Quantitative Analysis of Physical Layer and Link Layer Measurements in WiFi Networks

Fakultät IV - Elektrotechnik und Informatik
Intelligent Networks / Intelligente Netze (INET)
Research Group of Prof. Anja Feldmann, Ph.D.

Julius Schulz-Zander
June 1, 2012

Prüfer: Prof. Anja Feldmann, Ph. D.
Betreuer: Çiğdem Şengül, Ph. D.
Betreuer: Steve Uhlig, Dr.
Betreuer: Ruben Merz, Ph. D.
Die selbständige und eigenhändige Anfertigung versichere ich an Eides Statt.

Berlin, den June 1, 2012

Julius Schulz-Zander
Abstract

In this thesis, we present a measurement study on the quantification of lost transmission opportunities in IEEE 802.11 networks, which is an increasingly more important problem due to a higher number of IEEE 802.11-based deployments. Lost transmission opportunities can arise from both unwanted traffic and link impairments. We perform passive measurements to gain information about unwanted traffic which lead to a reduction of available airtime and active measurements to gain information on link impairments which lead to packet loss or unfairness. In both cases, we follow a cross layer approach, since measuring lost transmission opportunities depends on information from two layers: (1) the physical (PHY) layer which provides signal quality and transmission statistics and (2) the medium access-control (MAC) layer which provides IEEE 802.11 protocol specific information, and throughput and loss statistics. We examine unwanted traffic to understand the impact on the environment and to support the development of mitigating mechanisms. To measure link impairments, we adapt an active measurement approach [17], which is, to the very best of our knowledge, the only approach that is able to estimate the majority of the reasons behind packet loss. We present our measurement methodology on how to collect network wide measurements with off-the-shelf WiFi hardware. All measurements are performed in a typical office environment located on the 16th floor in the TEL building at the Technische Universität Berlin. Using our measurement framework, we collect several network wide measurements that contain, for example, the hardware states of the WiFi card, link statistics and packet traces, and store everything on a central server. Using these measurements, we show that both active and passive cross-layer measurements are valuable in measuring lost transmission opportunities, and that this cannot be done purely on just a single layer. From passive measurements, we show that a significant overhead of IEEE 802.11 control and management traffic exist, reducing the wireless network capacity. In addition, we show that, with active measurements by using the link impairment estimator from [17] a close estimation of the root-cause of loss due to link impairments is possible.
## Contents

1 Introduction 2

2 Background 4
   2.1 Link Impairments in IEEE 802.11 Networks 4
   2.2 Sending a Frame in IEEE 802.11 6
      2.2.1 The Structure of an IEEE 802.11 Frame 6
      2.2.2 Frame Airtime Calculation Based On IEEE 802.11 PHY 9
      2.2.3 Transmission of a Frame at IEEE 802.11 MAC 12

3 Related Work 16
   3.1 Quantifying Wireless Link Quality 16
   3.2 Measuring Interference 17
   3.3 Identifying Lost Transmission Opportunities 18

4 Design and Implementation 20
   4.1 Description of the General Measurement Environment 20
      4.1.1 Tools and the Configuration Environment 20
      4.1.2 Storage 21
      4.1.3 Synchronization of Measurement Nodes 22
      4.1.4 Testbed 23
   4.2 Passive Measurements of Lost Transmission Opportunities 23
      4.2.1 Identifying IEEE 802.11 Traffic 23
      4.2.2 Calculation of the Airtime of IEEE 802.11 Traffic from Measurements 24
   4.3 Active Measurements of Lost Transmission Opportunities 24
      4.3.1 LIME in a Nutshell 25
      4.3.2 Validation of LIME 27

5 Network Measurements 30
   5.1 Identifying IEEE 802.11 traffic overhead 30
   5.2 Active measurement of lost transmission opportunities 37

6 Conclusion and Future Work 43

A Appendix 1 44

B Appendix 2 45

C Appendix 3 46
1 Introduction

In this thesis, we present a measurement study on the quantification of lost transmission opportunities in IEEE 802.11 networks, which is an increasingly more important problem due to a higher number of IEEE 802.11-based deployments. Lost transmission opportunities are typically caused by (1) unwanted traffic, which may reduce the available link capacity due to management and control traffic overhead or (2) due to link impairments. Importantly, link impairments can either lead to packet loss or unfairness situations between stations and hence, the performance degrades due to a reduction of achievable throughput or increasing delays, if stations experience loss or are prevented from sending. On the other hand, unwanted traffic results in reduction of available airtime and hence, in a reduction of available bandwidth.

Research on measuring lost transmission opportunities in IEEE 802.11 (WiFi) deployments is still in its infancy. The majority of the literature is actually on the quantification of the usage and operational state of the network using physical layer and link layer measurements. However, in this area, most research has been done on quantifying the impact of wireless interference and using this information for improving channel selection, transmission power control or even changing the physical location of nodes. In doing so, most literature on measuring interference assumes a flat network to create an interference map at a central point. Very little research has been done on developing approaches that help differentiate between root causes of lost transmission opportunities in real networks (e.g., networks with access point-client topology), which typically result from link impairments or the presence of unwanted traffic.

To address this issue, we perform (1) passive measurements to gain information about unwanted traffic which lead to a reduction of available airtime and (2) active measurements to gain information on link impairments which lead to packet loss or unfairness. In both cases, we follow a cross-layer approach, since measuring lost transmission opportunities depends on information from two layers: (1) the physical (PHY) layer which provides signal quality and transmission statistics and (2) the medium access-control (MAC) layer which provides IEEE 802.11 protocol specific information, and throughput and loss statistics. We examine unwanted traffic to understand the impact on the environment and to support the development of mitigating mechanisms. To measure link impairments, we adapt an active measurement approach \cite{17}, which is, to the very best of our knowledge, the only approach that is able to estimate the reasons behind packet loss. We present our methodology on how to collect network-wide measurements with off-the-shelf WiFi hardware. All measurements are performed in a typical WiFi office environment located on the 16th floor in the TEL building of the Technische Universität Berlin. We have deployed a five node test-bed exclusively for the purpose of this thesis, due to a hardware issue with the existing networks. Using our measurement framework, we collect several network wide measurements that contain for example, the hardware states of the WiFi card, link statistics and packet traces, and store everything on a central server. Using these measurements, we show that both active and passive cross-layer measurements are necessary to measure the majority of lost transmission opportunities, and that this cannot be done purely on the link (MAC) or networking (IP) layer.

We can consider several practical applications of the approaches and measurement environment which we present in this thesis. Today’s rate control algorithms lack the ability to identify the root-cause of different link impairments, which in turn result in counter-intuitive decisions. For instance, the rate is reduced in the presence of loss due to collisions or hidden terminals, which would aggravate the problem rather than improve it. For instance, the approach can support rate adaption algorithms with additional
information about link impairments, which are used to predict the quality of a link under different conditions. Another application area is channel selection, where the goal is to select the channel with the least lost transmission opportunities. Finally, our measurement framework can aid network operators to understand the performance of their network and use as an input to network configuration management.

This thesis is structured as follows. We first discuss in Chapter 2 the link impairments and unfairness issues in standard IEEE 802.11 (WiFi) networks and provide the necessary detail about sending a frame based on the IEEE 802.11 standard to understand the proposed passive and active measurement approaches. In Chapter 3 we discuss the related work on characterizing wireless links and approaches to identify lost transmission opportunities. We present the measurement methodology in Chapter 4 which is used to perform the active and passive measurements. Chapter 5 presents our measurement results and in Chapter 6 we conclude and present future work.
2 Background

The goal of this thesis is to identify the measurement methodology to characterize the link performance under different link impairments. Therefore, in this chapter, we first discuss the link impairments and unfairness issues in standard IEEE 802.11 (WiFi) networks. Next, to quantify the lost transmission opportunities, we explain how a frame is sent in a WiFi network. The background information presented in this chapter provides the necessary detail to understand the proposed passive and active measurement approaches introduced in Chapter 4.

2.1 Link Impairments in IEEE 802.11 Networks

IEEE 802.11 networks are composed of fundamental building blocks called basic service set (BSS). There are two types of BSSs: (1) an infrastructure BSS (BSS) (see Figure 1 (a)) and (2) an independent BSS (IBSS) (see Figure 1 (b)). Both types of networks use an identifier that is at most 32 characters long (service set identifier (SSID)), making it possible to identify different networks in a given environment.

**Infrastructure BSSs** consists of an access point (AP) and all its associated stations (STA). The AP is the central point of the network and acts as a central controller of all its STAs (e.g., provides synchronization, authentication). Each BSS has a unique identifier (BSSID) which is the MAC address of the radio interface of the AP. APs periodically broadcast Beacons that contain the SSID, WiFi settings such as the used encryption method, and optional vendor-specific features. STAs use these announcements for several purposes such as BSS detection, as a signal strength indicator, synchronization or as a “keepalive”. Multiple interconnected BSSs form an extended BSS with a common extended service set identifier (ESSID). This is usually the case for large production networks like enterprise or campus networks.

**Independent BSSs** are the foundation for WiFi “Mesh” networks. In this network type, STAs form a independent network in the absence of a central controller. Hence, each station acts as an AP and sends Beacons to perform neighbour discovery.

In the recent years, a high number of IEEE 802.11 networks of both types were deployed. Due to the increasing density of networks and nodes, a number of link impairments occur that degrade the communication performance. Our goal in this thesis is to identify

---

In case the SSID is configured as hidden, it is omitted from the Beacons.
the differences between such impairments. This differentiation enables the design of specific optimizations under different conditions. To this end, we focus on five types of most commonly observed link impairments:

- **Noise errors**: Noise error refer to frame corruption due to a too low signal-to-noise ratio (SNR). In this thesis, we define noise as the white noise as well as random signals caused by non-IEEE 802.11 stations (e.g. micro waves, radar). A low SNR can lead to frame corruption and can be detected by checksum errors.

- **Hidden Terminals**: Hidden terminal (aka. hidden station or hidden node) problem occurs when STAs cannot sense each others’ transmissions (see Section 2.2.3 for more information on carrier sensing). For instance, in Figure 2 STA A and STA B are hidden from each other. When both STA A and STA B transmit concurrently to their respective APs (i.e, AP A and AP B), due to the broadcast nature of the wireless medium, this will lead to the corruption of STA B’s frames at AP B (i.e., STA A’s frames to AP A and STA B’s frame to AP B will collide at AP B). Hidden terminal effect can be observed more severely in the case of longer transmissions compared to shorter transmissions, as the probability that two transmissions coincide increases as the frame length increases. Therefore, frame length is expected to play an important role in identifying hidden terminals.

- **Collisions**: Collisions happen when two or more stations simultaneously access the medium at the same time. This is different than hidden terminal problem as, even though nodes are able to carrier sense each other, based on the random medium access protocol (see Section 2.2.3) their transmissions coincide. The famous Bianchi model [6] computes the probability of a packet transmission failure due to collisions assuming an ideal channel (e.g., there are no hidden terminals). Based on this model, the loss probability due to collisions increases as the number of active stations in the network and the traffic load increase. For instance, in an IEEE 802.11b network, the collision probability with four active stations is more than 16%, with 20 stations more than 32% and with 40 stations more than 40%. Hence, with increasing node density and traffic load, the probability that link impairments are caused by collisions increase.

