DIPLOMA THESIS

Interference Analysis of WLAN Communication

supervised by

Univ. Ass. Dipl.-Ing. Bernhard Pichler
and
Assoc. Prof. Dipl.-Ing. Dr. techn. Holger Arthaber

performed at the
Institute of Electrodynamics, Microwave and Circuit Engineering

by

Christian Spindelberger, BSc.
Matr. Nr. 01226143
Erndtgasse 9–11/4, 1180 Wien

Vienna, July 2019

Abstract

The IoT is a strongly growing branch, aiming to connect all kinds of devices exchanging information. As there is a heavy rush to this market, IoT became one of the biggest success stories in the last decade. In order to lower financial access barriers to this sector for small companies, such as start-ups, this thesis addresses low-cost implementations for device-testing methods in terms of interference. As IoT devices utilize ISM band-related communication standards, e.g., WLAN and BLE, a measurement campaign was held to describe the interference scenario of WLAN channels. Two methods have been found to analyze the traffic (time-quantized analysis and energy detection) and extract parameters for interference emulation. These parameters randomly alter the power level and the respective frame length of the interfering signals. Furthermore, an interference measurement setup based on a WLAN communication system was implemented. This setup was used to verify the introduced emulation techniques by measuring two typical performance parameters, i.e., PER and throughput. In addition to this, the performance of alternative low-cost interference sources was compared to professional RF equipment. As the ISM band traffic was dominated by WLAN communications, the first approach was to emulate interference by utilizing a WLAN module. However, because of configuration limitations of such systems, this attempt has been dropped. Therefore, similarities between common modulation techniques, such as OFDM, and noise, regarding their amplitude distributions were exploited. It was shown that it is possible to emulate ISM band traffic with a simple-structured noise source. The best performance results, compared to RF instruments, were achieved by applying the time-quantized method for modeling noise bursts.

Kurzfassung

Die immens wachsende IoT Branche zielt darauf ab, möglichst viele Geräte miteinander zu vernetzen und Informationen auszutauschen. Da ein großer Ansturm auf diesen Markt zu verzeichnen ist, stellte IoT sich als einer der größten Erfolgsgeschichten der letzen Jahre heraus. Thema der vorliegenden Diplomarbeit ist somit, die finanziellen Einstiegsbarrieren der genannten Branche auch für kleinere Firmen, wie zum Beispiel Start-ups, zu reduzieren. Es wurden kostengünstige Implementierungen für die Charakterisierung von IoT Geräten unter Einfluss von Interferenzen erarbeitet. IoT verwendet für das ISM Band typisch zugelassene Kommunikationsstandards, z.B. WLAN und BLE. Aus diesem Grund wurde eine Messkampagne abgehalten, um das Interferenz Szenario von WLAN Kanälen statistisch zu beschreiben. Zwei Methoden, die Zeit-Quantisierung sowie die Energiedetektion, ermöglichten es, den aufgezeichneten Datenverkehr zu analysieren. Die daraus gewonnenen Parameter bestimmen die Leistung und die Länge des Inteferenzsignals. Des Weiteren wurde ein Messaufbau errichtet, der auf einem WLAN Netzwerk basiert. Hiermit wurden die Auswirkungen von Interferenzen mit typischen leistungsbezogenen Netzwerkparametern, wie Paketfehlerraten und Datendurchsatz, zu beschrieben. Zuzüglich wurden alternative preiswerte Interferenzquellen mit professionellem Hochfrequenz Equipment verglichen. Da die ISM Band Daten überwiegend WLAN basierende Kommunikationen aufwies, erfolgte vorerst die Annahme, Inteferenzen mit einem WLAN Modul zu emulieren. Wegen limitierter Konfigurationsmöglichkeiten solcher Systeme wurde dieser Ansatz jedoch nicht weiterverfolgt. Vielmehr wurde die Ähnlichkeit der Amplitudenverteilung von Modulationstechniken, wie OFDM, und Rauschen ausgenutzt. Es konnte gezeigt werden, dass es möglich ist, ISM Band Interferenzen mit einer simpel aufgebauten Rauschquelle nachzuahmen. Verglichen mit Hochfrequenz Equipment erzielte die Anwendung der Zeit-Quantisierungs Methode, zur Modellierung von Rauschsignalen, die besten Ergebnisse.

Acknowledgements

First of all, I would like to thank Assoc. Prof. Dipl.-Ing. Dr. techn. Holger Arthaber. He is one of the most talented and experienced people accompanying me in my young career as an engineer. With his comprehensive knowledge, he guided me through my thesis, spending me support whenever I needed it.

Furthermore, the EMCE microwave engineering working group deserves my highest gratitude and respect. Especially, Bernhard Pichler, Jure Soklic, Daniel Neunteufel, and Florian Galler enriched me with innovative ideas and a high technical expertise.

Lastly, I would like to emphasize the unconditional encouragement by my parents and my better half, Lena. It would not have been possible to reach my diploma without their constant support in times of doubt.

Contents

1 Introduction **1**

2 Theory **4**

2.1 IEEE 802.11 Wireless LAN 4

2.1.1 Medium Access Control Layer 5

2.1.2 Physical Layer 10

2.1.3 Higher-Layer Protocols 15

2.2 Interference Sources 16

2.2.1 Bluetooth Low Energy 17

2.2.2 Short Range Devices 20

2.2.3 Microwave Ovens 20

2.2.4 Radar Systems 21

3 Measurement Campaign **23**

3.1 Setup and Location 23

3.2 Data Analysis 26

3.2.1 Time-Quantized Analysis 27

3.2.2 Energy Detection 29

3.2.3 Classifying Frames 31

3.3 Results 33

4 Measuring Interference **40**

4.1 Test Setup 40

4.1.1 Measurement Setup Considerations 42

4.1.2 Performance Measurements 44

4.2 Injecting WLAN Frames with Linux 45

4.3 Modulated Noise 46

4.3.1 Simulation Model 47

4.3.2 Implementation of a Low-Cost Interference Source 52

5 Interference Emulation Results **54**
5.1 Reference Measurements 54
5.2 Modulated Noise 56
5.3 Low-Cost Noise Source 62
5.3.1 Memory Depth 62
5.3.2 Reference Power Levels 62
5.3.3 Time Critical Behavior 63
5.3.4 Comparison of Results 63

6 Conclusion **65**

References **67**

A WLAN Setup Instruction Guide **69**

Abbreviations

ACK acknowledgment

AWGN additive white Gaussian noise

BLE Bluetooth low energy

BPSK binary phase shift keying

CCA clear channel assessment

CP cyclic prefix

CTS clear to send

DIFS distributed interframe space

DSSS direct spread spectrum sequence

ED energy detector

EIRP effective isotropic radiated power

FCS frame check sequence

FEC forward error correction

FFT fast Fourier transform

FH frequency hopping

GFSK Gaussian frequency shift keying

IoT internet of things

IP internet protocol

ISM industrial, scientific, and medical

IFS interframe space

IFFT inverse fast Fourier transform

LAN local area network

LLC logical link control

LTF long training field

MAC medium access control

MF matched filter

MTU maximum transmission unit

NAV network allocation vector

NF noise figure

OFDM orthogonal frequency division multiplexing

OSI open systems interconnection

PAPR peak-to-average power ratio

PDF probability density function

PER packet error rate

PHY physical

PLCP physical layer convergence procedure

PMD physical medium dependent

QAM quadratue amplitude modulation

Q-Q quantile-quantile

RF radio frequency

RIFS reduced interframe space

RSSI received signal strength indicator

RTS request to send

SIFS short interframe space

SINR signal to interference plus noise ratio

SRD short range device

STF short training field

TCP transmission control protocol

UDP user datagram protocol

VSA vector signal analyzer

VSG vector signal generator

WLAN wireless local area network

Chapter 1

Introduction

The internet of things (IoT) has been one of the biggest success stories in the 21$^{\text{st}}$ century. Because of a huge amount of applications in several branches, IoT gained an exponentially growing device density. The main idea behind this technology is to link all kinds of smart "things", as for instance, smart wearables or electric meters in buildings. Clearly, such devices exchange information with main base stations, but they are also capable to create wireless meshes among each other. Hence, IoT can be divided into two sections, called critical- and massive IoT. The critical applications require a low latency, a high throughput, and reliability. For instance, use cases like healthcare and autonomous driving insist on these properties. The second section, massive IoT, refers to less critical applications, but demanding low-cost, low-power-consumption devices on a network covering wide areas [1]. As this sector is probably the fastest growing one, this thesis relates to the corresponding use cases.

Several technologies have been implemented to meet the mentioned requirements of a high device density and geographical reach. Figure 1.1 indicates the use case of the respective communication standards. Currently, 5G MTC[1] is not available but is likely to play an important role for high device densities and ranges in the future. Because of these high performance aspects, it seems natural that 5G MTC will demand high financial investments. Sigfox, NB-IoT, and LoRa[2] are made for communication systems penetrating buildings over large distances. The remaining standards, i.e., wireless local area network (WLAN) and Bluetooth low energy (BLE), are the most common ones in the IoT sector. Because of an exponential increase of IoT devices, WLAN, for instance, can be utilized to create a

[1] 5G machine type communications (MTC) is based on the 5$^{\text{th}}$ generation of cellular networks beyond LTE, optimized for high performance applications in the IoT branch.

[2] Sigfox, narrowband (NB)-IoT, and long range wide area (LoRa) networks are standards using the ISM band with a narrowband spectrum.

wireless mesh extending the respective range. Consequently, WLAN and BLE communication systems will be examined in terms of co-existence issues and interference capabilities. Furthermore, details about working principles and further interference sources will be given in Chapter 2.

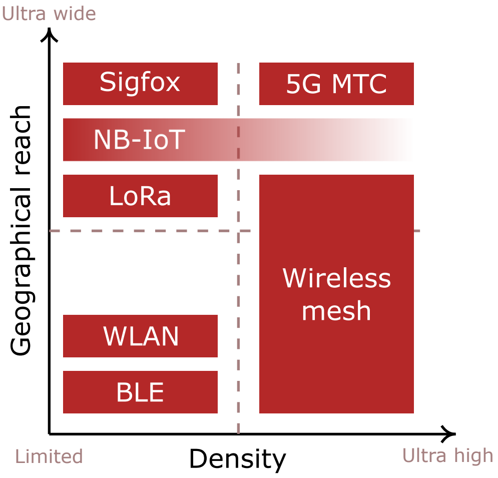


Figure 1.1: Standard applications depending on geographical reach and density [1]

As a consequence of high device densities, interference mitigation became of huge interest. Investigating such scenarios scientifically, with appropriate radio frequency (RF) equipment, is one of the biggest obstacles for financially weak companies. Therefore, alternative solutions will be presented to overcome this problem. Firstly, a measurement campaign, described in Chapter 3, will provide data for characterizing interference signals. Especially Linux based systems offer many tools for eventually replacing expensive measurement instruments through their open source character. Therefore, this campaign will also indicate whether a vector signal analyzer (VSA), utilized as a baseband signal recorder, can be replaced by such a Linux operating system.

Analyzing interference effects basically works by measuring established traffic speed and error numbers, such as retransmissions and the packet error rate (PER), while an interference source perturbs the data exchange. In order to describe these effects, a measurement setup consisting of a WLAN network is created in Chapter 4. As a reference interference source, a vector signal generator (VSG), which is capable of replaying customized digital data with a high time resolution and dynamic range, is utilized. As these devices tend to be expensive, further cost-saving solutions obtaining a sufficient interference source will be treated. At last, in Chapter 5, the presented techniques and implemented devices are compared to professional RF equipment, utilizing interference signal recordings from the measurement campaign.

In conclusion, the main goals of this thesis are to describe interference signals in the industrial, scientific, and medical (ISM) band statistically and utilize the observed model to emulate the recorded scenario by different techniques. In order to lower financial investments, the presented techniques will be applied to alternative interference sources. For verification of the introduced concepts, a measurement setup based on WLAN, describing interference in terms of throughput and error rates, is utilized. Finally, Chapter 6 summarizes the whole thesis and gives a forecast to further improvements and ideas.

Chapter 2

Theory

This chapter provides a theoretical background of WLAN communication systems. The aim is to make the reader familiar with relevant concepts and parameters utilized in this thesis. It mainly consists of a WLAN related part, discussing important topics such as layer stack up, collision avoidance, and transmission parameters, while common interference sources (BLE, microwave ovens, SRDs, and radars) affecting WLAN are presented in the second part.

Starting with Section 2.1, an introduction about the WLAN-related layer stack up and similarities to other transmission standards is given. Proceeding with this topic, the two most important parts concerning this work are discussed. The medium access control (MAC) is described in terms of timing constraints, collision avoidance, and layer interaction. Furthermore, physical (PHY)-layer parameters, such as modulation techniques (OFDM, DSSS) and carrier sensing, are examined. In order to explain these sublayer interactions, practice relevant examples utilizing the user datagram protocol (UDP) and transmission control protocol (TCP) are given. Lastly, basics about potential interference sources will be presented in Section 2.2. BLE is one of the most serious interferers concerning WLAN. Therefore, details about the PHY layer and channel access schemes are investigated. The final subsections are devoted to further interferers, i.e., microwave ovens, short range devices (SRDs), and radars.

2.1 IEEE 802.11 Wireless LAN

WLAN is a widely deployed standard for data transmission in the ISM band. Usually, it is intended to be used as a wireless access point to a network (router). Since the first launch in 1997, the IEEE 802.11 standard has undergone several improvements [2]. Today, different versions are available, such as IEEE 802.11a/b/g/n/ac and many more. In the IoT branch, the state-of-the-art WLAN standard is IEEE 802.11n, implemented with one single antenna. Therefore, this work will focus on this respective version. The current section about IEEE 802.11 WLAN properties is based on *802.11 Wireless Networks: The Definitive Guide*, written by Matthew Gast [3].

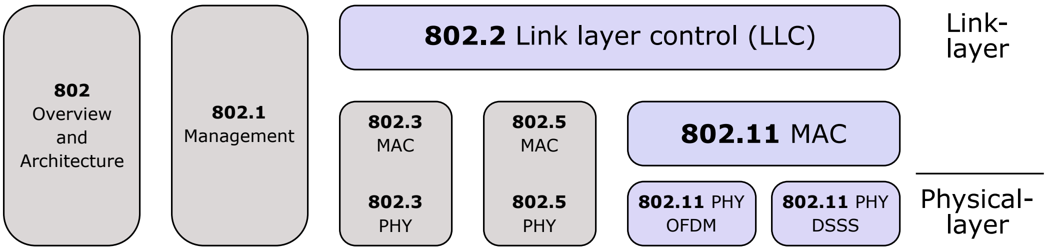


Figure 2.1: OSI model of IEEE 802.11 embedded in IEEE 802.2 LLC [3]

As the layer composition of IEEE 802.11 strongly relates to IEEE 802-based networks, i.e., local area network (LAN), the low level constructs (MAC & PHY) must fit into the required open systems interconnection (OSI) model (Figure 2.1). The most important parts, considering WLAN, are marked in violet. It is obvious that all IEEE 802 networks have a MAC and a PHY component. The classical data link layer is responsible for an error free transmission by calculating checksums or making use of channel coding. While the MAC determines specific rules for collision avoidance and how to access the medium, the PHY layer is related to details about data reception and transmission. Thus, WLAN uses the IEEE 802.2 logical link control (LLC) encapsulation, which makes it very powerful because it can utilize higher layer protocols (UDP, TCP). Details about the sublayer realizations will be discussed in the following.

2.1.1 Medium Access Control Layer

The access medium of IEEE 802.11 systems is a wireless radio channel and demands specific MAC properties to ensure stable data transmissions. Especially for unlicensed radio channels, e.g., the ISM band, many different interferers such as BLE and microwave ovens occur. Therefore, to check if a transmission was successful, WLAN uses acknowledgment (ACK) frames for confirmation. Figure 2.2 depicts, how a simple frame transmission between two stations works. Station one (*STA 1*) transmits the desired frame to receiver station two (*STA 2*). If the transmission was successful, *STA 2* replies with an ACK frame. Detecting a damaged packet forces *STA 2* to omit an answer. Depending on the utilized protocol, a retransmission will be sent if the ACK frame was missing.

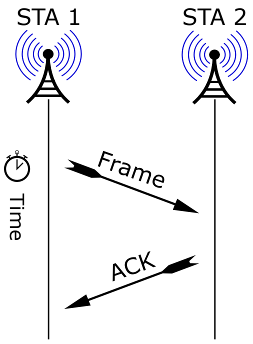


Figure 2.2: Valid WLAN frame transmission [3]

Hidden Node Problem

A major problem arises concerning multi-user scenarios. Because of limited transmit powers, free space losses, channel variations, and several other effects, the range of a station is limited. This makes it impossible to reach those users that are too far from the station. Assume the following scenario: One access point (server) and two further stations (clients) are present. The first client (*client 1*) wants to start a data exchange with the *server*, but the second client (*client 2*) is too far away to receive any message from *client 1*. What now happens, is that the transmission is corrupted if *client 2* also starts a conversation with the *server* at the same time. This is called the hidden node problem.

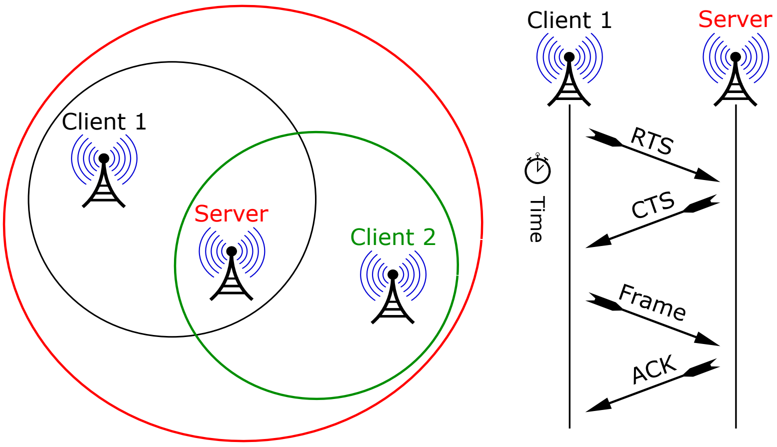


Figure 2.3: Hidden node problem: distribution of the involved stations and their ranges sketched by colored circles (left), utilization of the RTS-CTS transaction over time (right) [3]

To prevent such errors, the MAC is able to introduce request to send (RTS) and clear to send (CTS) frames. Now, before *client 1* starts a transmission, it first sends an RTS frame and waits for a CTS reply by the *server*. Then, *client 2* also receives this frame and holds the channel free. Figure 2.3 shows this scenario in detail. On the left side, three WLAN stations are visible. The colored circles mark the respective ranges. Obviously, *client 1* and *2* cannot reach each other. Consequently, the RTS-CTS transaction is performed (right side of Figure 2.3).

Generally, long data frames tend to be corrupted in a noisy environment. Therefore, WLAN offers two ways to mitigate this problem. The first one is to fragment long sequences into short ones and the second is to implement an RTS-CTS exchange. The indicator for such a transaction is given by the RTS threshold. If the frame length is larger than the defined threshold, an RTS is utilized. Usually, the fragmentation and RTS threshold are of the same size.

Network Allocation Vector

The MAC header of WLAN frames (Figure 2.6) introduces the signal duration field, also called network allocation vector (NAV), which can be used to hold the channel free from WLAN interference for a defined time period. This method is called virtual carrier-sensing. Figure 2.4 depicts a transmission sequence between sender and receiver with an RTS-CTS transaction.

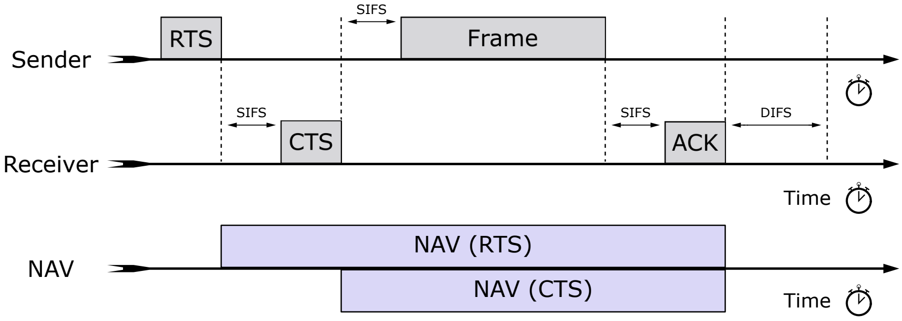


Figure 2.4: Signal duration indicator [3]

When a frame transmission between two stations starts, the NAV is also received by all other reachable stations and each of them starts a timer. In this time, the channel is held free from WLAN interference until the timer elapses. Furthermore, the NAV can be updated if the frames have been fragmented before. It must be mentioned that the NAV is also sent for transmissions without an RTS and CTS sequence.

