. Luethi, and M. Wenk. Deliverable D2.3.1: Hardware extension and MAC upgrade for MIMO testbed, April 2007. IST-026905 MASCOT. [6](#)
- [5] P. Luethi et al. VLSI Implementation of a High-Speed Iterative Sorted QR Decomposition. In *Proc. IEEE Int. Symp. on Circuits and Systems*, pages 1421–1424, May 2007. [40](#)
- [6] H. Friederich. Multi-User MIMO MAC Demonstration, January 2008. ETH semester thesis. [10](#)
- [7] P. Luethi, M. Wenk, and D. Wagner. Deliverable D2.3.2a: Report on MIMO MAC extensions, December 2007. IST-026905 MASCOT. [6](#), [10](#)
- [8] NLANR/DAST. Iperf, 2005. <http://dast.nlanr.net/Projects/Iperf/>. [33](#)
- [9] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980. [15](#)

- [10] J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFC 3168. [14](#)
- [11] J. Postel and J. Reynolds. File Transfer Protocol. RFC 959 (Standard), October 1985. Updated by RFCs 2228, 2640, 2773, 3659. [14](#)
- [12] Tim Schenk. *RF Imperfections in High-rate Wireless Systems*. 2008. [27](#), [29](#)
- [13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. [15](#)