HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

ELEKTROTECHNIK

Student Research Project

802.11g Realtime Audio

Aleksandar Milosevic, Martin Lüssi

July 5, 2005

Supervisor: Fabian Meier, CTO G–Innovation "Geneva" GmbH

Abstract

The popularity of wireless local area networks has been constantly growing for the last few years. As the data rates supported are rising, new areas of application are explored. One such application is transmitting audio over the wireless network. Todays products, like Apple's Airport Extreme, designed for audio transmission over WLAN, show a relatively high latency. This latency is mainly induced by the lossy compression used and the jitter buffer of the receiver.

In this work, research is conducted to find ways to minimize the latency. For the design of a low latency system, it is important to understand the characteristics of the channel over which the audio packets are transmitted. Therefore, an extensive measurement campaign is carried out. For the measurements, a 802.11g test system is built. All measurements are conducted in an anechoic chamber and the influence of various kinds of interferers, like microwave ovens, is explored.

Analysis of the gathered data provides interesting results. Recommendations can be made about the length of the jitter buffer, packet size and the 802.11g data rate.

In addition, an audio transmission system is developed with the aim to minimize the latency introduced by the audio system of the computer.

Aleksandar Milosevic, Martin Lüssi

Contents

1. Introduction **1**
1.1. Motivation and Subject Description . . . 1
1.2. Original Task . . . 1

2. Low Latency Audio Transmission **5**
2.1. Connectionless vs. Connection-Oriented Protocol . . . 5
2.2. Packetization . . . 5
2.3. Compression . . . 6
2.4. Playout Buffer . . . 6

3. Introduction to 802.11g **9**
3.1. Medium Access Control . . . 9
3.1.1. The Distributed Coordination Function . . . 11
3.1.2. MAC Frame Formats . . . 13
3.2. Physical Layer . . . 15
3.2.1. Frequencies and Channels . . . 15
3.2.2. Operational Modes . . . 16
3.2.3. OFDM . . . 16
3.2.4. Data Rates . . . 17
3.2.5. PHY Frames . . . 18
3.3. Example Chipset . . . 20

4. Limiting Factors **23**
4.1. Electromagnetic Propagation . . . 23
4.2. Interference from ISM Band Devices . . . 24
4.3. Packetization Overhead . . . 24

5. Measurements **27**
5.1. Goals . . . 27
5.2. Measurement Setup . . . 27
5.3. Software . . . 29
5.3.1. UDP Sender & Receiver . . . 30
5.3.2. Packet Capturing At The Reference Receiver . . . 30
5.3.3. Offline Data Merging . . . 31
5.4. Scenarios . . . 31
5.4.1. Normal Signal . . . 32
5.4.2. Weak Signal . . . 32
5.4.3. Microwave Oven Interference . . . 32
5.4.4. Bluetooth Interference . . . 33

6. Data Analysis and Interpretation **35**
6.1. Data Analysis . . . 35
6.2. Interpretation of the Results . . . 36
6.2.1. Changing the Retry Limit . . . 36
6.2.2. Data Rate and Packet Size . . . 38

7. Recommendation for Audio Transmission System **41**

8. Proof of Concept Implementation **43**
8.1. Frame Formats . . . 43
8.2. Sender Application . . . 44
8.3. Receiver Application . . . 44
8.4. Review and further Development . . . 45

9. Conclusion **47**

A. Measurement Plots **49**
A.1. Scenario Normal Signal . . . 49
A.2. Scenario Weak Signal . . . 50
A.3. Scenario Microwave Oven . . . 51
A.4. Scenario Bluetooth (BTDH1) . . . 52
A.5. Scenario Bluetooth (BTDH3) . . . 53
A.6. Scenario Bluetooth (BTDH5) . . . 54

B. Various Information **55**
B.1. Tricking the Network Stack . . . 55

List of Abbreviations **57**

List of Figures **59**

List of Tables **61**

Bibliography **63**

1. Introduction

1.1. Motivation and Subject Description

Wireless local area networks are becoming popular for the transmission of audio data. Todays products like Apple's[1] Airport Extreme show high latency. Especially for audio and video applications where human perception plays a significant role, latency is annoying. An example is to see someone speaking and the movement of the lips appears not to be synchronized with the voice heard. A possible application of real time audio over WLAN could be a home entertainment system.

Because the audio data is transmitted over wireless a network like 802.11g, the communication gets undesired interference of devices operating in the same band. 802.11g shares the frequency band with Bluetooth devices and microwave ovens. Since many households use microwave ovens, the probability that it influences the 802.11g communication is high.

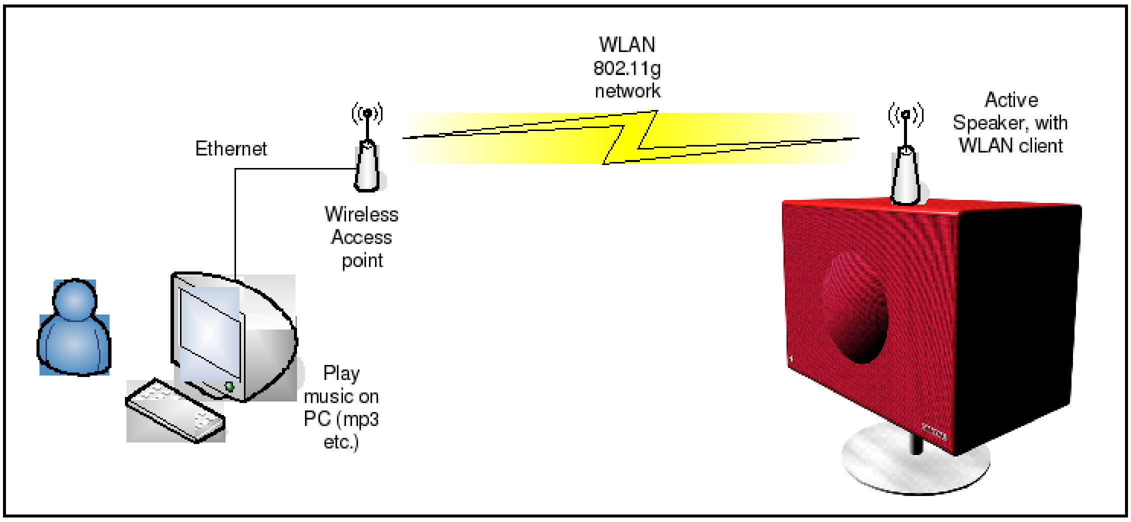
To be able to design a robust audio transmission systems, the channel must be understood. The undesired latency must be held as low as possible. Since the transmission is not only influenced by disturbers operating in the same band, factors like buffer size, packet length, path propagation etc. must be considered too.

In order to understand the channel as good as possible, theoretical and practical consideration must be made. Measurements must be conducted to simulate a transmission which is disturbed by other signals for example by spurious emission from microwave ovens.

1.2. Original Task

Description G-Innovation GmbH ("Geneva") in Zurich will release a wireless speaker product for the consumer market in summer 2005. There are plans to add a product on the same hardware platform that would address the semi-professional audio market (home studios) with a low latency wireless speaker product. This project will provide G-Innovation with fundamental data how feasible such a product is. Another reason is this project can identify where the problems are and how to solve them.

[1] http://www.apple.com



Theoretical Part

- Investigate how 802.11g WLANs work.
- Investigate what real-world WLAN problems exist (other WLANs, noise caused by mobiles, microwave ovens, etc) – describe the disturbances.
- Research existing and proposed protocols for real–time transfer.
- Propose a protocol (without audio compression), with features such as scalability, redundancy, latency control.

Practical Part

- Simulate the protocol in a simulator.
- Implement the protocol on a Linux or Windows platform using two PC's, write a sender and a receiver application.
- Add monitoring capabilities to measure and analyze latencies, jitter, packet loss throughput, etc.
- Come up with ways to make the wireless transmission "bad", i.e. drop packets, add latency to packets, create extra load on the real network.

Expected Results

- Documentation of theory, simulations and implementation.
- Simulation scripts/programs.
- C/C++ programs for PC (Linux or Windows).
- One laboratory journal each.

How to Work Each student has a personal laboratory journal. The results of the work carried out, how much time is spent for a particular task and when it is conducted is logged in this journal. Before a meeting, the students send an agenda to the supervisor. After a meeting the students send a protocol.

2. Low Latency Audio Transmission

To transmit real time audio over wireless networks there are a few things which have to be considered. One important factor is to keep the latency low as possible.

2.1. Connectionless vs. Connection-Oriented Protocol

Speaking of connectionless protocols may confuse. The meaning is that the protocol does not need to establish a connection with the recipient. The sender just sends the data to a destinatin host in hope the message will arrive. *User Datagram Protocol* (UDP) is an famous example for such a kind of protocola.

Connection-oriented protocols require a channel to be established before any message is sent. The most popular example is the *Transmission Control Protocol* (TCP). The establishment is performed by *synchronize/acknowledge* (SYN/ACK) handshake.

For situations where data must be received rapidly, connection-oriented protocols are not qualified. The reason is: For each packet sent an acknowledgment must be received, which means more latency. TCP for example needs an extra round trip to synchronize the TCP window sequence number. All modern TCP stacks implement *congestion control algorithms* (as specified in RFC 1122 and updated in RFC2001). This algorithms interpret packetloss as a hint that the network is congested. Reaction to that is slowing down the bit rate at which TCP sends the data. By reason of latency, connectionless protocols are preferred. In meny connectionless applications UDP has reached a big popularity.

One problem of connectionless protocols is not knowing if a packet has arrived or not (on the basis of the sequence number the receiver can determinate if a packet is missing or not). Solution to that can be sending redundancy, for example: A packet contains previously sent data and the actual one (quiet simple). An other approach is to detect the missing packet and to reconstruct it by a interpolated curve.

2.2. Packetization

Since there is always a fix size of header information in a packet, the transmitted data (information and header) cannot be constructed as small as desired. Thus, only the size of the information (audio data) can be varied. Header information belong to different protocols: *Internet Protocol* (IP), *Logical Link Control* (LLC) and for real time data *Real Time Protocol* (RTP) and of course UDP header information. Figure 2.1 shows a possible composition of an MAC Frame Header described in 3.1.2.1.

- Remark: RTP does not guarantee quality-of-service, it provides minimal control and identification functionality.

To reach low latency, the data must not be too large but it also can not be too short (sending useful data is still the aim). Sending large packets will increase the transmission time. An other point of view is the behavior of packet loss for large packets. Sending small

Size (Bytes):	8	20	8	20	N_A
	LLC	IP	UDP	RTP	Audio Data

Figure 2.1.: MAC Frame Body

packet sizes means that there is a lot of header information than audio, what does not make sense. But small packet sizes qualify for busy networks. Thus, audio communication is possible even if a low data transmission is present.

2.3. Compression

Todays consumer products which are deigned to transmit audio data over wireless LAN use some sort of lossy compression to limit the amount of data. The compression scheme used is similar to the famous MP3 (MPEG1 layer 3) compression scheme. Compression is useful because it reduces the amount of data and therefore relaxes the bandwith requirements. On the other hand, compression adds additional delay, which in case of low latency transmission is undesireable. How much delay is introduced by compression/decompression depends on the codec. When MP3 is used, the frame length is 26ms [1], the framing delay is therefore 26ms. Additional time is needed to compress the audio data at the sender and decompress it for playout. All together, the additional delay lies in the range of 40ms to 100ms. Some recent compression schemes, like the Frauenhofer ULD codec [2], are designed to keep the delay low (ULD stands for *Ultra Low Delay*). The delay introduced by such codecs is in the range of 10ms to 20ms. In general, the delay introduced by a lossy compression scheme is a tradeoff between compression quality (perceived quality / bit rate) and computional complexity.

When the audio transmission system is used as a virutal sound card, meaning a sound driver on the computer transmits the audio data to a remote device for playout, and additional problem exists. In order to conserve disk space, the music is stored in a compressed form (for example MP3) on the computers hard drive. To playback the music, it has to be decompressed and compressed again for transmission. Doing so requires additional computing power and degrades the quality. An other option is to transmit the audio data in the compressed form available from the hard drive. The problem with this approach is, that a variety of compression formats exist and it is impossible for the playout device to support all formats. Thus it limits the users freedom by requireing them to use certain supported formats.

When plenty of bandwith is available for transmission, it is possible to use no compression at all. This way, no additional delay is introduced and less computing power is needed. In addition, the audio frames can be made arbitrarily small without any loss in audio quality. Because of this, it is the goal of this project to research if transmission of uncompressed CD quality audio (44.1kHz, 16bit, stereo) over 802.11g is possible.

2.4. Playout Buffer

An other important part to keep latency low is the design of the buffer. To give the feeling that the data is played in real time, an small buffer size is needed. Even if the data is transmitted rapidly over the network, as long as it stays in the buffer the data is delayed. But to small buffer sizes lead to buffer overflows and packets get lost. Therefore, an ideal buffersize must be found that works even when a worse network connection exits. The synchronization between writing and reading to/from buffer is an other challenging part of buffer design. In mean, it must be read as fast as it was written to the buffer. This guarantees that there is always data available for output, otherwise the danger exists that no data is available (it is read faster than written) or data gets lost (bufferoverflow, it is written faster than read).

3. Introduction to 802.11g

This chapter provides an introduction to 802.11g technology. Since the original IEEE standards [6] [7] [8] are very lengthy, it can not cover every aspect of 802.11g. For a full reference, the reader is advised to consult the original standard documents.

Components of a 802.11 WLAN are stations and *Access Points* (AP). A station is a device containing a WLAN adapter, for example a laptop or a PDA. Multiple stations register with an AP and together they build a *Basic Service Set* (BSS). The BSS meant here is known as infrastructure BSS. In such a BSS, frames can only be exchanged between the AP and a station. When a station has to send a frame to an other station, the AP acts as a relay. The AP provides services such as authentication to control access to the network. In small networks, for example at home, the network is comprised only of stations and one AP. Larger networks however, need more then one AP because an AP can only cover a limited area. In this case, multiple BSS are linked together by a *Distribution System* (DS) and form an *Extended Service Set* (ESS). In an ESS a station can roam from one BSS to an other with handovers occuring transparently to the user.

Today, a variety of different 802.11 standards exists. Extensions like 802.11a or 802.11g describe the operating mechanism like encoding, bandwidth etc. Table 3.1 shows the currently available standards.

Standard	**Description**
802.11a	54Mbit/s - 5GHz-Band
802.11b	11Mbit/s - 2,4GHz-Band
802.11c	Wireless Bridging
802.11d	World Mode
802.11e	QoS and Streaming Extension for a/g/h
802.11f	Roaming According to IAPP for a/g/h
802.11g	54Mbit/s - 2,4GHz-Band
802.11h	54Mbit/s - 5 GHz-Band
802.11i	Authentication and Encryption for a/b/g/h

Table 3.1.: WLAN Overview

The IEEE 802.11g standard covers aspects of the *Data Link Layer* and the *Physical Layer* in the OSI/ISO reference model. An overview of the aspects covered and other IEEE standards involved is provided in Figure 3.1.

The functionality of the *Medium Access Control* (MAC) layer and *Physical Layer* (PHY) are described in the following. At the end of this chapter, an emxaple of a 802.11g chipset is shown.

3.1. Medium Access Control

As the name suggests, the MAC controls access to the medium. The functionality of the MAC can be divided into three categories:

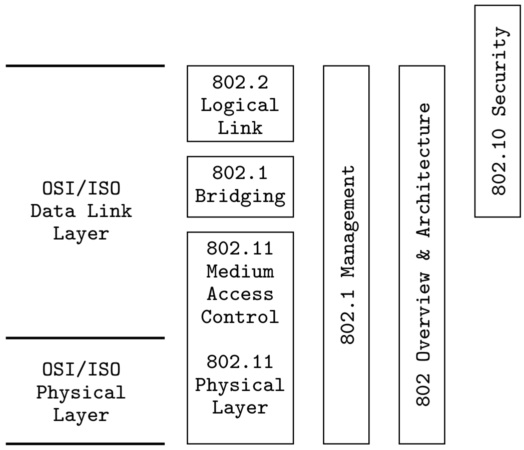


Figure 3.1.: Aspects of the OSI/ISO model coverded by 802.11

- Fairly control access to the medium. In 802.11 there are two different schemes that can be used for doing so. When the *Point Coordination Function* (PCF) is employed, the AP centrally controls access to the medium by assigning time slots to each station. The *Distributed Coordination Function* (DCF) on the other hand, needs no central coordinator. Because the PCF is rarely used, only the functionality of the DCF will be described in this section.

- Provide reliable transmission of frames to the users of the MAC. Wireless links are prone to errors. In order to provide a reliable transmission to the user, 802.11 can resend a frame when it was corrupted. This process works transparently to all higher level protocols.