- **Exposed Nodes**: The exposed node problem does not lead to frame loss but refers to a situation where a transmission could take place but the node is prevented from sending due to sensing transmissions from another transmitter. This leads eventually to a reduction in achievable throughput. Figure 3 illustrates the problem, where the stations B, C and D are exposed. Essentially, even though they can concurrently transmit without affecting the reception at their respective APs (e.g., AP A and AP B), they defer from sending unnecessarily since they sense each other’s transmissions. In the case that all stations have always data to send, this situation causes unfairness at STA B; that is STA B shares the medium with STAs A, C and D, whereas STA A only competes with STA B. Hence, STA A senses the medium idle more often than STA B and finds more chance to transmit.
Figure 3: Example of an exposed node problem. Here, STA B, STA C and STA D are exposed nodes. Even though they can transmit concurrently without causing any problems, they defer from sending as they sense each others’ transmissions.

Figure 4: Example of a capture effect problem. Here, the AP always “captures” the frames from STA A as it receives the frames from STA A with a higher signal-to-noise ratio, when the transmissions of STA A and STA B coincide. This does not lead to a packet loss but may cause unfairness.

- **Capture Effect**: The capture effect is another impairment which does not lead to a frame loss, but might lead to an unfairness situation. Capture effect happens if two transmitters transmit at the same time and the receiver captures only the frames with the higher signal. For instance, in Figure 4, two stations A and B are physically located in such a way that transmissions by STA A are always received with a higher signal-to-noise ratio by the receiver AP. Thus, if their transmissions collide, A’s frames are more likely to be received compared to B, causing an unfairness situation.

### 2.2 Sending a Frame in IEEE 802.11

To identify link impairments, we need to understand how a frame is sent in IEEE 802.11, which is the topic of this section. To this end, we first explain the structure of an IEEE 802.11 frame. Next, we explain how to calculate the duration of a frame which is the time of how long the medium would be busy for a transmission, the so-called airtime. A high duration might have a huge impact on the link impairments (e.g., the loss probability due to hidden terminals may increase). Therefore, we present the relevant parts of IEEE 802.11 PHY and MAC layers that impact the duration and transmission of a frame.

#### 2.2.1 The Structure of an IEEE 802.11 Frame

The IEEE 802.11 standard body has published several amendments over the years (see Table 1), which use different PHY technologies. The original IEEE 802.11b standard primarily adopts DSSS, which is a single-carrier technology. On the other hand, IEEE 802.11ag uses OFDM, which uses 52 subcarriers. From a hardware implementation point of view, OFDM makes it easier to address multipath fading. Generally, the IEEE 802.11 PHY specification splits the PHY layer into two sub-layers: (1) PHY Layer Convergence
Table 1: List of IEEE 802.11 Amendments. FHSS: Frequency-hopping spread spectrum, DSSS: Direct Sequence Spread Spectrum, ER-DSSS: Extended Rate-DSSS, OFDM: Orthogonal Frequency-Division Multiplexing, ER-OFDM: Extended Rate-OFDM

<table>
<thead>
<tr>
<th>Standard</th>
<th>Frequency</th>
<th>Technology</th>
<th>Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11 (1997)</td>
<td>2.4 GHz</td>
<td>FHSS/DSSS</td>
<td>1-2 Mbps</td>
</tr>
<tr>
<td>802.11b (1999)</td>
<td>2.4 GHz</td>
<td>ER-DSSS</td>
<td>1-11 Mbps</td>
</tr>
<tr>
<td>802.11a (1999)</td>
<td>5 GHz</td>
<td>OFDM</td>
<td>6-54 Mbps</td>
</tr>
<tr>
<td>802.11g (2003)</td>
<td>2.4 GHz</td>
<td>ER-OFDM</td>
<td>1-54 Mbps</td>
</tr>
<tr>
<td>802.11 (2007)</td>
<td>2.4,5 GHz</td>
<td>ER-OFDM/OFDM</td>
<td>1-54 Mbps</td>
</tr>
<tr>
<td>802.11n (2009)</td>
<td>2.4,5 GHz</td>
<td>OFDM</td>
<td>1-600 Mbps</td>
</tr>
</tbody>
</table>

*ISM (Industrial Scientific Medical) band is a frequency band originally for industrial, scientific and medical purposes (e.g. microwave ovens) but today also used for wireless date communications (e.g. 802.11, Bluetooth), cordless phones (e.g. DECT phones) etc. The band is free of charge and need no prior registration at the regulation authorities. However the emitted power in those bands is limited.

Procedures (PLCP) which are used to aid the receiver in demodulation and reception of the actual MAC layer payload and (2) Physical Medium Dependent (PMD) sub-layer, which interacts directly with the physical medium and converts data in a suitable format for the PLCP. All packets in IEEE 802.11 contain a physical layer convergence procedure (PLCP) preamble and header used to synchronize the receiver to transmissions. The Physical Protocol Data Unit (PPDU) is formed during data transmissions by appending the Physical layer Service Data Unit (PSDU) to the PLCP preamble and header. On the receiver side, the PLCP preamble and header are used for demodulation and delivery of the PSDU which is the actual MAC layer frame or another PLCP header in backwards compatibility mode.

Figures 5(a) and 5(b) show the PPDU for IEEE 802.11b with DSSS technology and PPDU for IEEE 802.11ag for OFDM technology. As seen from the figures, each technology uses different PLCP preamble and headers, which affect the airtime of a frame. The (DSSS-based) IEEE 802.11b defines a mandatory long PLCP with a duration of 192 µs shown in Figure 5(a) and an optional short PLCP with a duration of 96 µs shown in Figure 5(b). For the long PLCP, the preamble and the header are sent at 1 Mb/s. The short PLCP consists of less training symbols and a header which is sent at a higher rate of 2 Mb/s. In contrast, Figure 6 shows the OFDM PPDU format. In OFDM the PLCP preamble
Figure 6: OFDM PPDU format
(Source: IEEE 802.11 Std. [3] (Fig. 17-1, Page 595))

Figure 7: Frame format of an IEEE 802.11 frame formats. Starting with the frame control (FC) field, the data frame format (MPDU) including the MSDU, the management frame format (MMPDU) and the ACK frame format.

consists of 10 short training symbols of 0.8 μs duration and two long training symbols a 4 μs duration followed by a 1 symbol long PLCP header of 4 μs duration. The PLCP header and preamble are sent at the lowest rate of 6 Mb/s.

The complete airtime of a frame is encoded in MAC layer Protocol Data Unit (MPDU), which contains a MAC header, the body of the frame and a Frame Check Sequence (FCS). Figure 7 shows the abstract IEEE 802.11 MAC layer frame structure common to all amendments. In IEEE 802.11, there are three main frame types: (1) “Data frames” are used for data transmissions; (2) “Control frames” are used to control the access to the shared medium and for reliability (e.g., RTS (Request-to-Send), CTS (Clear-to-Send) and ACK (acknowledgment)) and (3) “Management frames” are used for management-specific functions such as maintaining connections. The type of a frame is defined through the type and subtype field in the frame control (FC) field. The MAC headers for data, management and ACK frames are shown in Figure 7. In this figure, the duration/ID field corresponds to the time to complete a transmission (represented in microseconds). The duration/ID
field is a 16 bit long field. In case of regular unicast transmissions which require ACKs, the duration field of a data frame covers until the completion of the corresponding ACK. On the other hand, the duration field in the ACK frame contains the remaining duration.

Fragmentation is the process of partitioning a MAC service data unit (MSDU) into multiple smaller MPDUs (illustrated in Figure 8). In the process of fragmentation only MPDUs with a unicast receiver address are fragmented whereas broadcast or multicast frames are never fragmented even if their length exceeds the fragmentation threshold. All fragments of an original MPDU have the same unique sequence number and all fragments except the last fragment have the “more fragment” bit set in the FC (see Figure 7). Each fragment is ACKed separately. Fragments can be sent in a burst, as illustrated in Figure 9. During a burst, a station gains exclusive access to the medium, since frames are inter-spaced by a SIFS. Hence, all subsequent frames in a burst are protected from collisions. We make use of this property in our active measurements.

2.2.2 Frame Airtime Calculation Based On IEEE 802.11 PHY

In this section, we present the formulas for calculating the “airtime” of a frame based on different PHY technologies following the terminology introduced in the previous section. These formulas are used in our measurement tools to calculate the airtime of the frames. This is necessary, as even though theoretically, MAC layer frames contain a duration field, this information is not always filled correctly by the Madwifi driver that we use in the testbed.

The crucial parts which affect the “airtime” of a frame are the modulation techniques, channel coding from the PMD and the different PLCPs. Channel coding and the modulation techniques depend on the used standard and the selected rate for transmission (see Table 1). According to the differences of DSSS (used by IEEE 802.11b) and OFDM (used by IEEE 802.11ag), the final bit-rate for transmissions is calculated differently. More specifically, each bit-rate uses forward error correction (FEC) with a code rate \( R = k/n \),
Table 2: Rate table for 802.11b with 20 MHz channel width.

<table>
<thead>
<tr>
<th>Rate</th>
<th>Modulation</th>
<th>R</th>
<th>nBPC</th>
<th>nSP</th>
<th>nSPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BPSK</td>
<td>1/11</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>QPSK</td>
<td>1/11</td>
<td>2</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>CCK</td>
<td>4/8</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CCK</td>
<td>4/8</td>
<td>2</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Rate table for 802.11ag with 20 MHz channel width.

<table>
<thead>
<tr>
<th>Rate</th>
<th>Modulation</th>
<th>R</th>
<th>nBPC</th>
<th>nSD</th>
<th>nSP</th>
<th>nCBPS</th>
<th>nDBPS</th>
<th>nSPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>BPSK</td>
<td>1/2</td>
<td>1</td>
<td>48</td>
<td>4</td>
<td>48</td>
<td>24</td>
<td>250K</td>
</tr>
<tr>
<td>9</td>
<td>BPSK</td>
<td>3/4</td>
<td>1</td>
<td>48</td>
<td>4</td>
<td>48</td>
<td>36</td>
<td>250K</td>
</tr>
<tr>
<td>12</td>
<td>QPSK</td>
<td>1/2</td>
<td>2</td>
<td>48</td>
<td>4</td>
<td>96</td>
<td>48</td>
<td>250K</td>
</tr>
<tr>
<td>18</td>
<td>QPSK</td>
<td>3/4</td>
<td>2</td>
<td>48</td>
<td>4</td>
<td>96</td>
<td>72</td>
<td>250K</td>
</tr>
<tr>
<td>24</td>
<td>16-QAM</td>
<td>1/2</td>
<td>4</td>
<td>48</td>
<td>4</td>
<td>192</td>
<td>92</td>
<td>250K</td>
</tr>
<tr>
<td>36</td>
<td>16-QAM</td>
<td>3/4</td>
<td>4</td>
<td>48</td>
<td>4</td>
<td>192</td>
<td>144</td>
<td>250K</td>
</tr>
<tr>
<td>48</td>
<td>64-QAM</td>
<td>2/3</td>
<td>6</td>
<td>48</td>
<td>4</td>
<td>288</td>
<td>192</td>
<td>250K</td>
</tr>
<tr>
<td>54</td>
<td>64-QAM</td>
<td>3/4</td>
<td>6</td>
<td>48</td>
<td>4</td>
<td>288</td>
<td>216</td>
<td>250K</td>
</tr>
</tbody>
</table>

where information is divided into blocks of length $k$ and $r$ parity bits are added to each block so that $n = r + k$ bits are transmitted.