Interframe Spacing

One can notice the marked spaces SIFS and DIFS in Figure 2.4. They are called interframe spaces (IFSs). IFSs are utilized to coordinate the channel access among multiple WLAN stations. In IEEE 802.11, several different types are defined depending on the frame type. In the following, the most important ones will be discussed. Figure 2.5 depicts how they relate to each other.

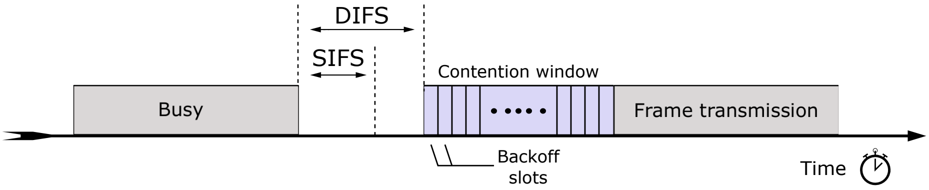


Figure 2.5: IFS: time constraints which have to elapse before proceeding with a new transmission [3]

Short interframe space (SIFS):
Only high-priority events, such as RTS, CTS, and ACK frames are allowed to transmit after one SIFS has elapsed. If such a high-priority transmission has started, the medium becomes busy. Thus, frames transmitted after SIFSs have a higher priority than, for example, frames that are transmitted after DIFS.

Distributed interframe space (DIFS):
Determines the minimum idle time which must elapse before a contention based transmission can be started. One DIFS is followed by a contention window with 31 slots. These slots are chosen randomly by a station and are equally distributed. Now, the station which chooses the first slot, relative to the others, wins and is allowed to transmit a frame afterwards. If two stations choose the same slot and interfere with each other, another greater contention window is utilized.

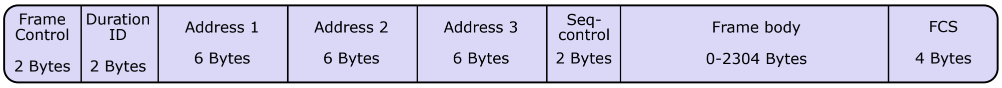
IFS	2.4 GHz	5 GHz
SIFS	10 µs	16 µs
DIFS	28 µs	34 µs
RIFS	2 µs	2 µs

Table 2.1: IFS durations for IEEE 802.11n

Table 2.1 provides information on typical IFS lengths in IEEE 802.11n. In order to decrease latency effects, another type is introduced, called the reduced interframe space (RIFS). It is the shortest space used in IEEE 802.11n. Since newer WLAN revisions launched to the market, RIFSs are often deactivated per default in networks to maintain compatibility.

Frame Format

Corresponding to Figure 2.1, WLAN utilizes the IEEE 802 LAN topology. Hence, it makes use of the internet protocol (IP) for linking with other stations. In order to address a specific receiver under the mentioned challenges in a wireless data link, the MAC layer generates defined frames. For instance, such sequences let the receivers know if they have to respond or not. Figure 2.6 depicts a generic MAC frame structure.



Frame Control	Duration ID	Address 1	Address 2	Address 3	Seq-control	Frame body	FCS
2 Bytes	2 Bytes	6 Bytes	6 Bytes	6 Bytes	2 Bytes	0-2304 Bytes	4 Bytes

Figure 2.6: Generic IEEE 802.11 MAC frame [3]

The MAC frame is a very powerful part with plenty configurable properties. Therefore, only the most important ones are presented in the following.

Frame control:
Gives the receiver information about the frame type. In the IEEE 802.11 standard, three different types exist: management, control, and data frames. Management frames are typically broadcasted by access points or association communications. In the section about hidden node problems, RTS and CTS have been introduced, which are control frames. The last type is, due to its name, self explanatory and simply transmits data.

Duration/ID:
The most important use case of the duration field is the virtual carrier-sensing called NAV. With this information each receiver knows for how long the medium will be busy.

Address fields:
The address fields contain the MAC addresses of the receiver and transmitter. The third address is used for filtering the basic service set ID to identify different networks (access points) in the same area.

Frame body:
Moves data between two stations corresponding to higher layer protocols, such as UDP or TCP. The maximum payload in IEEE 802.11 is a data amount of 2,304 Byte. However, in conventional systems, the maximum amount of data is 1,500 Byte. This length is determined by the maximum transmission unit (MTU) size of a system, which will be discussed

in Section 2.1.3.

FCS:
The frame check sequence (FCS) provides the receiver with information whether the received data packet is damaged. The transmitter sends the FCS within the MAC frame and the receiver recalculates the FCS. The transmission is expected to be correct if the two check sums match. Otherwise, the ACK frame is not sent back and a retransmission is forced, depending on the used protocol.

At the end, the total MAC frame is passed to the PHY layer, which applies further operations onto data.

2.1.2 Physical Layer

The lowest sublevel of the IEEE 802.11 architecture is the PHY layer. In this section, the working principle and further common topics, which appear in the PHY-layer management, will be discussed in detail.

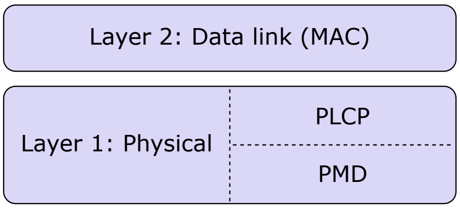


Figure 2.7: Interaction between MAC and PHY [3]

Figure 2.7 depicts the PHY-layer structure. It consists of two main parts, the physical layer convergence procedure (PLCP) sublayer and the physical medium dependent (PMD) sublayer. The PLCP sublayer can be interpreted as the interface between MAC and PHY layers. It adds a specific header for frame detection and synchronization to the frame passed by the MAC. Lastly, the PMD sublayer transmits the provided data from the PLCP sublayer by using the RF front-end. Furthermore, the PHY layer utilizes another important application to mitigate interference perturbations, called clear channel assessment (CCA). The CCA indicates if the medium is idle or busy. This information is directly passed to the MAC for collision avoidance.

Modulation

The PMD sublayer of current WLAN systems is capable of two different modulation types, called direct spread spectrum sequence (DSSS) and orthogonal frequency division multiplexing (OFDM). DSSS is the older version of the two, but still in use in the IEEE 802.11b standard. With this technique, a bitrate of 11 Mbits/s at 22 MHz bandwidth is achieved. In newer standard revisions, the OFDM modulation has carried through due to its higher data rates and smaller bandwidth. The main advantages of OFDM are robustness against multipath fading and simplicity of channel estimation. As IEEE 802.11n utilizes the OFDM modulation only, the focus in this section will lie on this technique.

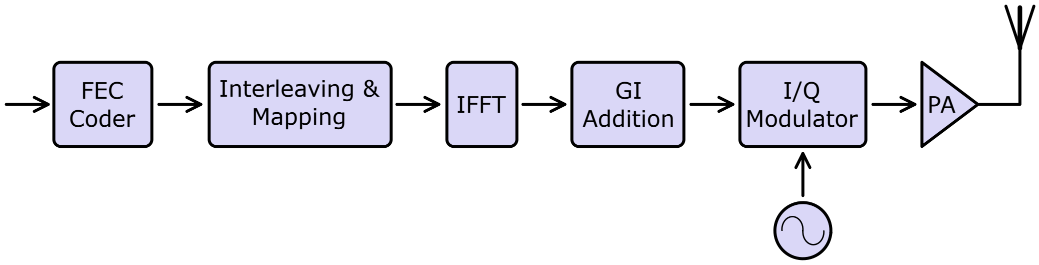


Figure 2.8: IEEE 802.11 transmission chain [2]

Figure 2.8 depicts the transmitter chain of the PMD sublayer in IEEE 802.11n (single antenna). At the beginning, a given bitstream is encoded by a forward error correction (FEC) coder. Afterwards the data is interleaved, together with the encoding, ensuring a better performance due to added redundancy and data reordering. The following path can be viewed as an OFDM modulator (Figure 2.9), which consists mainly of two parts, the OFDM modulation and the cyclic prefix (CP) extension.

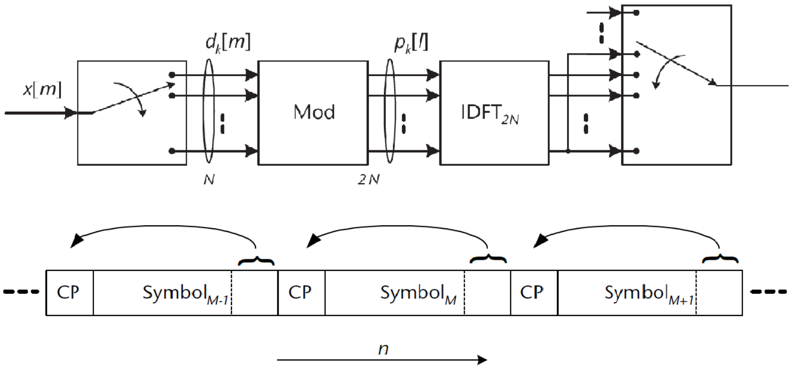


Figure 2.9: OFDM modulation (top) and applying the cyclic prefix extension (bottom) [4]

The encoded bitstream is mapped onto a desired scheme, for instance, quadratue amplitude modulation (QAM) or binary phase shift keying (BPSK). Afterwards, the serial symbol sequence is transposed into a column vector. Now, the next step is to apply an inverse fast Fourier transform (IFFT). At last, the CP is added to the OFDM symbol. The CP is a fractional part of the created OFDM symbol after applying the IFFT. It is appended to the beginning of the respective symbol, yielding several advantages discussed in the following. After another transpose to a row vector, the baseband signal is mixed up to the desired frequency and passed through the channel.

The cyclic prefix, together with the fast Fourier transform (FFT), offers a major advantage. The receiver stage detects a WLAN frame, strips off the CP and demodulates the signal accordingly with the FFT. Thus, the detected signal is in the frequency domain and the channel estimation results in a simple mathematical division. More details about applying these techniques will be discussed in Section 4.3.1. Nonetheless, the FFT ensures orthogonality only if the signal is periodic, which is established with the CP. Through the periodicity introduced by the CP, the convolution with the channel is circular. Therefore, with its use, it is possible to absorb all multipath components up to a delay which is not longer than the CP itself. This technique makes OFDM robust against multipath fading.

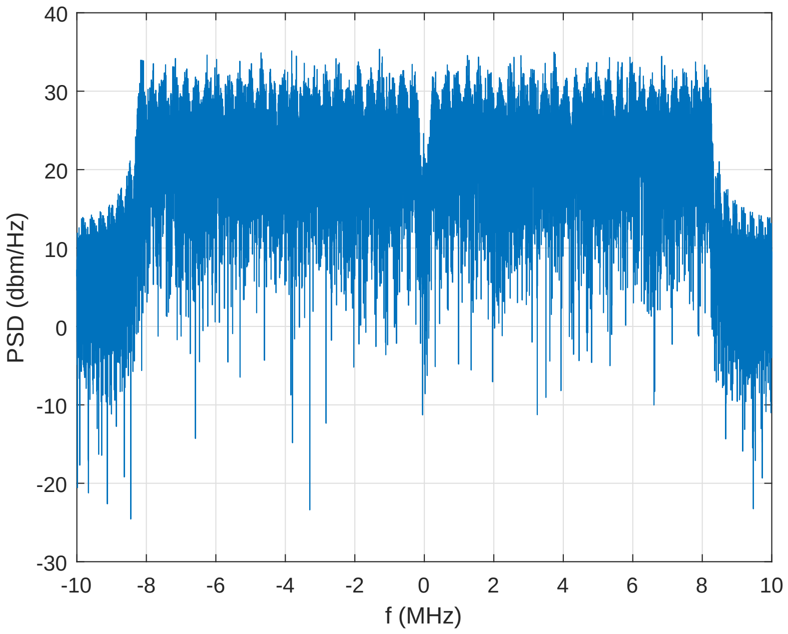


Figure 2.10: Spectrum of an IEEE 802.11a WLAN frame with BPSK mapping

WLAN shares the ISM band with lots of different standards, such as BLE, Sigfox, HiperLAN, and many more. Due to splitting the data into K parallel streams, according to the FFT size, the spectrum of OFDM is broadband ($\geq$ 20 MHz). Figure 2.10 depicts the spectrum of a WLAN frame according to the IEEE 802.11a standard. This broadband characteristic makes WLAN susceptible for interference perturbations. Therefore, this work will examine how different interference sources affect the overall performance of communication systems based on WLAN.

Preamble

As already discussed in the previous section, the PLCP sublayer introduces a preamble for frame synchronization and initial channel estimation. In Figure 2.11, one can see the legacy preamble of an IEEE 802.11a WLAN frame. This sequence consists of a short training field (STF) and a long training field (LTF). The STF is used for coarse signal detection, diversity selection, etc., while the LTF is utilized for fine timing-, frequency-offset synchronization, and channel estimation.

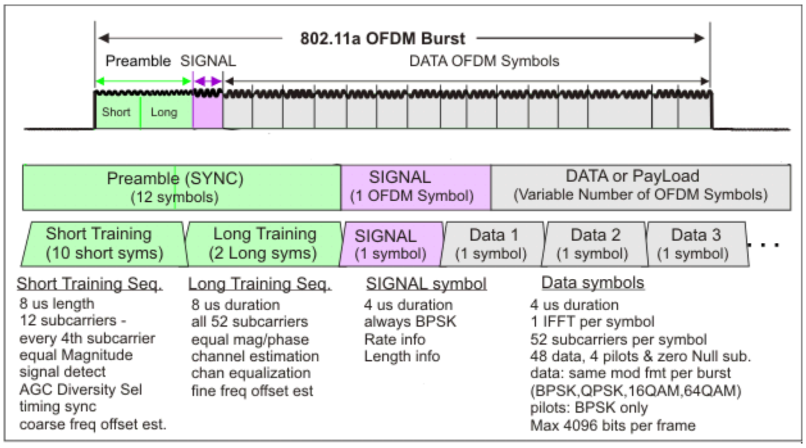


Figure 2.11: Preamble of an OFDM IEEE 802.11 frame [5]

The STF field consists of a ten times repeated 0.8 µs long training sequence. These fields are generated according to a Barker sequence. This type ensures that the autocorrelation function is minimal at off-peak values. Due to this repeated structure, a certain algorithm, invented by Schmidl and Cox, can be implemented for coarse frame detection [6].

The LTF field is again 8 µs long, but consists of two repeated sections including a long guard interval. This field is used for fine timing- and frequency-offset synchronization by implementing, for instance, a matched filter. More details about detection and synchronization schemes will follow later.

The signal (SIG) field gives the receiver information about the used modulation, coding rate, frame length, and used standard. In order to provide compatibility for all standards, the symbol mapping of the SIG field is always done by utilizing BPSK. Since newer standards, like the IEEE 802.11n, make use of multiple antennas, beamforming, and several diversity techniques, the preamble becomes longer to provide the required information.

Clear Channel Assessment

In the previous section about the MAC layer, the NAV was introduced. This vector is used for virtual carrier sensing, providing a technique for collision avoidance in a multiuser scenario. In order to detect further interfering signals, such as BLE or microwave ovens, the PHY layer adds another method to mitigate interference effects.

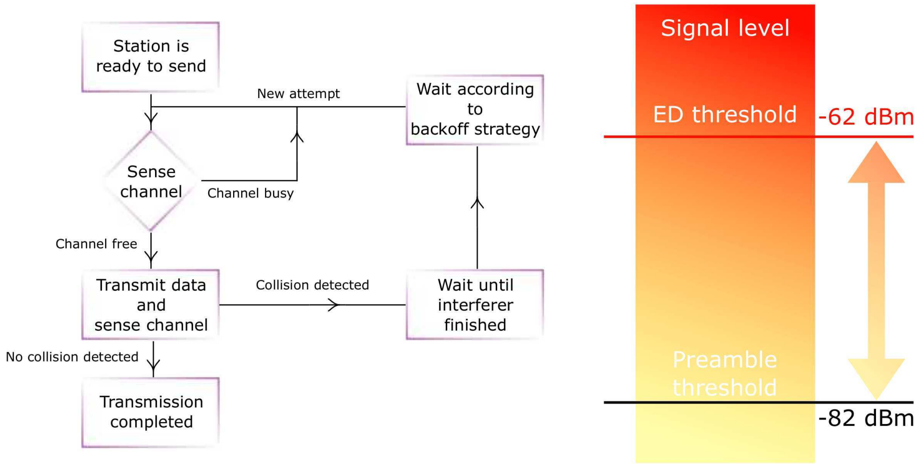


Figure 2.12: CCA: flow diagram (left), channel sensing thresholds (right) [2]

Before every transmission, the physical layer starts a CCA sequence according to Figure 2.12. Before a station is allowed to transmit, it listens to the channel, finding out if it is busy or not. The channel sensing consists of two parts. Firstly, a preamble detection with the Schmidl and Cox algorithm is performed. Secondly, an energy detector, which samples data within 4 µs, is implemented. In the IEEE 802.11a standard, it is specified that the preamble detection must work down to a power level of -82 dBm and the energy detection down to -62 dBm in order to detect every other kind of interference. It must be mentioned here, that state-of-the-art WLAN modules do achieve a much better preamble detection power level of about -92 dBm. If the channel is determined to be free for at least a distributed interframe space (DIFS), a contention window is started. After transmitting the desired frame, the channel is sensed for collision detection again. In the case of detecting interference, violating IFS timing constraints, the backoff strategy is started again. Otherwise, the transmission is completed.

It must be emphasized that the channel sensing scheme becomes a threshold decision problem and influences the performance of WLAN systems. The throughput between two stations might be reduced significantly if the threshold is set to low values in crowded areas. This problem could be skipped if the threshold is set to the maximum of −82 dBm. Consequently, the received signal power of the two stations must be high enough to neglect the residual interference beyond this power level. One must notice that this solution will only work if the stations are close to each other, since the transmit power in WLAN is limited to about 20 dBm. In addition to this aspect, the decision threshold becomes important for power-critical applications, such as battery-powered devices. Typically, stations adjust their transmit power according to a received signal strength indicator (RSSI) for saving energy. Naturally, this tactic succeeds only if the receiver does not neglect the utilized power level, attenuated by the channel.

2.1.3 Higher-Layer Protocols

WLAN utilizes the IEEE 802 LLC encapsulation. Therefore, it is able to work with higher OSI layers. In Figure 2.1, only the first two layers are presented, but the general model consists of seven layers in total. As the goal of this work is to describe interference mechanisms in WLAN systems, it is necessary to characterize them. In order to measure interference effects in real world systems, PERs, throughput, and retransmissions are investigated. Hence, to understand how interference effects on WLAN systems can be characterized, two more layers have to be considered.

Network Layer

This layer mainly routes the best logical paths between two nodes for an efficient data exchange. Typical hardware devices which relate to the network layer are, for instance, routers, bridges, and switches. In addition to this, it is capable to transmit variable data lengths. If the message is too large for the utilized communication standard, it is fragmented into frames. The maximum data length is defined by the MTU. The MTU size in Ethernet networks is defined to be 1,500 Byte long. Therefore, every payload data that is to be transmitted is fragmented into frames if it exceeds the MTU size. Furthermore, it is able to send these fragments independently and reassemble them again at the receiver side.

Transport Layer

The fourth instance of the OSI model establishes a reliable data link to a desired destination by maintaining quality of service functions, such as flow control, segmentation, and error detection. Depending on the used protocol, the transport layer can enforce retransmissions of packets due to a missing ACK frame. The two most important protocols for this work are the UDP and the TCP.

User datagram protocol (UDP):
UDP is a simple transport layer protocol. Because of the FCS introduced by the MAC frame, UDP is capable to detect errors. However, there is no guarantee that the message has been transmitted successfully. The transmission procedure starts by simply sending the frame. This means that no handshake with the receiver side is necessary. If the transmission was successful, an ACK frame is replied, but if not, the transmitter does not force a retransmission. Hence, UDP always sends packets with maximum data rate and will be used for PER and throughput tests in this work. It must be mentioned that this protocol is not applicable for services which require a high reliability. Therefore, it is often used for applications with an error tolerance, such as video streaming or gaming servers.

Transmission control protocol (TCP):
TCP is a more complex protocol compared to UDP. It implements a linking management, flow control, and error detection. These properties will be explained by a simple data exchange example. First of all, the transmitter establishes a connection with the receiver before a data transfer is started. After each packet, an ACK frame is sent back or not, depending on the error detection. If the ACK frame is missing, a timer is started, after which the damaged packet is retransmitted. During payload data transmission, TCP controls the throughput and adapts it according to occurring retransmissions. At the beginning, it starts with a small window size and increases it continuously until a retransmission is necessary. Thus, it reacts dynamically to interference effects. At the end of a data exchange, the connection is terminated. Consequently, the two involved stations are always in contact with each other to ensure a highly reliable connection. The main applications of TCP are, for example, email exchanges. In this work it will be used for throughput and retransmission tests for WLAN systems.