- Protect the data that are delivered. Because wireless networks can not be contained to a single room or office, this is very important. Without encryption an attacker would be able to receive the data and possibly steal sensitive information. The protection mechanism provided by 802.11 is *Wired Equivalent Privacy* (WEP), a more secure technique is *Wi-Fi Protected Access* (WPA) which is not part of the standard but supported by many devices. For real time audio transmission, security is not one of the key problems, therefore the protection mechanisms provided by 802.11 will not be further described in this report.

For an IEEE 802.11 device it is not possible to transmit and receive simultaneously. For this reason, the *Carrier Sense Multiple Access / Collision Detection* (CSMA/CD) scheme employed by wired ethernet (IEEE 802.3) can not be used to detect frame collisions. For wireless LAN, a different strategy to avoid and detect frame collisions is needed, the one used by 802.11 is known as *Carrier Sense Multiple Access / Collision Avoidance* (CSMA/CA). CSMA/CA is a combination of a physical and a virtual carrier sense mechanism. Before a frame is transmitted, the medium has to be idle for a specified amount of time. This is the functionality of the physical carrier sense mechanism. For virtual carrier sense, the MAC implements a *Network Allocation Vector* (NAV). The NAV indicates the time until the medium becomes idle. To obtain this information, a 802.11 station always listens for frames of other stations and updates its NAV according to the duration values indicated in the PHY frame headers received. Unfortunately, in a wireless network it is possible that not every station is able to receive the frames of every other station, because the stations are located too far apart. This leads to the so called hidden node problem. To combat this problem, the 802.11 MAC supports the *Request To Send / Clear To Send* (RTS/CTS) mechanism (its functionality is described below). Finally, when a station successfully receives a frame, it indicates this to the sender by sending an acknowledge (ACK) frame. The frame sent and the following ACK build an atomic unit in the 802.11 MAC and can not be interrupted by other frame transmissions. When no ACK is received by the sender, it assumes that the frame has been lost during the transmission. In this case, the frame is resent. In order to avoid the occupation of the medium by one station sending the same frame again and again, the number of times a frame can be resent is limited.

In the rest of this section, the DCFs functionality and the MAC frame formats are described.

3.1.1. The Distributed Coordination Function

The DCF allows to fairly share the medium without the need of any central coordination. Two different modes can be employed, the RTS/CTS mode is used when hidden nodes are present and for sending large frames. Basic access does not provide this kind of protection but incurs also less overhead because fewer frames have to be exchanged. Which mode is used, depends on the frame length. When the frame is shorter or equal than dot11RTSThreshold, basic access is employed, for longer frames RTS/CTS. The default value of dot11RTSThreshold is 2347 Bytes for 802.11g (and also for 802.11b and 802.11).

3.1.1.1. Basic Access

Basic access employs CSMA/CA together with a binary exponential backoff mechanism. When a frame has to be sent, physical and virtual carrier are used to determine if the the medium is currently in use. If either of these mechanisms indicate the medium to be busy, binary exponential backoff is employed, otherwise the frame is transmitted. The binary exponential backoff chooses a random number, the *Contention Window* (CW), which represents the time the medium must be idle until transmission is attempted again. The range of the random number is doubled every time the transmission fails, from the initial value (aCWmin) up to the upper limit (aCWmax), which are 31 and 1023 for 802.11g respectively. The increase of the range for CW is shown in Figure 3.2.

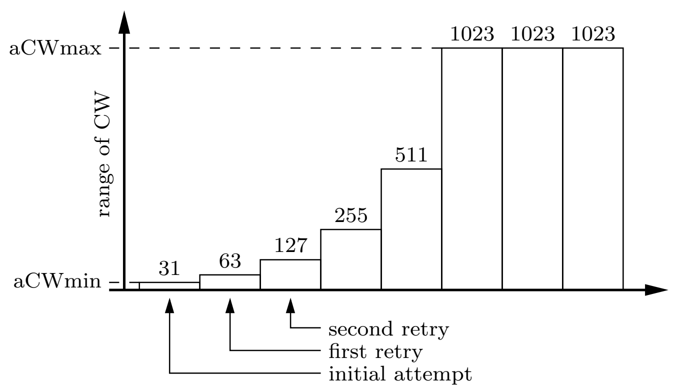


Figure 3.2.: Binary exponential backoff algorithm

In addition to this, the retry counter associated with short frames is increased. The maximal number of transmission attempts is defined by the dot11ShortRetryLimit, which has a default value of 7 for 802.11g. Finally, when the transmission is either successful or the maximal number of retries is reached, the retry counter and the value of CW are reset to their initial values. A transmission can only be successful when an ACK from the destination has been received.

The time intervals and their values for 802.11g are given in table 3.2.

Interval	**Duration**
Slot Time (SLOT)	9µs (short), 20µs (long)
Short InterFrame Space (SIFS)	10µs
DCF InterFrame Space (DIFS)	SIFS + 2SLOT

Table 3.2.: DCF timing intervals

In addition to the intervals shown here, an other interval, the *Extended InterFrame Space* (EIFS) exists. EIFS is needed to recover from PHY error when a frame is sent, its value depends on the current data rate. For 802.11g, two different SLOT times exist, 20µs are needed when 802.11g and 802.11b devices are operated in the same BSS. The short SLOT time (9µs) is employed for an 802.11g-only BSS.

When a 802.11g station tries to send a frame, the medium has to be idle for the duration of DIFS (physical carrier sense). If the medium appears to be busy, CW is calculated by the binary exponential backoff algorithm explained above. During the deferral period, a backoff counter, with an initial value of CW, is decremented by one every time the medium is sensed to be idle. The physical carrier sense during this period takes place every SLOT. If the medium is sensed busy, the device has to wait for DIFS until the backoff procedure can be resumed. Finally when the backoff timer reaches zero, transmission of the frame is attempted again. When employing this procedure, it can take a very long time until the backoff counter reaches zero because the medium is sensed to be busy all the time. To avoid this, the 802.11 MAC limits the total time allowed for the transmission of a frame.

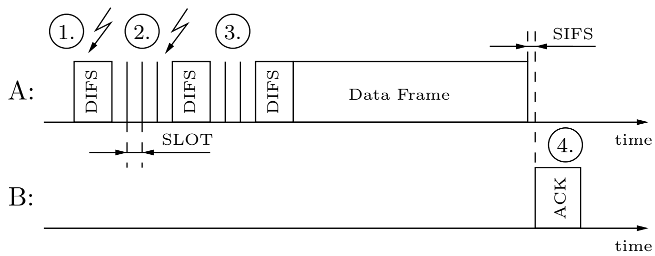


Figure 3.3.: Basic access mechanism

In Figure 3.3 an example for the basic access mechanism is shown. In this scenario, A tries to send a frame to B, the events that happen are the following:

1. The NAV of A indicates an idle medium, therefore A senses the medium physically. The physical carrier sense mechanism signals a busy medium. Binary exponential backoff is employed which chooses a CW of 7.
2. During the deferral period, A senses the medium every SLOT and decrements its backoff counter. The fourth time, the medium appears to be busy. A has to wait for DIFS until it can proceed.

3. Finally, the backoff counter reaches zero. A waits for an additional DIFS, during which physical carrier sense indicates an idle medium. After the DIFS, A starts sending the data frame.

4. B has successfully received the frame. After a SIFS, B sends an ACK frame back to A. A receives the ACK and the transmission of the data frame is completed.

3.1.1.2. RTS/CTS

When RTS/CTS is used, the source station sends a *Request To Send* (RTS) frame to destination. If the destination is ready to receive the frame, it answers with a *Clear To Send* (CTS) frame. After the reception of CTS, the source can start to transmit the data frame. As with basic access, the destination sends an ACK when the data frame has been successfully received. If the frame exchange procedure fails at any point, the binary exponential backoff algorithm described in 3.1.1.1 is used. Frames sent with RTS/CTS have a different retry limit, the dot11LongRetryLimit, it has a default value of 4.

In a BSS, it is assumed that every station is able to receive frames sent from the AP. Because the AP participates in every frame exchange in the BSS, it always sends either a RTS or a CTS frame. The RTS/CTS frames contain a duration field indicating the time required for the upcoming frame transmission. This allows all stations in the BSS to update their NAV and refrain from starting a transmission during this period. A frame exchange using RTS/CTS is shown in Figure 3.4. Between the frames a SIFS is inserted.

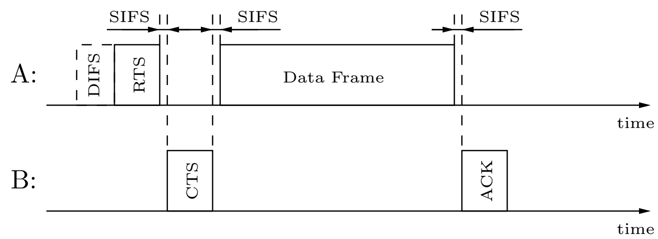


Figure 3.4.: RTS/CTS mechanism

The RTS/CTS scheme is beneficial when hidden nodes are present and for transmitting long frames. For short frames however, the additional frame exchange adds a significant overhead.

3.1.2. MAC Frame Formats

The 802.11 MAC describes various frame types which are grouped into three categories:

Management Frames This frames are used for the management of the BSS. One example is the beacon frame which is sent periodically by the AP to announce the presence of the BSS.

Control Frames Control frames are frames such as CTS, RTS and ACK.

Data Frames Data frames are the frames used to transmit higher level protocol data.

In this section, only some of the most common frame types are explained.

3.1.2.1. Data Frames

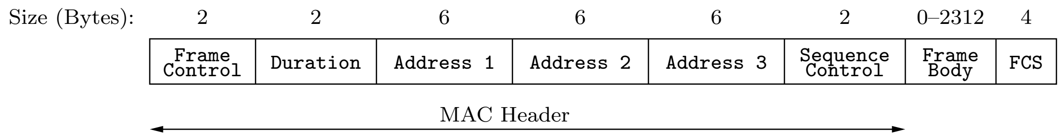


Figure 3.5.: MAC data frame

The data frame shown here is used in an infrastructure BSS. When a wireless DS is in use, data frames have an additional address field (Address 4). The functions of the different fields are explained in the following:

Frame Control The Frame Control field consists of several subfields. Information about the type of the frame, the encryption in use (WEP), the fragmentation of the frame (see below) and if it is the first attempt to transmit the frame is provided in the subfields.

Duration This field contains information about the duration of the ongoing transmission, it is used for updating the NAV.

Address Fields A data frame has three address fields. Each address field can hold a 48bit MAC address (IEEE 802-1990). Three addresses are needed, because the AP acts as a relay. An explanation of the contents of the address fields is given in 3.3. In this scenario, the BSS consits of two stations (A, B) and an AP (AP).

Source	Destination	Address 1	Address 2	Address 3
AP	A	MAC(A)	MAC(AP) = BSSID	MAC(AP) = BSSID
A	AP	MAC(AP) = BSSID	MAC(A)	MAC(AP) = BSSID
A	B	MAC(AP) = BSSID	MAC(A)	MAC(B)

Table 3.3.: Address fields

The MAC address of the AP is also known as *Basic Service Set ID* (BSSID). The first address field always contains the MAC address of the station (or AP) which is meant to receive the frame. The second address field holds the address of the sending station (or AP). The third address field can either contain the destination or the source MAC address. When A wants to send a frame to B for example, it fills Address 3 with MAC(B) (destination) and sends the frame to the AP. The AP then relays the frame by sending it to B, Address 3 now holds MAC(A) (source). When the AP is connected to wired ethernet, Address 3 can also hold the address of a device on the wired side. In this case, the AP acts as a bridge between the wired and the wireless network.

Sequence Control This field holds a 4bit fragment number and a 12bit sequence number, they are needed to decide if the same frame has been received multiple times.

Frame Body All higher level protocol data handed over to the MAC for transmission is embedded in the frame body field. When WEP is disabled, it can hold up to 2312Bytes of data. With WEP enabled 2304Bytes. The maximal size of 2312Bytes allows the transmission of 2048Bytes user data together with 256Bytes higher level protocol headers. When the size of the frame is larger than 2312Bytes (or larger than aFragmentationThreshold), the frame is fragmented. Fragmentation complicates the frame exchange procedure and is only useful for large frames, therefore it shall not be explained here.

FCS In order to decide if a frame has been correctly received, the receiver calculates a 32bit *Cyclic Redundancy Check* (CRC) number, known as *Frame Check Sequence* (FCS), from the data of the MAC header and the frame body. The calculated FCS is compared with the FCS sent with the frame. If they match, the possibility for errors is very small and the frame is handed over to the higher layer.

3.1.2.2. Control Frames

Size (Bytes):	2	2	6	4
ACK:	Frame Control	Duration	RA	FCS

Size (Bytes):	2	2	6	6	4
RTS:	Frame Control	Duration	RA	TA	FCS

Size (Bytes):	2	2	6	4
CTS:	Frame Control	Duration	RA	FCS

Figure 3.6.: MAC control frames

In an ACK frame, the RA field holds the MAC address of the station the previous data frame has been received from. The RA field of an RTS frame holds the MAC address of the destination, the TA field the MAC address of the sending station. CTS is sent in response to a RTS, the contents of the RA field are copied from the previously received RTS TA field.

To limit the overhead incurred by the exchange of control frames, their size is kept small. At high data rates however, the size of a frame has only a limited impact on the time needed to transit the frame. This is explained in the next section.

3.2. Physical Layer

When a packet has to be transmitted, the MAC layer finally hands it over to the 802.11g *Physical Layer* (PHY). The PHY is where the data are transmitted over the wireless medium. 802.11g uses a modern technique called *Orthogonal Frequency Division Multiplexing* (OFDM), the use of OFDM leads to the high data rates (up to 54Mbit/s) supported by 802.11g.

Operating in the ISM band at 2.4GHz, 802.11g shares the medium with other unlicensed devices, for example Bluetooth. This leads to unwanted interference which can cause packet loss. Effects such as multipath propagation may lead to degradation of the signal quality and the high data rates required for real time audio can not be achieved.

3.2.1. Frequencies and Channels

Like many other wireless standards, 802.11g WLANs are operated in the ISM band at 2.4GHz. The ISM band is divided into 14 channels with a spacing of 5MHz, ranging from

2.412GHz to 2.483GHz. How many channels can be used depends on national regulations, in most parts of Europe channels 1 to 13 are available. Since the bandwith required for a 802.11g wireless LAN is bigger than the channel spacing, not all channels can be assigned to a network at the same location. For noninterfering operation, the occupied channels have to be located at least 25MHz apart. This reduces the maximal number of networks at the same location to 3 (one on channel 1, one on channel 6 and one on channel 11). The center frequencies of the channels and the spacing requirement are shared by all 802.11 standards operating at 2.4GHz and are therefore not unique to 802.11g.

3.2.2. Operational Modes

Devices implementing the 802.11g standard support different modes of operation. Some of them make 802.11g fully backward compatible with 802.11 and 802.11b. The modes are described in the following.

ERP-DSSS In this mode data is transmitted using a technique called *Direct Sequence Spread Spectrum* (DSSS). ERP-DSSS is required for backward compatibility with 802.11 devices supporting data rates of 1 and 2Mbit/s.

ERP-CCK The ERP-CCK mode is used for compatibility with 802.11b devices. CCK stands for *Complementary Code Keying*, the data rates supported are 5.5 and 11Mbit/s.

ERP-PBCC This mode is optional and rarely used. PBCC means *Packet Binary Convolutional Coding* and is used together with DSSS. The data rates achieved by this mode are 5.5, 11, 22 and 33Mbit/s.

DSSS-OFDM As the the name suggests, a combination of DSSS and OFDM is used in this mode. DSSS is employed for transmitting the header of a PHY frame. Doing so allows 802.11b devices to receive this information and update their NAVs accordingly. Therefore 802.11b devices and 802.11g devices can be operated in the same network (BSS). The actual data is OFDM modulated and can not be received by 802.11b devices. The data rates are 6, 9, 12, 18, 24, 36, 48, and 54Mbit/s.

ERP-OFDM In this mode all the data is sent by OFDM and can only be received by 802.11g devices. It is therefore known as 802.11g-only mode. The data rates are also 6, 9, 12, 18, 24, 36, 48, and 54Mbit/s.

Since the transmission of uncompressed CD-quality audio requires high data rates, only DSSS-OFDM and ERP-OFDM are viable modes. As will be shown later, ERP-OFDM is better suited for transmitting small packets at high rates, which is characteristic for low-latency audio transmission.

3.2.3. OFDM

OFDM is a relatively new technique and can in some ways be regarded as a channel access scheme or a modulation scheme [4]. By the use of the *Inverse Discrete Fourier Transform* (IDFT) a number of equally spaced subcarriers is created. These subcarriers are distributed over the available bandwith W and are orthogonal to each other. Orthogonality makes OFDM a very bandwith efficient technique because no guard bands between the subcarriers are needed. Information is transmitted by modulating each subcarrier by a conventional digital modulation scheme like *Quadrature Amplitude Modulation* (QAM).