Table 2 shows a summary of the coding rates, modulations, coded bits per signal carrier and symbols sent per second for DSSS. For OFDM, this information is presented in Table 3. The bit-rates of 802.11b are calculated multiplying the coding rate ($R$), the number of coded bits per signal carrier ($nBPC$) and the number of complex data numbers per symbol ($nSPS$):

$$Rate_{DSSS} = nBPC \times R \times nSPS.$$ (1)

For OFDM the bit-rate is calculated slightly differently since it allows the transmission of more data bits per symbol. The bit-rate is calculated by multiplying the number of data bits per OFDM symbol ($nDBPS$) with the number of symbols per second ($nSPS$):

$$Rate_{OFDM} = nDBPS \times nSPS,$$ (2)

where $nDPS = R \times nBPC \times nSD$ and, $nSD$ is the number of complex data numbers per ODFM symbol and $nSP$ is the number of pilot values per OFDM symbol.

In the following, we list the formulas for airtime calculation which are affected by the aforementioned rate selections. Specifically, we focus on for IEEE 802.11b, IEEE 802.11g when there are also IEEE 802.11b nodes the network (compatibility mode), pure IEEE 802.11g and IEEE 802.11a. In all the equations below, Ceiling is a function that returns the smallest integer value greater than or equal to its argument value.

**IEEE 802.11b HR-DSSS**

In pure 802.11b networks the duration of the PLCP depends on whether the operator has chosen to use short preamble or long preamble. Therefore, the airtime of an IEEE 802.11 frame, $TXTIME_{HR-DSSS}$, is:
\[ \text{TXTIME}_{HR-DSSS} = \]
\[ \text{PreambleLength}_{DSSS} + \text{PLCPHeaderTime}_{DSSS} \]
\[ + \text{Ceiling}(\frac{(LENGTH \times 8)}{\text{Rate}_{DSSS}}), \]  

(3)

\[ \text{PreambleLength}_{DSSS} \]
\[ 144 \, \mu s \text{ for a “long preamble”} \]
\[ \text{or } 72 \, \mu s \text{ for a “short preamble”} \]

\[ \text{PLCPHeaderTime}_{DSSS} \]
\[ 48 \, \mu s \text{ for a “long preamble”} \]
\[ \text{or } 24 \, \mu s \text{ for a “short preamble”} \]

\[ \text{LENGTH} \]
\[ \text{in units of octets} \]

\[ \text{DATARATE} \]
\[ \text{in units of Mb/s} \]

**IEEE 802.11bg (IEEE 802.11b compatibility mode)-DSSS-OFDM**

If an operator sets its network in “compatibility mode”, the duration of frames also change. In other words, the duration of the PLCP changes in 802.11g networks, if 802.11b stations are present. In this case, the airtime of an IEEE 802.11bg frame, \( \text{TXTIME}_{DSSS-OFDM} \), becomes:

\[ \text{TXTIME}_{DSSS-OFDM} = \]
\[ \text{PreambleLength}_{DSSS} + \text{PLCPHeaderTime}_{DSSS} \]
\[ + \text{PreambleLength}_{OFDM} + \text{PLCPSignal}_{OFDM} \]
\[ + \text{Ceiling}(\frac{(16 + 8 \times LENGTH + 6)}{\text{Rate}_{OFDM}}) \]
\[ + \text{SignalExtension}, \]  

(4)

\[ \text{PreambleLength}_{OFDM} \]
\[ 8 \, \mu s \]

\[ \text{PLCPSignal}_{OFDM} \]
\[ 4 \, \mu s \]

\[ \text{PLCPServiceBits} \]
\[ 16 \text{ bits} \]

\[ \text{NumberOfOctets} \]
\[ \text{number of data octets in the PSDU} \]

\[ \text{PadBits} \]
\[ 6 \text{ bits} \]

\[ \text{SignalExtension} \]
\[ 6 \, \mu s \text{ (due to extending } 10\mu s \text{ slot length of DSSS to } 16 \text{ in OFDM. )} \]

**IEEE 802.11a - OFDM**

The airtime of an IEEE 802.11a frame, \( \text{TXTIME}_{OFDM} \) is:

\[ \text{TXTIME}_{OFDM} = \]
\[ T_{PREAMBLE} + T_{SIGNAL} \]
\[ + \text{Ceiling}(\frac{(16 + 8 \times LENGTH + 6)}{\text{Rate}_{OFDM}}), \]  

(5)

\[ T_{PREAMBLE} \]
\[ \text{PLCP preamble duration } (16 \, \mu s) \]

\[ T_{SIGNAL} \]
\[ \text{Duration of the SIGNAL BPSK-OFDM symbol } (4 \, \mu s) \]

11
IEEE 802.11g - ERP-OFDM

The pure IEEE 802.11g frame airtime (i.e., not in compatibility mode) is:

\[ TXTIME_{ERP-OFDM} = TXTIME_{OFDM} + \text{SignalExtension}, \]

2.2.3 Transmission of a Frame at IEEE 802.11 MAC

Until this section, we have presented the WiFi frame structure and the IEEE PHY properties that affect the airtime of a frame. In this section, we will explain the relevant parts from the MAC layer that defines (1) the necessary conditions to be able to transmit a frame (basic medium access) and (2) the time difference between two consecutive frames (the inter-frame spacing (IFS) and the exponential back-off algorithm).

Basic Medium Access  Contention-based basic medium access in IEEE 802.11 is achieved by a distributed coordination function (DCF) based on the carrier sense multiple access with collision avoidance (CSMA/CA) scheme. The CSMA protocol works as follows: If a station carrier senses the medium busy, it defers its transmission until the medium is idle. Only if the medium is sensed idle, the station is allowed to transmit.

Carrier sensing (CS) is done in two ways to determine the idle or busy state of the carrier (1) by comparing the current energy level of the wireless medium to a defined threshold and (2) through a virtual mechanism called Network Allocation Vector (NAV).

In particular, the NAV is used in the four-way handshake of a transmission. During the handshake, a station first sends a request to send (RTS) and if possible, the receiver responds with a clear-to-send (CTS). In this case, the CTS contains the requested duration to complete the frame transmission. Then, the DATA and ACK transmissions can proceed. Note that using RTS/CTS, a station can obtain exclusive access to the medium as neighbor nodes perform virtual carrier sensing by setting their NAVs and stay silent during this
period (see Figure 10). More specifically, the NAV is set whenever the STA receives a valid frame where the receiver address (RA) is not its own address and the frame contains a non-zero duration/id field (see Section 2.2.1 for frame structure). Before a station sets the NAV, the time to receive the frame is subtracted from the value of the duration/ID field. While RTS/CTS messages protect frames against collisions and hidden terminals (see Section 2.1), they incur a high overhead of control frames and a reduction of available airtime. Furthermore, airtime might be wasted due to increasing exposed terminal situations.

IEEE 802.11 defines a different medium access method for QoS networks (IEEE 802.11e), which is called enhanced distributed channel access (EDCA). EDCA provides differentiated, distributed access to the wireless medium using different Access Categories (ACs). EDCA is part of a hybrid coordination function (HCF) and should be implemented by all QoS-enabled STAs. All WiFi-certified IEEE 802.11 compatible devices produced from 2005 on are QoS-enabled due to the WiFi Alliance.

**Interframe Space (IFS)** IEEE 802.11 defines several different interframe spacing (IFS), which is the time gap between consecutive frame transmissions. In other words, the time is slotted and has well defined boundaries at which transmissions can occur due to interframe spacings. Essentially, IFS provides different priorities for different types of frame transmissions and plays an important role in our measurement approaches to differentiate different link impairments. The following lists the types of IFS, which are fixed for each PHY technology.

- **Short Interframe Space (SIFS)** is the minimum time a station has to defer its transmission before sending control frames or sending the remaining frames of a burst. All frames sent after a SIFS interval are protected from collisions. The SIFS for OFDM is 16 $\mu$s and for DSSS is 10 $\mu$s. A duration of a SIFS must stay within the bounds of one SIFS $\pm$ 10%.

- **PCF interframe space (PIFS)** is an interframe space used only for contention-free access (Point Coordination Function (PCF)), which is not supported by current drivers. It is mentioned here for completeness.

- **DCF interframe space (DIFS)** is used by the DCF in the exponential backoff algorithm (explained in the next section). The DIFS is a SIFS plus twice a SlotTime: $DIFS = SIFS + 2 \times SlotTime$, where $SlotTime$ is the duration of one time slot specific to each PHY.

- **Arbitration interframe space (AIFS)** was introduced by the IEEE 802.11e. The AIFS helps a station to gain different medium access opportunities depending on different access categories (ACs). $AIFS[AC] = SIFS + AIFSN[AC] \times SlotTime$.

- **Extended interframe space (EIFS)** is the longest interframe space. It includes a DIFS and a SIFS and is the time a station defers its transmission if it senses the medium idle following a reception of an erroneous frame. Hence, the EIFS gives the other station enough time to acknowledge a frame.

The relationship between the interframe spacings is shown in Figure 11.
Figure 11: The relationship between different IFS settings. Each node waits at least a DIFS/AIFS interval to transmit. If the medium is busy, nodes defer access. Based on the exponential backoff algorithm, nodes back off from sending and decrement their back-off counters as long as they sense the medium idle at each slot time.

(Source: IEEE 802.11 Std. [3] (Fig. 9-3, Page 258))

Figure 12: The STAs have simultaneously decremented their backoff counter and start transmitting and both transmissions collide at the receiver. A STA can determine the unsuccessful transmission through the missing ACK.
**Exponential Backoff Algorithm** IEEE 802.11 uses an exponential backoff procedure to mediate access to the medium when collisions occur. An unsuccessful transmission due to a collision is illustrated in Figure 12, where two stations have simultaneous decremented their CW and transmit at the same time. The initial backoff value is randomly chosen from a congestion window (CW) and multiplied with the slot time. The backoff counter is decremented after the DIFS as long as the medium is sensed idle during each slot time. In case the medium is busy, the station defers decrementing its back-off counter until it senses the medium idle again. Hence, before a station is able to transmit, the medium must have been idle for at least a DIFS and the back-off counter must have reached zero (see Figure 11).

If unsuccessful transmission occurs, the CW is exponentially increased until it reaches the maximum CW ($CW_{\text{max}}$) value of 255. Unsuccessful unicast transmissions can be determined by the lack of an immediate positive acknowledgment (ACK) (retransmissions are scheduled after an ACK timeout). Broadcast transmissions are never ACKed, hence the backoff counter is always at the initial value. The standard DCF defines a minimum CW ($CW_{\text{min}}$) of 7. For QoS enabled stations the $CW_{\text{min}}$ and $CW_{\text{max}}$ values depend on the access category allowing to give different priorities of medium access to different traffic classes.
3 Related Work

This chapter discusses related work on characterizing wireless links. We observe that today’s studies are more often carried out in testbeds with off-the-shelf WiFi hardware, hence, we focus primarily on measurement studies in such testbeds. We believe this trend will continue as larger measurement frameworks [10, 39, 27, 43] that facilitate measurement collection, merging, and analysis are adopted by the research community.

The first attempts to characterize wireless links focused on creating metrics to quantify link quality, which are summarized in Section 3.1. The main goal was to be able to represent wireless links better in upper-layer protocols such as routing. While these metrics are able to capture link diversity in a network to a certain extent, they do not provide any insight about the reasons why a certain link quality is achieved. To this end, the community turned its attention to interference measurements as interference is seen as the major reason behind degradation of wireless link quality. We discuss the related work on interference measurements in Section 3.2. Finally, the most recent approaches take a deeper look and try to differentiate conditions for lost transmission opportunities, typically resulting from unwanted traffic and link impairments. There is relatively less work in this area, and we give a brief summary of the current state-of-the-art in Section 3.3.