2.2 Interference Sources

Although WLAN is capable of various techniques for interference mitigation, the performance in terms of throughput, PER, and retransmissions is still limited. In the following, common interference effects and the respective sources will be described in detail.

Under ideal conditions, when a given channel is used only by WLAN systems with sufficient transmit power, no interference effects occur because of MAC- and PHY-layer constraints. Unfortunately, in real world scenarios, many distributed access points and their local associated clients share the same channel and the hidden node problem is inevitable. As already explained, this effect is handled by an RTS-CTS transaction. Nevertheless, this problem leads to transmission perturbations for packets that are smaller than the RTS threshold. Furthermore, if a DIFS (Figure 2.5) has elapsed after a transmission, it is statistically possible that some stations pick the same time slot of the backoff window. This would also lead to interference effects. Since these events do not consider co-existence problems with other standards or microwave ovens, the most important sources are investigated in the following.

2.2.1 Bluetooth Low Energy

At the beginnings of Bluetooth the main focus was put on connecting cell phones to laptops. Later on, its main application became establishing an audio link between headphones and a smart phone. As these applications required an increased throughput, the data rate of newer Bluetooth-standard revisions was increased from a basic rate of 1 Mbit/s to hundreds of megabits per second. It is clear that an increased throughput causes a reduced battery lifetime.

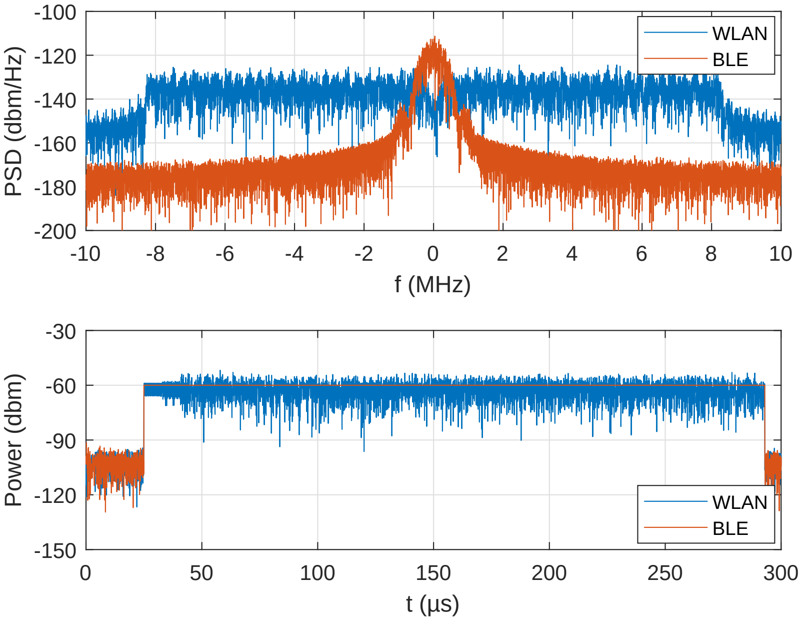


Figure 2.13: BLE vs. WLAN: spectrum (top) and envelope (bottom)

Therefore, BLE has been launched for low-power applications. It was designed for short lasting communications, for instance, incorporated by wireless sensors. Because of this low-power consumption, it is widely used in the IoT branch besides WLAN. Since BLE does not perform collision avoidance in higher layers (MAC layer), the subject of this section is how the PHY layer of BLE works and which co-existence problems arise [7].

Physical Layer

The PHY layer of BLE utilizes Gaussian frequency shift keying (GFSK) with a bitrate of 1 or 2 Mbits/s and an according bandwidth of 1 or 2 MHz. This modulation keeps the spectral efficiency high by optimizing transitions between symbols. Figure 2.13 depicts the spectrum and envelope of a BLE- compared to a WLAN packet. Because of the used modulation, the side lobes are decaying fast and ensure low co-channel interference. In comparison with OFDM, the envelope of GFSK signals appears to be almost constant while OFDM shows a noise-like behavior. As a consequence, these two modulation types presumably behave differently as interferers.

BLE uses up to 36 channels in the 2.4 GHz ISM band for data transmission and partly coincides with WLAN bands corresponding to Figure 2.14. Furthermore, it must be noticed that static advertiser channels are placed for device coupling communications outside of the main WLAN channels (1,6,11).

In the ISM band, the maximum transmit power is regulated to 10 dBm, which makes it capable of transmitting over long distances. Since modern BLE receivers achieve a receiver sensitivity down to −90 dBm, a minimum SNR of approximately 20 dB, which equals a distance of 250 m, is necessary for correct demodulation [7].

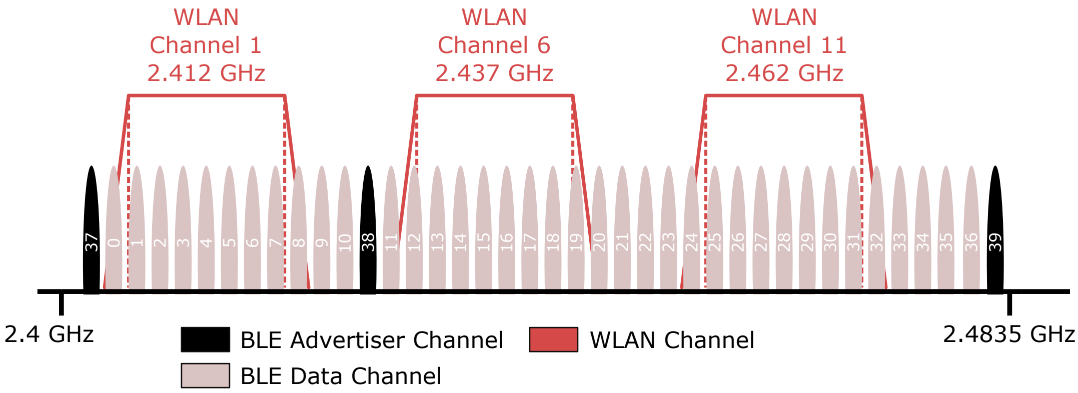


Figure 2.14: BLE channel separation interfering with main WLAN channels

Since the topic of this work is to investigate interference effects on WLAN communication systems, it is noteworthy that BLE does not perform any CCA before sending data. The reason for that is the implementation of frequency hopping (FH). Corresponding to Figure 2.14, 36 channels are used in a random fashion during data transmission (hopping). In the ISM band it is allowed to access one single channel for a certain time period. This ensures, that one channel is not blocked over a long time. Because of this frequency hopping technique, a co-existence problem between WLAN and BLE arises. The BLE channels are chosen randomly, but they might interfere with WLAN packets. In order to mitigate transmission perturbations, BLE improved its PHY layer for collision avoidance.

Adaptive Frequency Hopping

The ISM band is used by various different standards. Therefore, a channel sensing scheme is indispensable. Bluetooth utilizes adaptive FH to improve collision avoidance. This simple technique identifies channels which are already in use and remaps them onto channels expected to be free of interference.

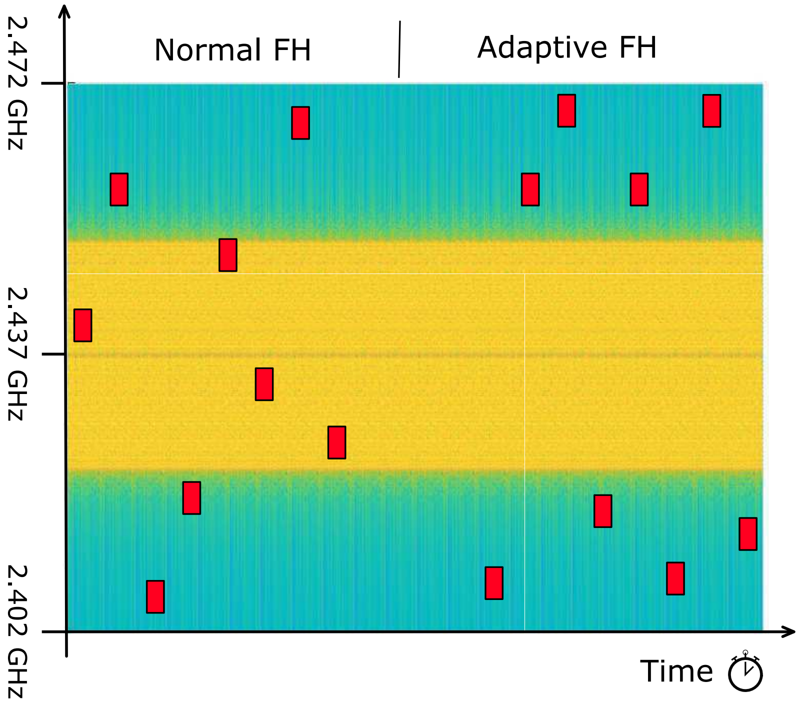


Figure 2.15: Adaptive frequency hopping: BLE (red), WLAN (yellow)

Figure 2.15 gives an idea, how this scheme works. Occupied channels are identified in terms of received signal strength indicator (RSSI), SNR, and PER. Nevertheless, it is necessary to maintain a minimum number ($N_{min} = 20$) of remaining channels in an interference scenario. Hence, if less than N_{min} channels are free of perturbations, also occupied ones are considered for transmitting data. Thus, this problem becomes more important for broadband systems, such as WLAN using modulations with a bandwidth of 40 MHz. Furthermore, because of fading effects, some channels may appear to be free even when they are not.

2.2.2 Short Range Devices

Besides BLE based communication systems, SRDs are becoming increasingly popular. Typical applications are, for instance, alarm systems, wireless audio, and wireless keyboards. SRDs are using almost all ISM bands. As this thesis relates to WLAN based systems, the 2.4 GHz band is investigated.

In the previous sections, different collision avoidance- and channel access schemes, such as carrier sensing (CCA utilized in WLAN) and FH (BLE), were discussed. Some SRDs use these techniques as well, but especially the CCA demands a high computational effort. In addition to this, low-power applications, such as wireless keyboards transmit very small packet sizes. Typical keyboards would get along with a 7-bit broad codebook, but through the utilization of additional encryption techniques and protocols the packets are about ~ 100 bit long. Hence, rudimentary channel access schemes, such as the duty-cycle based random access, are utilized. Stations are sending within one time frame T for a maximum time interval $T_s \ll T$. In order to prevent static collisions with other stations, the access of the channel is random. As collisions are likely to happen for high device densities, for instance, the non-slotted ALOHA protocol can be utilized. In order to enhance the reliability of data exchanges, successful transmissions are confirmed by ACK replies. Because of this simple working principle, the duty-cycle based random access scheme suits low data rate applications, such as wireless keyboards [8].

State-of-the-art SRD transceiver chips, such as the *CC2500* from *Texas Instruments*, are capable of all common modulation types (GFSK, QAM) and channel access schemes (FH, CCA). Depending on the application, an occupied bandwidth (99 %) between 91 kHz and 489 kHz at a maximum transmit power of 1 dBm is achieved. Furthermore, RSSI measurement functions are implemented. As these kind of integrated chips are highly flexible, SRDs are a serious interference source for WLAN communication systems. Especially low-power applications, using rudimentary channel access schemes, neglecting carrier sensing, presumably affect data exchanges extensively. In addition to this, the transmit power is high enough to perturb WLAN receivers close to SRDs.

2.2.3 Microwave Ovens

The widespread use of microwave ovens, for instance, in hospital environments affects WLAN systems operating in the 2.4 GHz ISM band. As microwave ovens usually have a high power level, compared to other communication standards, they can be detected easily.

The spectrum of microwave ovens spreads over a wide frequency range for long time intervals. It is possible to describe the emission characteristics by a narrowband signal with a bandwidth smaller than 1 MHz and a center frequency of 2.45 GHz. The broadband spectrum is modeled by significant variations of the center frequency. The radiation pattern can be approximated isotropic with a maximum effective isotropic radiated power (EIRP) level up to 33 dBm, while the mean EIRP is about 5 dBm. Furthermore, the periodically emitted RF power depends on several individual device characteristics, such as type, load, and make of the oven.[9].

As the emission pattern strongly depends on the type of the implemented power supply, Table 2.2 indicates some examples describing the time periodic behavior for different realizations. Obviously, the radiation pattern for the first two types extends over long time intervals and for the inverter types over short times, but the off-time is also much shorter. Hence, if the collision avoidance of a WLAN system would detect a microwave oven burst and no PER occurs, still the throughput would be decreased. As a consequence of these properties, it is clear that the throughput as well as the PER of WLAN systems may suffer significantly from this type of interference.

Type	**Emission pattern**
Transformer type	emits once per AC power cycle, every 20 ms
Switching type	emits twice per AC power cycle, every 10 ms
Inverter type 1	emits once per inverter switching-cycle typically, every 35 µs
Inverter type 2	emits twice per inverter switching-cycle typically, every 17.5 µs

Table 2.2: Switching characteristics of microwave ovens depending on power supply realization [9]

2.2.4 Radar Systems

Until now, interference sources of the 2.4 GHz ISM band have been discussed. But the 5 GHz band also suffers from perturbations caused, for instance, by weather radar systems. Radar signals are typically very short duration pulses. As no specific timing constraints, such as IFSs, have to be fulfilled it is a challenging task to identify such signals.

Refering to *ETSI EN 301 893 V2.1.1 (Annex D)*, radar signals can be described by different patterns [10]. The short pulses are sent periodically within one burst (L), demonstrated in Figure 2.16. Typical pulse width parameters are between 0.5 µs and 30 µs.

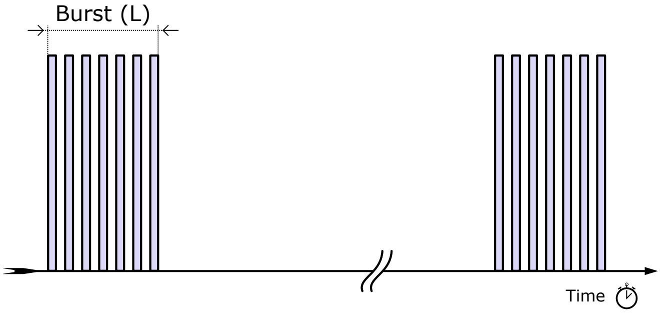


Figure 2.16: Example of a radar pattern

In order to mitigate such interferers, the dynamic frequency selection scheme has been implemented for WLAN systems working in the 5 GHz band. This technique forces an access point to change the transmit channel randomly if a radar signal is detected by the CCA. Consequently, the data transfer between two stations is interrupted during this change. Especially wrongly detected events identified as radar signals cause a downgrade of the throughput. Furthermore, not all channels utilize the dynamic frequency selection scheme, i.e., channel 36. Therefore, radar signals can influence WLAN communication systems considerably.

Chapter 3

Measurement Campaign

In order to describe interference effects in the ISM band, a measurement campaign was held in June of 2018. The goal of this campaign was to record traffic in WLAN channels localized in the 2.4 GHz and 5 GHz band, obtaining parameters such as traffic load and power level distributions. Further investigations will show if it is possible to abstract such a scenario by using different kinds of interference sources.

Section 3.1 introduces the measurement scenario, setup, and how the measured data is recorded. Subsequently, the observed records are analyzed utilizing several techniques, concerning typical traffic parameters (Section 3.2). Firstly, a rudimentary model is considered, describing the main issues regarding detection problems. Based on this point of view, a more complex method called energy detector (ED) is presented. This technique offers the opportunity to detect transmitted frames, characterizing them by their duration and mean power. This section is completed by determining the communication standard of detected events. At last, results utilizing the explained techniques are given in Section 3.3.

3.1 Setup and Location

The measurement campaign took place during a lecture called “Computer supported Japanese” at the TU Wien. Due to a large number of attendees, a high device density of laptops, smart phones, and smart wearables, based on WLAN and BLE, was expected. Consequently, this scenario offered the opportunity to record multiple users, and an above average traffic load. Furthermore, the existing devices were distributed over the whole lecture hall, resulting in a varying power level of the received frames. Therefore, an omnidirectional antenna with broadband receive characteristics was used to cover the desired frequency range from 2.4 GHz to 5 GHz, capturing ISM band traffic. The described scenario is depicted in Figure 3.1. Due to space limitations, the measurement equipment was placed outside of the lecture hall. The antenna position is marked by the white arrow in the left subfigure. As the remaining equipment was placed outside, a 30 m long cable was utilized to connect the antenna.

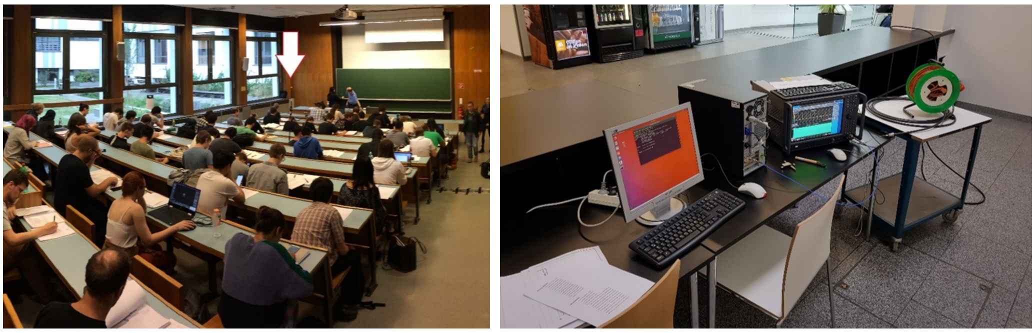


Figure 3.1: Lecture hall (left), measurement equipment (right)

Receiver Chain

Since the utilized RF cables introduce losses and, consequently, increase the noise figure (NF) as well, an amplifier chain was utilized to conquer these effects.

According to Figure 3.2, a low-noise amplifier was connected directly to the connector of the antenna to establish a calculated low total NF of $\sim 1.53\,\text{dB}$. While the antenna was mounted on a tripod, an additional cable down to the floor was utilized to connect another power amplifier. The overall gain of these two amplifiers overcomes the insertion losses induced by the long transmission cable to the measurement equipment. Details about the used components can be found in Table 3.1.

At the end of the 30 m long cable, a power splitter connects a VSA and a Linux PC to the receiver chain. The VSA records the received data with a sampling rate of 25.6 MSa/s up to a maximum length of approximately 10 s. In parallel, the PC demodulates all received WLAN frames and records details about each, which can be analyzed with software tools, such as Wireshark[1]. It will be verified if it is possible to replace expensive measurement equipment (VSA) with a Linux PC for data recordings. At the beginning of the measurement campaign it was evaluated which WLAN channels were occupied most frequently. It turned out that channel 1 ($f_c = 2.412\,\text{GHz}$) and channel 36 ($f_c = 5.18\,\text{GHz}$) were the best

[1] Wireshark is a network protocol analyzer tool. It enables decoding frames on a microscopic level. Regarding WLAN, it is possible to record live data exchanges, revealing details about MAC and PHY properties.

choice in terms of traffic load. During the whole measurement scenario 40 records in the 2.4 GHz and 5 GHz band have been made.

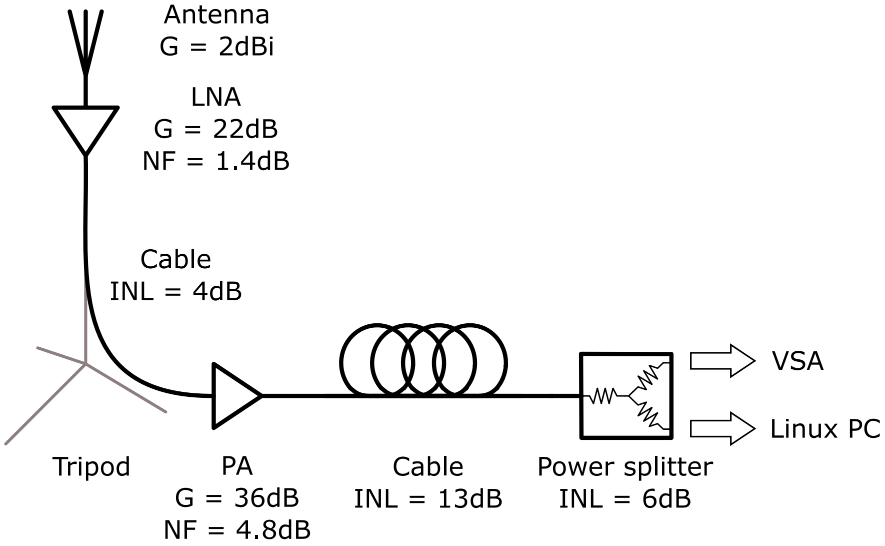


Figure 3.2: Receiver chain of the measurement setup, specified for 2.4 GHz (G...Gain, NF...Noise figure, INL...Insertion loss)

Component	Manufacturer	Description
Antenna	Huber und Suhner	SENCITY Omni-S 1399.17.0224
LNA	Mini-Circuits	ZX60-83LN-S+
PA	Mini-Circuits	ZVE-8G
Power splitter	Mini-Circuits	ZFRSC-183+
Filter	K&L	CAV-01311
WLAN module (PC)	Atheros	WLE900VX
VSA	Keysight	UXA N9040B

Table 3.1: List of the utilized components corresponding to Figure 3.2

The receiver chain may cause blocking effects of the WLAN module implemented in the Linux PC. Therefore, an additional filter for measurements in the 5 GHz band was added. One can see the respective S-parameters in Figure 3.3. In order to further adjust the received power levels to an appropriate value, additional attenuators were applied.