An other advantage of OFDM is the small bandwith of a single subcarrier, it is smaller than the coherence bandwith B_{coh} of the channel. This leads to a simplified channel eagalisation in the receiver because no frequency selective fading occurs within the bandwith occupied by an individual subcarrier. In addition, the OFDM symbol duration T_{sym} has to be smaller than the coherence time T_{coh} of the channel, which characterizes its time invariance. The number of subcarriers N_{c} is chosen to satisfy Eq. 3.1 (Source : [4]).

$$\frac{W}{B_{\mathrm{coh}}} \ll N_{\mathrm{c}} \ll W T_{\mathrm{coh}} \tag{3.1}$$

For 802.11g the number of subcarriers is 52, 4 of them are pilot channels and therefore 48 subcarriers are available for data transmission ($N_{\mathrm{c}} = 48$). The subcarrier spacing is 312.5kHz, this leads to a total bandwith of about 16.25MHz. The OFDM symbol duration T_{sym} is 4μs. Having this information, it is straightforward to calculate the data rate as will be shown in 3.2.4.

3.2.4. Data Rates

As mentioned before, the ERP-OFDM and DSSS-OFDM modes allow data rates up to 54MBit/s. This data rates however are raw data rates, at the application layer the achievable throughput is only a fraction of the raw rate. Table 3.4 shows the data rate, the kind of modulation and the code rate the different modes employ.

Mode	**Modulation**	**Code Rate R_{c}**	**Data Rate R_{d}**
1	BPSK	1/2	6Mbit/s
2	BPSK	3/4	9Mbit/s
3	QPSK	1/2	12Mbit/s
4	QPSK	3/4	18Mbit/s
5	16QAM	1/2	24Mbit/s
6	16QAM	3/4	36Mbit/s
7	64QAM	2/3	48Mbit/s
8	64QAM	3/4	54Mbit/s

Table 3.4.: 802.11g data rates

In order to transmit more bits per symbol the complexity of the applied subcarrier modulation increases with the data rate. In Figure 3.7 the constellations of the different modulation schemes are shown.

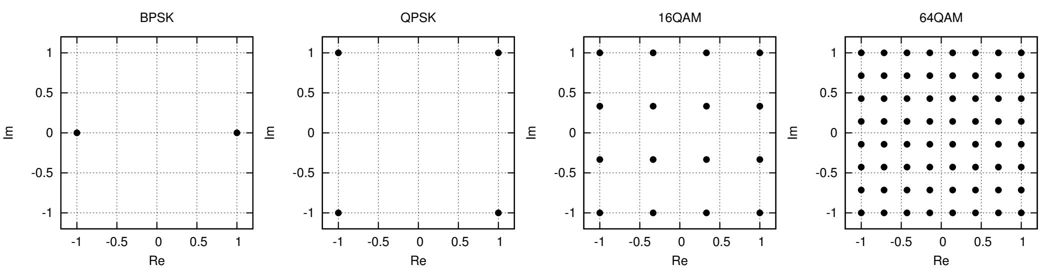


Figure 3.7.: Constellations of the different modulation schemes

It is apparent that a complex modulation like 64QAM needs a much higher SNR than for example BPSK to be successfully received. Figure 3.8 shows the theoretical *Bit Error Rates* (BER) assuming the channel is AWGN (since no frequency selective fading occurs for an individual subcarrier this is a reasonable assumption). For the calculation formulas given in [4] were used.

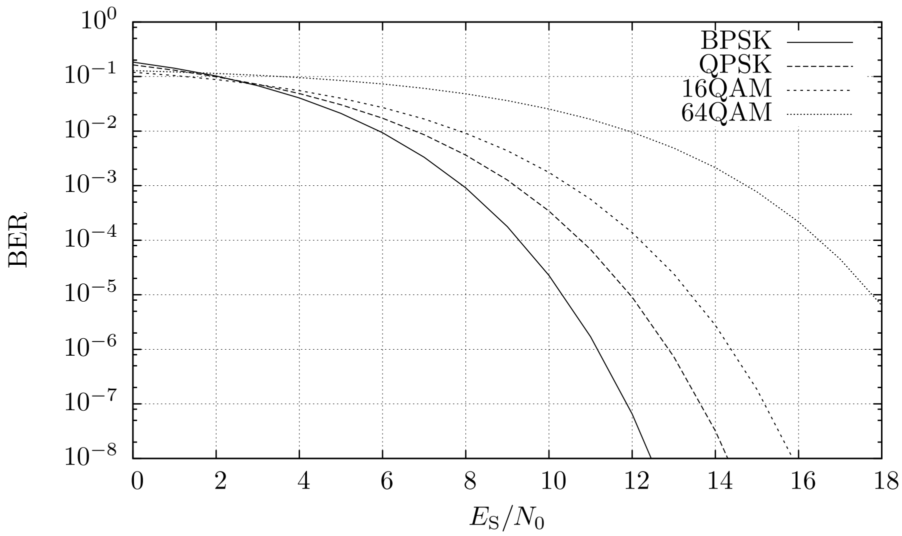


Figure 3.8.: Bit error rates of the modulations used by 802.11g

To make the transmission more resilient to errors, convolutional coding is employed. The code rate R_c in Table 3.4 describes how many data bits are required to build a number of transmission bits. A R_c of $3/4$ for example means that for 3 data bits 4 bits are transmitted. The data rate for each mode can now be calculated using Eq. 3.2.

$$R_d = \frac{1}{T_{sym}} N_c R_c N_{bit} \tag{3.2}$$

N_{bit} stands for the number of bits transmitted per symbol and depends on the subcarrier modulation. For QAM64 it is 6 (in general the logarithm to the power of 2 of the number of symbols available for a constellation in Figure 3.7).

3.2.5. PHY Frames

In order to transmit packets over the wireless link, MAC frames are encapsulated into PHY frames. A PHY frame consists of a preamble, a header and the modulated MAC data. Each mode of operation (see Chapter 3.2.2) has its specific PHY frame definition. For both DSSS-OFDM and ERP-OFDM the payload is OFDM modulated, the differences lie in the way the header information and the preamble are transmitted.

3.2.5.1. DSSS-OFDM Frames

When a 802.11g device sends a MAC frame in DSSS-OFDM mode, the header information is transmitted by employing DSSS modulation. Doing so makes concurrent operation of 802.11b and 802.11g devices in the same BSS possible. The structure of the MAC frame is given in Figure 3.9.

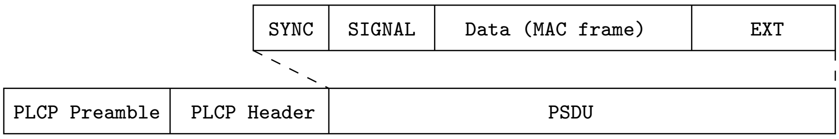


Figure 3.9.: DSSS-OFDM PHY frame

Two different submodes for DSSS-OFDM exist, the "short preamble" and the "long preamble" mode. The kind of modulation used and the transmission times required for the different frame fields are shown in Table 3.5. The time needed for the DATA field depends on the size of the encapsulated MAC frame The SYNC and SIGNAL fields require two and one OFDM symbol respectively.

Mode	Field	Modulation	Data Rate	Bits	transmit time
short preamble	PLCP Preamble	DBPSK (DSSS)	1Mbit/s	72	72 µs = T_{pre}
short preamble	PLCP Header	DBPSK (DSSS)	2Mbit/s	48	24 µs = T_{head}
short preamble	SYNC	OFDM			8 µs = T_{sync}
short preamble	SIGNAL	OFDM			4 µs = T_{sig}
short preamble	Data	OFDM	R_{d}		
short preamble	EXT	OFDM			6 µs = T_{ext}
long preamble	PLCP Preamble	DBPSK (DSSS)	1Mbit/s	144	144 µs = T_{pre}
long preamble	PLCP Header	DBPSK (DSSS)	1Mbit/s	48	48 µs = T_{head}
long preamble	SYNC	OFDM			8 µs = T_{sync}
long preamble	SIGNAL	OFDM			4 µs = T_{sig}
long preamble	Data	OFDM	R_{d}		
long preamble	EXT	OFDM			6 µs = T_{ext}

Table 3.5.: DSSS-OFDM frame fields

From the figures given, the total time required to transmit a frame in DSSS-OFDM mode can be calculated (Source :[8]):

$$T_{\text{tx}} = T_{\text{pre}} + T_{\text{head}} + T_{\text{sync}} + T_{\text{sig}} + T_{\text{sym}} \cdot \left\lceil \frac{22 + 8N_{\text{mac}}}{N_{\text{dbps}}} \right\rceil + T_{\text{ext}} \tag{3.3}$$

N_{dbps} denotes the number of data bits per OFDM symbol and N_{mac} the size of the MAC frame (in Bytes). The additional 22 bits are needed for the SERVICE and PAD fields not shown here. Knowing the number of subcarriers N_{c}, the number of bits per subcarrier symbol N_{bit}, the OFDM symbol duration T_{sym} and the code rate R_{c}, Eq. 3.3 can be written as:

$$T_{\text{tx}} = T_{\text{ovh}} + 4\mu s \cdot \left\lceil \frac{22 + 8N_{\text{mac}}}{48 N_{\text{bit}} R_{\text{c}}} \right\rceil \tag{3.4}$$

whereas T_{ovh} is 114µs for the "short preamble" mode and 210µs for the "long preamble" mode.

3.2.5.2. ERP-OFDM Frames

In ERP-OFDM mode, the whole PHY frame is OFDM modulated. Except from the carrier frequencies this mode is identical to the transmission scheme employed by 802.11a. To

PLCP Preamble	SIGNAL	Data (MAC Frame)

Figure 3.10.: ERP-OFDM PHY frame

Field	Modulation	Data Rate	Bits	transmit time
PLCP Preamble	OFDM			16 µs = T_{pre}
SIGNAL	OFDM	6Mbit/s	24	4 µs = T_{sig}
Data	OFDM	R_{d}		

Table 3.6.: ERP-OFDM frame fields

send a frame in this mode, less time is required than in DSSS-OFDM mode. The PHY fields of an ERP-OFDM frame are shown in 3.10.

The transmission time for a PHY frame in ERP-OFDM mode can now be calculated:

$$T_{\text{tx}} = T_{\text{pre}} + T_{\text{sig}} + T_{\text{sym}} \cdot \left\lceil \frac{22 + 8N_{\text{mac}}}{N_{\text{dbps}}} \right\rceil = 20\mu s + 4\mu s \cdot \left\lceil \frac{22 + 8N_{\text{mac}}}{48 N_{\text{bit}} R_{\text{c}}} \right\rceil \tag{3.5}$$

3.3. Example Chipset

In wireless technology, Broadcom Corporation provides common chipsets. The leader of wireless communications semiconductors supplies also the chipset for the Linksys Wireless - G Broadband Router (WRT54G) used in this work.

A chipset includes all mentioned parts, the MAC (Chapter 3.1), the PHY (Chapter 3.2) etc. Figure 3.11[1] shows the block diagram with the BCM2050 and BCM4309 chips – The used access point comes also with the BCM2060 chip. It has the same functionality as BCM2050 but for the 802.11a standard.

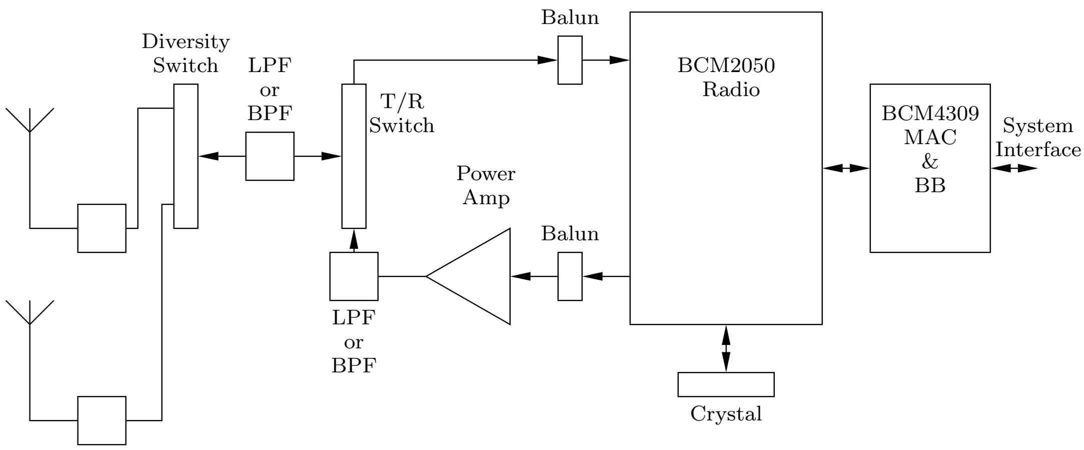


Figure 3.11.: Block Diagram

The roles of the shown components in Figure 3.11 is the following [3]:

- BCM4309 implements the baseband part and the MAC. Timing-critical funcitions are handled by the MAC within microseconds of an event or at precise intervals.

[1] modified from [3]

Examples are the channel access mechanism such as random backoff and listen before talk, checksum generation and verification and hardware-level frame retry (non timing-critical functions are performed on the host CPU in the driver software for example frame fragmentation and defragmentation, frame buffering etc.). The baseband transceiver implements the 802.11g physical layer function. An other part of the MAC and baseband is the host interface unit (HIU) that provides connectivity to the host processor over a PCI bus.

- The data received from the BCM4309 chip is passed for transmission to the BCM2050. As it is shown in figure 3.12, BCM2050 is also responsible for the reception of the signals (data). Thus, the modulation and demodulation is done by it.
- The diversity switch achieves to receive the strongest signal available, there are two signals received in this example (two antennas). Since the access point can be placed anyhow, the use of two (or more) antennas helps to receive a better signal. – The comming generation of WLAN products will use multiple input multiple output (MIMO) devices.

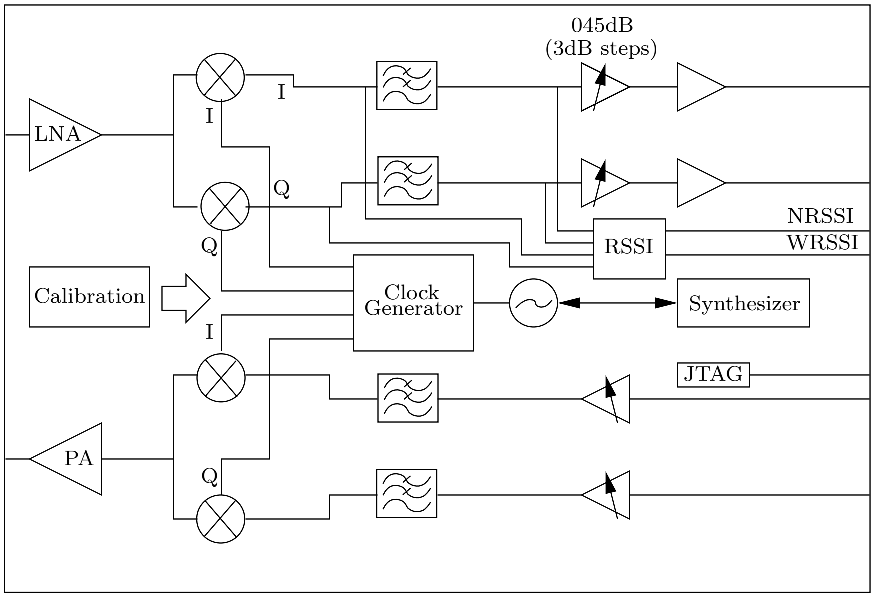


Figure 3.12.: BCM2050 Block Diagram. Source: [3]

Table 3.7 shows the data sheet [3] of the BCM2050 chip.

Parameter	**Value**
NF	4dB typ
Receiver IIP3 (max. gain)	-16dBm typ
Receiver IIP3 (min. gain)	4dBm typ
Transmitter output power	5dBm typ
Transmitter OIP3	18dBm
Transmitter output power range	5dBm to -15dBm typ
Transmitter EVM	-27dB min. at 54Mbit/s
Receive-mode current consumption	110mA typ (1.8V)
Transmit-mode current consumption	80mA typ (1.8V)
V_{dd}	1.8V

Table 3.7.: BCM2050 Specifications

4. Limiting Factors

When a wireless network is used to transport audio data, certain limiting factors exist which are not present in a wired network. Electromagnetic propagation leads to a degradation of signal quality. In indoor enviroments, where 802.11g networks are used, multipath effects are a major problem. Those effects are caused by signal reflections from walls, ceiling and other objects. An other problem are other devices operating in the ISM band. Such devices are for example Bluetooth devices but also microwave ovens which emit spurious radiation. In addition, the 802.11g PHY adds a significant overhead when small packets are sent. In this chapter, electromagnetic propagation is reviewed and the packetization overhead incured by the 802.11g MAC and PHY is outlined.

4.1. Electromagnetic Propagation

As the name it already says the wireless is a kind of communication without needing a physical wiring. This means not that a wired technique is better. Starting from a certain distance, the loss of a signal is growing more for a wired environment unlike for wireless.