We classify the measurement approaches used in the related work using the following taxonomy:

- Measurement approach: (1) passive, (2) active, and (3) hybrid
- Information source: (1) PHY layer - SNR and bit error rate (BER), (2) MAC layer that rely on throughput and delay statistics, and (3) cross-layer (MAC/PHY)

It must be noted that majority of the work we present assumes a flat network topology of APs (and are applicable to only a wireless backbone). In this thesis, we consider more general networks, for example, an enterprise network with several access points and STAs, which may or may not be managed by a single operator.

3.1 Quantifying Wireless Link Quality

Typically, wireless link quality is quantified in two ways: MAC layer estimations based on packet loss rates and PHY layer link quality measured in terms of SNR and bit error rate (BER). MAC layer based approaches are typically used to construct link-quality based routing metrics or used in rate control algorithms. For example, the most widely-known routing metrics are ETX (Expected Transmission Count) [13] and ETT (Expected Transmission Time) [14], which attempt to measure the cost of transmitting over a given link in terms of the number of retransmissions and the time it takes to transmit a packet, respectively. These algorithms measure link quality through the exchange of broadcast messages periodically and multiply the percentage of successful measurements in each direction to determine the link cost. However, such measurements are decoupled from unicast data traffic, and hence, may not be representative of the link quality during data communication [12].

In a multi-rate IEEE 802.11 network, rate adaptation is an important technique to select the best rate for transmission under current channel characteristics. Indeed, the selection of a poor transmission rate leads to lost transmission opportunities. In literature, the rate adaption algorithms use either MAC or PHY layer measurements to obtain packet loss information and eventually select the best transmission rate. In MAC-layer approaches, lost frames are detected by lack of MAC-layer ACKs, and hence, these approaches operate only on bidirectional links. An earlier rate adaptation algorithm, ARF
(Auto-Rate Fallback) [21], use passive measurements of a fixed number of frames to estimate the channel quality, and decrease the rate in case of consecutive frame losses (e.g., 4 lost frames in a row). Nowadays, active sampling approaches are widely used in practice. These approaches continuously sample multiple rates to estimate the best rate based on packet loss (SampleRate [7] or Minstrel [41]). The limitation of these algorithms is their low effectiveness in the presence of fast channel fading.

To represent the channel conditions better, another type of rate control algorithms rely on PHY layer information. For example, the error probability at a chosen rate is mapped to the signal-to-noise ratio (SNR). However, in practice, wireless network interface cards provide only a received-signal-strength indicator (RSSI), which is computed from the received signal strength over a calibrated but fixed noise floor. Hence, the samples of the RSSI may vary over packet reception, be mis-calibrated or be corrupted by interference [7, 34]. To alleviate this problem, Acharya et al. present WOOF [1], which performs passive PHY layer measurements of the medium busy and idle time. The algorithm outperforms the MAC layer based SampleRate approach and shows a throughput improvement of 300 percent and it is able to differentiate congestion-related packet loss and packet loss due to link quality degradation. This is very important as transmission rate should be adapted only in case of link degradation and not congestion. In [18, 33], it was shown that selecting rates based on a pure PHY layer approach through SNR measurements per frequency (or channel state information provided by IEEE 802.11n-based cards) is comparable to the MAC layer approaches based on packet loss probability and outperforms RSSI approaches. Nevertheless, pure PHY layer approaches lack the ability to estimate the cause of link impairments e.g., the presence of hidden terminals [5].

The authors of [15] argue that link quality should be estimated passively using feedback from multiple layers: PHY, MAC and network. Using 4-bits (1 from PHY, 1 from MAC and 2 from network), the authors show that it is possible to identify which links are valuable to higher layers. Similarly, EAR [22] proposes a cross-layer approach, which combines active and passive online measurements and represent link quality as a smoothed delivery ratio over time. Both works focus on the use of link quality information for routing.

3.2 Measuring Interference

There exist several works that specifically focus on measuring interference, as it can severely degrade performance. In these works, interference is typically classified as carrier sense (sending) and hidden terminal (receiving) interference.

The first type of approaches are active network monitoring approaches that work either on PHY or MAC layers and require a downtime of the network as they perform measurements in the absence of real traffic. In [30] and in [29], an interference map is computed using sender, receiver, single interferer measurements. More specifically both approaches populate a delivery ratio matrix, which contains for each sender, receiver, single interferer triplet, the delivery ratio in the presence of the interference normalized with the delivery ratio without interference. Both approaches measure only “good” links (i.e., links with an ETX value higher than a threshold). In [29], the interference map is extended to include carrier sense and hidden terminal relationships. In contrast, the authors of [11] investigate the interference resulting not only from a single sender but from multiple interferers and show that even though this is not widespread, it does have an impact on the link quality, when it occurs. Lee et al. [25] propose a RSS-based interference prediction that takes into account carrier sensing and physical layer capture. Compared to [30], this approach allows understanding the conditions, in which one of the two interfering links has an unfair advantage over the other (e.g., due to capture).
The second type of approaches are online and hence, do not require a network down time. For example, one example is PIE (Passive Interference Estimation) framework [39], which uses an online passive monitoring approach. Given an enterprise network, PIE assumes all APs monitor the network in a time-synchronized fashion and send their measurements to a centralized controller. It must be noted this approach might not extend to a multi-hop wireless network as timing requirements cannot be guaranteed. Furthermore, PIE cannot deal with external interference caused by non-WiFi or hidden WiFi sources. Finally, in [8], an active online measurement approach is used to characterize interference assuming it follows a probability distribution function (e.g., pulse interference as in the case of microwave oven). It is shown that, by varying the packet transmission rate, which changes the duration of a packets, and monitoring the corresponding MAC layer loss rates, the timing of interference pulses can be fully estimated.

3.3 Identifying Lost Transmission Opportunities

In this thesis, we do not directly measure interference but the effects of interference that lead to lost transmission opportunities. We will focus on two types of lost transmission opportunities: (1) arising from PHY and MAC layer issues (e.g., protocol overhead) and (2) due to link impairments as described in Section 2.1.

Unwanted traffic: In WiFi networks, the presence of unnecessary or unwanted traffic can have a significant impact on performance. Unwanted traffic can be categorized as malicious traffic, and traffic specific to the IEEE 802.11 protocol and in this thesis, we focus on the 2nd category as unwanted traffic. Several studies use passive MAC layer approaches to monitor the unwanted traffic. For example, in [32], IEEE 802.11 management traffic was identified to lead to inefficient medium utilization and lost transmission opportunities using the data collected at the 67th Internet Engineering Task Force (IETF) meeting. A high percentage (40%) of the frames was shown to be management traffic: Specifically, keep-alive traffic, which is used to maintain the wireless link in the absence of data traffic and, the probing mechanism to gain information about the neighbourhood are identified as the two main sources. Furthermore, the authors show that a high amount of unwanted traffic is created by the mechanisms that initiate, maintain, and update client-AP connectivity (e.g., incorrectly initiated hand-offs under high load conditions). Their results also show that 15% of the clients use RTS/CTS handshake, which is expected to play a minor role. However, the results from [19] contrasts these findings and show that the RTS/CTS mechanism can create up to 37% and 27% overhead on throughput for IEEE 802.11b and IEEE 802.11g, respectively, in heterogeneous MAC environments [20]. In Chapter 5, we also show that CTS-based overhead does exist and might lead to lost transmission opportunities. In [37], using the measurements from the SIGCOMM 2004 conference, it was discovered that only 40% of the transmissions corresponded to data traffic, whereas 35% of the airtime was consumed by retransmissions, 15% by acknowledgements and 10% by management traffic. The high number of retransmissions indicates a high number of link impairments, which we discuss next.

Link impairments: In the presence of certain link impairments such as hidden and exposed terminals, the approaches that try to quantify link quality and measure interference do not perform well as they lack the ability to identify the root cause of lost transmission opportunities [14]. Several works try to address this problem by performing more in depth measurement studies but mostly focusing on a single type of impairment. In [16, 24], a passive PHY layer approach is used to determine when physical layer capture occurs. In [23], a cross-layer approach is used to detect hidden terminals using the fact that a hidden station would only hear an ACK frame without detecting the preceding data
In [7], Giustiniano et al. present a cross-layer approach which is the only approach capable of measuring the majority of link impairments in real WiFi environments. They perform both active and passive measurements to distinguish between lost transmission opportunities due to collisions, noise errors, hidden terminals, capture effect and exposed nodes. Active measurements are performed at the MAC layer whereas passive measurements are performed at the PHY layer. As this is the only extensive approach, we also adopt it in this thesis and present it in detail in the next section.

\footnote{Even the ACK packet cannot be decoded correctly, the PLCP header length can be used to identify ACK packets}


<table>
<thead>
<tr>
<th>Tool</th>
<th>What it measures</th>
<th>Example</th>
<th>Layer</th>
<th>Granularity (Sampling Frequency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>register logger</td>
<td>Hardware register states</td>
<td>energy detect, idle</td>
<td>PHY</td>
<td>high (max. 4 KHz)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PHY busy, RX busy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TX busy, NAV, TSF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>athstats</td>
<td>Frame types and PHY errors</td>
<td>retransmissions, OFDM timing errors</td>
<td>PHY/MAC</td>
<td>low (max. 100 Hz)</td>
</tr>
<tr>
<td>tcpdump</td>
<td>packet trace</td>
<td>radiotap header</td>
<td>MAC</td>
<td>per-frame</td>
</tr>
</tbody>
</table>

Table 4: Table of tools for passive measurements

4 Design and Implementation

As described in Chapter 3, lost transmission opportunities can arise from both unwanted traffic and link impairments. In this chapter, we will present the methodology to measure these: (1) passive PHY and MAC layer measurements to capture unwanted traffic and (2) active PHY and MAC layer measurements to estimate the impact of different link impairments (see Section 2.1). The reason why we use both PHY and MAC layer information is that link impairments that result in packet loss need to be measured at the MAC layer, whereas exposed nodes or physical layer capture can be measured by the combination of PHY and MAC layer measurements. For the active MAC layer measurements, we adapt the approach proposed in [17] to our testbed. The remainder of this section is organized as follows. First, we present the general measurement setup. Next, we describe passive measurements and finally, present how we combine active PHY layer and active MAC layer measurements.

4.1 Description of the General Measurement Environment

In this section, we describe our general measurement set-up, which consists of (1) several tools to collect measurements such as packet traces and a configuration environment for the measurement tool-chain, (2) the measurement storage, and (3) the synchronization mechanism for merging measurements from different nodes. Finally, we present the testbed which was exclusively created for the purpose of this thesis.

4.1.1 Tools and the Configuration Environment

We use several measurement tools to record the wireless environment, and the hardware state and, collect packet traces. Table 4 summarizes the measurement tools we used and provides examples of what could be measured with each tool at which layer. Additionally, the table shows the granularity in terms of the maximum sampling frequency that can be used by the corresponding tool.