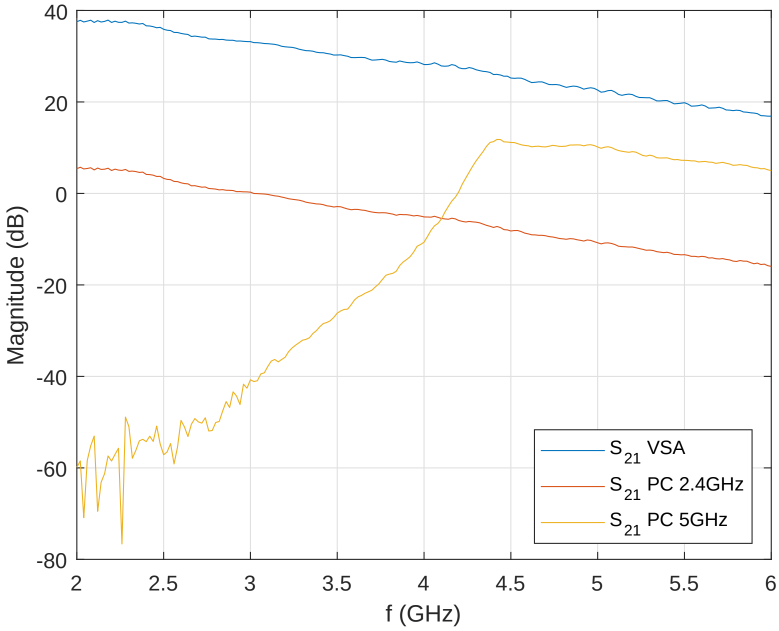


Figure 3.3: Gain of the receiver chain between antenna and: VSA (blue), PC 2.4 GHz (red), PC 5 GHz (yellow)

3.2 Data Analysis

As the goal of this work is to emulate real ISM band traffic utilizing different techniques and interference sources, the captured data must be analyzed to abstract an appropriate channel model. The VSA saves the data as a complex baseband signal with a sample rate of 25.6 MSa/s. The highest expected bandwidth is 22 MHz (WLAN DSSS), therefore, the whole measurement data is further processed by a filter with the desired frequency range. In order to create a statistical model of the observed scenario, the baseband signal is investigated in terms of channel on-/off-times and power levels. On- and off-times are specified as immediately consecutive samples with the respective decision state. The power levels are obtained by the mean value within the estimated on-times, recalculated by the measured S-parameters from Figure 3.3.

The detection of on- and off-times is a nontrivial problem for various reasons. First of all, the received signal modulation is unknown. As already discussed in Section 2.2, BLE and WLAN behave differently regarding their envelope signal in time domain. Later on, it will be shown that OFDM has a noise-like amplitude distribution (Section 4.3). Because of strong amplitude variations, it becomes rather difficult to detect an OFDM modulated signal without interruptions by simply setting a power threshold. Figure 3.4 depicts the power envelope of a short sequence from the recorded ISM band data. Obviously, the envelope shows a high dynamic range for some frames, which makes them hard to detect correctly, even when the mean SNR is high enough ($> 10\,\mathrm{dB}$). The lower subfigure of 3.4 emphasizes the detection problem by a power threshold of $-80\,\mathrm{dBm}$. In this case, the mean SNR is $\geq 20\,\mathrm{dB}$ (theoretically calculated noise power of $-100.6\,\mathrm{dBm}$ for a bandwidth of $22\,\mathrm{MHz}$) and the decision is still ambiguous even for frames with a high power level. These uncertainties result in a wrong on- and off-time characterization. In the following, methods will be discussed to overcome these problems.

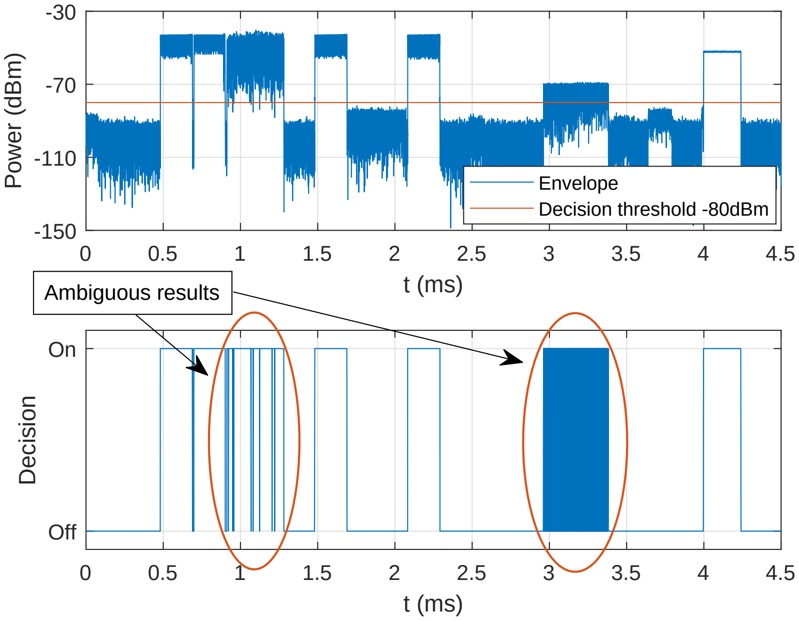


Figure 3.4: ISM band recording, WLAN channel 1, BW = 22 MHz: envelope (top), decision according to power level detection (bottom)

3.2.1 Time-Quantized Analysis

A simple approach to omit the detection problem (Section 3.2) is to slice the recorded power envelope in time domain into equally stepped intervals and determine the mean power of the respective slots. This model circumvents detection problems by neglecting an estimation of start- and end point of an event. Hence, it is possible that multiple interferer types occur in one slot, making it impractical to determine the respective communication standard. Therefore, only power levels are investigated.

Before proceeding with the data analysis of this technique, a statistical concept will be introduced. Quantiles are often used to describe random data sets by arranging the obtained samples according to their size. One popular quantile, for instance, is the median, dividing the sorted data set into two parts with an equal amount of samples (50 %/50 %). The separation of the data into groups of equal size can be arbitrarily extended. It is possible to examine similarities of two data sets regarding their probability distributions with a graphical method called quantile-quantile (Q-Q) plot. In order to obtain such a graph, the quantiles of the two sets are assigned to separate axes. Furthermore, the quantile separation has to be determined. A point on the Q-Q plot corresponds to an intersection between the same quantiles of the x- and y-axis. The points will lie approximately on the line $y = x$ if the two data sets being compared have the same probability density function (PDF).

Figure 3.5 depicts the power level distribution of two records from the 2.4 GHz ISM band with a reference step size of 1 µs. The corresponding Q-Q plots show a comparison between different step sizes (100 µs and 1 ms) and the reference. Since the preamble of a WLAN frame lasts for more than 16 µs, a resolution of 1 µs is assumed to be fine enough to neglect clipping effects of high power levels through averaging. As one goal of this work is to find ways of complexity reduction utilizing different interference sources, longer step sizes of 100 µs and 1 ms are investigated to meet these requirements. More details about the implementation will follow in Chapter 4.

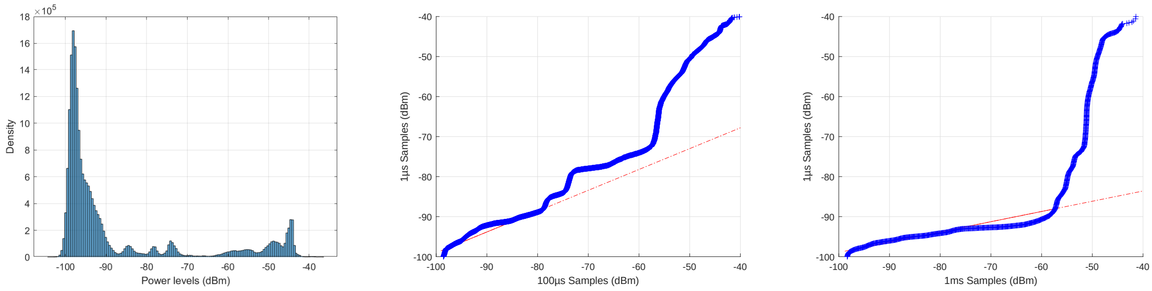


Figure 3.5: Histogram of measured power levels with a step size of 1 µs (left), respective Q-Q plots with longer time intervals of 100 µs (middle) and 1 ms (right)

The Q-Q plots give an estimate on how good the power levels, averaged over longer step sizes, fit to the ones with a step size of 1 µs. If the plot appears to be linear, the compared samples tend to have the same distribution. It is noteworthy that the quantiles for a step size of 100 µs deviate already for power levels greater than −75 dBm, while the samples corresponding to the step size of 1 ms seem to be linear up to a power of −60 dBm. Furthermore, one can see that the red linear line has a smaller slope than $y = x$. This means, that the quantiles of the x-axis have a higher density of larger power levels than the quantiles of the y-axis (1 µs). Hence, the power levels do not suffer from clipping effects caused by averaging for bigger step sizes, as it was expected.

3.2.2 Energy Detection

The previously presented technique is a simple approach to describe the power level characteristics of the captured ISM band records. Unfortunately, this model does not consider frame lengths and off-times of the channel. Thus, a more complex method must be utilized to conquer the detection problems explained in Section 3.2.

Because of the large amount of different communication standards using the ISM band, it would cause an intolerable effort to demodulate each detected event and determine the frame length. In addition to this, interferers such as microwave ovens cannot be detected by this method. Therefore, another concept, optimized for single antenna receivers called ED is introduced. This so-called ED calculates the received energy over a time interval T and utilizes an appropriate threshold ξ to decide if a signal is present. The main parameters which determine the performance of the ED are the decision threshold, the number of samples ($N = Tf_s$), and the estimated noise power [11, 12].

The ED just decides if a signal is present or not. Therfore, this scheme can be broken down into a binary hypothesis-testing problem [11]:

Hypothesis 0 ($\mathscr{H}_0$:) *signal is absent*

Hypothesis 1 ($\mathscr{H}_1$:) *signal is present*

Hence, the received complex baseband signal is divided into two states over time index i (n_i...noise, x_i...transmitted signal, y_i...received signal):

$$y_i = \begin{cases} n_i & : \mathscr{H}_0 \\ x_i + n_i & : \mathscr{H}_1 \end{cases} \tag{3.1}$$

In order to be independent of absolute power levels, the ED estimates the mean SNR over time. Equation 3.2 defines the calculation of the ED by averaging the signal power (y_i^2) over N samples. Since the noise power is not perfectly known, a maximum likelihood estimation of $\hat{\sigma}^2$ is utilized over M samples. In conclusion, the ED is a sliding window that calculates the mean SNR and decides according to a given threshold (ξ) if a signal is present or not [12].

$$\Lambda(y) = \frac{1}{2\hat{\sigma}^2 N} \sum_{i=0}^{N-1} |y_i|^2 \underset{\mathscr{H}_0}{\overset{\mathscr{H}_1}{\gtrless}} \xi, \qquad \hat{\sigma}^2 = \frac{1}{2M} \sum_{i=0}^{M-1} |n_i|^2 \tag{3.2}$$

The desired ED performance will be described in terms of false alarm probability (P_f) and detection probability (P_d). P_f is the probability of falsely detecting a signal while it is actually absent ($\mathscr{H}_0$) and P_d is the probability of correctly detecting a present signal ($\mathscr{H}_1$). Assuming an additive white Gaussian noise (AWGN) scenario and a Gaussian signal, e.g., OFDM, to be detected, these probabilities may be expressed for large N and M by [12]:

$$P_f \approxeq Q\left(\frac{\xi - 1}{\sqrt{\frac{N+M}{NM}}}\right), \qquad P_d \approxeq Q\left(\frac{\frac{\xi}{1+\mathrm{SNR}} - 1}{\sqrt{\frac{N+M}{NM}}}\right). \tag{3.3}$$

Eliminating the threshold ξ through the two equations from 3.3, it is possible to calculate the minimum achievable SNR, satisfying the desired P_f and P_d:

$$\mathrm{SNR}_{\min} = \frac{1 + Q(P_f)^{-1}\sqrt{\frac{N+M}{NM}}}{1 + Q(P_d)^{-1}\sqrt{\frac{N+M}{NM}}} - 1. \tag{3.4}$$

The solution for the optimum decision threshold ξ_{opt}, meeting the desired probabilities (P_f, P_d), is then given by utilizing $\mathrm{SNR}_{\min}$ in equation 3.2 (right):

$$\xi_{\mathrm{opt}} = \left(Q(P_d)^{-1}\sqrt{\frac{N+M}{NM}} + 1\right)(1 + \mathrm{SNR}_{\min}). \tag{3.5}$$

Figure 3.6 again shows the ISM band sequence from Figure 3.4, now based on the ED. Because of averaging effects of the sliding window and appropriate calculation of a threshold, the decision states are unambiguous. If the size of the sliding window (N) is too large, the off-times of the channel tend to grow over and it is not possible anymore to define start and end points of two frames that are closer to each other than the defined window length. In the IEEE 802.11n standard, the shortest IFS is the so-called RIFS, which is 2 µs long. Therefore, the sliding window size is set to $t = 2\,\mu\mathrm{s}$. At a sample rate of $f_s = 25.6\,\mathrm{MSa/s}$, this equals a number of $N = f_s t \approx 52$ samples.

It is necessary to define a set of noise samples to calculate the ED output of Equation 3.2. Examining Equation 3.4 indicates that, for the desired probabilities P_f and P_d, the parameters N and M are available to optimize the $\mathrm{SNR}_{\min}$. As the window size is $N = 52$ samples long, the noise sample size M can be chosen arbitrarily large to make the decision threshold as low as possible. One might think that the threshold can be made infinitely low. However, because of uncertainties caused by parameter estimation methods, the minimum achievable SNR becomes asymptotic for large values of M and N. Hence, a signal-free sequence in the recorded baseband data with $M = 10^5$ samples led to a satisfying noise power estimation ($\hat{\sigma}^2$). It must be mentioned that the noise power estimation suffers from several effects and influences the performance of the ED. The four most important factors are [12]:

1. temperature variation
2. change in low-noise amplifier gain due to thermal variations
3. initial calibration error
4. presence of interferers

It is possible to overcome the first three mentioned issues by using a sufficient amount of samples for noise power estimation. Nevertheless, the influence of interferers close to the noise floor, which makes them hard to detect, cannot be canceled out by increasing the amount of noise samples. Consequently, it must be taken into account that the ED performance may deviate from theoretical calculations.

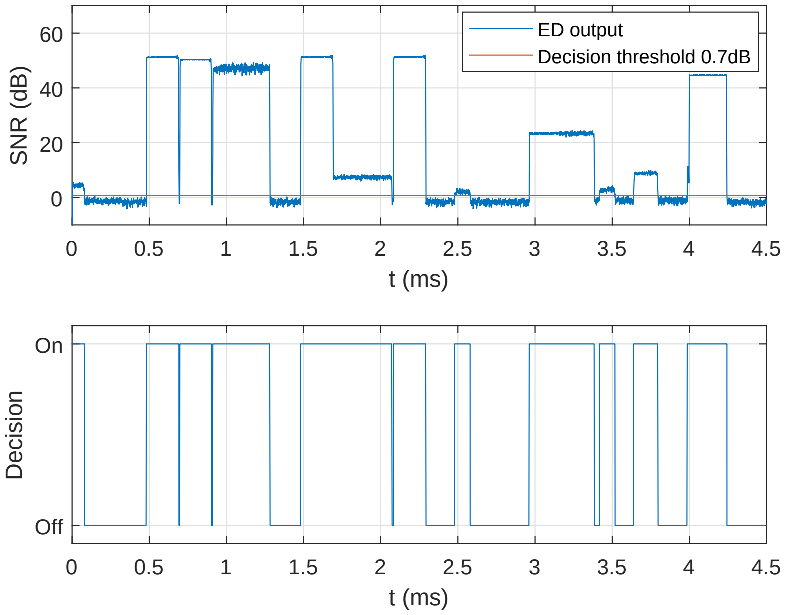


Figure 3.6: ED outptut in term of SNR over time of the sequence from 3.4 (top), decision according to threshold (bottom): $f_s = 25.6\,\mathrm{MSa/s}$, $N = 52$, $M = 10^5$, $P_f = 0.1$, $P_d = 0.9$, $\xi = 0.7\,\mathrm{dB}$

3.2.3 Classifying Frames

With the ED, it is possible to identify channel occupations independent of standard and modulation. This approach can be further extended by classifying the detected frames in order to draw conclusions about different modulation schemes influencing the behavior of interfering signals. Hence, the amount of existing standards (BLE, WLAN-DSSS, WLAN-OFDM) in the captured ISM band data is estimated. This section treats the detection of WLAN frames with DSSS and OFDM modulation. The remaining signals, such as BLE, are classified as general interference.

WLAN DSSS

The DSSS modulation is only used in the 2.4 GHz ISM band corresponding to the IEEE 802.11b standard. Even though this technique is obsolete by now, it is still frequently used to provide compatibility between old and new WLAN-standard revisions.

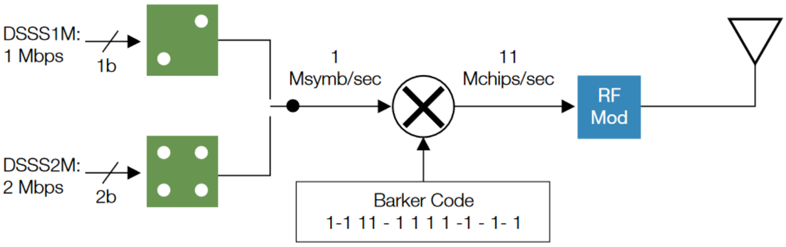


Figure 3.7: WLAN DSSS modulation scheme [13]

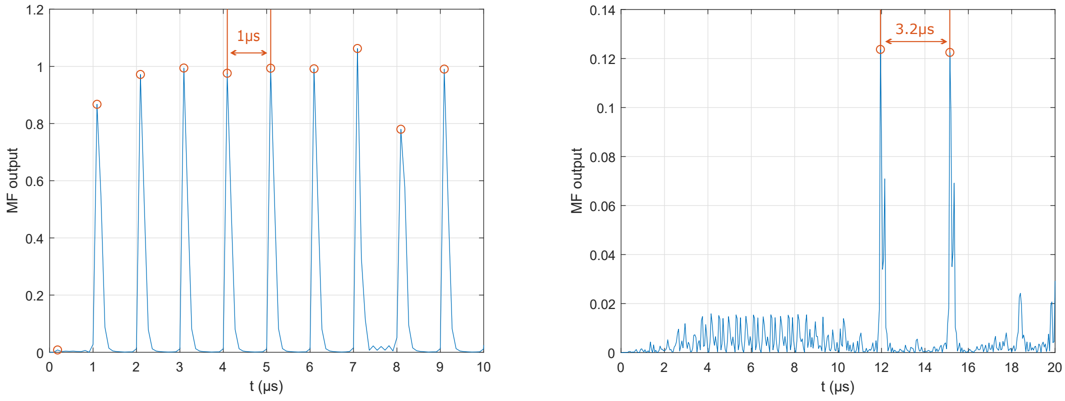
The legacy preamble has a bandwidth of 1 MHz and is broadened by multiplying the baseband data with a spreading sequence up to 11 MHz. Figure 3.7 depicts the modulation technique in detail. The spreading sequence is realized with a Barker code and ensures that the autocorrelation function has a minimum at off-peak values with the following BPSK symbols: $g_i = [1, -1, 1, 1, -1, 1, 1, 1, -1, -1, -1]$. Since the sequence stays the same for the whole preamble, it is possible to detect DSSS-modulated frames by exploiting this repeated structure. This can be realized, for instance, with a matched filter (MF) ($h_i = g^*_{N-i}$, N...impulse response length), which maximizes the SNR for the optimum decision point.