For better understanding of loss in wireless communication systems the consideration of a free-space environment simplifies the topic (Source: [4]). Its loss can be computed as follows:

$$L = \left(\frac{4\pi r f}{\mathrm{c}}\right)^2 \tag{4.1}$$

or in logarithm,

$$L[dB] = -147.6 + 20\log_{10}(r) + 20\log_{10}(f) \tag{4.2}$$

where r is the distance, f the frequency and c the velocity of light.

As we can see the free-space loss grows with r^2 (in logarithmic form $20\log(r)$). This exponent "2" is called the path-loss exponent. For indoor environment this exponent has an other value. Table 4.1 gives an overview of the different path-loss exponents for a given environment.

Environment	Path-Loss Exponent
free space	2
open field (long distance)	4
cellular radio, urban area	2.7 - 4
shadowed urban cellular radio	5 - 6
in building, line-of-sight	1.6 - 1.8
in building, obstructed	4 - 6

Table 4.1.: Path-loss exponents for different environments. (Source: [4]).

It is quiet difficult to say how a signal is affected in a room. A lot of factors must be considered: the walls and its material, path propagation, antenna position, reflexion, scattering, diffraction, and absorb radiation.

Referring to [5], it is well-known that objects whose dimensions are small compared to a wavelength act as weak scattering centers. Objects much larger than a wavelength can be treated by the familiar approach of phase-insensitive ray tracing learned in classical optics. The dominant scatterers in indoor environments fall into neither of these simplistic categories: their typical sizes are from two to three centimeter to human dimensions of two to three meters vs. wavelengths of roughly 12cm in the considered ISM band. For indoor propagation, experiments have shown that the avarage values fall as much as 30dB below ideal free-space propagation.

4.2. Interference from ISM Band Devices

Other devices operating in the unlicensed ISM band can interfer with the 802.11g WLAN. Prominent interferers are microwave ovens, Bluetooth devices and other wireless LANs. The characteristics of interference caused by Bluetooth devices and microwave ovens are explained in Chapter 5.

4.3. Packetization Overhead

The time needed to transmit a frame in ERP-OFDM and DSSS-OFDM mode has been shown in 3.2.5 and can be calculated by the means of Eq. 3.5 and Eq. 3.4 respectively. Headers are transmitted together with the audio data, when UDP/RTP the size of the header information is 84Bytes (RTP, UDP, IP, LLC, MAC). In order to achieve low latency, small packets are sent. In this case the headers and preambles cause a significant overhead. The occupation of the medium is related to the time required to transmit a frame. When audio data is transmitted at regular intervals t_{int}, the time available for transmission is smaller than t_{int}. In reality, additional time is needed to sense the medium, resend the frame and send the ACK frame. The number of times a frame needs to be resent is dificult to predict because it depends on various interference effects which are stochastic. In Figure 4.1 the time needed to transmit audio data of a certain size N_{A} together with 84Bytes header information is shown.

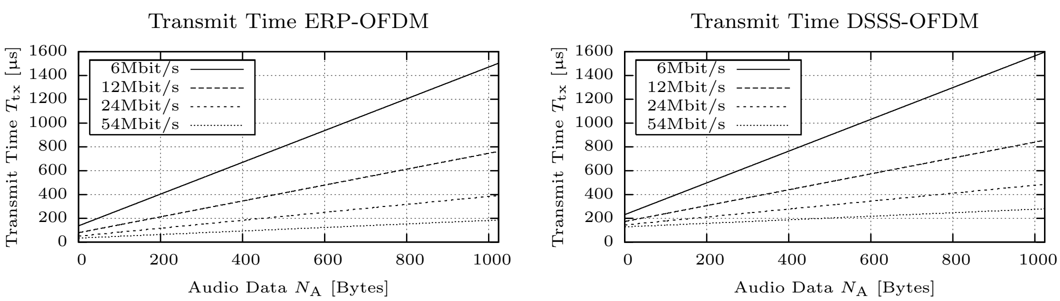


Figure 4.1.: Transmit time for N_{A} Bytes audio data

The difference between the transmit time needed in DSSS-OFDM mode and in ERP-OFDM mode is 94µs. For small packets this can be a significant difference in the overhead incured. For low latency audio transmission it is therefore recomendable to use the ERP-OFDM mode (802.11g–only mode).

5. Measurements

In order to implement an efficient and robust audio transmission scheme, it is important to understand the channel over which the packetized audio data is sent. In this project, the channel is an 802.11g wireless network. During the transmission, the network can lose packets or delay their delivery. Unlike in internet telephony, where audio packets are sent over a multi hop network, the network in this work consits of only few hops. One of these hops however, is a wireless link. When the internet is used for packet transmission, loss mainly occurs because packets are dropped by congested routers and delay is induced by the time the packets stay in router queues. In a 802.11g wireless LAN on the other hand, loss and delay occur because of interference caused by other devices and weak signals. In other words, the underlying process, which leads to packet loss and delay, is completely different from that of the internet. For audio transmission, this means that schemes implemented for the internet are not certain to work well with wireless LAN.

For this reason, a series of measurements are conducted. In this chapter, the goals of the measurements, the setup and the different scenarios are explained.

5.1. Goals

The goal of the measurements is to obtain the following data:

One Way Delay Packets are sent over the wireless link in one direction. The packet contains a timestamp from the sender, having this information the receiver can calculate the one way delay.

Packet Loss Each packet contains a sequence number, this allow the receiver to decide if a packet has been lost.

Number of Retries The 802.11 MAC can resend a packet multiple times. The number of times a packet is resent can have a direct influence on the delay of the packet.

For each measurement, a number of packets are sent over the wireless link. The parameters above are saved for each packet. Later, offline analysis of the data can be performed to obtain information such as burst loss statistics.

5.2. Measurement Setup

A common problem when measuring the one way delay of packets, is time synchronization of the sender and the receiver. Synchronization is required because absolute times from sender and receiver are needed to calculate the one way delay. Synchronization techniques are complicated and are an additional source for errors. To avoid these errors, it is better to use the same computer to send and receive the packets. This way, both the sending and the receiving application use the same clock and synchronization is no problem. When conducting measurements with wireless LAN, the environment influences the results significantly. It is therefore important to have a controlled environment, where no random events such as people moving around happen. Otherwise the results of the different measurements can not be compared. In order to provide such an environment, all measurements are conducted in an anechoic chamber. The chamber is constructed of metal and blocks radiation coming from the outside. Inside, absorber pyramids prevent the reflection of microwaves from the walls. The multipath effects, which are a major problem for wireless LAN, are therefore not present in the chamber. In Figure 8.1 the top view of the setup is shown.

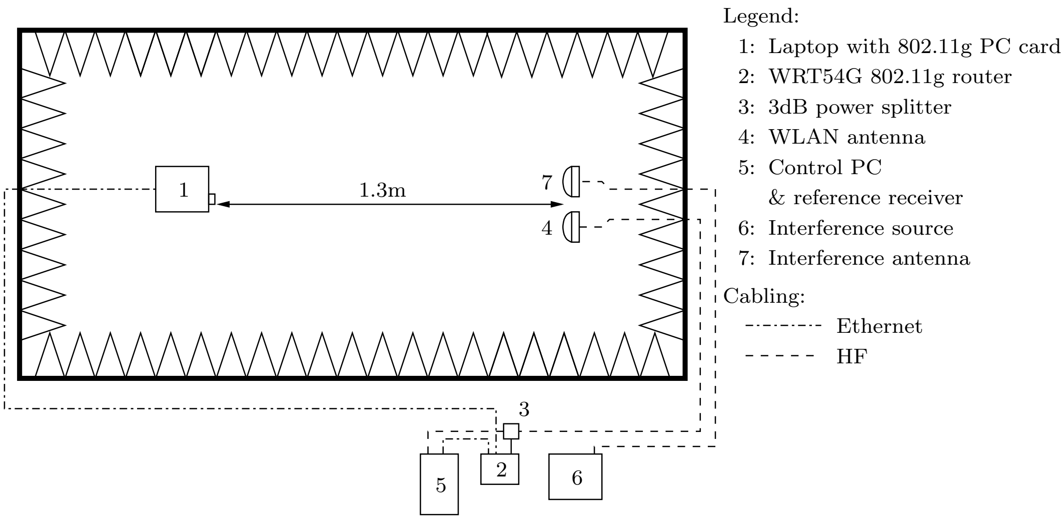


Figure 5.1.: Measurement setup

The tasks of the different parts of the setup are the following:

1: Laptop with 802.11g PC card The laptop is the core piece of the setup. It sends the packets over wired ethernet, the destination of the packets is the 802.11g PC card inserted in the laptop. The packets are bridged by the 802.11g router (2) from wired ethernet to wireless LAN and sent over the WLAN antenna (4). Sending packets from one network interface to an other inserted in the same computer involves some tricks (see Appendix B.1). The operating system running on the laptop is linux (kernel 2.6.11.6). The laptop is a HP omnibook xt1500 with a 1.6GHz Pentium 4M processor and 512MB RAM. Access to the wired ethernet is provided by the built–in Realtek RTL–8139 network interface. The 802.11g PC card is a MSI CB54G2 with a RaLink rt2500 chipset. Drivers for the PC card are compiled from the CVS version of 14.04.2005 provided by the rt2x00 Open Source Project[1]. In order to control the measurement, the laptop can change 802.11g parameters of the 802.11g router (2) by accessing it with telnet and and control the reference receiver (5) by using *Secure Shell* (SSH). The whole measurement sequence is controlled by a shell script running on the laptop.

2: WRT54G 802.11g router The Linksys WRT54G is a very versatile device, it acts as AP in the setup. Using a modified firmware (hyperWRT 2.0[2]), it can be accessed with

[1]http://rt2x00.serialmonkey.com
[2]http://www.hyperwrt.org

telnet and settings can be changed from the shell. The operating system which runs on the WRT54G is linux. Parameters that can be changed include the data rate, retry limits and the transmit power. Only one antenna port of the router is used, it is directly connected to the power splitter (3). For all measurements the transmit power P_{tx} is 10dBm.

3: 3dB power splitter In order to split the power sent to the WLAN antenna (4) and the reference receiver (5), a power splitter is needed. The splitter is a Mini–Circuits ZFSC–2–2500, it has an insertion loss L_{split} of 3.5dB at 2.4GHz.

4: WLAN antenna The antenna is a Huber & Suhner SPA 2400/75/9/0/V, it has a gain G_{ant} of 8.5dBi. It is mounted 90 degrees rotated (horizontal polarization). The cabling from the splitter (3) to the antenna has a loss L_{cabwl} of 4dB. Between the port of the antenna and the cable is an attenuator, which has an insertion loss L_{attwl} of either 23dB or 33dB depending on the measurement scenario.

5: Control PC & reference receiver To be able to control the measurement process, the operator can access the laptop (1) with SSH. In addition, the control PC is used as reference receiver. A 802.11g PCI card captures the frames sent by the router (2) to the laptop (1). During a measurement, the capturing application can count how many times the same frame is transmitted over the PHY. The card is a MSI PC54G2, it uses the same chipset an driver as the PC card in the laptop. In order not to overload the receiver, the signals coming from the splitter (3) are attenuated. To do so, a 40dB attenuator is inserted between the card an the cable. Including a cable loss of 2dB, the signal power at the receiver input is -35dBm.

6: Interference source Depending on the scenario, a different interference source is used. For the microwave oven scenario it is a microwave oven, for the Bluetooth scenarios a signal generator simulating Bluetooth signals.

7: Interference antenna The signals generated by the interference source (6) are transmitted inside the chamber by the interference antenna. The antenna type is the same as the type of the WLAN antenna. It is mounted directly above the WLAN antenna, also 90 degrees rotated (not next to the WLAN antenna as shown in Figure 8.1). Depending on the measurement scenario, an attenuator is attached to the port of the antenna. The cabling between the interference source (6) and the antenna has a loss L_{cabint} of 8dB.

A picture of the measurement setup inside the anechoic chamber is shown in Figure 5.2.

5.3. Software

To perfrom the measurements various software is used. On the laptop, a program sends UDP packets with a specified rate and size. On the same laptop, an other program receives the packets and saves timing information for each of them. The reference receiver captures all UDP packets sent by the WRT54G router and stores information about how many times the same packet has been sent. Finally, the files generated on the laptop and by the reference receiver are merged together. The resulting file can be read by matlab where data analysis is performed.

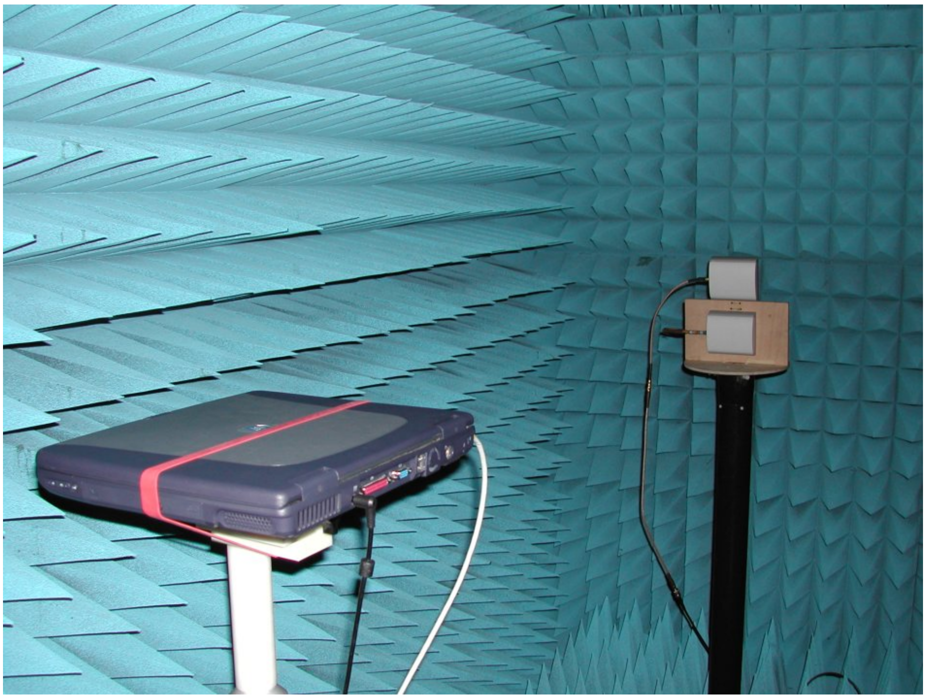


Figure 5.2.: Measurement setup inside the anechoic chamber

5.3.1. UDP Sender & Receiver

The program used for sending UDP packets from the laptop is *Real–time UDP Data Emitter* (RUDE)[3] and the program which receives the packets is *Collector for RUDE* (CRUDE)[3]. For the measurements, RUDE generates one flow with a constant bit rate. The packets sent by crude contain a 64bit timestamp with 1µs resolution, the receiver CRUDE stores the timestamp contained in the packet together with the time the packet is received (also 64bit, 1µs resolution) and the sequence number in a binary file. In order to obtain precise results, RUDE and CRUDE are run with real time scheduling policy (round robin scheduling, SCHED_RR). For all measurements, the scheduling priority for RUDE is 85 and for CRUDE 90. This means, that the receiver has a higher priority than the sender. The other way around packets are lost because RUDE uses a busy loop for timing and CRUDE gets not enough time from the scheduler to receive the packets.

5.3.2. Packet Capturing At The Reference Receiver

The reference receiver uses a 802.11g PCI card in monitor mode. In this mode, all received frames can be processed by user space programs. The program used for capturing the packets is programmed in C and uses libpcap[4] for packet capturing. Libpcap provides a very simple and platform independent interface. When a packet is received, libpcap calls a previously registered callback function of the user space program. In the program used, the callback function performs some checks on the packet headers to decide if the packet is a UDP packet sent from RUDE. If the packet is such a packet, the program stores the sequence number (generated by RUDE) and the arrival time in a binary file. When the

[3]http://rude.sourceforge.net
[4]http://www.tcpdump.org

same UDP packet is sent multiple times, the file contains more than one entry for the same packet. This allows the data merging application to calculate how many times the same packet has been sent.

5.3.3. Offline Data Merging

For each measurement two files are generated. One file is generated by CRUDE on the laptop, the other by the packet capturing application on the PC used as reference receiver. For each measurement 100'000 packets are sent (see below), the files are about 2MB in size and contain about 100'000 entries. For performance reasons, the data merging application is programmed in C and performs the following tasks:

- Normalize the 64bit timestamps from the file generated by CRUDE. The timestamps use a 64bit format (POSIX struct timeval), 32bits are spent for the seconds and 32bit for the microseconds. Since the times are absolute, the value for the seconds is very high. This leads to resolution problems when the timestamps are stored in 64bit floating point numbers by matlab. In order to avoid these problems, the transmit time of the the first packet is subtracted from all other time values. To do so, fixed point arithmetic is used. The resulting timestamps are all relative to the start time of the measurement and floating point resolution is no issue.
- Count how many times the same packet has been sent over the wireless link. This is done by counting how many entries for a packet exist in the file generated by the packet capturing program.