Specifically, we use the following measurement tools:

- **register logger** is a patch to the madwifi [1] driver to take snapshots of the register states (shown in the “Example column” in Table 4). The patch is maintained by Thomas Hünn. The information obtained by this tool is used for airtime calculation for erroneous frames during passive measurements (see Section 4.2 and validation of the active measurement approach in Section 4.3).
• **athstats** provides statistics of the hardware state, PHY layer and MAC layer frame types. For the purpose of this thesis, athstats has been modified to dump its output in a machine readable format based on a user-defined sampling frequency. In addition, we have incorporated the statistics collected by the active measurements.

• **tcpdump**[^3] is used to capture packets on a network interface that match a boolean expression and dumps them to a file for offline analysis. We configure *Tcpdump* to capture only the necessary protocol information, for example, for passive measurements (described in detail in Section 4.2). Essentially, capturing full packets does not scale: it leads to packet loss due to increased packet processing time, a bottleneck at the measurement collection point and, increases storage requirements. Hence, we capture only the first 200 bytes of a packet, if not stated otherwise.

Tcqdump is based on the packet capture library *libpcap*. Pcap traces from a radiotap enabled WiFi interface contain the following headers in the following order:

<table>
<thead>
<tr>
<th>1st frame</th>
<th>2nd frame ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Pcap</td>
<td>Pcap Radiotap MAC + Packet Data Pcap Radiotap ...</td>
</tr>
</tbody>
</table>

Here, *Global Pcap* header contains information about the maximum length of the captured packets in octets, the network type and the GMT for local time corrections. *Packet Pcap* (pcap) header contains time and length information of each frame. This information is used to calculate the airtime of a frame. Next, the *Radiotap* header contains PHY information for each received packet, used to again identify and feed the right formula for calculating the airtime with information about the transmission rate and PHY technology (see Section 2.2.2 for the formulas for different PHY technologies). The *IEEE 802.11 MAC* header contains information about the frame type. This is used to categorize the IEEE 802.11 MAC related traffic for passive measurements described in Section 4.2.

• **cpusage** is used to understand the CPU utilization due to measurements and to verify that measurements are not CPU-bound. *cpusage* records the percentages of the CPU state tics counter twice a second. The output is written in a machine readable format to a file. The tool is maintained by Fabian Schneider[^4]. In this thesis version 0.4 is used.

These tools are used in a toolchain for automated measurements. To configure the tool chain, we use the UCI interface, which is the general configuration system of OpenWrt[^2]. The goal of UCI is centralize and simplify the configurations. In particular, we have (1) a configuration file to specify the measurement tool, the granularity of the measurement, where to store the measurement, (2) a parser script to parse the configuration file, (3) a management tool to prepare, start and stop measurements. FIFO pipes are used to buffer the content of the measurement source and to block until another program (e.g., *cat*) reads from the FIFO. The blocking allows the preparation of all measurement tools before initiating a measurement.

### 4.1.2 Storage

For storing measurements, we use a network file system due to the absence of local storage on the nodes. We configured NSFv3 network file system to use small cache sizes to avoid

[^3]: http://www.tcpdump.org/tcpdump_man.html
[^4]: http://www.net.t-labs.tu-berlin.de/ fabian/software_en#cpusage
burstiness of data stream through the network, and also high CPU consumption during bursts. We also reduced the read and write sizes used for file system transaction to avoid fragmentation in the networking layer. Instead of using a network file system, we could also choose to send a continuous data stream (a) via a tool like `netcat` from each measurement tool or (b) via a measurement framework that uses server daemon process like OML [40] or IPFIX [31]. We did not use the approach (b), as it requires patching all measurement tools and we wanted to avoid this. In the approach (a), the output of a measurement tool is piped to `netcat` and then transferred over the network. On the receiver side, the output from the corresponding `netcat` instance is piped to another program or redirected to the disk. However, this requires one `netcat` instance for each data source, since each `netcat` instance establishes one individual connection. Even though we used this approach in the beginning for a small number of measurement sources, it did not scale as we increased the number of sources, and then, we decided to use the network file system.

4.1.3 Synchronization of Measurement Nodes

Synchronization of wireless measurement nodes is achieved by the network time protocol (NTP) due to the absence of a high precision time source like GPS. All nodes in the network run the `ntpd` daemon to synchronize to a central server within the LAN before the measurements take place to achieve higher accuracy. Clock synchronization is disabled during measurements due to possible unwanted clock jumps. To examine the time drift of clocks during the experiments, a 64-bit precise time synchronization function (TSF) of a second wireless card with an accuracy no worse than 0.01% is used (see Section 11.1.2.4 in [3]). NTP version 4 achieves an accuracy error of less than 1 millisecond [28, 36] under the assumption of a low network load and kernel support. Hence, a low network load is mandatory before taking measurements. Furthermore, only a single server should be used to avoid clock jumps due to differences of the time and delay between the servers. The precision and accuracy achieved by NTP suits the requirements for all performed measurements in this thesis. Specifically, we used `ntpd` with the following settings:

```shell
server 172.17.255.254 iburst minpoll 4 maxpoll 4 enable kernel tinker allan 16
```

In case that only chronological ordering of events matters, another approach is to timestamp all incoming packets on the access interface with the TFS of the second WiFi interface, within the wireless driver. However, this approach is expected to introduce unpredictable delays in packet receptions. Another well-known problem is that TSF does not work in multi-hop wireless networks [19, 44]. Hence, this approach would not be applicable to all our testbeds (e.g., the BOWL outdoor network). In case one common channel is used across the whole network, it is also possible to create two “virtual” interfaces over the same physical interface, one for measurement and one for synchronization purposes. However, in practice, most WiFi environments use adjacent channels across all access points within an ESS (Extended Service Set) and hence, this approach is typically not applicable. Nevertheless, in both cases, the TSF can only be used as a logical clock to order events chronologically.

\[ \text{The precision is calculated by ntpd and can be determined by the command ntpq -c rl and calculating } 2^{\text{precision}}. \text{ The precision of nodes within the testbed is around } 2\mu s. \]
4.1.4 Testbed

The active measurements in this thesis require a x86 testbed due to a bug in the proprietary HAL of the OpenWrt madwifi-ng driver. The bug causes the duration/id field of the IEEE 802.11 header to be recalculated and overwritten with wrong values and leads to wrong statistics. The analysis of the bug is further described in Appendix B. Figure 13 shows the testbed that was exclusively created for the purpose of this thesis and was integrated into the existing BOWL indoor testbed in the 16th floor of the TEL building at TU-Berlin. It consists of five x86 based PC Engines WiFi APs equipped with two cards. The first card is a legacy IEEE 802.11abg Atheros AR5212/AR5213 WiFi card used with the madwifi-ng driver. For synchronization, a second IEEE 802.11abgn AR9220/AR9280 WiFi card used with the ath9k driver. To both cards, so-called 2dBi "rubber duck" antennas are attached. The operating system is the release of OpenWrt called "Backfire" subversion revision 26741. We operate this testbed in the 2.4 GHz band, which is also used by other wireless networks in the environment. The next two sections detail our measurements in this testbed.

4.2 Passive Measurements of Lost Transmission Opportunities

We use a passive approach to measure lost transmission opportunities due to unwanted traffic through tcpdump packet traces taken by “sniffing” out all received frames and the output of the register logger (see Section 4.1.1). In particular, such “sniffing” is possible as most of today’s off-the-shelf hardware allows the creation of a monitoring interface (e.g., a virtual interface that only listens to the medium). To determine unwanted traffic, we first measure the amount of IEEE 802.11 related traffic and next, the airtime of this traffic. In the remainder of this section, we first show how frame types are identified and finally show how the airtime of frames from the available information is calculated.

4.2.1 Identifying IEEE 802.11 Traffic

In Section 3, we discussed several measurement studies that show a high overhead from certain types of control and management frames in different networking environments. Hence, it is important to identify such unwanted traffic, as it is possible to reduce this overhead by reducing the interval of keepalive traffic or changing the PHY layer mode (e.g., by disabling IEEE 802.11b and not allowing compatibility mode). To this end, we filter two IEEE 802.11 frame types from PCAP traces (by examining the MAC header): management and control traffic, which are either used for specific situations (e.g., when performing handover) or is a part of the standard unicast transmission (e.g., ACK messages).
To measure the amount of IEEE 802.11 management and control traffic, we examine the MAC header for each frame for the following information:

- **Valid FCS**: We check if a frame has a valid FCS to only account for correctly received frames. Otherwise the frame control (FC) in the MAC header (see Section 2) might be corrupted, which would lead to wrong statistics.

- **Type of IEEE 802.11 frame**: This field is used for differentiation between the three categories: management, control and data of the IEEE 802.11 frame types.

- **Subtype of IEEE 802.11 frame**: The subtype field is used to differentiate between all variations of frames within each type. For example: a frame of management type and association request subtype.

### 4.2.2 Calculation of the Airtime of IEEE 802.11 Traffic from Measurements

To relate the amount of IEEE 802.11 management and control traffic to the overall available capacity, we compute the airtime of IEEE 802.11 frames, as described in Section 2. To this end, we require the following information from the packet trace:

- **Packet length of a frame from the PCAP header**

- **Transmission rate from the Radiotap header**

- **Modulation of the payload from the Radiotap header**

- **Channel Type information, whether the frame was sent in OFDM, DSSS or compatibility mode from the Radiotap header**

- **Length of the preamble (i.e., whether it is a short or a long preamble)**

Again, we only consider correctly received frames from the pcap traces. The overall consumed airtime is also passively measured by the register logger trace. Due to incorrectly received frames we derive the overall channel utilization from the energy detection, receive busy, transmit busy and idle slots which are accessible through the register logger trace every 10\(ms\) (100 Hz).

### 4.3 Active Measurements of Lost Transmission Opportunities

Our active measurements are performed using the approach proposed by Giustiniano et. al in [17] and by Leith and Malone [26]. From this point on, we will refer to this approach as LIME (Link IMpairment Estimator). LIME is able to distinguish losses caused by noise, collision and hidden terminals by using MAC layer measurements. With additional information from the PHY layer, the approach can also distinguish between unfairness caused by exposed nodes and the capture effect. Note that, lost transmission opportunities usually arise from both the PHY and MAC layer, hence, the estimation of the link impairments require measurements of both layers.

Our implementation is partly based on driver modifications on the LIME code provided by the Hamilton Institute, which was used in [26] and [17]. We ported LIME to the madwifi-ng of OpenWrt [1]. This was necessary to get LIME to run on the BOWL testbeds, which contain different types of embedded hardware (ARM, MIPS and x86). The original driver also does not support Linux kernels after 2.6.24, thus making it impossible to run on the hardware architectures used in the BOWL testbeds. The porting of the LIME code took a considerable amount of time as the HAL and the driver in madwifi-ng contain...
several improvements and stability fixes and so, the vast majority of the code-base of the MadWIFI-ng in OpenWrt were changed compared to the original MadWIFI v0.9.4 release [1].

In addition to porting LIME to madwifi-ng, we extended it to collect statistics from user space. Specifically, all the MAC layer packet statistics from LIME are collected from the driver with the athstats utility. Originally, printk was used to export LIME statistics, however, on some of our hardware this caused delays in the transmission of packets in a burst due to a lack of computing power. Next, we summarize the main idea of LIME and the validations of the new LIME implementation for the madwifi-ng.