The output of such an MF according to a DSSS preamble is given by Figure 3.8(a). One can see that the peaks are spaced equidistantly by 1 µs. Consequently, WLAN frames, which are modulated according to the IEEE 802.11b standard, can be classified with the explained timing constraint. It must be mentioned that channel variations may distort the MF output and peak distances deviate about a few samples.

WLAN OFDM

State-of-the-art WLAN systems utilize OFDM as a modulation scheme. The bandwidth is allowed to occupy a range of up to 160 MHz (IEEE 802.11ac). Such broadband transmissions are required for high data rate applications. As IoT focuses on wireless sensors and wearables, which are satisfied with low data rates, a single WLAN channel with a bandwidth of 20 MHz is analyzed.

As explained previously (Figure 2.11), OFDM-modulated WLAN utilizes a preamble for setting channel parameters and synchronization in frequency and time domain. The legacy preamble of every OFDM-modulated WLAN frame consists of a repeated structure, which makes them easy to detect. Figure 3.8(b) depicts the matched filter output of an OFDM-modulated WLAN frame from the measurement campaign, correlated with the long training sequence, separated by a time interval of 3.2 µs. Hence, a timing constraint can be set to identify such frames. Obviously, the main two peaks have a side lobe each, which presumably appear due to multipath components. For low-SNR frames, the main peaks may not be detected correctly. Thus, a timing tolerance has to be taken into account. Further investigations showed that an empirically defined deviation of 10 % led to an appropriate functionality of the MF detection scheme. Therefore, an absolute deviation of up to 100 ns for DSSS and 320 ns for OFDM is accepted.



(a) WLAN DSSS

(b) WLAN OFDM

Figure 3.8: Matched filter output of corresponding WLAN preambles

3.3 Results

Using the techniques which were presented in the previous sections, the ISM band data is analyzed according to the ED scheme. In addition to this, the results are interpreted and parameter fittings of respective probability densities are investigated.

The boxplot (Figure 3.9) is another statistical method for graphically describing numerical data utilizing quantiles. They are defined by the median (yellow), interquartile range

$\text{IQR} = Q_3 - Q_1$ (box in red) and the upper ($Q_3 + 1.5 \cdot \text{IQR}$) and lower whisker ($Q_1 - 1.5 \cdot \text{IQR}$). Samples detected outside the range defined by the whiskers, which is 99.3 % for normal distributions, are outliers (green). Since the example relates to a normally distributed function, which is symmetric, the median is placed inside the IQR symmetrically as well. Furthermore, the length of the IQR indicates how the observed data is concentrated. For large IQRs, the samples are widespread and vice versa for short ranges. In conclusion, the boxplot offers an efficient way to describe data distributions graphically.

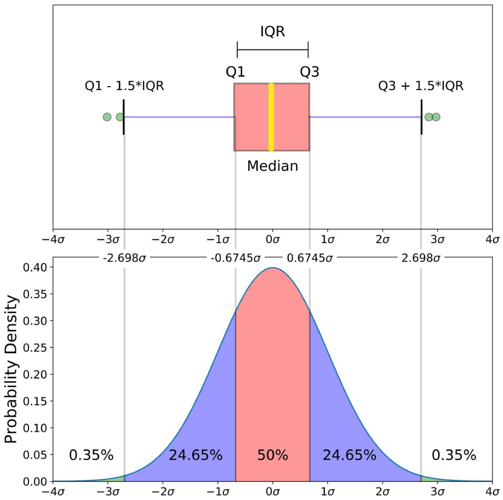
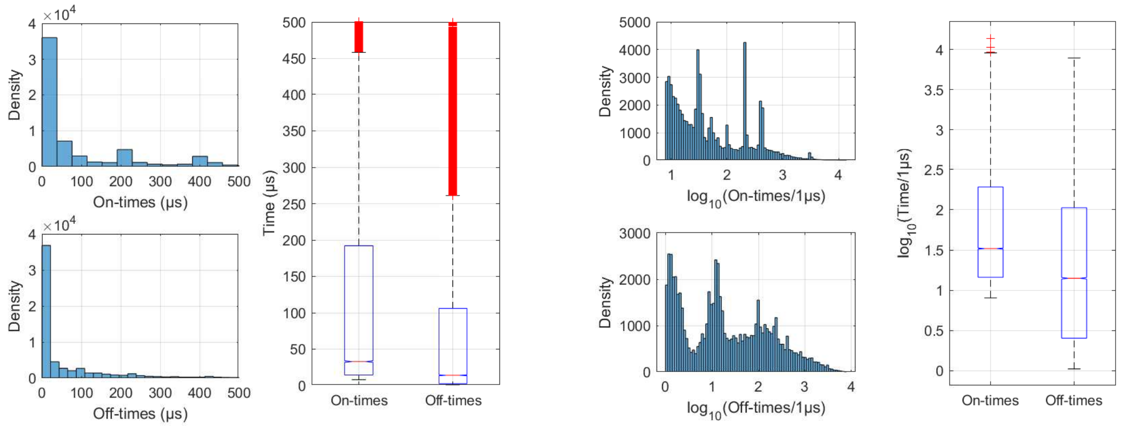


Figure 3.9: Example of a boxplot (top) according to a normal probability density function (bottom) [14]

In this section, two 2.4 GHz ISM band records of WLAN channel 1 are considered for explaining the process of data analysis. Because of several aspects concerning interference characterization, the two recordings with the highest traffic load are analyzed. Details about these aspects follow in Chapter 5. The data has been analyzed with the following parameters: $N = 52$, $M = 10^5$, $P_d = 0.9$, $P_f = 1 - P_d = 0.1$, $\xi = 0.7\,\text{dB}$. Through the false alarm probability of $P_f = 0.1$, outliers are likely to appear. In order to maintain a more stable decision output of the ED, an outlier detection has been implemented. The minimum off-times regarding WLAN systems last for 2 µs (RIFS). Because of several effects, such as channel variations, this minimum idle time may be violated. Thus, all off-times between two on states, which are smaller than 1 µs, have been dedicated to be on-times. Examining the preamble of BLE and WLAN standards indicates that preambles last for more than 8 µs. Therefore, all on-times which last shorter than 8 µs have been set to off state.

Figure 3.10(a) depicts the histogram of the on- and off-times and the related boxplot.

Samples that are not within the whiskers range are outliers and marked in red. One can notice the huge amount of outliers and that the median values of on- and off-times are between 15 µs and 33 µs. Furthermore, the histograms show a high density for low values and a nonsymmetric shape. In addition to this, the median value is close to the lower quantile Q_1 and the IQR is relatively large. Consequently, the data is spread over a wide range, which is also reflected in the big amount of outliers. Due to these properties, parametric distributions like gamma or beta may not fit sufficiently to cover the whole data range.



(a) Linear scale

(b) Logarithmic scale

Figure 3.10: Histograms and boxplots of on- and off-times corresponding to their scale

Since the on- and off-times spread over a wide range, it seems natural to transform the data logarithmically in order to gather the samples closer to each other. Figure 3.10(b) shows the same data from 3.10(a), in a logarithmic scale. As expected, the data samples are closer to each other and the boxplot shows only two outliers for on-times. Nevertheless, it still must be clarified if it is possible to fit a distribution to the current histograms. Obviously, the structure of the densities is asymmetric and sharp spikes appear over the whole range. As a consequence, classical parametric distributions like Gaussian or Laplace will not fit. Therefore, a nonparametric kernel distribution is utilized to create PDFs of the respective random variables. The main parameters to describe such a PDF are the type of the smoothing function, e.g., Gaussian, triangle, and the width (variance) influencing the smoothness of the resulting approximation. Equation 3.6 describes the kernel estimator function [15]:

$$\hat{f}(y) = \frac{1}{n} \sum_{i=1}^{n} w(y - y_i; h). \tag{3.6}$$

The probability density w is the so called smoothing function and the corresponding variance is defined by parameter h. The kernel estimator function will be explained by a short example.

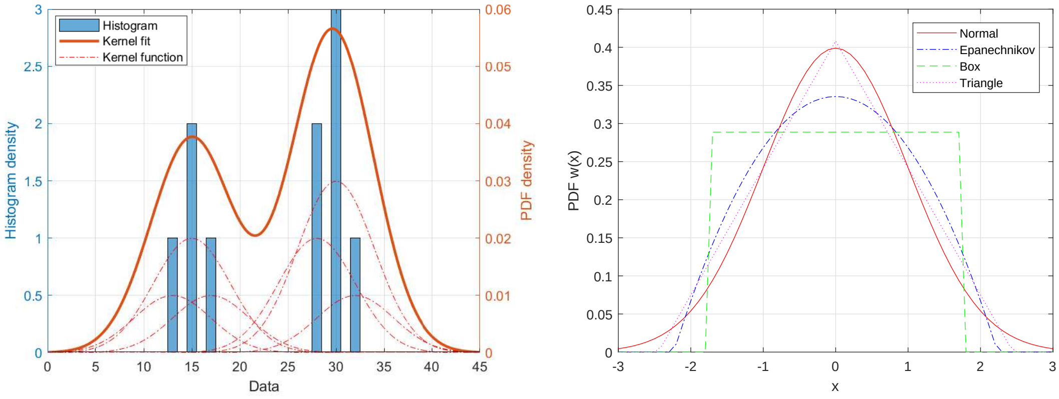


Figure 3.11: Kernel fit example: densities (left), smoothing functions (right)

Figure 3.11 depicts the histogram of some test data on the left. The kernel function is realized as a normal distribution with a variance of $\sigma = 4$. According to Equation 3.6, at every sample, a smoothing function is placed in red dashed lines. The overall sum of this ensemble, normalized by the amount of data points n, is the estimated kernel fit $\hat{f}(y)$, marked in red. On the right of Figure 3.11, a variety of smoothing functions is presented. Typically, the normal distribution is utilized, but for some use cases other shapes, such as the box, triangle, or Epanechnikov, yield better results.

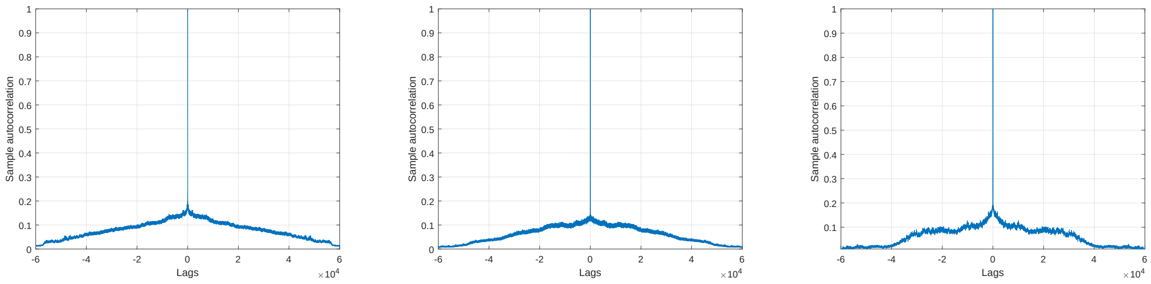


Figure 3.12: Autocorrelation of on- (left), off-times (middle), and power levels (right)

Before obtaining random variables for on-/off-times and power levels, dependencies in terms of periodicities must be considered. In order to describe such a behavior, the autocorrelation function can be examined through identifying periodic peaks. Figure 3.12 depicts the respective autocorrelation functions in the linear regime. As no periodic peaks appear in the plots, the data sets are assumed to be independently distributed. One must note that correlations between power levels and corresponding on-times have been neglected.

Since the data densities have been described by their respective autocorrelation functions, random variables can be created by utilizing kernel distributions. It turned out that the Gaussian smoothing function yields the best results for parameter fittings. Furthermore, this technique offers the opportunity to specify a desired range of the PDF. Figure 3.13 depicts the fitted PDFs of on-/off-times and power levels. It must be mentioned that the fitting strongly depends on the histogram resolution and smoothing function variance. If the bin width is chosen too large, significant values such as maxima and minima get lost. For very high resolutions, it becomes a challenging task to fit strong variations of data densities by a proper variance. Consequently, the bin width must be chosen carefully, but a lot of different rules for calculating the resolution exist.

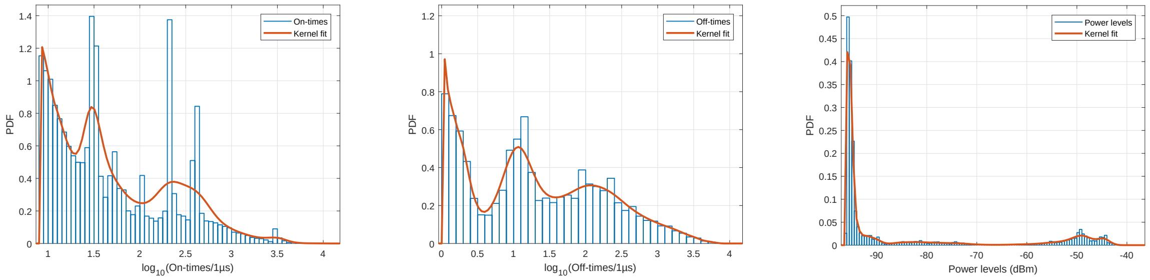


Figure 3.13: Parameter fitting: on-times (left), off-times (middle), power levels (right)

In this work, the Freedman-Diaconis rule led to satisfactory results and will be explained briefly [16]. The bin width is defined by the interquartile range (IQR) of the observed data and is normalized by the amount of samples (n) as:

$$\text{Bin resolution} = 2\frac{\text{IQR}}{\sqrt[3]{n}}. \tag{3.7}$$

The interquartile range relates to the boxplot representation, explained previously. It is defined by the distance between the upper- and lower quantile, surrounding the median and covering 50 % of data (IQR $= Q_3 - Q_1$).

Unfortunately, the on-times density (Figure 3.10(b)) shows significant spikes through the data which appear noncontinuously. Even for a small variance, these spikes cannot be approximated ideally. One big advantage of the kernel fit is the opportunity to define the data range for parameter fitting. Unlike parametric functions, such as the beta or gamma function, exclusion rules can be set arbitrarily. Table 3.2 refers to the details about the bandwidth of the smoothing function and data ranges located on the x-axis of the respective histograms. Due to simulation settings, utilizing all recorded data sets would cause an intolerable increase of time intervals for measuring interference perturbations. Hence, only two data sets with the highest traffic density, using WLAN channels 1 and 36, are investigated. The fitted distributions will be used in Chapter 4.

It has been mentioned previously that it is of interest, which modulations and standards are involved in the captured data. With the detection methods from Section 3.2.3, it is possible to distinguish between WLAN modulations (which are OFDM and DSSS). Otherwise, the detected frame is classified as general interference, such as BLE, Sigfox, or HiperLAN. Although these presented correlation techniques work out well, it may happen that multiple standards are detected in a single on-time, which will be classified as WLAN if a valid preamble is detected. In addition to this, the observed statistical model utilizes the mean power of the whole frame. It probably occurs that the power level changes during a detected event for several reasons. Due to superposition of multiple users (hidden node) or, for instance, a BLE frequency hopping sequence, the power levels may change rapidly. As a consequence of averaging, strong power level variations will be clipped.

	Channel 1 2.412 GHz		**Channel 36** 5.18 GHz	
Parameter	**Bandwidth**	**Range**	**Bandwidth**	**Range**
On-times ($\log_{10}(t/1\,\mu s)$)	0.18	$0.9 < x < 4.15$	0.15	$0.9 < x < 4.1$
Off-times ($\log_{10}(t/1\,\mu s)$)	0.18	$0 < x < 3.95$	0.11	$0 < x < 4.2$
Power levels (dBm)	0.13	$-96 < x < -37$	0.45	$-96 < x < -37$

Table 3.2: Kernel fit parameters: x-axes ranges (investigated x-axis range of the respective histogram) are in logarithmic scale

Table 3.3 states the detailed amount of involved modulations and how many on-times have been detected in channels 1 and 36. The power threshold is empirically set to −90 dBm to ensure a proper functionality of the frame classification techniques. This means that frames below this power level are not taken into account. It is noteworthy to see that the amount of IEEE 802.11b frames (WLAN DSSS) is still high compared to newer revisions, which are using OFDM. In addition to this, the percentage of interferers is in the same range as for the 5 GHz band. It must be emphasized that the true amount of interferers will be higher. Every on-time detected with a valid WLAN preamble is classified as such. Hence, frames will only be identified as an interferer when they are free of WLAN signals.

	Channel 1 2.412 GHz	**Channel 36** 5.18 GHz
WLAN DSSS	38.41 %	0 %
WLAN OFDM	52.4 %	92.74 %
Interference	9.19 %	7.26 %
Recordings	2	2
On-times	19,360	24,740

Table 3.3: Frame classification: power threshold $> -90\,\mathrm{dBm}$

In Section 3.1, the attempt of replacing a VSA by a Linux PC system is discussed. It turned out, for several reasons, that this is not feasible. The Linux system demodulates the received data by a WLAN card, which is not able to detect BLE or other kinds of standards. According to Table 3.3, this would cause a data loss of about 9 % in channel 1 and 7 % in channel 36. At an amount of more than 19,000 detected events, this seems to be bearable. One must note that the percentage of interferers changes up to 20 % if all records of 2.4 GHz band are investigated. This would cause an unacceptable data loss. Furthermore, Linux offers information about demodulated frames in terms of frame length, modulation, power level, and further details about MAC properties. As one important key parameter describing channel characteristics is the power level, inaccuracies have to be taken into account. There is no transparency about how power levels are measured and for which time period. In addition to this, capturing data by WLAN cards significantly suffers from crosstalk problems due to inadequate isolation. If an ideal receiver is examined, one would expect detecting data only when connected to a proper antenna. Unfortunately, WLAN cards still receive signals when the respective inputs are left open or terminated by a load. As already mentioned, the Linux system was placed outside of the lecture hall. Consequently, the PC-demodulated traffic does not fit to the one from the VSA. In summary, it can be said that a real-time device for recording ISM band traffic cannot be replaced by a Linux PC system in this configuration.

Chapter 4

Measuring Interference

In the previous section, ISM band traffic has been described by on-/off-times and respective power level distributions. The next step is to build a test setup, which is capable of repeating such a scenario and measuring interference effects.

At the beginning, the setup and related components are characterized in Section 4.1. Note that an instruction guide for an appropriate installation and use of the test setup can be found in Appendix A. One goal of this work is to find alternative interference sources, replacing expensive RF equipment. Section 4.2 treats the implementation of a Linux PC system injecting interfering signals and emerging issues. Furthermore, another interference source, utilizing modulated noise, is introduced in Section 4.3. A simple simulation model will compare different interferers, i.e., WLAN DSSS, OFDM, and BLE, to modulated noise. At last, a low-cost implementation of an interference source, modulating noise with the corresponding random variables from Section 3.3, will be presented.

4.1 Test Setup

Regarding WLAN systems, interference causes perturbations by impairing the throughput, PER, retransmissions, and packet delay (jitter). Hence, a test setup (Figure 4.1) has been created to examine different interference sources by measuring typical performance parameters. The test arrangement consists of two independent Linux PC systems (server and client) with implemented WLAN modules. They are connected with coaxial cables, combined by a directional coupler and a variable attenuator. Lastly, an interference source, which is connected through the coupler, completes the setup. Details about the utilized components can be found in Table 4.1. The coupler guides the signal from the interference source to the server. This scenario recalls the hidden node problem explained in Section 2.1.1. The client starts a transmission and the server receives available packets while the data exchange is corrupted by interference. Through the additional insertion loss maintained by the attenuator, the range between client and server can be changed. The adaptation of the range can be interpreted by changing the distance between transmitter (client) and receiver (server). Small ranges relate to close distances and vice versa. Thus, the setup realizes a hidden node scenario with variable ranges between client and server.