The resulting binary file can be read by matlab and contains the following information for each packet:

- The sequence number N_{seq}.
- The transmit time t_{tx}.
- The receive time t_{rx}.
- The number of times the packet has been sent N_{send}.

If a packet is lost during transmission and not received by CRUDE, no entry for the packet exist in the file.

5.4. Scenarios

In order to simulate various real world environments, measurements with different types of interference are conducted. A series of measurements with the same conditions (interference source) is called a scenario. For each scenario 48 measurements are carried out. Which parameters are varied for the measurements is shown in Table 5.1.

The packet size is the size of the UDP body and does include the RUDE/CRUDE header. Each measurement consists of 100'000 packets, the time between two consecutive packets is 1ms resulting in a total time of 100s required for a measurement. A short retry limit L_{s} (srl in the WRT54G router) of 1 means, that transmission of a packet is attempted only one time (no retries). The channel used for the measurements is Nr. 9, it has a center frequency of 2.452GHz. For all measurements the ERP-OFDM PHY mode is used.

Parameter	Values
Packet Size (Bytes) S_p	128, 256, 512, 1024
Short Retry Limit L_s	1, 2, 4, 7
Data Rate (Mbit/s) R_d	6, 24, 54

Table 5.1.: Parameters for each measurement

5.4.1. Normal Signal

In this scenario, a good connection with no interference is simulated. In order to have a strong signal, an attenuation L_attwl of 23dB is used at the port of the WLAN antenna. The resulting power P_ant at the port of the antenna is calculated the following way (all figures in dB):

$$P_\mathrm{ant} = P_\mathrm{tx} - L_\mathrm{split} - L_\mathrm{cabwl} - L_\mathrm{attwl} = 10\mathrm{dBm} - 30.5\mathrm{dB} = -20.5\mathrm{dBm} = 8.9\mu\mathrm{W} \quad (5.1)$$

Knowing the gain of the antenna used G_ant and the distance from the antenna to the antenna of the PC card r, the power density at the location of the PC card can be calculated (all figures linear):

$$S_\mathrm{wl} = \frac{P_\mathrm{ant} G_\mathrm{ant}}{4\pi r^2} = 3.0\mu\mathrm{Wm}^{-2} \quad (5.2)$$

By using this power density, a certain distance r_s from the access point is simulated. For an access point with a dipole antenna ($G_\mathrm{dip} = 2.15\mathrm{dBi}$) and a transmit power P_ap of 10dBm, the simulated distance can be calculated:

$$r_\mathrm{s} = \sqrt[3]{\frac{P_\mathrm{ap} G_\mathrm{dip}}{4\pi S_\mathrm{wl}}} = 7.6\mathrm{m} \quad (5.3)$$

For the calculation a path loss exponent of 3 is assumed. This distance is used for all scenarios except the weak signal scenario.

5.4.2. Weak Signal

In order to simulate a weak signal, the signal is attenuated 10dB more than in the normal signal scenario ($L_\mathrm{attwl} = 33\mathrm{dB}$). The resulting power density at the location of the PC card is therefore 10 times lower ($S_\mathrm{wl} = 0.3\mu\mathrm{Wm}^{-2}$). From this power density, the resulting simulated distance can be calculated by means of Eq. 5.3, $r_\mathrm{s} = 16.3\mathrm{m}$.

5.4.3. Microwave Oven Interference

Microwave ovens use microwaves to heat food. The microwaves are generated by a magnetron tube in the oven. When 50Hz ac power is used, the magnetron produces a burst of microwaves every 20ms. The duration of a burst is about 8ms and its center frequency is 2.45GHz. The channel used for the measurements has a center frequency of 2.452GHz. It can be assumed that this channel is the most affected by the microwave oven because the center frequencies of the channel and the microwave bursts nearly coincide. The shielding of microwave ovens is not perfect, according to DIN EN 60335-2-25 microwave ovens are allowed to emit up to $50\mathrm{Wm}^{-2}$ at a distance of 5cm. Signal produced by magnetron tubes have complex spectral and transient characteristics. It is therefore difficult to generate such signals artificially. For this reason, a microwave oven is used as interference source. The oven is an old MioStar model with a nominal power of 700W. Through the side wall of the oven, a small $\lambda/4$ (3cm) monopole antenna is mounted. The antenna picks up radiation generated by the magnetron tube. The received signals are attenuated outside the oven (6dB). HF cabling is used to connect the oven with the interference antenna. For the measuremets, a bowl filled with 1l water is placed inside the oven and maximal power is used. After an initial warm up period, when the water begins to boil, the power measured at the the antenna port is 14dBm. For this power measurement a HP435B power meter with a 8482A sensor is used. The power density of the interference signals at the location of the PC card is the following:

$$S_{\text{int}} = \frac{P_{\text{int}} G_{\text{ant}}}{4\pi r^2} = 8.3\text{mWm}^{-2} \tag{5.4}$$

5.4.4. Bluetooth Interference

Bluetooth is a widely used technology to form a network between consumer devices, called piconet. The frequency band used for Bluetooth is the same as for 802.11g, the unlicensed ISM band at 2.4GHz. Due to the widespread use of Bluetooth, the possibility for a 802.11g wireless LAN and a Bluetooth piconet to coexist at the same location is high. Because Bluetooth uses *Frequency Hopping Spread Spectrum* (FHSS) technology with hopping frequencies distributed over the whole ISM band, all 802.11g channels are affected by interference from Bluetooth devices. The frequencies of the 79 channels used by Bluetooth can be calculated the following way:

$$f = (2042 + k)\text{MHz}, \ k \in \mathbb{N} \wedge k = [0..78] \tag{5.5}$$

Which frequency is used, is determined by a pseudo random hopping sequence. The sequence is calculated from the MAC address of the master. Normally, the frequency is changed after one slot, which has a duration of 625µs. This leads to a hop rate of 1600 hops per second. Because the hopping sequence is very long ($2^{27} - 1$), the time until the sequence is repeated is more than 23 hours. By the use of different hopping sequences, several piconets can coexist at the same location. A piconet consists of a master and one or more slaves. Channel access is realized by *Time Division Duplex* (TDD), this way the master is only allowed to transmit at even numbered slots and the slaves at odd numbered ones. For certain applications, a device can use more than one slot to transmit data. When multi slot packets are sent, the frequency is changed after the whole packet has been sent. Multi slot packets have a length of either 3 or 5 slots.

For this scenario, Bluetooth signals are generated using Rohde&Schwarz AMIQ and SMIQ signal generators. According to application note 1GP38_0E [9], the frames are designed on a computer using the WinIQSIM software and transfered to the AMIQ, which generates the baseband signal. The frequency hopping is accomplished by the SMIQ modulator. To do so, a random list with 4000 Bluetooth frequencies is generated on a computer and transfered to the SMIQ modulator. Frequency hops are initiated by trigger signals from AMIQ. Because the SMIQ modulator needs time to change the frequency, it is not possible to hop every slot. One empty slot is needed to change the frequency. Measurements for three different subscenarios are conducted, the first uses one slot packets (BTDH1), the others 3 and 5 slot packets (BTDH3, BTDH5). The slots used for the subscenarios are shown in Figure 5.3.

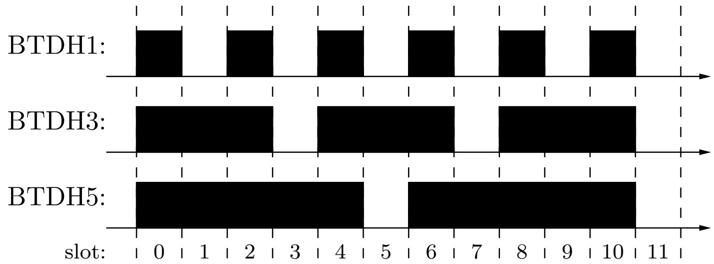


Figure 5.3.: Used transmit slots for the three Bluetooth scenarios

According to the Bluetooth standard, not the whole slot can be used for packet transmission, because time is needed for hopping to the next frequency. For the three scenarios, the maximal allowed transmit times are used. The transmit time, the hop rate and the duration of the random sequence are given in Table 5.2.

Scenario	**Transmit Slots**	**Transmit Time**	**Hop Rate**	**Sequence Duration**
BTDH1	1	366µs	800s^{-1}	5s
BTDH3	3	1616µs	400s^{-1}	10s
BTDH5	5	2866µs	266.7s^{-1}	15s

Table 5.2.: Bluetooth scenario parameters

The transmit power of the Bluetooth interference signal is chosen to result in about the same power density at the location of the PC card as the power density of the wireless LAN. To do so, the transmit power is set to 0dBm and an attenuation of 13dB is inserted at the port of the interference antenna. Together with 8dB cable loss, this results in a power of -21dBm at the port of the interference antenna. The power density at the location of the PC card is calculated by means of Eq. 5.2, the resulting S_{int} is $2.65\mu\text{Wm}^{-2}$.

6. Data Analysis and Interpretation

In total, measurements for six different scenarios are conducted. Every scenario consists of 48 measurement. This leads to 288 data files which need to be analyzed. The analysis of the data is conducted with matlab. Because of the high number of measurements, it is important to find a way to display the results in a condensed form. In the first section of this chapter, the data analysis is explained. An interpretation of the results is given in the second section. Detailed measurement results and plots for the different scenarios can be found in Appendix A.

6.1. Data Analysis

In order to perform the data analysis, the combined measurement file is read into matlab. The resulting vector contains the sequence number N_i^{seq}, the transmit time t_i^{tx}, the receive time t_i^{rx} and the number of times the packet was sent N_i^{send}. The vector holds this information for all N received packets. Lost packets can be detected because their sequence number is missing. From this vector, the sample mean of the delay ($\overline{d}$) and the sample variance of the delay (j, also knwon as jitter) are calculated the following way:

$$\overline{d} = \frac{1}{N}\sum_{i=1}^{N} t_i^{\text{rx}} - t_i^{\text{tx}} \tag{6.1}$$

$$j = \frac{1}{N}\sum_{i=1}^{N}((t_i^{\text{rx}} - t_i^{\text{tx}}) - \overline{d})^2 \tag{6.2}$$

In addition, the *Packet Error Rate* (PER) can be calculated from the number of sent and the number of received packets. This figures are simple to calculate but they do not provide much useful information about the characteristics of the channel. For the design of a low latency audio transmission system, it is important to know the characteristics for a specific playout buffer size. The size of the playout buffer (also known as jitter buffer), limits the maximal allowed delay of a packet. If a packet arrives too late, it is dropped because the scheduled playout time of the packet has already passed. When a fixed playout buffer size is used, it is important to know how many consecutive packets are lost. The longer a burst loss is, the more difficult it is for the audio transmission system to correct or conceal the loss effciently. The goal of the data analysis is therefore to calculate a delay dependant burst statistic from the measured data. This statistic is calculated the following way:

1. Build a delay vector $d(n)$ from the initial vector by calculating $t_i^{\text{rx}} - t_i^{\text{tx}}$ for each packet. Packets which have been lost during the transmission are given a high delay (1s).
2. Calculate a loss pattern from the delay vector $d(n)$ and a maximal allowed delay d_{limit}. Packets with a delay bigger than d_{limit} are treated as lost.

3. From the loss pattern calculate a burst statistic. This statistic is a histogramm calculated form the number of consecutive lost packets (burst lengths).

This procedure is repeated several times, each time a differnet d_{limit} is used. The result is a delay dependant burst statistic. An example of such a statistic is shown in Figure 6.1.

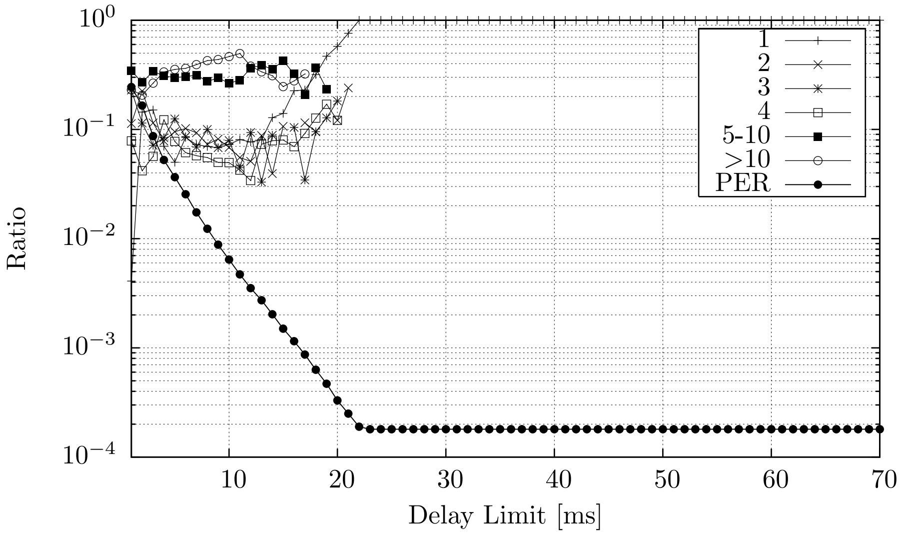


Figure 6.1.: Delay dependant burst statistic

The first six lines show the fraction of the lost packets which have been lost in burst of a specific size (burst size 1, 2, 3, 4, 5 to 10 and bigger than 10). A burst starts with a successfully received packet folowed by a number of lost packets (this denotes the burst length), it ends with a received packet. In addition to the burst statistic, the total PER is given by the line labeled PER.

6.2. Interpretation of the Results

In this section the results of the measurements are interpreted. From all the measurements condcuted, selected plots are shown and the influence of the different parameters are discussed. The parameters which influence the burst statistic are the scenario, the data rate R_{d}, the retry limit L_{s} and the packet size S_{p} (in Bytes).

6.2.1. Changing the Retry Limit

Changing the retry limit to a lower value than recommended in the standard seems to be a good solution to decrease the mean delay of the transmission. This is of course true, because the number of times a packet can be sent and therefore the total time needed for the transmission is lower. But the lower retry limit does also increase the PER. The mean delay, the jitter and the PER are shown in Table 6.1. The values are calculated from the microwave oven interference scenario. The data rate is 54Mbit/s and the packet size 512Bytes.

The PER increases up to 24% when a retry limit of 1 is used (this means the packet is only transmitted one time). A high PER does not automatically mean, that the transmission is not suited for low latency audio transmission. When *Forward Error Correction*

Retry Limit L_s	**Mean Delay $\overline{d}$**	**Jitter j**	**PER**
7	6.4ms	29µs^2	0.5%
4	4.3ms	19µs^2	8.0%
2	3.1ms	10µs^2	15%
1	1.8ms	4.1µs^2	24%

Table 6.1.: Transmission characteristics for different retry limits

(FEC) is used, it is possible to recover from packet loss. But this is only possible when the lost packets are more or less uniformly distributed between the successfully received ones. When the packets are lost in bursts, FEC does not work and low latency audio transmission is not possible. The bursts statistics for the same measurements are shown in Figure 6.2.

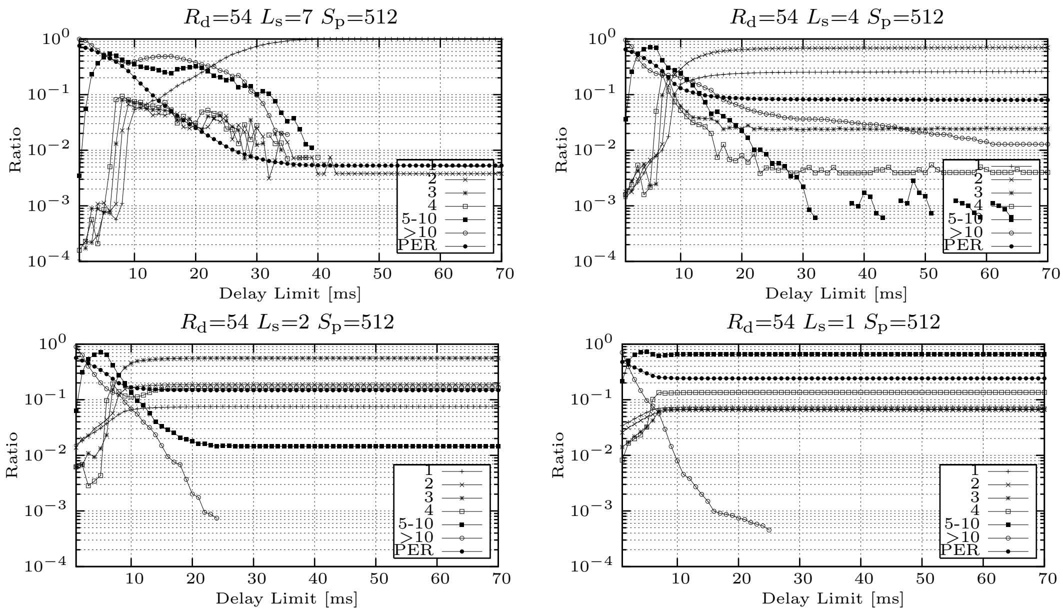


Figure 6.2.: Different retry limits

It can be seen, that the loss characteristics become more bursty when the retry limit is decreased. The 802.11g standard setting, which is L_s=7, seems to be a good choice. When this retry limit and a buffer size of about 45ms is used, almost all packets are lost in bursts of size 1 and very few in bursts of length 2. The microwave oven, which is used as interfernence source for this scenario, causes more packet loss than all other interference sources used. It can be seen as the worst case scenario. Although the increasing burstyness when lowering the retry limit is not unique to the microwave oven scenario. Measurements from all other scenarios provide similar results. It is therefore reasonable to conclude, that lowering the retry limit is not a viable solution. As a consequence, all other measurement plots shown in this section and in Appendix A are generated from measurements with L_s=7.