4.3.1 LIME in a Nutshell

LIME builds on the fact that certain packet transmissions in IEEE 802.11 standard are subject to different types of loss. Specifically, it uses a packet-pair technique, implemented via fragments in IEEE 802.11, and exploits the fact that IEEE 802.11 uses short inter-frame spacing (SIFS) and ACK messages to protect fragment frames. In particular, in LIME, frames are fragmented into two fragments which are sent in a burst inter-spaced by a SIFS, hence all the secondary frames are protected from collisions. This is due to the basic medium access mechanism (based on DCF) of IEEE 802.11, which requires each station to wait for at least a DIFS (which is greater than SIFS) before transmitting. In order to protect secondary fragments against loss due to hidden terminals, for all the primary frames in a burst, the duration/ID field of the first fragment is set to a value in microseconds which covers the transmission of fragments and their ACKs (see Section 2.2.1). Importantly, the remaining duration from the first fragment is echoed in the corresponding fragment ACK resulting in updating of the NAV at surrounding stations (see Section 2.2.3). Accordingly, the duration field in the fragment ACK is akin to the RTS/CTS mechanism from the perspective of hidden terminals. In particular, first fragment of a burst is subject to collisions, noise and hidden terminals whereas second fragment is only subject to noise. To differentiate between collisions and hidden terminals, LIME alternately sends second fragments in an unprotected manner (i.e., by not setting the duration/ID field). This way, the second unprotected fragments are subject to both noise and hidden terminals, which allows to extract the collision probability.

The burst of protected fragments is shown in Figure 14 whereas a burst of unprotected frames is shown in Figure 15. In Figure 14, the sender gains access to the medium and transmits its first fragment containing the duration for the entire transmission sequence (i.e., 2 fragments, 2 ACKs and inter-frame spacing). The receiver echoes the duration in the ACK to the first fragment, which updates the NAV of hidden stations correctly until the end of second ACK. On the other hand, in Figure 15, the first fragment does not contain the duration of the entire transmission sequence (i.e., only covers its transmission duration), and so, the corresponding ACK does not update the NAV of the hidden stations and, hence, the second fragment is subject to interference from these stations.

In the remainder of this section, we explain how LIME estimates the probability of loss due to noise, hidden terminals and collisions based on this packet-pair technique. Table 5 shows the measurement counter and descriptions. As stated before, protected second fragments of a burst are only subject to noise, hence we can estimate the success probability of these with

\[
P[\text{success of 2nd protected fragment}] = (1 - p_n) = \frac{A_S}{T_S}, \tag{7}\]

where \(p_n\) is the probability of loss due to noise, \(T_S\) are all protected subsequent data frames sent in a burst and \(A_S\) of such were successfully transmitted. Hence, \(p_n\) is:

\[
p_n = (1 - \frac{A_S}{T_S}). \tag{8}\]
Figure 14: Burst of protected fragments

Figure 15: Burst of unprotected fragments
Table 5: Measurement counters and estimators

<table>
<thead>
<tr>
<th>Counter</th>
<th>Description</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_0$</td>
<td>First data frames in a burst</td>
<td>Normal DCF frames</td>
</tr>
<tr>
<td>$A_0$</td>
<td>ACK of first frames in a burst</td>
<td>Normal DCF frames</td>
</tr>
<tr>
<td>$T_1$</td>
<td>Unprotected subsequent data frames</td>
<td>Unprotected burst</td>
</tr>
<tr>
<td>$A_1$</td>
<td>ACK of unprotected subsequent data frames</td>
<td>Unprotected burst</td>
</tr>
<tr>
<td>$T_S$</td>
<td>Protected subsequent data frames</td>
<td>Protected burst</td>
</tr>
<tr>
<td>$A_S$</td>
<td>ACK of protected subsequent data frames</td>
<td>Protected burst</td>
</tr>
<tr>
<td>$I$</td>
<td>idle slots</td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>rx slots busy</td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td>tx slots busy</td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td>medium busy</td>
<td></td>
</tr>
</tbody>
</table>

On the other hand, unprotected second fragments are only subject to noise and hidden terminal interference. Accordingly, the successful transmission of such frames is:

$$\Pr[\text{success of 2nd unprotected fragment}] = (1 - p_h)(1 - p_n) = A_1/T_1,$$

where the station transmits $T_1$ unprotected second fragments and $A_1$ of such were successfully transmitted. We insert now the probability of loss due to noise $p_n$ from Equation 8 and get the probability of loss due to hidden terminal interference, $p_h$:

$$p_h = 1 - (A_1 \times T_S)/(A_S \times T_1).$$

Finally, we calculate the probability of loss due to collisions. From Equation 8 and Equation 12, we already have the probabilities of loss due to noise and hidden terminal interference. Note that, all first fragments in a burst use the standard DCF, thus these frames share the same collision probability as all other frames that use the standard DCF. In addition, all first fragments are also subject to noise and hidden terminal interference. Hence, all first fragment transmissions are successful if:

$$\Pr[\text{success of 1st fragment}] = (1 - p_c)(1 - p_h)(1 - p_n) = A_0/T_0,$$

where the station transmits $T_0$ first fragments in a burst and $A_0$ of such were successfully transmitted. We now insert again the probability of loss due to noise $p_n$ from Equation 8 and the probability of hidden terminal interference $p_h$ from Equation 12 to get the probability of loss due to collisions, $p_c$:

$$p_c = 1 - (A_0 \times T_1)/(A_1 \times T_0).$$

4.3.2 Validation of LIME

For LIME, it is crucial that the Interframe Spacing (IFS) between two fragments sent in a burst is not more than a SIFS (see in Figure 14 and Figure 15) and the NAV is set correctly in surrounding stations. We first validate the duration of the inter-frame spacing by using a spectrum analyzer. We needed to perform the validation tests twice due to implementation changes in two critical parts of the transmit path of the driver. The first validation was done at the Hamilton Institute using a Rohde & Schwarz (R&S)
Figure 16: Burst of two fragments with their corresponding ACK and the measured intermediately gap of a SIFS. The second frame is protected against PHY layer collisions.

FSL6 spectrum analyzer[^6], whereas the second validation was done with a R&S FSV7 spectrum analyzer[^7]. Both measurements were taken with R&S IQWizard 4.8[^8], which provides remote control of the spectrum analyzer to capture the I and Q data of a wireless transmission. The output can be captured in a data format suitable for Matlab. The Matlab code and device settings are further described in Appendix 6.

The measurement setup consisted of one STA sending unicast data traffic at the maximum power level of 17dBm to an AP. The antenna of the spectrum analyzer was located close to the antenna of the STA. For all experiments, an empty channel was chosen, which was also verified by the spectrum analyzer. Since both experiments produced the same results, we only present the newer results in this section. Older results of the first measurement can be found in the Appendix 6.

Figure 16 shows the measured amplitude of the transmission of two fragments sent in a burst with their corresponding ACKs. The higher amplitude represents data transmissions by the STA, the less high amplitude shows ACK transmissions by the AP. Each transmission follows a gap of a SIFS interval of 16µs duration. As stated before, the most important finding is the SIFS interval between the first ACK and the second fragment. All traces contain several burst transmissions and all have correct SIFS intervals. Based on this experiment, we confirm that the inter-frame spacing of two fragments are of length SIFS and that second frames are protected against collisions.

Next, LIME relies on correct setting of NAV to distinguish hidden terminals and noise. Therefore, we validate whether the NAV is set correctly based on the information in the duration/ID field of fragments and ACKs. Specifically, we need to validate the following:

[^8]: http://www2.rohde-schwarz.com/en/service_and_support/Downloads/Application_Notes/?type=20\&downid=1439
• Duration field of the first fragment contains the correct duration including the duration of the second fragment and ACK.

• ACKs of the first fragment echo the correct duration until the end of second fragment and its ACK.

• NAV value at the Hidden Terminal (HT) is updated by the ACKs.

The validation was done by passively capturing the frame exchange between the sender and the receiver in a clean channel. In particular, absence of interference was confirmed by register logger traces that record the energy detection register of the hardware. Within the trace, the correct setting of the duration fields of the frames and the corresponding ACKs were verified. Furthermore, a register logger trace of the NAV register at the hidden terminal was taken, and it was verified that the NAV was set.

With these validations, we conclude our design and implementation section, and present measurement results in the next section.
5 Network Measurements

In this chapter, we present passive and active measurements carried out using the measurement framework described in Chapter 4. In the first section, we present passive cross-layer measurements to quantify unwanted IEEE 802.11 traffic. In the second section, we discuss our active measurement results using the LIME estimator.

5.1 Identifying IEEE 802.11 traffic overhead

To collect passive measurements, we use five measurement points in the testbed described in Chapter 4.1.4. These nodes monitor all traffic (including from other networks) at the 802.11bg channels 1, 6 and 11 at 2.4 GHz (which are orthogonal) at different periods of time. We did not measure the 5 Ghz frequency spectrum, since active scans have shown no IEEE 802.11a APs. The environment in which we perform the measurements consists of an unknown number of APs and STAs. To our surprise, we found that only a small portion of the captured traffic are data frames. This can be explained by the availability of wired Ethernet connectivity for desktop PCs as well as for laptops. Our environment therefore consist of users who do not tend to rely on wireless connectivity for their daily work, despite the availability of two wireless networks.

In our traces, we see that all APs announce that they operate in the compatibility mode with short preambles (see Section 2.2.3). Note, this is not state of the art anymore. In the compatibility mode, all management frames are sent at the basic rate, which leads to a significant amount of airtime consumption. Control frames are often also sent at basic rate, but might be also sent at higher rates. STAs in such environments have three solutions to send data frames: (1) send frames in compatibility mode where an IEEE 802.11b compatible preamble precedes the OFDM encoded frame, (2) send only OFDM encoded frames in the absence of 802.11b stations or (3) use a CTS-to-self or RTS/CTS mechanism which precedes the OFDM encoded frame transmission. Even though RTS/CTS frame sizes are small, they incur a constant physical layer overhead which results in an additional airtime consumption for each transmitted data frame [9]. The IEEE 802.11 standard warns that RTS/CTS might not be useful short data frames, but does not explicitly define when RTS/CTS mechanism should be used. Furthermore, the use of RTS/CTS may result in false blocking of nodes and temporary deadlock situations in ad-hoc networks [35]. Therefore, both CTS-to-self or RTS/CTS mechanisms should be disabled for better throughput performance.

In the remainder of this section, we first present our analysis based on a 36 hour packet trace and register logger taken at channel 6 on the 18th of September 2011 starting at 20:11:34 UTC (22:11:34 CEST) and then, analyze a 4 hour subtrace in detail. We have identified 175 STAs over the 36 hours trace and 95 in the 4 hours afternoon trace. Roughly 50 percent of the data transmissions are sent using IEEE 802.11g rates. We first analyze the channel utilization in terms of airtime consumption from the register logger and packet traces. Previous work identified a high overhead of management traffic. Hence, we further look into the airtime consumption of specific subtypes of the three IEEE 802.11 frame type categories, including also the control traffic and NULL-data traffic (data frames with no payload). Finally, we present the proportion of frame subtypes which depend on the number of active STAs. All figures show the time in UTC.

Figure 17 depicts the energy detection (medium busy), receive (rx) busy and transmit (tx) busy slots measured from the register logger and the airtime consumption of all packets (including packets with checksums errors) from our packet trace. All data is aggregated
Figure 17: Illustrates the ratio of medium, receive (rx) and transmit (tx) busy measured by the register logger as well as the airtime of all frames captured on channel 6 over 36 hours.