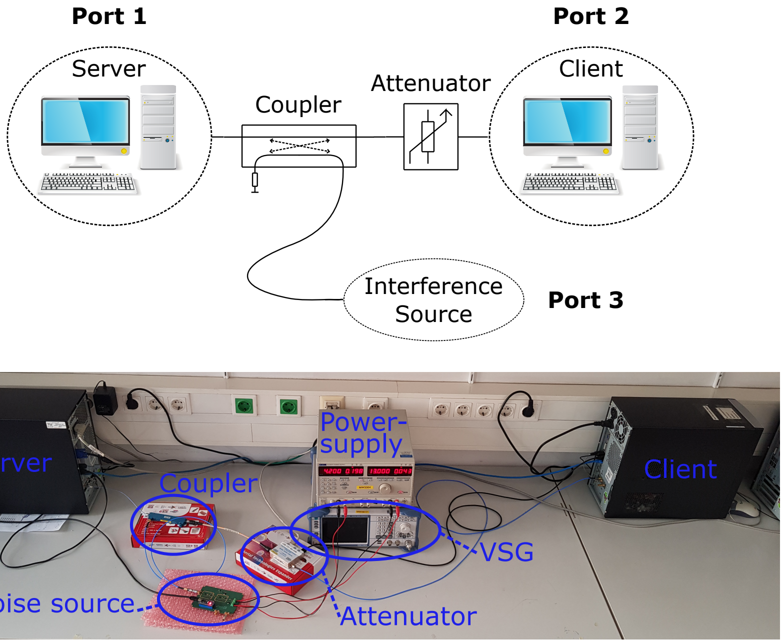


Figure 4.1: WLAN test setup: block diagram (top), realized arrangement (bottom) with the utilized interference sources (noise source with additional power supply & VSG)

Component	Manufacturer	Description
WLAN modules	Atheros	WLE900VX
Coupler	Krytar	MODEL 1850
Attenuator	Mini Circuits	RCDAT-6000-60
VSG	Rohde und Schwarz	SMBV100A

Table 4.1: WLAN test setup: list of utilized components

Furthermore, the interference source is assumed to be blind to any surrounding traffic established between server and client. It simply injects interference signals without any collision avoidance schemes. IoT systems typically exchange a low amount of data, resulting in a small packet size. The RTS threshold, explained in Section 2.1.1, is therefore deactivated. The fact that the traffic, caused by the interference source, does not have to respond to RTS-CTS transactions, justifies the hidden node model. It must be mentioned that the isolation realized by the coupler is not infinite. Consequently, the transmitter will receive an amount of interference signals even for a high attenuation. Figure 4.2 depicts the relevant S-parameters according to Figure 4.1 for an attenuation of 0 dB.

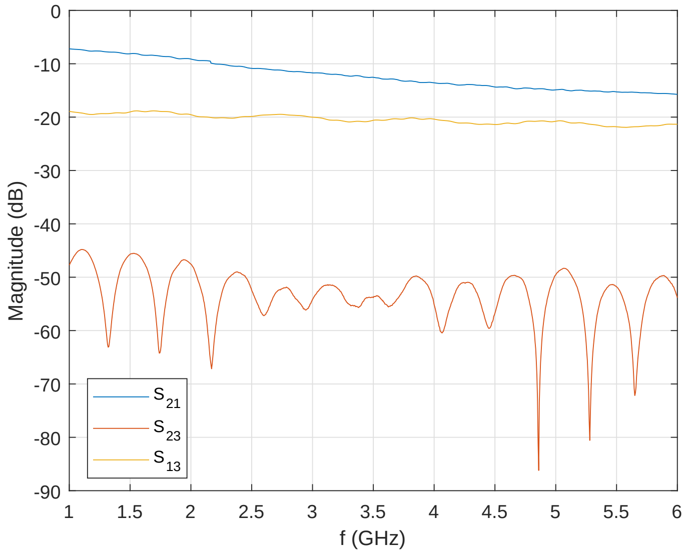


Figure 4.2: S-parameters of the WLAN test setup: server (port 1), client (port 2), interference source (port 3)

4.1.1 Measurement Setup Considerations

In the following, occurred issues, leading to the actual test setup (Figure 4.1), will be discussed. First of all it will be explained, why two separate Linux PCs have been used. Indeed, common PCs offer several PCIe[1] interfaces to connect multiple module extensions. Unfortunately, some issues made it impossible to realize the measurement setup with one PC.

[1] Peripheral component interconnect express (PCIe) is a high-speed serial computer expansion bus standard, connecting extensions like WLAN modules.

Firmware Issues

As already explained, to establish a connection between two WLAN modules, an access point (server) and a station (client) have to be configured. In Appendix A, it is explained how an association can be established, but utilizing these instructions does not succeed using a single PC, for several reasons.

First of all, it was not possible to mount two identical modules within one PC and configure them individually. The firmware boot loader had problems with loading the desired firmware onto the WLAN modules separately. Therefore, network namespaces were implemented circumventing this problem. Namespaces have the capability to isolate single modules seen by the kernel, making it possible to configure them individually. By implementing this technique, it is possible to set up the two WLAN modules as server and client establishing a valid connection. Unfortunately, the Linux PC system was not able to accomplish performance tests with typical programs. Starting such a test always led to an immediate shut down of the whole system. As the speed of the shut down was very fast, similar to unplugging the power supply, the hardware functionality has been inspected carefully. In addition to this, the IP routing table and the MAC addresses have been verified correctly, but this behavior could not be clarified. Consequently, two separate Linux PC systems were inevitable.

Crosstalk Issues

Using two separate Linux PC systems made it possible to establish a valid connection and to accomplish performance tests, but another problem arose concerning crosstalk. The two systems were capable to associate with each other and exchange data without any cable or antenna connected. The transmit power of WLAN modules in Europe is limited to 20 dBm in the 2.4 GHz ISM band. Utilizing this high power level makes it possible to overcome large distances of a few hundred meters. Therefore, it is necessary to decrease the output power to a minimum of 0 dBm (see in Appendix A). At this power level, the two Linux PC systems are not able to maintain an association anymore. Nevertheless, further crosstalk effects due to insufficient decoupling, causing signal leakage over power supplies and wired LAN routers, are presumed.

Scenario Issues

The introduced measurement setup refers to the hidden node problem (Figure 4.1). Another interesting scenario is to simulate WLAN stations which are close to each other, receiving the same interference signals. Therefore, an additional test arrangement was created and analyzed. Figure 4.3 depicts the setup in detail. It consists of two couplers, two circulators, one attenuator, and a power splitter. The power splitter divides the interference source signal into two paths. Each path is followed by a circulator and a coupler.

As before, the coupler provides isolation towards the counterpropagating path to separate the injected interference. Furthermore, the circulators ensure, through their nonreciprocal behavior, that the established traffic between server and client is not passed through the interference source connection network.

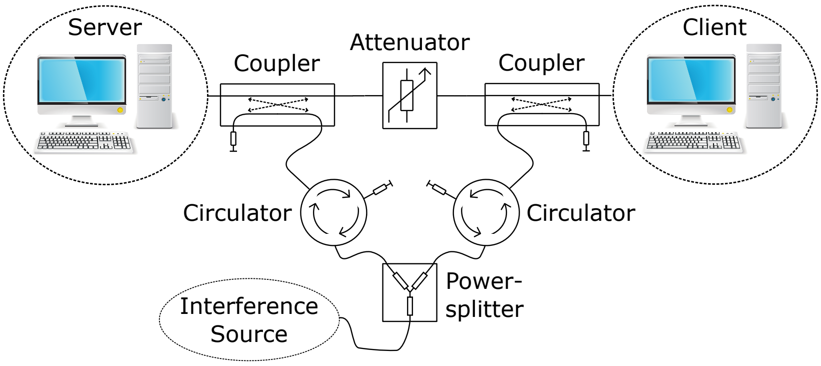


Figure 4.3: Alternative WLAN test setup

It turned out that this realization is impractical. The insertion loss between the interference source and the WLAN stations had an intolerable ripple, which could not be calibrated. The variations within one WLAN channel, at a bandwidth of 20 MHz, led to critical distortions of the interferer power spectral density. Therefore, this approach was discarded and the focus was put on the hidden node scenario.

4.1.2 Performance Measurements

In order to analyze interference effects, a network performance tool called *iperf* was utilized to establish a data transfer between server and client. *iperf* enables measuring, for instance, the throughput of an IP network using different protocols, such as UDP and TCP. It is possible to modify various parameters in terms of packet size, buffers, delay, protocols, and many more [17]. A performance test works as follows: the *iperf* client is started on both sides, server and client. While the server listens to a specified port for incoming packets, the client addresses the receiver (server) by its IP address. Then, a data stream is sent to the receiver side for one second and values like the throughput and PER are recorded. In order to gain stable performance results, this sequence is repeated ten times within one test cycle. It was mentioned in Section 2.1.3 that the MTU defines the maximum frame length of an IP network related transmission. Therefore, depending on the packet length, the data to be transmitted is fragmented into frames which fit the MTU best. As the desired packet size is smaller than 1,500 Byte, one transmitted frame consists of several packets. Through resizing data, a block ACK frame is utilized to confirm all valid packets within one reply.

As state-of-the-art modules have plenty of different WLAN standards on board, it is of great interest to have full control over utilized transmission parameters, such as modulation index and transmit power. Indeed, it is possible to control these settings by implementing a specific kernel version from *Candela Technologies* on the respective Linux PC systems [18]. For details about the installation and applications of the whole test setup, refer to the instruction guide in Appendix A.

Concerning the interference source, the captured baseband data from Section 3.3 is utilized to perturb the communication with a VSG during a performance test. Hence, the output of the reference source is a digitally modulated signal, repeating real ISM band traffic. The VSG is started independent of performance measurements and repeats the actual recording endlessly to eliminate timing-related issues. The measurement results of this configuration serve as reference for further verification of different interference sources, discussed in Section 4.2 and Section 4.3.

4.2 Injecting WLAN Frames with Linux

Remembering the results from Table 3.3, the total amount of classified WLAN frames is more than 90 % for both channels. Consequently, it seems to be natural utilizing another Linux PC system with an implemented WLAN module to emulate ISM band traffic. The main requirements on the desired source are an arbitrary transmit power level and a sufficient timing resolution to fulfill minimum off-times of 1 µs. In the following, two techniques realizing an interference source injecting WLAN frames will be discussed.

The first technique is based on sockets, enabling IP-based communications by utilizing, for instance, UDP or TCP. In order to make use of such higher layer protocols, an association between the two respective nodes, server and interference source, must be established. An association between two WLAN nodes can be understood as connecting a WLAN device to an available access point (server) for entering internet platforms. Examining this working principle of sockets yields a violation of the scenario explained in Section 4.1. The coupler from Figure 4.1 provides the same transmission characteristics towards interferer as vice versa. Hence, the interference source will receive some traffic established between server and client during a performance test. Because of collision avoidance schemes, the interferer will stop injecting frames when surrounding traffic is detected. In conclusion, it turned out that the investigated sockets are not sufficient as interference source.

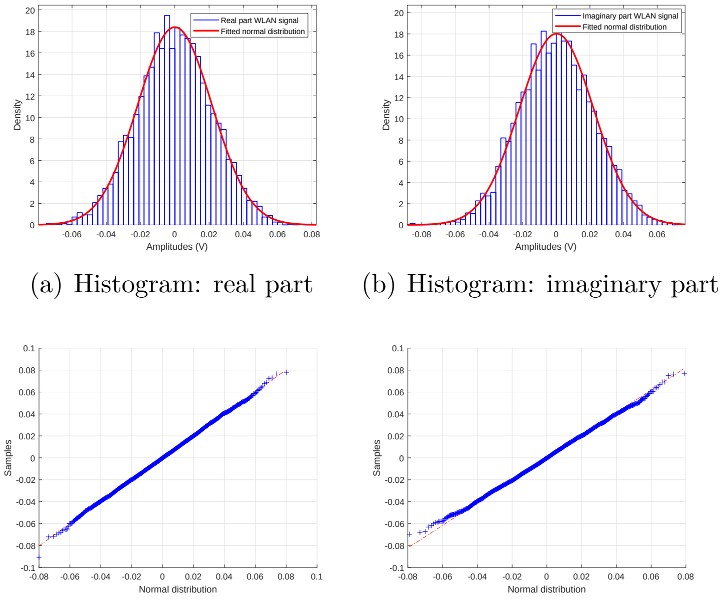
A further approach for interference injection is a Python based program called *Scapy*. It is capable of forming specific WLAN frames and transmitting them independently of any

MAC- and PHY-layer constraints explained in Section 2.1. Another important key parameter of *Scapy* is the unrestricted access to MAC-header properties. It is possible to form all kind of sequences, such as management-, control-, and data frames [19]. Since the desired interference source is blind to any surrounding traffic, it is necessary to omit specified transmission rules for medium access. In order to meet these requirements, the operating mode of the respective WLAN module must be changed. The monitor mode offers the opportunity to demodulate all received WLAN packets passively and store their transmission properties. As its name implies, this mode is designed for demodulation only and no rules regarding a valid data exchange have to be fulfilled. Unfortunately, PHY properties cannot be changed. The utilized modulations stick to legacy rates which are 1 Mbit/s (DSSS) regarding the 2.4 GHz band and 6 Mbit/s (OFDM) for 5 GHz-related channels.

With *Scapy*, a promising interference source has been found, replacing expensive RF equipment, like a VSG. Nonetheless, several issues have been identified. It is not possible to change transmission parameters like modulation and data rate. Although the monitor mode makes use of OFDM in the 5 GHz band, another issue makes this source impractical. As already mentioned, *Scapy* is a Python based program, which is an interpreted, high-level programming language. Unlike procedural languages such as C, Python induces latency effects when it comes to time-critical applications. Linux operating systems offer priority levels for executed scripts, but the occurred timing variations are still too large for the highest priority level. Sockets were also examined regarding timing properties. Since the respective scripts are written in C and the required timing resolution could also not be reached, it is obvious that the problem is caused by the operating system. Furthermore, the actual available firmware and driver of the utilized WLAN module do not support a dynamic regulation of the output power. Thanks to the kernel and adapted firmware from *Candela Technologies*, it is possible to set the power level, but just initially for one single time. However, the dynamic range, which could be achieved (20 dB) is not sufficient to emulate the power level distribution depicted in Figure 3.13 demanding a range of $\sim$ 60 dB. Because of the mentioned aspects, a different approach for realizing a low-cost interference source must be found.

4.3 Modulated Noise

As already mentioned, OFDM signals have a similar amplitude distribution compared to white Gaussian noise. In order to prove this statement, an OFDM modulated WLAN frame, captured by the measurement campaign in channel 36, is investigated. Figure 4.4 depicts the corresponding densities of real- and imaginary parts of the sampled amplitudes. The mean and variance of the observed amplitudes have been calculated to fit a normal distribution to the data. The evaluated distributions are shown in a Q-Q plot to compare the OFDM amplitudes to a Gaussian density function.



(a) Histogram: real part (b) Histogram: imaginary part

(c) Q-Q plot: real part (d) Q-Q plot: imaginary part

Figure 4.4: Amplitudes of a recorded OFDM WLAN frame: histograms and corresponding Q-Q plots of real and imaginary part

The Q-Q plots appear to be linear, which means that the observed samples fit a Gaussian distribution function well. In addition, they are uncorrelated and zero-mean. The imaginary- and real parts can be assumed jointly Gaussian and, therefore, independently distributed.

Hence, it is possible to create an interference source by utilizing white Gaussian noise. Instead of injecting WLAN frames with a Linux PC system, noise will be modulated in terms of mean power and burst length, according to the defined random variables of Section 3.3. As the 2.4 GHz ISM band is also utilized by other communication standards, e.g., IEEE 802.11b (DSSS) and BLE, it will be examined in the following if noise can be used as an interference source, yielding the same performance test results as ISM band signals.

4.3.1 Simulation Model

In the following, a simulation model based on the IEEE 802.11a standard, in presence of different kind of interference signals, will be investigated. Recapitulating Section 2.1.2, OFDM splits a symbol sequence into $K = 64$ orthogonal parallel data streams and passes them through an IFFT. Figure 4.5 depicts the subcarrier mapping of the considered communication standard. Binary data is mapped onto a symbol constellation (BPSK, QAM, etc.) and converted into OFDM symbols consisting of 64 subcarriers. The subcarrier mapping shows that an amount of 48 bins is reserved for payload data (deep blue), four are pilots (green), and the remaining 11 subcarriers (cyan) are utilized for oversampling as guard band. Furthermore, the DC carrier in the middle (grey) is not used for data transmission. The OFDM symbols are then passed through the IFFT and at last a CP of 0.8 µs is added.

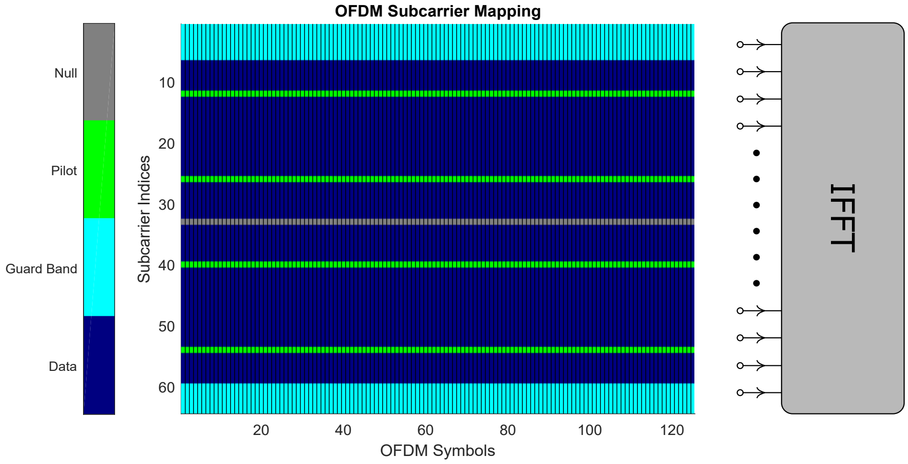


Figure 4.5: Subcarrier mapping according to IEEE 802.11a

As already mentioned, a preamble is added at the beginning of every frame. Due to its repeated structure, it is possible to detect packets and proceed with further operations, such as channel estimation and frequency offset correction. The short preamble consists of ten repeated sequences with a total length of 8 µs. Utilizing the metric invented by Schmidl and Cox [6], coarse frequency offset estimation and packet detection can be realized. Equation 4.1 defines the required timing metric $M(d)$ for packet detection, the frequency offset relevant function $P(d)$, and the signal energy $R(d)$. Due to implementing a sliding autocorrelation window with a size of L samples, the described metric is independent of absolute received amplitudes of r. As one short training sequence is 0.8 µs long and assuming a sample rate of $f_s = 20\,\text{MSa/s}$, the correlation window consists of $L = 16$ samples.

$$P(d) = \sum_{m=0}^{L-1} (r^*_{d+m} r_{d+m+L}), \qquad R(d) = \sum_{m=0}^{L-1} |r_{d+m+L}|^2, \qquad M(d) = \frac{|P(d)|^2}{R(d)^2} \tag{4.1}$$

Through function $P(d)$, it is possible to calculate a coarse frequency offset estimate. According to Equation 4.2, a shift in frequency domain (δ_k) relates to a multiplication with an exponential function in time domain.

$$e^{j\frac{2\pi\delta_k}{N}n}x_n \quad \circ\!\!-\!\!\bullet \quad X_{k-\delta_k} \tag{4.2}$$

Furthermore, the autocorrelation function yields an output equal to one within the short training field. Assuming a signal model of $r_i = x_i e^{j\frac{2\pi\delta_k}{N}i}$, the argument of function $P(d)$ can be calculated ($N = N_{\text{fft}} = 64$):

$$\begin{aligned} P(d) &= \sum_{m=0}^{L-1} x^*_{d+m} e^{-j\frac{2\pi\delta_k}{N}(d+m)} x_{d+m+L} e^{j\frac{2\pi\delta_k}{N}(d+m+L)} \\ &= \sum_{m=0}^{L-1} x^*_{d+m} x_{d+m+L} e^{j\frac{2\pi\delta_k}{N}L} \\ &= \sum_{m=0}^{L-1} e^{j\frac{2\pi\delta_k}{N}L}, \\ \delta_k &= \frac{N}{2\pi L}\arg\left(P(d)\right). \end{aligned} \tag{4.3}$$

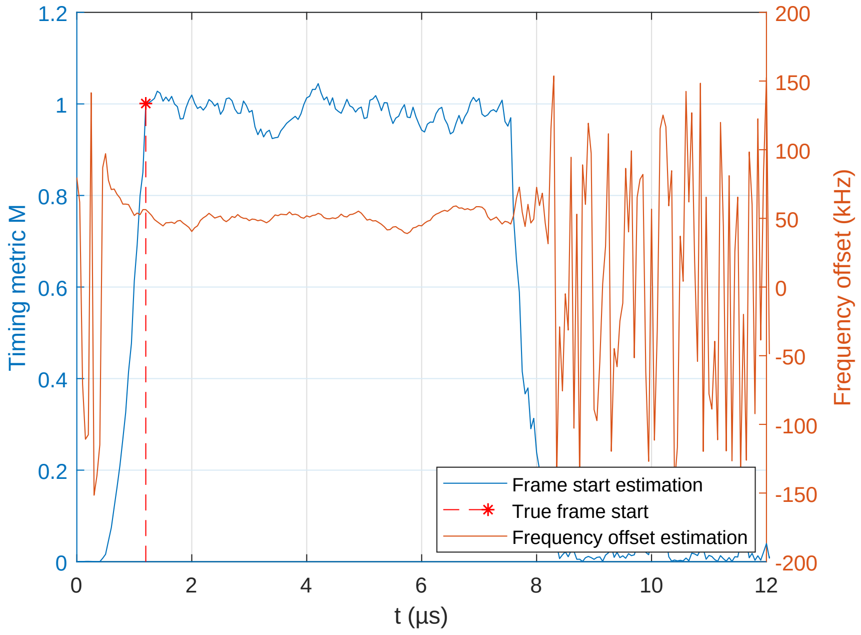


Figure 4.6: Coarse packet detection and frequency offset estimation: SNR = 20 dB, $f_{\text{offset}} = 50\,\text{kHz}$, $t_{\text{offset}} = 1.2\,\mu\text{s}$

Figure 4.6 depicts the respective output of timing metric $M(d)$ and frequency offset $f_{\text{offset}} = \delta_k \frac{f_s}{N_{\text{fft}}}$ for the following example: The investigated frame starts after 1.2 µs and has a frequency offset of 50 kHz. Furthermore, AWGN has been added with an SNR of 20 dB.