6.2.2. Data Rate and Packet Size

When a 802.11g devices detects a high PER, it lowers the data rate to make the transmission more robust. This is normally a good choice, because a less complex modulation scheme is employed and the SNR needed to receive the signal is therefore lower. The increased performance of the transmission by decreasing the data rate can be clearly seen in Figure 6.3. The plots for this Figure are from the weak signal scenario, where no additional interference is present. For the plots at the top a data rate of 54Mbit/s is used, for the lower two a data rate of 24Mbit/s.

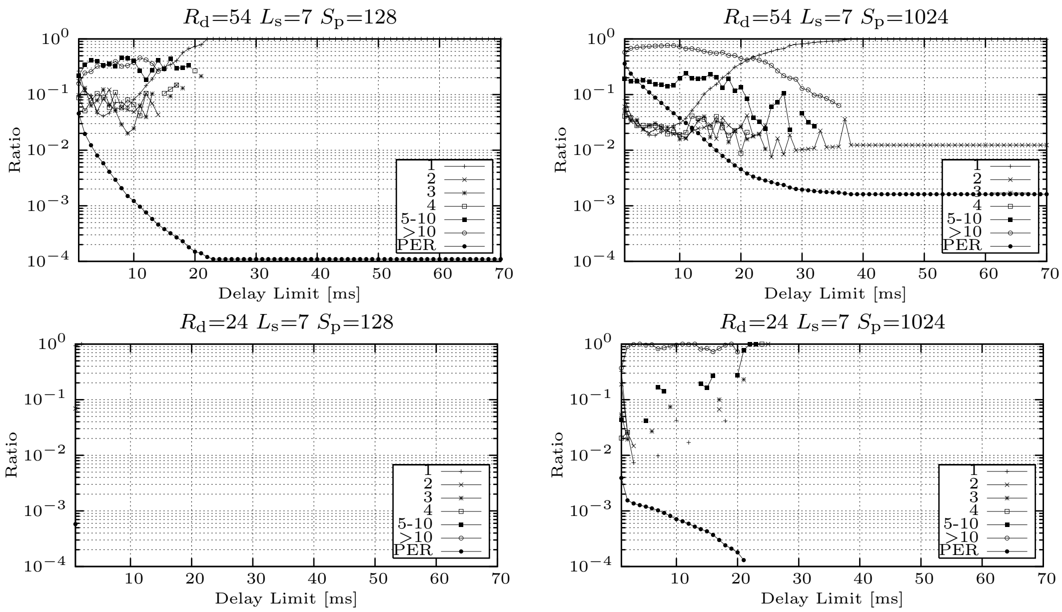


Figure 6.3.: Data rate and packet size with weak signal

This behavior is the reason why most 802.11g devices automatically decrease the data rate when a bad connection is detected. The problem is, that the kind of interference is unknown to the 802.11g device. When the interference is just an increased noise level, as in the weak signal scenario, lowering the data rate is reasonable. On the other hand, when bursty interference is present, a lower data rate causes the transmission to become even more unreliable. The problem is not the actual frame transmission but the time needed to transmit the frame. When small frames are sent with a lower data rate, the PER is smaller than with a high data rate. But time needed to transmit a frame at a low rate is higher. When the size of the frame is above a certain limit and retries occur frequently, packets are dropped by the access point because the link is congested.

When small packets are used (128Bytes), data rates of 54Mbit/s and 24Mbit/s provide similar results. The reason for this is the time needed to transmit the PHY frame. Together with higher level protocol headers (UDP,IP,LLC,MAC) the size of the MAC frame is 192Bytes for S_p=128Bytes. According to Eq. 3.5, the time needed to transmit the frame is 52µs for 54Mbit/s and 136µs for 24Mbit/s. When the size of the packet is increased to S_p=1024Bytes, the transmit time is 184µs and 384µs respectively. Although the transmit time increases more for small packets (because of the headers), the data rate has a more significant impact on the performance when large packets are sent. For R_d=24 and S_p=1024 nearly all packets are lost.

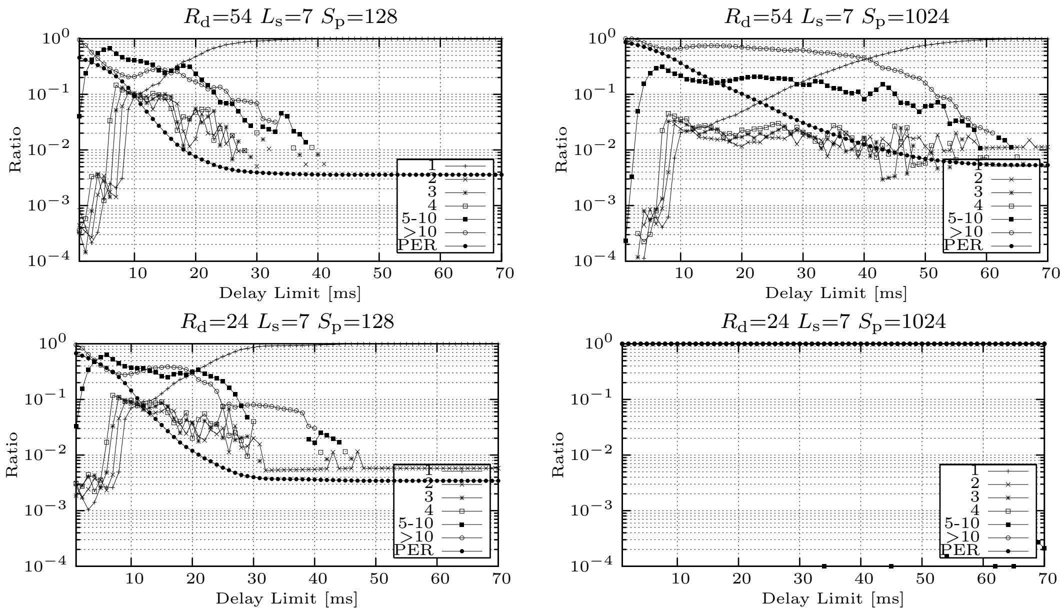


Figure 6.4.: Data rate and packet size with microvawe oven interference

When a data rate of 6Mbit/s is used and bursty interfernce is present, only very small packets can be sent successfully. Figure 6.5 shows this for two different interfernce sources. With Bluetooth interfernce (BTDH5), packets with sizes of 128Bytes and 256Bytes can be transmitted with acceptable performance. Microwave oven interfernce on the other hand, affects the performance more dramatically. Consequently, it can be conclueded that a data rate of 6Mbit/s should not be used for low latency audio transmission. The plots from measurements with data rates of 6Mbit/s are therefore not shown in Appendix A.

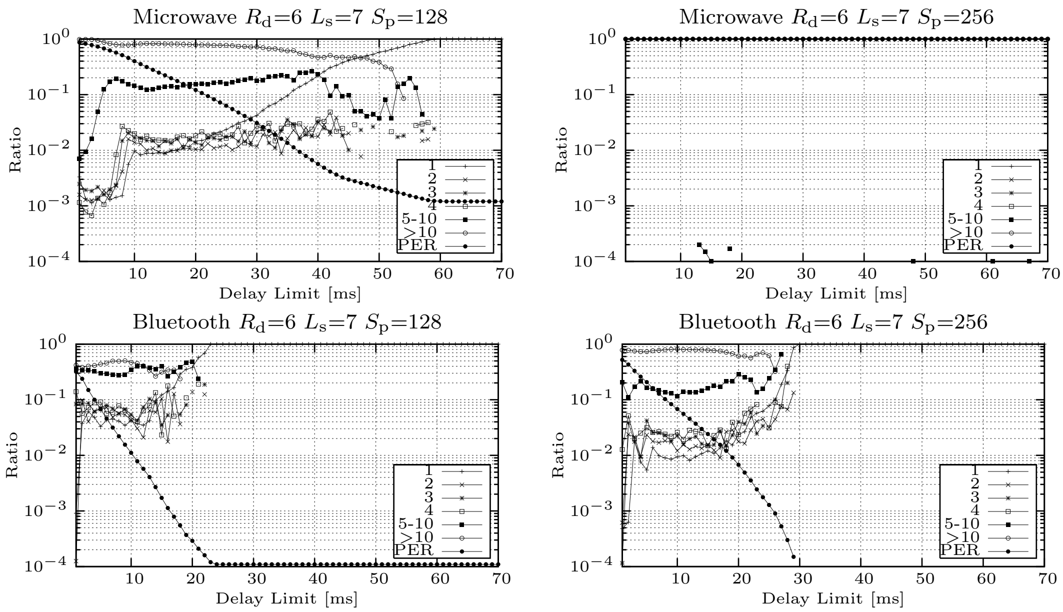


Figure 6.5.: 6Mbit/s data rate with microwave oven and Bluetooth interference

7. Recommendation for Audio Transmission System

According to the analyses in Chapter 6, important statements about the design of an audio transsmission systenm can be made. As already mentioned in Chapter 2, data size plays an important role. To guard against jitter, buffersize gets a major role.

By reason of noisy environments (microwave ovens in close proximity and switched on) an audio system must be able to survive such situations. Thus, to design a robust mechanism the buffer must be able to compensate the jitter introduced by the 802.11g connection and should have a length of about 50ms. This is large for real time audio applications. It must be clear that this result is based on selected interferences. Since there is not always a microwave oven or other disturber around, the buffer size must be adjustable by the user.

As already mentioned in Chapter 6, the data rate must be held constant otherwise if interferences occur the risk of link congestion rises. Even though the measured PER is sligthly higher when bursty interference is present and a high data rate is used, the link is still better suited to transmit frames at a rapid rate because no packets are dropped by the access point. It is therefore reasonable to conclude, that the data rate of the access point should be set to a fixed value. Without bursty interference from other devices, the connection should be as good as possible (low PER). For a good connection, the distance between the access point and the receiving station must not be too high. In addition, multipath effects can cause spots with low signal quality. The distance between the locations with high and low signal quality is in the range of centimeters because the wavelength of the wilreless LAN is only 12cm. A good way to combat these effects is to use multiple antennas (diversity antennas) at the sender and the receiver side.

Packet sizes of 256 to 512Bytes are a good compromise between information, redundancy and data rate. If the data size is higher, the risk of access point congestion rises when a low data rate is used. If the data size is smaller, the transmitted data is dominated by header information of protocols and the audio information is only a small part of the transmitted data.

The mentioned audio data size must also contain redundancy. Considering a long period of data transmission, packet loss bursts occur rarely. Measurements have shown that losses of one packet occurs more often as loss in form of bursts. Therefore, FEC can be used to reconstruct the missing data by sending the previous data part with the actual one.

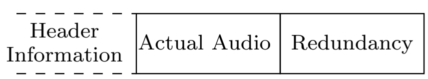


Figure 7.1.: Redundancy

Since bursts can occur anyway, action must be taken on this. Additionally to redundancy there should be a correction mechanism in the receiver application. Experimental trials have shown, that missing packets cause annoying clicks. A computation shows how long a missing audio data packet is:

- Audio Data Packet Size: 256Bytes
- UDP Body Size: 512Bytes (256Bytes Redundancy)
- Audio Resolution: 16bit
- Sampling Frequency: 44100Hz

$$\text{For stereo: } \frac{\frac{256\text{Bytes}}{2 \cdot 16\text{bit}} \cdot 8}{44100\text{Hz}} = 1.45\text{ms}$$

The drop out of packets which produces loud clicks, can be concealed by approximated Bezier Curves [10]. With this approximation, the missing part of an audio part can be stitched. Tests have shown, that the stitched part can hardly be heard. If the stitched part can be heard, depends on its length. For small lenghts (1.45ms), the stitching does not produce annoying artefacts. When longer parts are approximated, the stiching is noticable. This is an onther reason to favourize small audio frames.

An other part witch must be considered is the retry limit factor of the access point. In situations where a frame is lost, this factor gives the maximum number of times it can be resent. In the standard setup this setting is set to 7. As measurements have shown, the best transmission results are achieved with a retry limit factor of 7.

In short, following statements can be made:

- To reach a high data rate, a good connection must exist (use of multiple antennas (spatial diversity), consider distance between sender and receiver)
- Buffersize about 50ms makes operation possible when bursty interferers are present
- 802.11g bit rate must be held constant
- Packet size between 256 and 512Bytes
- Redundancy, sending last audio data with the actual one (FEC)
- Ability to conceal missing audio data which cannot be corrected by FEC
- Adjustable setting of the buffer size according to the environment (usage of microwave ovens, Bluetooth devices etc.)

8. Proof of Concept Implementation

In order to implement the findings of the conducted research, an audio transmission system is developed. The system is composed of a sender and a receiver application. The sender application reads uncompressed audio data from a wave file (44.1kHz, 16bit, stereo) and sends it to the receiver. At the receiver the audio data is stored in a ring buffer which is used to compensate jitter. For play out, audio data is read from the ring buffer and handed over to the audio card. The system is implemented using the C programming language. One goal of the application is to keep the latency as low as possible. To do so, an audio system is needed which supports small frame sizes. The audio system used is the *Advanced Linux Sound Architecture* (ALSA). Because of the lack of time, only a basic audio transmission system is developed. At the day of writing, it supports neither FEC nor loss concealment.

8.1. Frame Formats

Two different frame formats are used by the audio transmission system. The first frame type is the synchronization frame. It is used to calculate the time offset between the sender and the receiver. At the application startup, the sender sends a number of synchronization frames to the receiver. The "tx timestamp" is filled in by the sender. When the receiver receives a synchronization frame (the frame type is indicated in the "Packet info" field), it fills in the "rx timestamp" and the correct sequence number and sends it back to the sender immediately. This way the sender can calculate the round trip time and estimate the time offset between the clock of the sender and the receiver.

Synchronization Frame:

31	0
Sequence Number	
Packet info	last P. info
Timestamp tx (seconds)	
Timestamp tx (micro seconds)	
Timestamp rx (seconds)	
Timestamp rx (micro seconds)	

Audio Frame:

31	0
Sequence Number	
Packet info	last P. info
Timestamp (samples)	
Timestamp (seconds)	
Timestamp (micro seconds)	
Audio Data (256Bytes)	

Figure 8.1.: Frame formats of the audio transmission system

The other frame used is the audio frame. This frame type is used to transmit the actual audio data. It contains two different timestamps. The timestamp in samples is required to place the audio data in the ring buffer at the receiver. The other timestamp in seconds and micro seconds is needed to calculate the latency when the audio data is played out. To be able to do so, the sender fills in this timestamp corrected with the time offset estimated by the previous synchronization frame exchange. The fields "Packet info" and "last P. info" are used in both frame types. This field describes the type of the frame and could later be used to indicate information like the FEC scheme is use etc. There are two such fields, the idea is to send the contents of the "Packet info" field of the previous frame with the actual frame. This allows the receiver to know what kind of frame has been lost when packet loss occurs. Having this information is useful for packet loss statistics.

8.2. Sender Application

When the sender application is started, it performs the time offset estimation described above. After that, it reads audio data from a wave file and sends it to the receiver. In order to send the frame over the wireless network, UDP is used. The amount of audio data encapsulated in an audio frame is 256Bytes. When CD quality (44.1kHz, 16bit, stereo) audio data is used, the audio data in a frame has a length of 1.45ms. This means that the sender has to send an audio packet every 1.45ms. Such short time intervals are a problem when a standard operating system is used because the timers provided are too imprecise. Therefore, real time scheduling is used for the sender application (SCHED_RR). Timing is accomplished by the means of the usleep function provided by glibc. This function uses a busy loop internally to provide precise time intervals.