Next, we analyze the airtime consumption differentiated by the IEEE 802.11 frame types from the packet traces. Figure 18 depicts the airtime consumed by management, control, data and frames with a wrong checksum (Undefined (CRCerr) in the figure). Furthermore, the figure shows the medium busy from the register logger as a reference and the total airtime consumption of all received frames in our packet trace. We see that the airtime consumption of management traffic shows a high variation in the presence of data traffic and slightly increases during daytime. This variation is expected since beacon transmissions are aborted if the medium is sensed busy (instead of being deferred as beacons need to respect beacon intervals). When the background traffic increases, the medium is sensed busy more often, and so, resulting in a higher number of aborted beacon transmissions. Furthermore, we observe that data traffic increases during daytime, but consumes only 5 to 10 percent of the airtime. In addition, we see that the airtime consumed by control traffic increases approximately as much as the data traffic. We also observe an increase of frame corruption during the peak hours, which is most likely due to the increase of interference.

Finally, we analyze the proportion of traffic depending on the presence of STAs and data traffic. Figure 19 depicts the proportion of management probe traffic, RTS/CTS and
Figure 18: The ratio of airtime consumption differentiated by the IEEE 802.11 frame types from the packet traces captured on channel 6 over 36 hours.

Figure 19: Proportion of management probe traffic, RTS/CTS and data NULL traffic over their respective frame categories in the packet traces captured on channel 6 over 36 hours.
Figure 20: The ratio of medium, receive (tx) and transmit (tx) busy measured by the register logger as well as the artime of all frames captured on channel 6 over 4 hours in the afternoon between 12:00 UTC and 16:00 UTC.

data NULL traffic over their respective frame categories. We see an increasing amount of airtime consumption of probe traffic up to 50 percent during business hours ranging from 6 UTC (8 CEST) until 18 UTC (20 CEST). This is due to an increasing number of STAs since management probe frames are used by the STA’s supplicant to discover the neighborhood (e.g., the number of BSSIDs which belong to an ESSID). Another observation is that we see a huge amount of RTS/CTS control frames confirming that some stations do not use the “two-way handshake”, which incurs less overhead. In addition, we see an increasing share of data NULL frames on the overall amount of data traffic. In particular, STAs which use the power save mode send unicast data frames to the associated AP with the ‘Power Management’ field in the frame control (FC) set to 1. This signals to the AP to buffer pending frames until the STA wakes up and asks the AP to send these frames. We have identified that, 130 STAs make use of the power save mechanism and use data NULL frames for signalling their wake-up and sleep status.

From our analysis of our 36 hours trace, we conclude, that the calculation of the remaining channel capacity is beyond our passive measurement capabilities. We further conclude, that probe traffic depends on the number of surrounding STAs and whereas RTS/CTS depends on the STA activity. In the remainder of this section, we analyze the 4 hour afternoon trace ranging from 12:00 to 16:00, which has a higher amount of data and control traffic (see Figure [18]). First, we analyze the channel utilization as in Figure [17] but aggregated into blocks of 60 seconds. In Figure [20] we observe a high averaged medium busy time between 13:30 UTC (15:30 CEST) and 14:45 UTC, during which the card received 20% less frames. This is most likely due to simultaneous transmissions where the WiFi card tried to capture the preamble, but failed to demodulate the transmissions.

Next, we analyze the airtime utilization differentiated by the different IEEE 802.11 frames as described earlier. We see in Figure [21] that the amount of corrupted frames increases during the previously identified peak time up to 10 percent. Furthermore, we
see a decrease (10%) of airtime consumed by management frames during high channel utilization periods. We also observe that the airtime consumption of control traffic is mostly flat, whereas the data traffic shows a high variation.

In addition, we identify management traffic which is affected during high channel utilization periods. Figure 22 depicts the different subtypes of management frame types. We observe that the transmission of beacons depends highly on the channel utilization. However, we could not identify the source of the drop of probe traffic. Another observation is that the amount of airtime consumed by authentication and association frames is nominal.

To understand the effect of control traffic, in Figure 23 we show the airtime utilization of RTS/CTS and ACK frames as well as the overall airtime utilization. The total control overhead depends purely on the activity of the nodes. From the 95 STAs roughly 60 percent (54 STAs) use RTS/CTS to protect frame transmissions, however, we could not identify whether the stations use CTS-to-self or RTS/CTS. This is due to the fact that CTS frames only contain the receiver address and the RTS frame might be missed. One observation for which we could not come with an explanation is that some access points send bursts of CTS frames which reserve the medium for tens of milliseconds, resulting in a medium reservation of around 10 percent of the total airtime. Figure 24 shows periodic bursts sent every 90 seconds for a few seconds. Due to the absence of data traffic and the periodicity of events we guess that this behaviour might be used by APs for noise floor calibrations.

Finally, in Figure 25, we look at the number of data NULL-frames. As expected, data NULL-frames are sent more often when there is no data traffic (i.e., the stations indicate the wake-up and sleep states but have nothing to send or receive). Overall, even though the airtime utilization of data NULL-frames is negligible, the number of these frames
Figure 22: The airtime consumption of the IEEE 802.11 management subtypes from the packet traces captured at channel 6 over 4 hours in the afternoon between 12:00 UTC and 16:00 UTC.
Figure 23: The airtime consumption of the IEEE 802.11 control subtypes from the packet traces captured at channel 6 over 4 hours in the afternoon between 12:00 UTC and 16:00 UTC.

Figure 24: Vertical lines represent the NAV register value and the dots the duration from received CTS frames.
are significant. Considering each frame incurs a IFS time, the effect of this overhead is also high. During the 4 hours, we have identified 67 STAs using data NULL frames for signalling their wake-up and sleep status while in power save mode.

In summary, our results indicate a significant overhead from control and management traffic, resulting in a high amount of consumed airtime. To mitigate this overhead, the network operator could use a channel selection algorithm to find the least crowded channel. However, the ideal option is that the operators of enterprise networks configure their networks properly. For example, disabling the 802.11b support would prevent stations from sending frames preceding with a long or short DSSS preamble.

5.2 Active measurement of lost transmission opportunities

In this section, we present active measurements using the LIME approach presented in Section 4.3.1. Our aim is to understand the behaviour of this approach under different conditions. We first present results from a controlled environment and then continue with the estimated loss probability of different link impairments.

In order to determine the loss probabilities due to link impairments, we ran our experiments in a controlled environment without external link impairments and interference. The absence of external impairments and interference was verified by using the register logger. To be able to reproduce experiments, we disabled the Atheros specific noise and interference mitigation algorithm and additionally disabled the signal detection mechanism based on correlation (weak signal detection) (see Appendix ??).

We have chosen channel 149 and adjusted the channel busy threshold (CCA) to the defined standard of -70 dBm. We injected UDP traffic using the traffic generation tool iperf. Note that in all experiments, the frame length of the first and second fragments is almost identical differing at most by two bytes.

**Noise probability measurements**: The setup consists of 2 nodes: one AP and one STA. Traffic is sent at 6 Mbps over a link whose transmission rate is set to 24 Mbps. This
transmission rate was chosen to remove an unwanted capture effect with lower rates at the AP. Specifically, due to the capture effect all frames at low rates were received by the AP due to a higher signal-to-noise ratio. ACKs are sent with 24 Mbps by the AP at the maximum transmit power.

We have carried out several experiments where each lasted exactly 300 seconds. The transmit power of the STA was reduced every 17 seconds due to the 17 possible power levels. In other words, this slowly increases the loss due to noise every 17 seconds by reducing the transmit power. We expect that the $p_h$ is 0, and that the noise probability increases over time while reducing the transmit power. Figure 26 shows the estimated loss probability due to hidden terminals, $p_h$ (line/blue/circle), noise, $p_n$ (red/dotted/plus), and collisions $p_c$ (green/short dashed/triangle). The estimated loss probability is compared to the real total loss (dashed/black/cross). Figure 26 shows that the noise probability $p_n$ is close to the real loss and follows the same trend. At around 7 dBm transmission power, the first loss occurs and at around 1 dBm we observe more than 90% total loss rate. In particular, due to the high loss rate and the low transmission success of frames the estimator is not able to perform a precise loss estimation. All measurement runs show the same behaviour and, the figure is one sample.

Note that, no contending STAs are in STA’s carrier sense (CS) range and no hidden terminals are present, hence the collision probability $p_c$ and hidden terminal probability $p_h$ should be zero. However, both show a huge variation with decreasing power. This can be explained by the low transmission success probability of all frame transmissions and high channel dynamics. Note, that frames are sent in packet pairs where the transmission of the second frame depends on the success of the first frame which leads to a small number of samples. Hence, the loss probability might not be completely independent and affects the sample distribution. This can lead to a wrong estimated loss probability, since LIME assumes independence between the different types of losses.

**Hidden terminal probability measurements:** Now we add to the scenario one hidden
terminal which completely utilizes the channel. We expect that the $p_h$ is around 90 percent, and that the noise probability increases over time while reducing the transmit power. The hidden terminal situation between STA (tel-16-5-alix) and the hidden terminal (tel-16-1-alix) was verified by taking a trace of the energy detection register with the register logger. The hidden terminal completely utilizes the channel by sending UDP broadcast-traffic at the maximum power with a transmission rate of 24Mbps. The STA sends a continuous stream of UDP packets at 6 Mbps to an AP with a transmission rate of 24Mbps. AP’s ACKs are echoed at maximum power with a transmission rate of 24 Mbps and are received by the HT. Hence, the NAV is up-to-date. The transmission power of the STA is again reduced every 17 seconds to slowly increase the loss due to noise.

Figure 28 shows the estimated loss probability due to hidden terminals $p_h$ (full line/blue/circle) probability, noise $p_n$ (dotted/red/plus) and collisions $p_c$ (short dashed/green/triangles). The hidden terminal probability $p_h$ is close to the real total loss (dashed/black/crosses), which exactly matches our expectation. Furthermore, the estimated loss probability due to noise $p_n$ slightly increases again at around 7 dBm transmission power when reducing the transmission power.

The cause of the increase in estimated noise probability can be due to collisions of the ACK of the first fragment and transmissions of the hidden terminal. This is shown in Figure 27. In this case the NAV at the hidden terminal is not updated and the second protected frame corrupted. Furthermore if the link between the HT and the AP is unidirectional, ACKs of the AP are not received by the HT. Hence, the HT does not update its NAV. The HT would always send and STA is unable to detect the HT. Additionally, CTS can result in updating the NAV at the AP which would prevent it from sending ACKs to the STA and resulting in wrong statistics. Once again, more random losses at low transmission power affect all frames, hence the estimation of loss increases.
Figure 28: The loss probability estimates from the LIME estimator over time. The transmission power of the STA is continuously reduced every 17 seconds. The hidden terminal probability $p_h$ (full line/blue/circle) follows the trend of the total loss (black/dashed line), but overestimates the loss due to hidden terminal probability. From 7 dBm on the loss due to noise probability $p_n$ increases whereas the hidden terminal probability decreases. The estimator shows clearly a operational range in which estimates are accurate.

Collision probability measurements: Now we incrementally add up to seven additional contending STAs to the environment which finally completely utilize the channel. We expect that the $p_c$ (short dashed/green/triangles) gradually increases with the number of contending STAs, and that the noise and hidden terminal probability stay zero. Each contending STAs sends a continuous stream of UDP traffic at 700 Kbps to the AP. The number of contending STAs is increased every 30 seconds. All STAs transmit at the maximum power with a rate of 6Mbps. AP’s ACKs are echoed at maximum power with a transmission rate of 6 Mbs. All STAs are within each others CS and, hence back-off in case of unsuccessful frame transmissions.