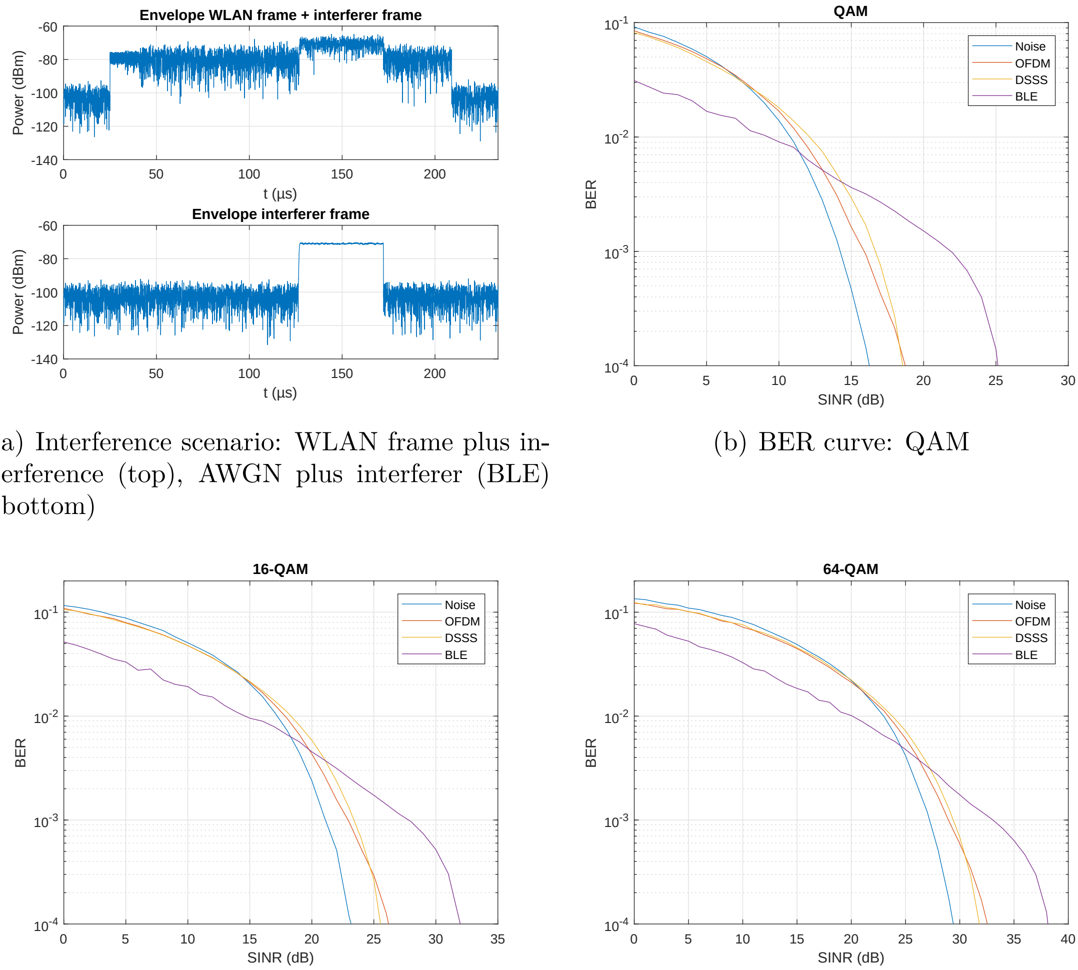
As the correlation window is just 16 samples long, the frequency offset estimation suffers from inaccuracies. Hence, the mean value is taken into account over a stable time interval. In order to find the beginning of this time interval, the timing metric $M(d)$ is utilized. The coarse detection settings have been set as follows: If more than 16 samples in a row are detected beyond a defined threshold of $M > 0.5$, the beginning of a frame is determined. Since more than two short preamble sequences get lost with this technique (autocorrelation window has a delay of L samples), a stable time of 6.4 µs remains. In order to get useful results also for low-SNR scenarios, a stable time of 2.4 µs is defined. The next step is a fine timing synchronization according to Figure 3.8. Utilizing an MF detection concerning the long training field, correlation peaks appear after the convolution. If two maxima are separated by 3.2 µs, the beginning of the frame is defined precisely. Subsequently, the long training symbols are also used for channel estimation. After synchronizing the received frame in time- and frequency domain, the CP is removed and the FFT is applied for demodulation. After the FFT, the outcoming OFDM symbols are in frequency domain. Therefore, channel estimation reduces to a simple division of the respective subcarriers (k) with the known preamble. Equation 4.4 yields a mathematical approach by simply dividing the received long training symbols $r_{k,n}$ by the true reference symbols $x_{k,n}$. As two long training fields exist (n), the mean of two channel estimates is taken into account ($n = 1..2$, $k = 1..48$).

$$\hat{H}_{k,n} = \frac{r_{k,n}}{x_{k,n}} = H_{k,n} + \frac{n_{k,n}}{x_{k,n}} \qquad \hat{H}_k = \frac{1}{2}(\hat{H}_{k,1} + \hat{H}_{k,2}) \tag{4.4}$$

The frequency offset estimation function of Figure 4.6 clearly depicts the problem of inaccuracies, caused by the short autocorrelation window. Consequently, the pilots (Figure 4.5), corrected by initial channel estimation, are investigated for further frequency offset correction. The rotation of the received pilots $\hat{p}_{k,n}$ around the true pilots $p_{k,n}$ is estimated with equation 4.5. The mean value e_n, is then multiplied with the respective OFDM symbol (n) [4].

$$e_n = \frac{1}{N_p} \sum_{k=0}^{N_p-1} p_{k,n} \hat{p}^*_{k,n} \tag{4.5}$$

In the following, the explained techniques will be applied to create a WLAN simulation model. Furthermore, three different communications standards (WLAN DSSS, OFDM, and BLE) will be compared to noise bursts as interferer. Figure 4.7 depicts the actual interference scenario in subfigure (a) and the respective BER curves (b)–(d) for different constellation mappings.



(a) Interference scenario: WLAN frame plus interference (top), AWGN plus interferer (BLE) (bottom)

(b) BER curve: QAM

(c) BER curve: 16-QAM

(d) BER curve: 64-QAM

Figure 4.7: (a) Current interference scenario: SINR of 0 dB and BLE as interferer, (b)–(d) BER curves over SINR for different modulations and interference sources (Noise, OFDM, DSSS, and BLE)

The sent WLAN frame consists of the legacy preamble (Figure 2.11) and payload data according to the MTU size of 1,500 Byte. In order to create realistic conditions, an equally distributed random time delay and frequency offset is applied. Furthermore, an interference signal of AWGN and a certain communication standard are added to the desired WLAN frame. The utilized interfering standard has an SNR of 30 dB, a quarter length of the transmitted WLAN frame, and changes the position anew, for every transmission. Regarding BLE as interferer, an equally distributed frequency hopping scheme, varying the center-frequency for every frame, is implemented. It must be mentioned, that the interfering standards were taken from the captured ISM band data to maintain effects like peak-to-average power ratio (PAPR) reduction.

Despite sweeping the SNR for BER curves like in an AWGN scenario, the signal to interference plus noise ratio (SINR) is investigated. Hence, the mean power of the whole interferer- and WLAN frame is divided to gain the respective SINR. Ensuring an appropriate packet detection and initial channel estimation, the utilized interfering standard has an additional time offset to avoid an overlap with the WLAN preamble. Consequently, only the payload data suffers from perturbations caused by additionally induced interfering signals.

Obviously, all BER curves (Figure 4.7) show a similar behavior regarding the interference source. Noise, WLAN DSSS, and OFDM result in BER curves which are close together. For higher QAM orders, the respective curves get even closer. Only distortions caused by BLE differ from noise completely. According to the observed simulation, it is possible to utilize modulated noise instead of digital data, such as WLAN DSSS and OFDM. Unfortunately, BLE cannot be described through this approach. Considering results from Table 3.3, the amount of this interferer is small. Hence, modulated noise is a promising technique for emulating ISM band traffic.

4.3.2 Implementation of a Low-Cost Interference Source

A low-cost interference source, modeling noise by a digitally-controlled output power level, has been realized [20]. This was done by using off-the-shelf components, such as noise diodes, power amplifiers, and digitally-controlled step attenuators (Figure 4.8).

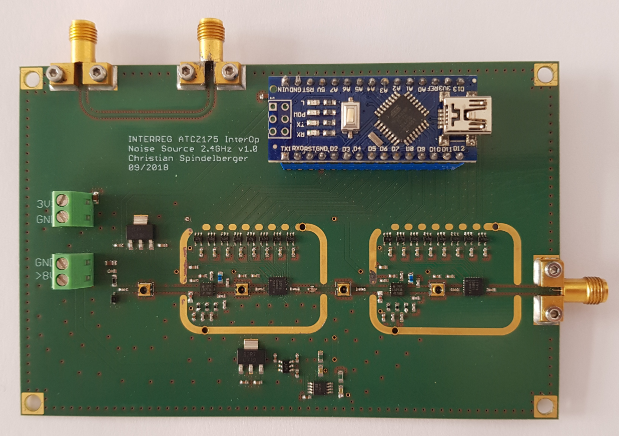


Figure 4.8: Low-cost interference source

The main system requirements are a high dynamic range DR > 50 dB and the opportunity to realize off-times down to $t_{\text{min}} = 1\,\mu s$. Furthermore, the power level must be large enough to overcome the insertion loss of the coupler, depicted in Figure 4.1. In order to meet these requirements, an overall gain of $\sim 100\,\text{dB}$, using one noise diode and two power amplifiers, is achieved. Figure 4.9 depicts the block diagram of the realized source. State-of-the-art noise sources utilize silicon avalanche diodes. Because of the avalanche effect, noise diodes are capable of enhancing the noise floor up to 30 dB for a wide frequency range (1 GHz...18 GHz). To describe the gain of such diodes, the excess noise ratio (ENR) is investigated. For instance, an ENR of 30 dB establishes a noise power density of $N_p = -174\,\text{dBm/Hz} + 30\,\text{dB} = -144\,\text{dBm/Hz}$ at room temperature $T_0 = 290\,\text{K}$.

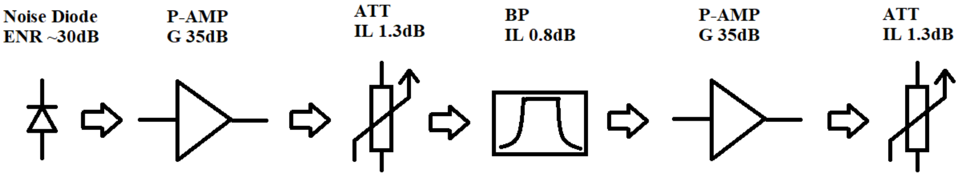


Figure 4.9: Low-cost interference source: block diagram [20]

In order to control the output power level, two digitally-controlled step attenuators, yielding a dynamic range of 63.5 dB, are implemented. A 14 bit broad parallel interface, controlled by a microcontroller, enables an output power resolution of 0.25 dB. A bandpass filter, optimized for WLAN applications in the 2.4 GHz band, is implemented to ensure a linear operation of the second power amplifier.

Arduino provides a simple structured integrated development environment (IDE) for prototyping applications. The observed noise source uses a serial interface to communicate over USB. Hence, a data sequence, realized by the respective random variables, created in Section 3.3, can be streamed to the microcontroller. Unfortunately, the memory depth is not high enough to generate sequences of the same length as a VSG. Therefore, it will be examined in Chapter 5, if this issue yields different performance results. In addition to this, the PSD of the noise source is optimized for the 2.4 GHz band. For WLAN channel 1, a maximum output power level of −2.67 dBm at a bandwidth of 20 MHz is achieved. Concerning the 5 GHz band, the output power is not high enough to overcome the insertion loss induced by the coupler. Consequently, only channel 1 at 2.412 GHz will be investigated.

Chapter 5

Interference Emulation Results

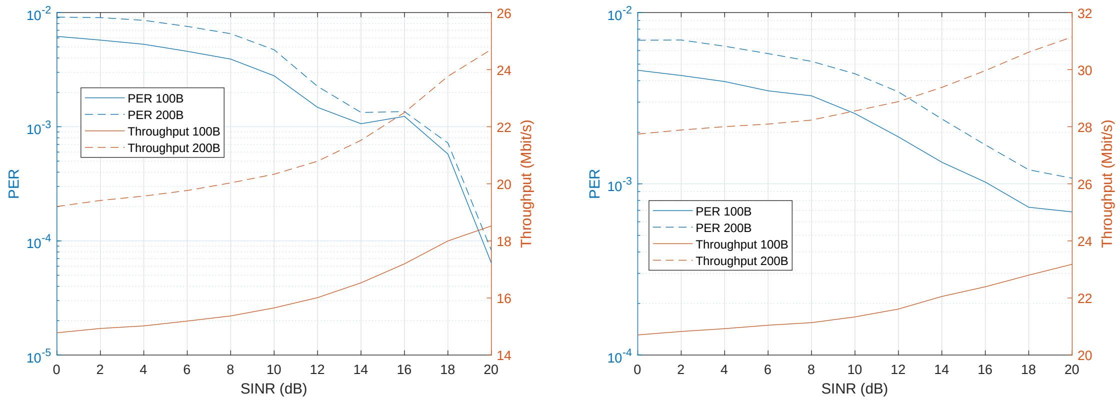
As the final step of this work, the introduced methods, explained in Chapter 3, will be examined using the WLAN test setup from Figure 4.1. Performance tests will give a graphical output to evaluate the observed scenarios.

First of all, reference measurements utilizing ISM band records, replayed by a VSG, are investigated in Section 5.1. These performance results are used to compare further techniques. Section 5.2 treats modulated noise as interfering signal. Two different models are applied and further improved by weighting power levels. It will be shown, that one decisive parameter influencing the comparison of interfering signals is caused by device dependent parameters. Furthermore, the capability of the low-cost interference source, described in Section 4.3.2, is analyzed. Therefore, the main output of Section 5.3 covers the replacement of a VSG by this low-cost implementation.

5.1 Reference Measurements

Based on the measurement setup from Figure 4.1, reference measurements are made utilizing a VSG as an interference source. Such a device enables replaying customized baseband data. Hence, two recordings with a total length of 20 s, analyzed in Section 3.3, are employed for interference injection. Because of memory depth limitations of the VSG, the investigated records are divided into four 5 s long sequences. Before that, the data is processed by a 22 MHz broad filter. In addition to this, the output power is adjusted such that the received power by the server equals the characterized power levels. The implemented variable attenuator of the setup controls the SINR by adjusting the received average power from the client relative to the mean power of the recordings, replayed by the interference source. In order to adjust the packet length of TCP performance tests using *iperf*, the maximum segment size must be set to the desired value. This adaptation limits the TCP window size, decreases the throughput drastically, and consequently, almost no retransmissions occurred for small packet sizes (< 200 Byte). As TCP performance tests did not lead to useful results, UDP is utilized in this chapter.

The PER and throughput are parameters which have to be estimated precisely. Therefore, the sample size ensuring an appropriate confidence interval has to be determined. Based on the central-limit theorem such statistical problems assume a normally distributed expectation value. These assumptions have to be dropped regarding the UDP performance tests. The WLAN test setup is surrounded by an unknown amount of devices influencing the obtained results. Consequently, the measurement scenario is affected by permanently changing ISM band traffic characteristics. Therefore, an empirical sample size of 100 performance tests, yielding a non-zero PER per SINR value, was set. This means that the PER computation for one operating point takes at least 16.7 min. Because of this time-consuming process, the two recordings with the highest traffic density have been investigated.



(a) HT-MCS-4

(b) HT-MCS-6

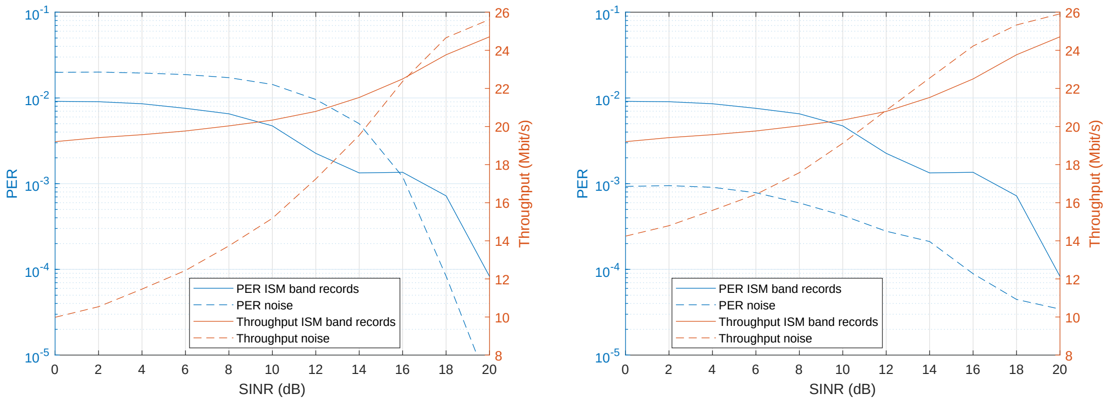
Figure 5.1: UDP performance test results for different modulations and packet lengths in WLAN channel 1

Figure 5.1 depicts the respective throughput and PER curves for a UDP performance test, varying the modulation index (HT-MCS-4/6: refer to Table A.1) and the packet length (100, 200 Byte). Due to longer time intervals, the probability of perturbations caused by interference becomes higher. Therefore, the PER is worse for longer packets, as expected. Although the PER of the short sequences is lower, the throughput is also decreased compared to the 200 Byte long sequences. The network layer is responsible for fragmenting data according to the MTU size. Hence, the amount of fragments increases for lower packet sizes, causing a higher data processing effort. Consequently, the throughput is decreased, while the PER is lower.

Another interesting outcome of these curves are the constant plateaus. The PER is constant for an SINR between 14 dB and 16 dB, depicted in Subfigure 5.1(a). Mainly, this behavior can be explained by CCA decision threshold settings. The preamble detection threshold has a maximum value of −82 dBm. Considering the measurement setup, the hidden node model is established. If the attenuator is set to low values for achieving a higher SINR, the client is able to detect some of the interfering signals. Due to the collision avoidance scheme, the transmitter (client) is capable to keep the PER constant in this SINR range. For higher attenuations, the transparency gets worse and the PER is increased. As the threshold is specified by an upper bound only, manufacturers set decision constraints individually.

5.2 Modulated Noise

As mentioned in Section 4.3, modulating noise is a promising technique to emulate ISM band interfering signals. According to the settings introduced in Section 5.1, two techniques creating random noise sequences are applied. Firstly, the time-quantized analysis (Section 3.2.1) will be investigated to model noise bursts.



(a) Step size: 100 µs

(b) Step size: 1 ms

Figure 5.2: Time-quantized noise bursts: UDP performance test results, packet length 200 Byte, HT-MCS-4, WLAN channel 1

Figure 5.2 depicts the measurement results of the corresponding step sizes for a UDP performance test in WLAN channel 1. The packet size is set to 200 Byte, utilizing the HT-MCS-4 (16-QAM-3/4) modulation. The performance results are compared to the ISM band recordings in terms of PERs and throughputs. Obviously, the ISM band-induced PER behaves differently compared to noise.