8.3. Receiver Application

The receiver application is much more complex than the sender application. It has to receive audio frames at a rapid rate and place it at the correct position in the ring buffer. At the same time, the audio needs to be played out. Two threads are used for those tasks. The first thread receives the data, the second transfers the audio data from the ring buffer of the audio card. Access to the ring buffer is serialized by a semaphore. The audio APO used is ALSA, in order to reach low latency, very small audio card buffers are needed. Most audio cards work with some kind of ping pong buffer. A ping pong buffer is a ring buffer with a size of two. When one part is played out, the other can be filled by the user application. In ALSA notation, one part of the ring buffer is called a period. How many periods must be used and what their size is depends on the audio hardware. The smallest period size current PC audio cards support is 16 frames. One frame is composed of one sample per channel. The audio card of the computer used for this work has an Intel 8x0 chip which supports such small periods. The receiver application uses two periods, each 16 frames long. At a sample rate of 44.1kHz one period has a length of 363µs, the total additional latency induced by the audio card buffer is therefore only 726µs. The small period size requires the receiver application to access the ring buffer every 363µs and fill the audio card buffer with new data for play out. When such small periods are used, the application must be run under real time scheduling policy (SCHED_RR) otherwise under runs occur.

In addition to the ring buffer used to store audio data, an other ring buffer is used for latency information. This buffer is filled with the timestamps from the received audio frames. The timestamps from this buffer are needed to calculate the latency when the audio data is transfered to audio card buffer. A fixed latency can be set by the user. Every time the audio card buffer is filled with new data, the actual latency is calculated and compared with the latency set by the user. If the latency is too high, the play out position is shifted back by 16 frames. When the latency is too small the play out position is advanced by 16 frames. Because very small buffers are used this is can hardly be heard. Shifting the play out position solves also an other problem which occurs because the clock of the sender and the clock of the audio card are not synchronized. This problem is known as clock skew and can cause buffer under- and overflows. The clock skew is automatically corrected by the shift operation because the latency is fixed.

8.4. Review and further Development

The audio transmission system described in this chapter is not complete. It shows however, that standard PC audio hardware can be used for low latency operation. The latency introduced by the jitter buffer of the receiver dominates the overall latency of the system. Depending on the audio hardware and the audio system, very small audio buffers can be used, the additional latency caused by these buffers is therefore negligible. For operation over 802.11g, the latency should be set to about 50ms. This includes the time needed to transmit the frame over the wireless link. The mean value of the transmission time however is very small (about 400µs) and can therefore be neglected.

Further development of the audio transmission system should consider the following points:

- Include a way to recover from packet loss. When a buffer length of about 50ms is used, the FEC should be able to correct single lost packets without any or only a small impact on audio quality.
- Implement loss concealment, this could be done by approximated Bezier Curves (see Chapter 7).
- For a professional system, a better technique for clock skew correction must be found. When custom audio hardware is used at the receiver side a tunable audio clock (PLL) could be used for this.
- The sender application could be implemented as a virtual audio device driver. The main challenge is to find an efficient way for timing. Since no audio hardware is used for the virtual driver, no interrupts can be used for timing. This is a major problem because normal operating systems do not provide timers for such small time periods.

9. Conclusion

With this work, essential statements about the influence of devices operating in the same band as WLAN can be made. It was shown than the microwave wave oven causes more packet loss than all other interference sources used. When the retry limit is decreased, the loss characteristics become more bursty. The standard setting of 802.11g (which is set to 7) is a good choice. When this retry limit and a buffer size of 50ms is used, almost all loss occurs in bursts of length 1. Only few packets are lost longer bursts. The solution to such a kind of packet loss is to add redundancy. Since the packets must not be too large, redundancy should only be sent for correcting loss bursts of length 1. For longer bursts a concealing mechanism can be used, for example approximated Bezier Curves. This solution stitches the missing part so that the difference to the original data is almost unhearable.

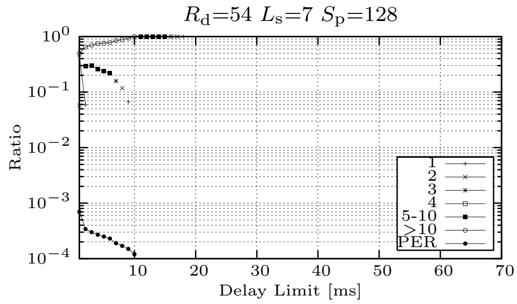
Lowering the data rate when bursty interference is present, can cause the transmission to become even more unreliable. This happens because the throughput of the wireless link is lower and packets are dropped by the access point. Not only the interfering devices must be considered to reach a good (low latency, low packet error rate friendly burst characteristics) audio transmission, the environment must be considered as well. Measurements conducted by others [5] have shown that indoor propagation falls 30dB below the ideal free-space propagation.

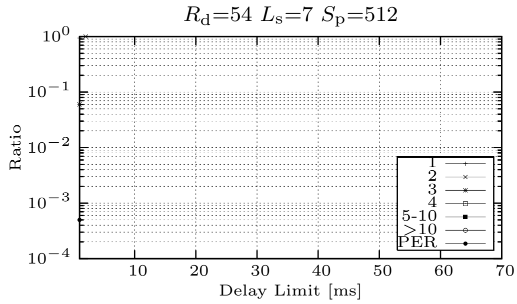
With the findings made, recommendations for the design of a low latency audio transmission system can be given. An important part of the system is the buffer size. For real time audio it must be as small as possible. When interferers are present, the buffer should have a length of about 50ms, otherwise packets are lost in long burst which can not be corrected by FEC. In addition, the buffer size must be adjustable according to the environment the system is used in.

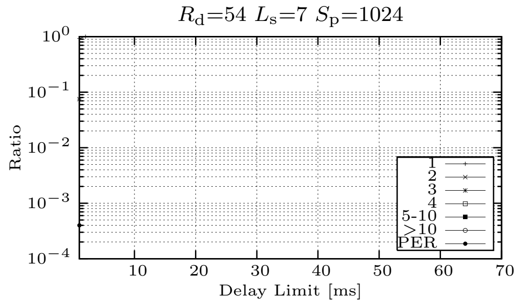
A. Measurement Plots

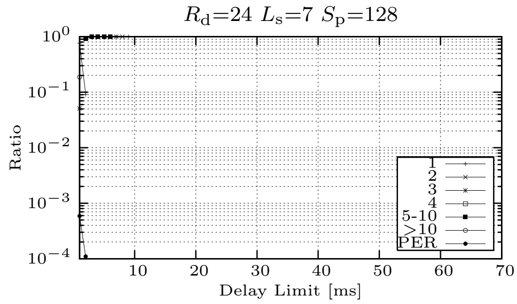
A.1. Scenario Normal Signal

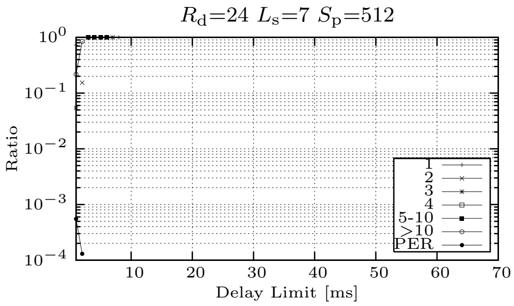
Data Rate R_{d}	Retry Limit L_{s}	Packet Size S_{p}	Mean Delay $\overline{d}$	Jitter j	PER
54	7	128	4.3e-04s	3.6e-08s^2	0
54	7	512	5.6e-04s	4.2e-09s^2	0
54	7	1024	7.4e-04s	4.4e-09s^2	0
24	7	128	4.6e-04s	7.7e-09s^2	0
24	7	512	6.6e-04s	8.1e-09s^2	0
24	7	1024	9.4e-04s	2.1e-07s^2	0

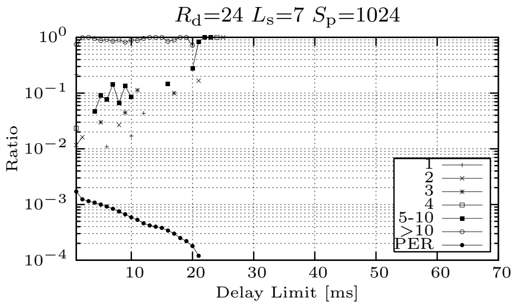






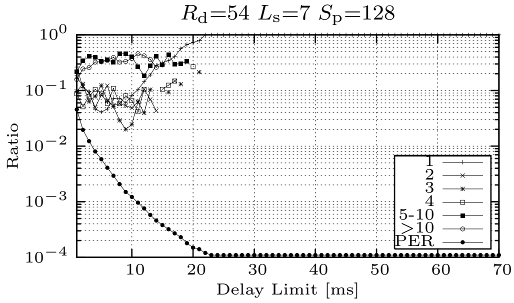


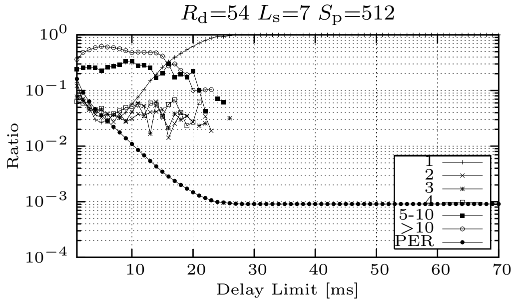


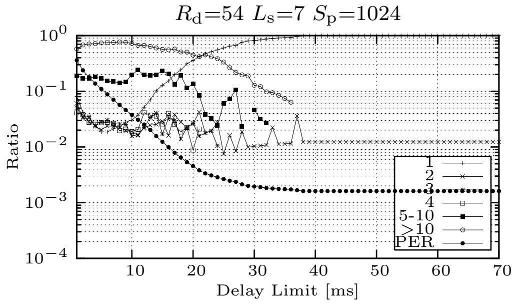


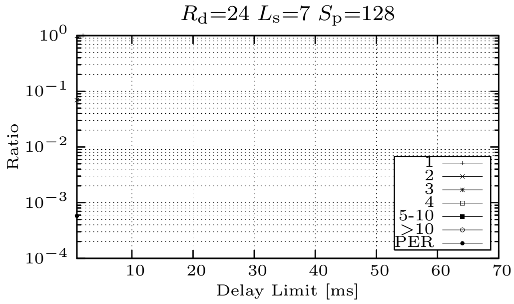
A.2. Scenario Weak Signal

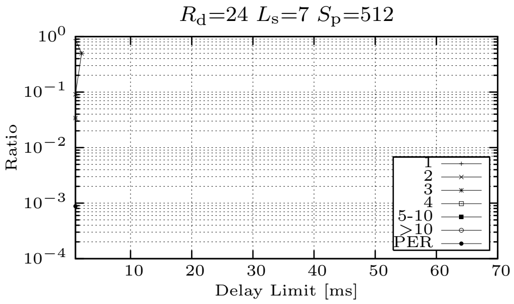
Data Rate R_d	Retry Limit L_s	Packet Size S_p	Mean Delay $\overline{d}$	Jitter j	PER
54	7	128	6.5e-04s	6.1e-07s^2	1.1e-04
54	7	512	1.3e-03s	3.5e-06s^2	9.1e-04
54	7	1024	2.5e-03s	1.0e-05s^2	1.6e-03
24	7	128	4.7e-04s	5.5e-09s^2	0
24	7	512	6.7e-04s	8.4e-09s^2	0
24	7	1024	9.7e-04s	2.5e-07s^2	0

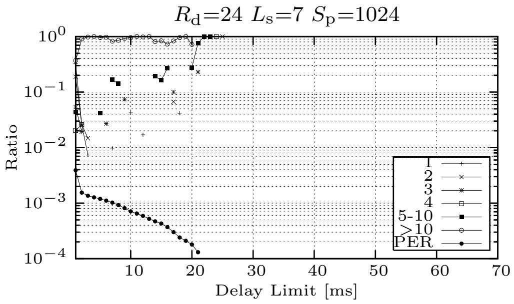






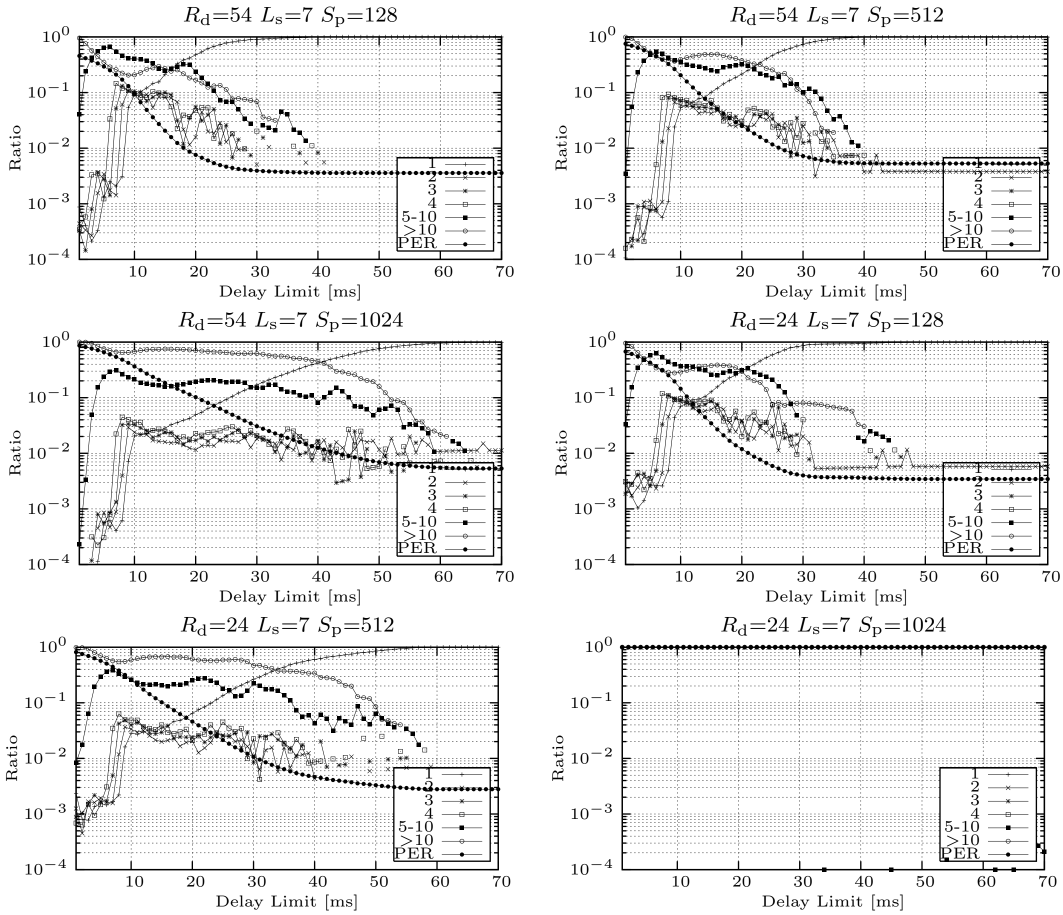






A.3. Scenario Microwave Oven

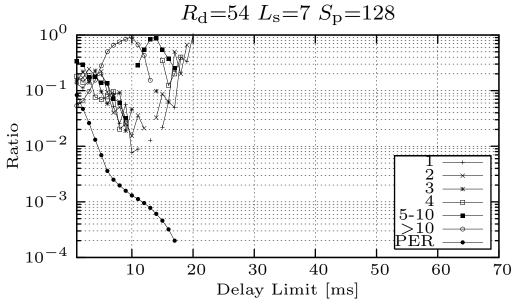
Data Rate R_d	**Retry Limit** L_s	**Packet Size** S_p	**Mean Delay** $\overline{d}$	**Jitter** j	**PER**
54	7	128	3.6e-03s	1.9e-05s^2	3.6e-03
54	7	512	6.4e-03s	2.9e-05s^2	5.3e-03
54	7	1024	9.9e-03s	6.6e-05s^2	5.3e-03
24	7	128	5.2e-03s	2.3e-05s^2	3.4e-03
24	7	512	7.7e-03s	4.1e-05s^2	2.8e-03
24	7	1024	1.6e-01s	2.9e-04s^2	1.7e-01



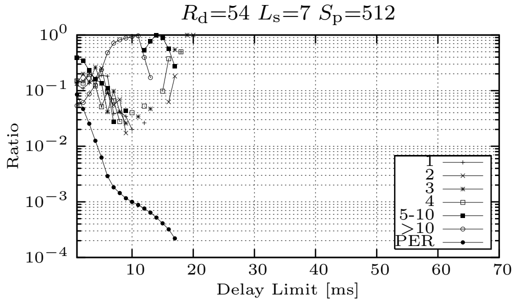
A.4. Scenario Bluetooth (BTDH1)

Data Rate R_d	Retry Limit L_s	Packet Size S_p	Mean Delay $\bar{d}$	Jitter j	PER
54	7	128	7.0e-04s	9.2e-07s^2	0
54	7	512	8.4e-04s	8.5e-07s^2	0
54	7	1024	1.1e-03s	1.2e-06s^2	0
24	7	128	7.3e-04s	8.1e-07s^2	0
24	7	512	1.0e-03s	9.2e-07s^2	0
24	7	1024	1.5e-03s	1.9e-06s^2	0