Figure 29 shows the estimated loss probability due to hidden terminals $p_h$ (full line/blue/circle) probability, noise $p_n$ (dotted/red/plus) and collisions $p_c$ (short dashed/green/triangles). The collision probability $p_c$ follows closely the trend of the real loss (dashed/black/crosses) while the noise and hidden terminal probability stay low. This exactly matches our expectation. However, the estimated collision probability shows a small drift when the channel becomes completely utilized. Furthermore, the estimated loss probability due to noise $p_n$ slightly increases over time, but stays below 5 percent. This can again be explained by the low transmission success probability of all frame transmissions and high channel dynamics which leads while sending packet pairs to a small number of samples. Hence, the loss probability might not be completely independent and affects the sample distribution.

Live measurement of loss probability: Finally, we measure the estimated loss probability in a live environment. The root-cause of loss in a live environment is unverifiable and hence, can only be estimated. Figure 30 shows the link impairments of an experiment which was taken during the night with existing live traffic. The setup is slightly different from the previous setups as we reduce the transmission power every 60 seconds until 8 dBm. Data was sent on a late evening between node tel-16-5-alix (STA) and tel-16-4-alix (AP). Figure 30 show that the by LIME estimated loss probability follows closely the
Figure 29: The loss probability estimates from the LIME estimator over time and the number of additional contending stations is continuously increased up to 7. The collision probability $p_c$ (short dashed/green/triangles) follows closely the trend of the total loss (dashed/black/crosses), while the noise and hidden terminal probability stay low.

Figure 30: The loss probability estimates from the LIME estimator over time and the transmission power of the STA is continuously reduced every 60 seconds. The estimation of loss follow the trend of the total loss (dashed/black/cross).
Figure 31: The link between the AP and the hidden terminal is asymmetric such that the ACKs are not received by the HT. This would lead to a wrong noise and hidden terminal estimation.

trend of the real loss (dashed/black/cross). Furthermore, we observe that the loss due to noise increases by reducing the transmit power, whereas the loss due to hidden terminals and collisions vary. In live environments, LIME could also suffer from an existing 802.11b or compatibility-mode network, since ACKs of an OFDM network are sent with up to 24 Mbps but frames from a compatibility-mode network are still sent at 1 Mbps (basic rate), which can lead to link asymmetries. In this case, ACKs are not received by the HT and hence, second fragments not protected anymore. LIME would falsely identify this as loss due to noise. This could be the case in our environment and would require another controlled experiment which we leave as future work.
6 Conclusion and Future Work

In this thesis, we presented a general framework for performing cross-layer measurements in WiFi networks. We have shown the advantages of combining active and passive cross-layer measurements to better understand lost transmission opportunities. From the passive measurements, we have shown that a significant amount of IEEE 802.11 control and management traffic exist, reducing the wireless network capacity. We have encountered several cases, where the root-cause of unwanted traffic lies in wrongly configured APs and STAs. For active measurements, we showed that the LIME [17] approach is able to estimate loss due to link impairments and achieves a good accuracy within a specific link quality range. Beyond this range, we believe that taking additional information, such as channel state informations into account could improve the estimation.

Future work includes evaluating the LIME [17] approach on the BOWL indoor and outdoor network with real clients. This requires addressing several challenges ranging from driver-based issues to ensuring client data performance during active measurements. For instance, due to a bug in the closed source hardware abstraction layer (HAL) of the OpenWrt MadWiFi driver, the duration field is handled incorrectly on the hardware platform of the BOWL indoor and outdoor testbed. This bug does not appear on the Open Source wireless driver ath5k, hence, the mechanism needs to be ported to this newer driver. Furthermore, this would require another validation before LIME can be used for BOWL networks. Another challenge is to perform measurements with live traffic. For instance, LIME uses fixed transmission rate measurements, which would be impractical for all clients. On the other hand, if different transmission rates are used, this would lead to different airtimes for the first and second fragments, and hence, different loss probabilities - making it difficult to identify the source of link impairment. Finally, it might be impractical to use all live traffic for the estimation of link impairments, hence, selecting a subset of frames for probing (sampling approach) might be beneficial (e.g, as in the case of “SampleRate” rate control algorithm[7]). However, this would require an understanding of how much of the data traffic should be used for measurements to be able to make reasonable estimations.
A Appendix 1

For an evaluation of signals in the time domain, the spectrum analyser is configured to sample on a center frequency in a mode that is called zero span (SP). Further parameters are:

- **Span (SP):** the distance between two frequencies if analysed in a frequency domain. If a signal is analysed in the time domain the span is 0 Hz called ”zero Span”.

- **Resolution bandwidth (RBW):** the bandwidth of the channel. For IEEE 802.11abg the bandwidth is by default 20 MHz. IEEE 802.11n supports 40MHz channel bandwidth.

- **Sweep time (ST):** the frequency in which the analyser sweeps between two given frequencies.

Table 6 shows the settings used in all experiments for the spectrum analyser.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Span (SP)</td>
<td>zero Span</td>
<td></td>
</tr>
<tr>
<td>Sweep time (ST)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution bandwidth (RBW)</td>
<td>20 MHz</td>
<td></td>
</tr>
<tr>
<td>Samples</td>
<td>523776</td>
<td></td>
</tr>
<tr>
<td>Sample Rate (SR)</td>
<td>40 MHz</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Table of tools for passive measurements

Matlab has been the first choice for three reasons: (1) the program IQWizard stores the measured data in a suitable format, (2) it allows the selection of measurement points within a plot and (3) it is faster than the open source alternative Octave which uses GNUplot.

The following code was used to plot the IQ values from the Spectrum Analyzer in Matlab.

```matlab
load('feb9_3_40MSps');
fs = 40e6;

Ts = 1/fs;
N=523776;
t = [Ts:Ts:N*Ts];
mag = abs(I.^2 + Q.^2);

figure(1);
% plot(t,mag);
semilogy(t,mag);
xlabel('time (sec)'); ylabel('Sample amplitude (|I^2 + Q^2|)');
```

The following code was used for measuring the IFS of the frame busts.
\[
\text{ans} = 1.6450 \times 10^{-5}
\]
\[
\text{ans} = 1.6404 \times 10^{-5}
\]

Figure 32 shows the measured IFS from the Spectrum Analyzer at NUIM.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{Figure32.png}
\caption{The measured Inter-frame Spacing (IFS) in the first implementation of the LIME mechanism}
\end{figure}

\section*{B Appendix 2}

From the STA (tel-16-5-asus) three ICMP ping frames are sent with transmit rate 12Mbps to an AP (tel-16-4-asus).

```
root@tel-16-5-asus:~# dmesg -c > /dev/null && ping 192.168.1.1 -s 1400 -c 3 && dmesg -c
PING 192.168.1.1 (192.168.1.1): 1400 data bytes
1408 bytes from 192.168.1.1: seq=0 ttl=64 time=43.292 ms
1408 bytes from 192.168.1.1: seq=1 ttl=64 time=43.105 ms
1408 bytes from 192.168.1.1: seq=2 ttl=64 time=42.672 ms
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 42.672/43.023/43.292 ms
```

The driver reports that the duration field is successfully overwritten to “555”.
1: duration was 48 and will be set to 48 current value is 0
1.1: new duration is 555 and will be set to 555 current value is 555
2: txdesc: time: 5179350.849389 pktlen 1466 hdrlen 26 atype 0 txpower 60
   txrate 10 try0 11 keyix 4 ant 0 flags 1 ctsrate 0 ctsdur 0 icvlen 0 ivlen 0 comp 3
3: flags=1 and bf->bf_flags=0
4: duration=555
ATH_DEBUG_TX_PROC: Updating frame’s sequence number from 854 to 910
1: duration was 48 and will be set to 48 current value is 0
1.1: new duration is 555 and will be set to 555 current value is 555
2: txdesc: time: 5179351.892726 pktlen 1466 hdrlen 26 atype 0 txpower 60
   txrate 10 try0 11 keyix 4 ant 0 flags 1 ctsrate 0 ctsdur 0 icvlen 0 ivlen 0 comp 3
3: flags=1 and bf->bf_flags=0
4: duration=555
ATH_DEBUG_TX_PROC: Updating frame’s sequence number from 855 to 911
1: duration was 48 and will be set to 48 current value is 0
1.1: new duration is 555 and will be set to 555 current value is 555
2: txdesc: time: 5179352.935969 pktlen 1466 hdrlen 26 atype 0 txpower 60
   txrate 10 try0 11 keyix 4 ant 0 flags 1 ctsrate 0 ctsdur 0 icvlen 0 ivlen 0 comp 3
3: flags=1 and bf->bf_flags=0
4: duration=555
ATH_DEBUG_TX_PROC: Updating frame’s sequence number from 856 to 912

However, the received frames at tel-16-4-asus shows that the duration field of the three
received frames is zero (shown in the second column). We assume, that the HAL overwrites
the duration field, since its neither a hardware bug (works with ath5k) nor a endianness
bug (issue shows up on ARM (big endian) and MIPS (little endian).

trace tel-16-4-asus-orig13.cap
|wlan.seq|wlan.duration|wlan.fc.ds|wlan.sa|wlan.da|wlan.bssid
|910|0|0x01|00:0b:6b:84:b3:88|00:0b:6b:84:b5:de|00:0b:6b:84:b5:de
|911|0|0x01|00:0b:6b:84:b3:88|00:0b:6b:84:b5:de|00:0b:6b:84:b5:de
|912|0|0x01|00:0b:6b:84:b3:88|00:0b:6b:84:b5:de|00:0b:6b:84:b5:de

C Appendix 3

The Atheros hardware can be configured by various parameters which have a huge impact
on the reception of a frame. Hence, it is critical for the repeatability of measurements to
deactivate all proprietary mechanisms of the atheros hardware and driver.

The Atheros hardware implements two different detection mechanisms for frame re-
ception.

**Strong Signal detection** is a mechanism defined by the standard and is based on the
signal strength of an incoming signal compared to the reference noise level. This
mechanism will trigger the reception of a frame in case of a SNR greater than 17
dB.

**Weak signal detection** is a correlation mechanism. For OFDM it tries to find a ”self-
correlation” of the periodicity of the preamble within an incoming signal. Since
Baker signals in CCK are multiplied by either -1 or +1 the multiplication is applied
to the signal to establish a CCK pattern. Thus this correlation mechanism could be described by the term "cross-correlation".

Both mechanisms can be adjusted by changing, e.g., receive sensitivity or noise immunity. We use a combination of driver default settings and recommendations from the IEEE 802.11 standard. We set the noise immunity to level 2 and the channel busy threshold (CCA) to the defined standard of -70 dBm for IEEE 802.11a and -62 dB for IEEE 802.11bg channels. Previous work has shown a -82 dB receive sensitivity of the Atheros hardware.

The proprietary Hardware Abstraction Layer (HAL) of the madwifi-ng contains a patented mechanism called “Adaptive Interference Immunity Control” [20] (aka “Ambient Noise Immunity” (ANI)) which permanently adjusts various parameters at the receiver to mitigate the effects of interference. However, in [38, 42] the authors have shown, that the mechanism can cause undesirable side effects and caused the discovered performance impairments. Hence, we disabled this for all the measurements, since makes the reproducibility of a measurement studies hard. In addition, we also disable the Atheros’ proprietary “Transmit Antenna Diversity” algorithm.
References