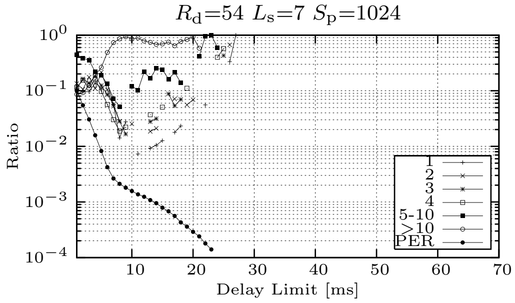
R_d=54 L_s=7 S_p=128



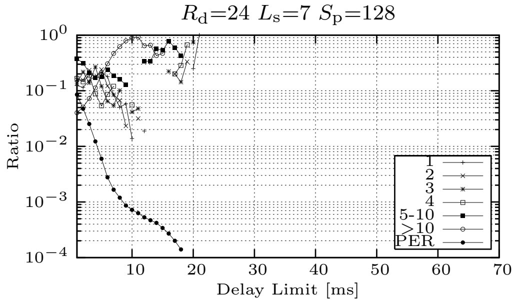
R_d=54 L_s=7 S_p=512



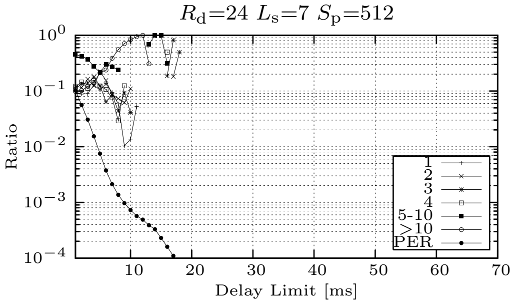
R_d=54 L_s=7 S_p=1024



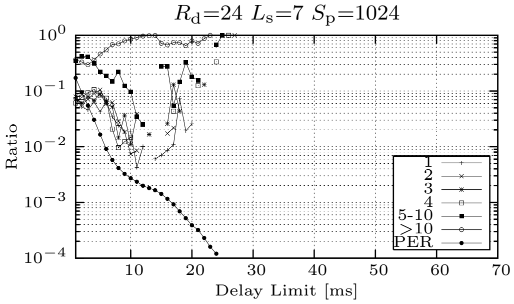
R_d=24 L_s=7 S_p=128



R_d=24 L_s=7 S_p=512

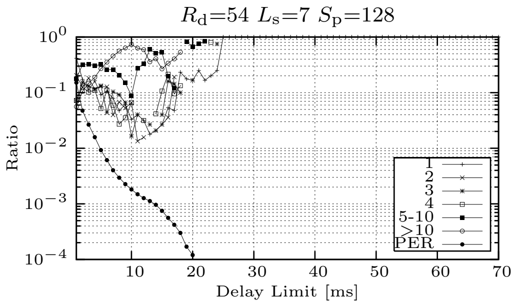


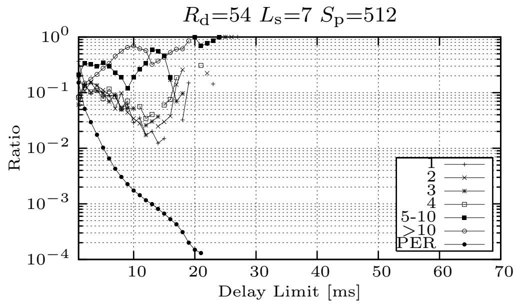
R_d=24 L_s=7 S_p=1024

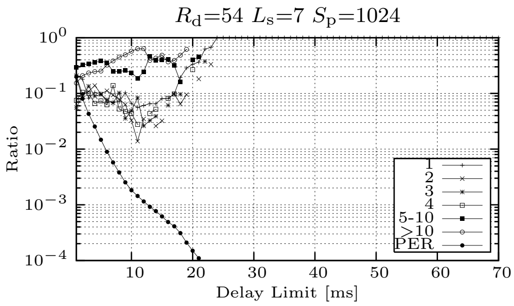


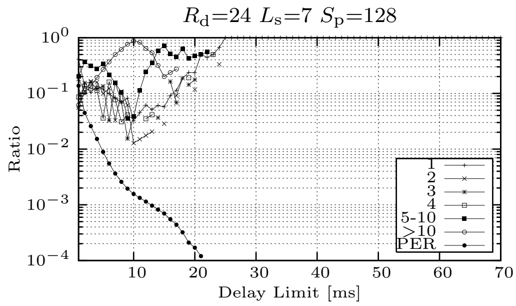
A.5. Scenario Bluetooth (BTDH3)

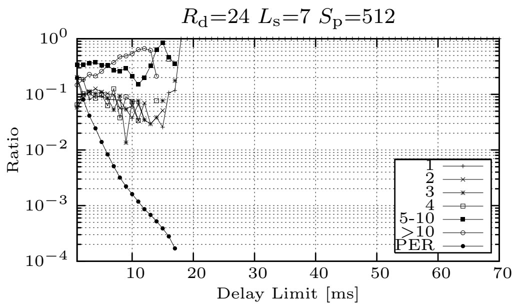
Data Rate R_{d}	Retry Limit L_{s}	Packet Size S_{p}	Mean Delay $\overline{d}$	Jitter j	PER
54	7	128	7.9e-04s	1.1e-06s^2	1.0e-05
54	7	512	9.7e-04s	1.3e-06s^2	0
54	7	1024	1.3e-03s	1.6e-06s^2	4.0e-05
24	7	128	8.3e-04s	1.1e-06s^2	4.0e-05
24	7	512	1.2e-03s	1.5e-06s^2	1.0e-05
24	7	1024	2.1e-03s	4.5e-06s^2	0

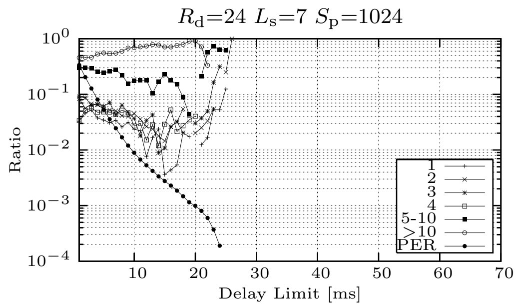






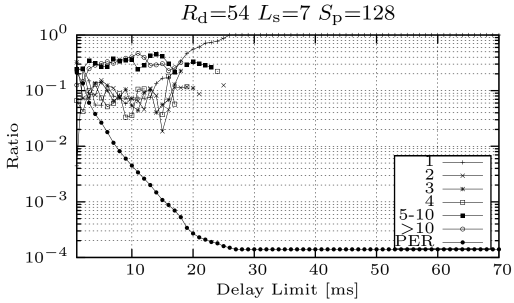


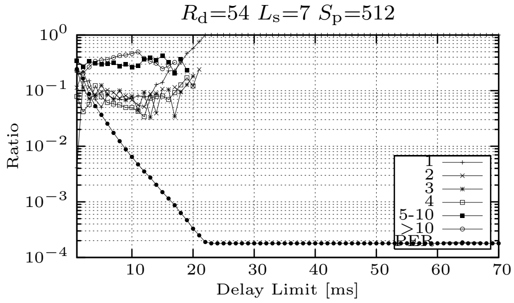


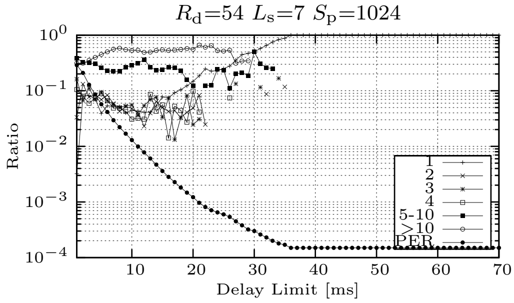


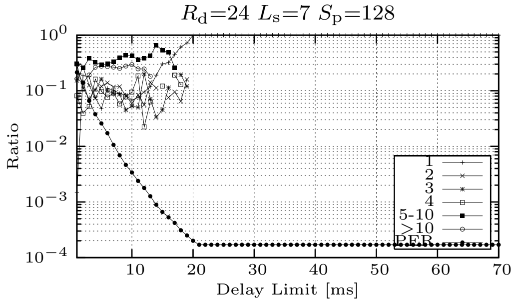
A.6. Scenario Bluetooth (BTDH5)

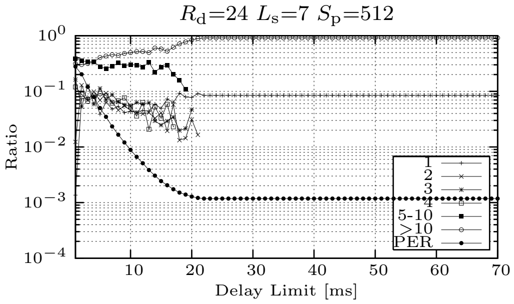
Data Rate R_{d}	Retry Limit L_{s}	Packet Size S_{p}	Mean Delay $\overline{d}$	Jitter j	PER
54	7	128	1.1e-03s	2.5e-06s^2	1.4e-04
54	7	512	1.4e-03s	3.3e-06s^2	1.8e-04
54	7	1024	1.8e-03s	5.4e-06s^2	1.5e-04
24	7	128	1.1e-03s	2.3e-06s^2	1.7e-04
24	7	512	1.7e-03s	4.0e-06s^2	1.2e-03
24	7	1024	2.8e-03s	9.7e-06s^2	6.0e-05

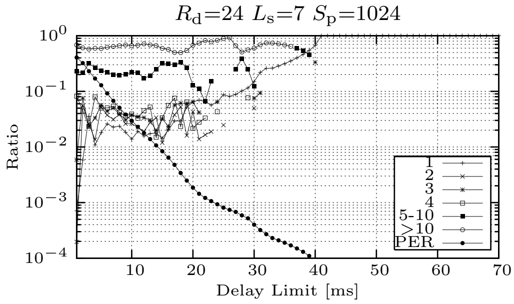












B. Various Information

B.1. Tricking the Network Stack

In the measurement setup used, packets are sent from one *Network Interface Card* (NIC) to an other. To avoid time synchronization problems, the two NICs are inserted in the same computer. When a packet is sent with a destination IP which belongs to a NIC in the same computer, it is routed inside the kernel. This means the packet is never transmitted over the wired or wireless link. This happens because for every NIC an entry in the kernel routing table exists. Deleting those entries is not an option, they are needed to determine if a packet is destined to a NIC of the computer on which the network stack runs.
The solution is to use two imaginary hosts with MAC addresses belonging to the NICs in the computer. Additionally, static *Address Resolution Protocol* (ARP) table entries together with source and destination *Network Address Translation* (NAT) are used to trick the linux network stack. The following code is an excerpt from the shell script used to set up the interfaces and the connection between them:

```
# 802.11g NIC:
WLANIF="ra0"
WLANIP="192.168.1.4"
WLANFAKEIP="192.168.1.14"
WLANMAC="00:11:09:9A:B1:C7"
ESSID="audiowlan"
# Ethernet NIC:
ETHIF="eth1"
ETHIP="192.168.1.2"
ETHFAKEIP="192.168.1.12"
ETHMAC="00:10:5A:72:93:88"

# set up both NICs:
/sbin/ifconfig $WLANIF netmask 255.255.255.0  -broadcast 192.168.1.255 $WLANIP up
/sbin/ifconfig $ETHIF  netmask 255.255.255.0  -broadcast 192.168.1.255 $ETHIP  up
/sbin/iwconfig $WLANIF essid $ESSID

# delete routing table entries generated by the ifconfig command
/sbin/route del -net 192.168.1.0 netmask 255.255.255.0 dev $WLANIF
/sbin/route del -net 192.168.1.0 netmask 255.255.255.0 dev $ETHIF

# add routing table entries for the imaginary hosts
/sbin/route add -host $ETHFAKEIP  dev $WLANIF
/sbin/route add -host $WLANFAKEIP dev $ETHIF

# add two static ARP table entries belonging to the imaginary hosts
/sbin/arp -i $ETHIF  -s $WLANFAKEIP $WLANMAC
/sbin/arp -i $WLANIF -s $ETHFAKEIP  $ETHMAC

# source NAT: change source of packet from imaginary IP to real IP. This happens
# when sending a packet, after the packet has been routed by the network stack.
/sbin/iptables -t nat -A POSTROUTING -o $ETHIF  -d $WLANFAKEIP -j SNAT --to $ETHFAKEIP
/sbin/iptables -t nat -A POSTROUTING -o $WLANIF -d $ETHFAKEIP  -j SNAT --to $WLANFAKEIP

# destination NAT: change destination of packet from imaginary to real IP. This happens
# when a packet is received, before the packet is routed by the network stack.
/sbin/iptables -t nat -A PREROUTING -i $ETHIF  -d $ETHFAKEIP  -j DNAT --to $ETHIP
/sbin/iptables -t nat -A PREROUTING -i $WLANIF -d $WLANFAKEIP -j DNAT --to $WLANIP
```

List of Abbreviations

ACK	Acknowledge
ALSA	Advanced Linux Sound Architecture
AWGN	Additive White Gaussian Noise
AP	Access Point
BER	Bit Error Rate
BSS	Basic Service Set
BSSID	Basic Service Set ID
CCK	Complementary Code Keying
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DIFS	DCF InterFrame Space
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EIFS	Extended InterFrame Space
ERP	Extended Rate PHY
ESS	Extended Service Set
FCS	Frame Check Sequence
FEC	Forward Error Correction
FHSS	Frequency Hopping Spread Spectrum
IAPP	Inter Access Point Protocol
IDFT	Inverse Discrete Fourier Transform
MAC	Medium Access Control
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
PBCC	Packet Binary Convolutional Coding
PCF	Point Coordination Function
PHY	Physical Layer
RTS	Request To Send
SIFS	Short InterFrame Space
SSH	Secure Shell
TDD	Time Division Duplex
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi–Fi Protected Access

List of Figures

2.1. MAC Frame Body 6

3.1. Aspects of the OSI/ISO model coverded by 802.11 10
3.2. Binary exponential backoff algorithm 11
3.3. Basic access mechanism 12
3.4. RTS/CTS mechanism 13
3.5. MAC data frame 14
3.6. MAC control frames 15
3.7. Constellations of the different modulation schemes 17
3.8. Bit error rates of the modulations used by 802.11g 18
3.9. DSSS-OFDM PHY frame 19
3.10. ERP-OFDM PHY frame 20
3.11. Block Diagram 20
3.12. BCM2050 Block Diagram. Source: [3] 21

4.1. Transmit time for N_{A} Bytes audio data 24

5.1. Measurement setup 28
5.2. Measurement setup inside the anechoic chamber 30
5.3. Used transmit slots for the three Bluetooth scenarios 34

6.1. Delay dependant burst statistic 36
6.2. Different retry limits 37
6.3. Data rate and packet size with weak signal 38
6.4. Data rate and packet size with microvawe oven interference 39
6.5. 6Mbit/s data rate with microwave oven and Bluetooth interference 39

7.1. Redundancy 41

8.1. Frame formats of the audio transmission system 43

List of Tables

3.1. WLAN Overview 9
3.2. DCF timing intervals 12
3.3. Address fields 14
3.4. 802.11g data rates 17
3.5. DSSS-OFDM frame fields 19
3.6. ERP-OFDM frame fields 20
3.7. BCM2050 Specifications 22

4.1. Path-loss exponents for different environments. (Source: [4]). 23

5.1. Parameters for each measurement 32
5.2. Bluetooth scenario parameters 34

6.1. Transmission characteristics for different retry limits 37

Bibliography

[1] S. Hacker, *MP3: The Definitive Guide*. O'Reilly, 2000.

[2] *Fraunhofer IIS ULD Audio Software*. Frauenhofer IIS. Available from: `http://www.iis.fraunhofer.de/amm/download/wp_iis_uld_1.4.pdf`.

[3] J. Trachewsky, A. Rofougaran, A. Behzad, T. Robinson, and E. Frank, eds., *Broadcom WLAN Chipset for 802.11a/b/g*, Broadcom Corporation, CA, USA, 2003.

[4] H. Mathis, *Mobile Communications (lecture notes)*. HSR, 2004.

[5] D. Dobkin, ed., *Indoor Propagation issues for wireless LANs*. RF EMC/RFI, 2002. Available from: `http://rfdesign.com/mag/radio_indoor_propagation_issues/` [cited July 03 2005].

[6] IEEE, *IEEE-Std 802.11-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 1999.

[7] IEEE, *IEEE-Std 802.11b-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*. 1999.

[8] IEEE, *IEEE-Std 802.11g-2003 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*. 2003.

[9] R. Desquiotz, *Application Note 1GP38_0E: Creating Test Signals for Bluetooth with AMIQ / WinIQSIM and SMIQ*. Rohde&Schwarz, 1999. Available from: `http://www.rohde-schwarz.com/appnote/1GP38.html` [cited June 28 2005].

[10] R. Sinha, C. Papadopoulos, and C. Kyriakakis, eds., *Loss Concealment for Multi-Channel Streaming Audio*, 2003.