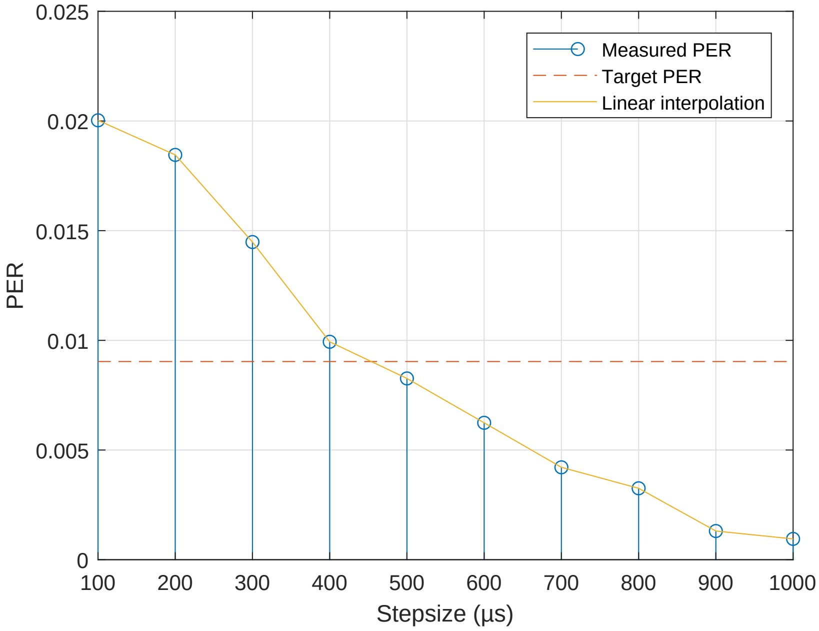
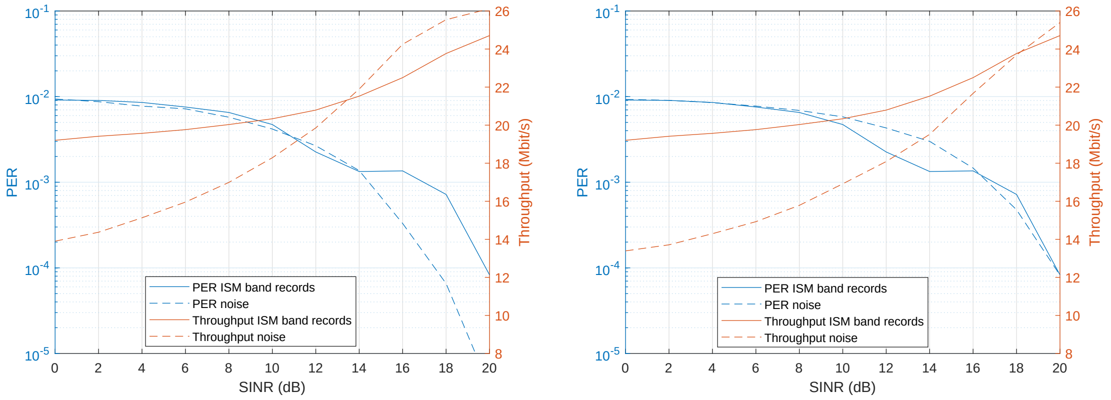


Figure 5.3: PER alignment function depending on the step size, SINR = 2 dB

The first issue concerning this problem is the missing preamble in noise bursts. For instance, the WLAN related PLCP sublayer introduces a header for frame detection. The interfering ISM band records force the server station to demodulate all incoming preambles lying beyond the CCA preamble detection threshold. Thus, interfering WLAN packets with a small power level relative to the desired frame may also cause packet errors. Another difference caused by CCA constraints is the missing PER plateau for modulated noise. The plateau indicates the area, where the isolation maintained by the coupler is in a range where the client is able to react on ISM band interfering signals, holding the PER constant. Unlike the preamble detection threshold (−82 dBm), the energy detection threshold is defined to be maximally −62 dBm. This difference of 20 dB causes the missing plateau within the observed SINR range for modulated noise. Furthermore, the noise related throughput has a higher slope for both step sizes. Considering throughputs from Subfigure 5.2(b), one can notice that the data rate is still lower than for digital data, although the noise related PER is smaller. This behavior emphasizes that time-quantized modulated noise is not capable to approximate the reference throughputs in this configuration. Nevertheless, it is possible to fit a fractional part of the PER with noise utilizing an appropriate step size.

In order to achieve a satisfying approximation, the main goal is to align the noise-induced PER curve with the one caused by ISM band traffic. Therefore, an SINR range has to be found where the PER curves have an almost constant shape, i.e., for SINR values below 6 dB. Figure 5.3 depicts the measured PER at an SINR of 2 dB for several step sizes and the target PER. Obviously, the alignment function shows a monotonically decreasing behavior in the observed interval. Hence, it is possible to find an appropriate step size to fit the desired PER. In order to also fit the PER curve for higher SINR values, it seems natural to enhance the mean power of the interference signal. As a consequence of increasing the power, the PER curve is shifted to the right. Corresponding to Figure 5.3, an appropriate step size of 455 µs has been found. Examining Figure 5.4 shows the concerning performance test results. In the left Subfigure 5.4(a) the PER is well aligned for SINR values between 0 dB and 14 dB. In order to further improve the fit, the mean power was enhanced by 2.2 dB. These adjustments lead to an approximation of almost the whole desired PER. The only issue that remains is the constant plateau which could not be emulated.

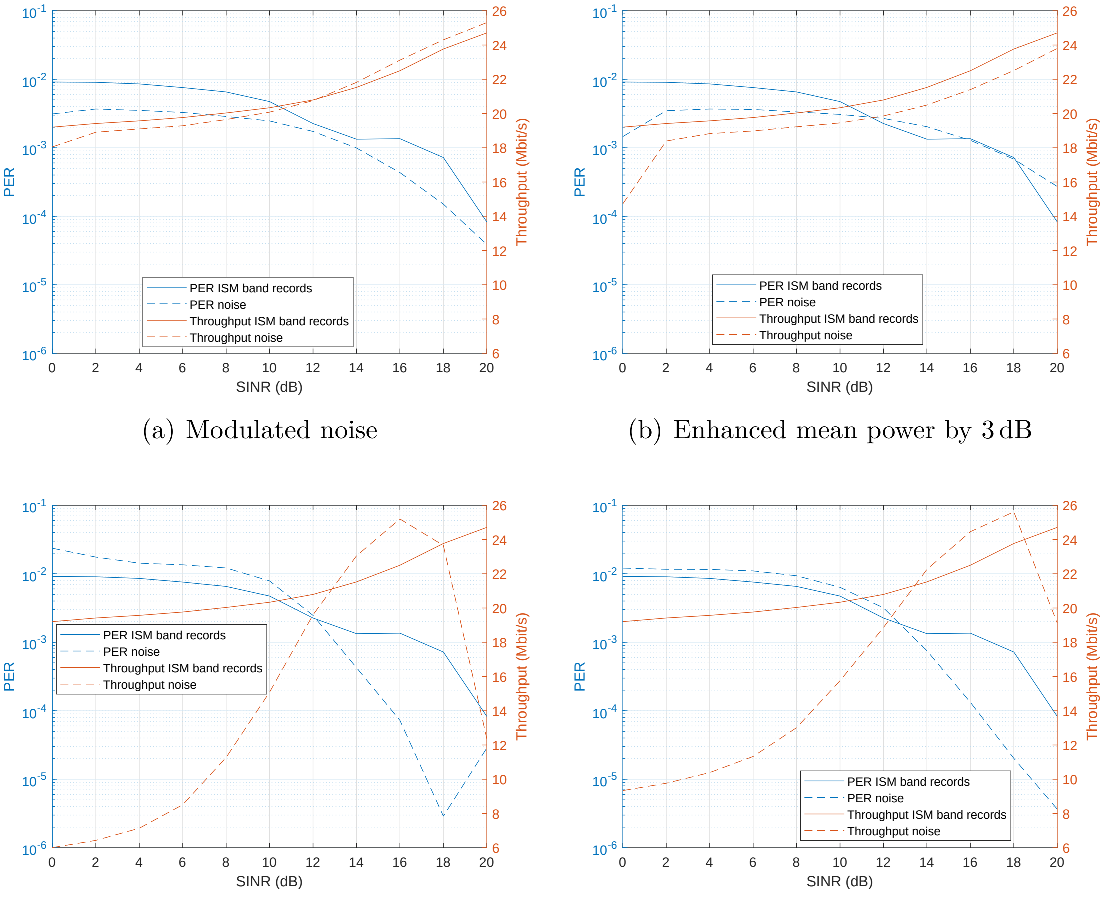


(a) Step size: 455 µs

(b) Enhanced mean power by 2.2 dB

Figure 5.4: Time-quantized noise bursts: UDP performance test results, packet length 200 Byte, HT-MCS-4, WLAN channel 1

The time-quantized analysis can be further extended by varying on- and off-times corresponding to the random variables gained in Section 3.3. The PER and throughput of this technique can be found in Subfigure 5.5(a). Varying on- and off-times yields a better fit concerning the throughput having a similar slope. This is not always the case, which will be discussed later in this section. Enhancing the mean power of the interfering signal by 3 dB results in a shift of the noise-induced PER to the right of the same range (Subfigure 5.5(b)). Hence, it is possible to fit the PER between 16 dB and 18 dB. One might notice the decay of PER and throughput for an SINR between 0 dB and 2 dB. Typically, an increased throughput is associated with a decay of the PER. As this is not the case in this SINR range, this behavior may be caused by crosstalk issues. WLAN modules based on MIMO have the opportunity to neglect receive paths when they are considered impractical. Presumably because of insufficient isolation, another received weak signal can be demodulated out of the receivers optimal operating point, resulting in a lower PER and throughput as well. Another technique to fit the PER curves is to omit power level values below a certain level with an exclusion rule. Subfigures 5.5(c) and 5.5(d) depict the respective performance results for an exclusion of power levels which are smaller than −70 dBm and −88 dBm.



(a) Modulated noise

(b) Enhanced mean power by 3 dB

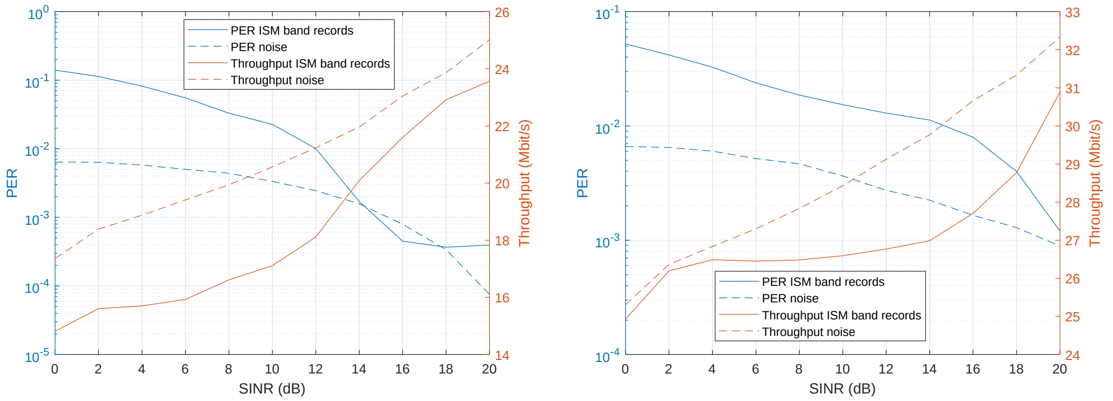
(c) Power threshold > −70 dBm

(d) Power threshold > −88 dBm

Figure 5.5: Modulated noise according to random variables from Section 3.3, packet length 200 Byte, HT-MCS-4, WLAN channel 1

Because of higher power levels, the PER becomes worse, enabling an approximation of the desired PER curve for SINR values between 0 dB and 12 dB. It is noteworthy to mention the strong throughput decay for an SINR beyond 16 dB. Adjusting a high SINR relates to a low attenuation between client and server. In this case, the power levels are high enough to be detected by the client yielding a lower throughput to keep the PER low. Thus, the working strategy of this implementation is oriented towards optimizing the PER by adjusting, for instance, the throughput. Due to blocking effects of the server, Subfigure 5.5(c) shows an increase of the PER between an SINR of 18 dB and 20 dB. At these power levels, the low-noise amplifiers are driven into their saturation region, causing an impractical performance.

As a consequence of saturation effects, it is not possible to align the noise-related PER curve by enhancing the mean power, like it was done in the time-quantized analysis. In addition to this, varying on- and off-times results in an even higher complexity, making this method impractical to approximate PER curves. Concerning all performance measurements in this chapter, the dilemma of throughput and PER approximation is present. Simply put, an approximation of the PER curve by noise does not lead to similar throughputs and vice versa. As already explained, this behavior is caused by CCA detection characteristics. Anyway, investigating Subfigure 5.5(a) might lead to the point that it is possible to fit throughput curves by modulating on- and off-times.



(a) HT-MCS-4, packet length 200 Byte

(b) HT-MCS-6, packet length 200 Byte

Figure 5.6: UDP performance test results: WLAN channel 36

Because of device-dependent characteristics, this is not always the case. Figure 5.6 depicts the performance measurement results compared to modulated noise in the 5 GHz band. Unlike the throughputs of Subfigure 5.5(a) in the 2.4 GHz band, the slopes appear in different shapes. The ISM band recordings show an area with a stagnation of traffic speed.

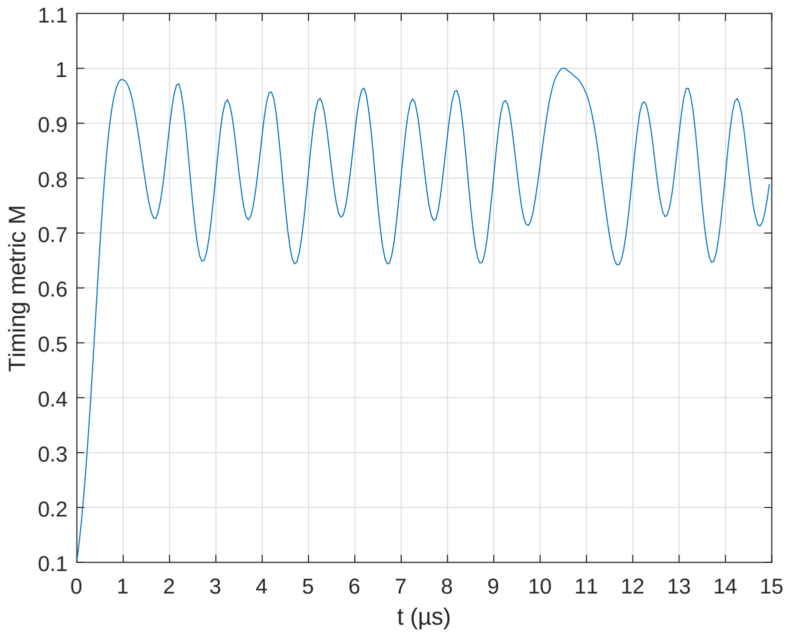


Figure 5.7: Coarse packet detection of a BLE packet utilizing Schmidl and Cox algorithm

This behavior gets even worse for higher modulations, while the noise related curve has a linear shape. Consequently, individual parameters, such as physical receiver design, or CCA threshold adaptation, influence the behavior of interfering signals in terms of PER and throughput stringently. For instance, one might think that the preamble detection threshold only works for WLAN related frames. Further investigations of the Schmidl and Cox algorithm showed that also BLE packets may cause a falsely detected WLAN packet. Figure 5.7 depicts the respective timing metric of a BLE packet from the measurement campaign. Although the signal is oscillating, the metric is beyond a level of 0.6. Hence, BLE packets can also be detected if they are beyond the preamble threshold causing a decreased throughput.

In conclusion, two methods modulating noise have been presented. Time-quantized analysis yields an appropriate technique by adjusting the step size and mean power of the interference signal to approximate the PER curve. In contrast to the PER curve, it turned out that the throughput could not be fitted with this method. If the on- and off-times are also varied, it is possible to gain similar throughputs. Unfortunately, this characteristic depends on several individual device dependent properties probably causing different shapes of the throughput curves.

5.3 Low-Cost Noise Source

In Section 4.3.2, a low-cost interference source was introduced. In order to verify this concept, performance measurement results will be compared to those made by a VSG. In the following, the main aspects comparing the VSG with the low-cost noise source will be discussed.

5.3.1 Memory Depth

The memory depth of the utilized VSG is limited to 256 MSa. At a sampling rate of, for instance, 51.2 MSa/s, the memory is capable to store a 5 s long recording. Considering the low-cost implementation, an *ATMEL 328P* microcontroller on an *Arduino Nano* board is used. The working principle of this device differs completely from the VSG. The noise source is not capable to repeat baseband data. It uses digitally-controlled attenuators to modulate amplified noise. Therefore, the memory depth of the microcontroller has to be examined. The on-/off-times, as well as the power level values, are generated by the corresponding random variables. The obtained values are then stored in the microcontroller. The total space for programming code is 30.72 kByte, and the dynamic storage used by variables offers 2,048 Byte. Obviously, the dynamic storage limits the maximum sequence length. The maximum amount of variables which can be stored without occurring instabilities is 500 values for the time-quantized method and 220 for modulated noise (on-/off-times). Due to these low values, several sequences have to be utilized to gain a fine resolution of the respective random variables. The measurement results will indicate if the reduced memory depth can be conquered by creating many smaller sequences and how high the resolution of the random variables must be to gain the same performance curves.

5.3.2 Reference Power Levels

As the low-cost noise source utilizes attenuators to modulate the output power over time, an absolute reference power level has to be defined. Therefore, the noise floor was investigated to calculate appropriate attenuation values relative to the randomly described power levels. It has been mentioned in Section 3.2.2, that the estimation of the noise floor density suffers from several effects. Hence, the calculation of the reference power level will include inaccuracies caused by interference signals. The survey of several sequences in the ISM band data, which were presumed to be noise, led to an average noise floor power level of −98 dBm (WLAN channel 1, BW = 22 MHz). Based on this reference level (P_{ref}), the attenuation values are calculated using the random power levels (P_{rand}) and the dynamic range of the source ($\mathrm{DR}_{\mathrm{ATT}} = 63.5$): $\mathrm{ATT} = \mathrm{DR}_{\mathrm{ATT}} - (P_{\mathrm{rand}} - P_{\mathrm{ref}})$.

5.3.3 Time Critical Behavior

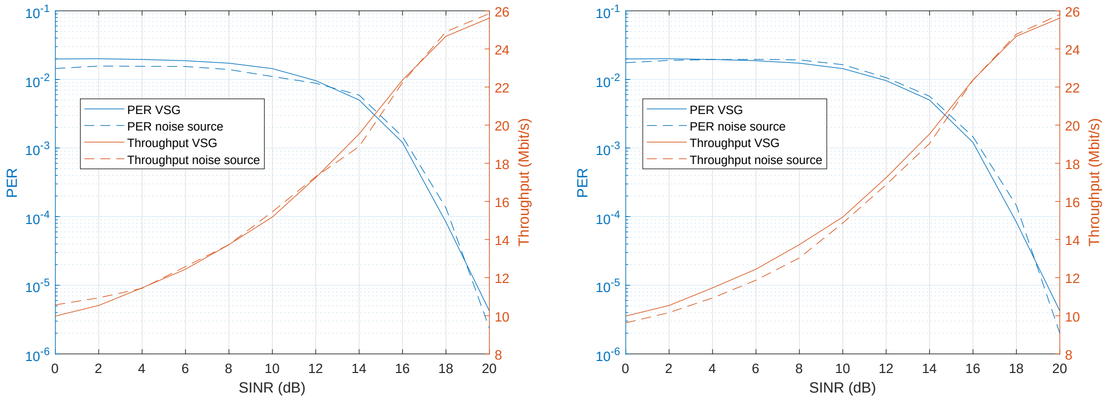
Usually, standard RF attenuators have an undefined output during state transitions, causing an overshoot of the envelope. Therefore, glitch-free attenuators have been utilized to conquer this effect. One negative aspect associated with such elements is the higher settling time compared to general purpose attenuators. The specified maximum settling time of the assembled attenuators is $< 400\,\text{ns}$.

Another important aspect concerning the timing resolution of the low-cost noise source is the microcontroller. The digital output pins, which are controlling the attenuators, are separated into eight bits long ports. They can be easily set by simply manipulating the respective bits. Unfortunately, these ports are not well synchronized, causing a switching delay between them up to a few hundred nanoseconds.

In contrast to the noise source, the VSG utilizes a digital-to-analog-converter with a resolution of 16 bits to replay baseband data directly from an internal storage. The sampling rate can be adjusted up to 120 MSa/s. In the following section, it will be investigated if such high timing resolutions are mandatory to achieve the same performance test results.

5.3.4 Comparison of Results

First of all, applying the time-quantized analysis to the low-cost noise source will be analyzed in contrast to the VSG. Therefore, several sequences with an amount of $N = 500$ samples each are utilized to jam the WLAN test setup.

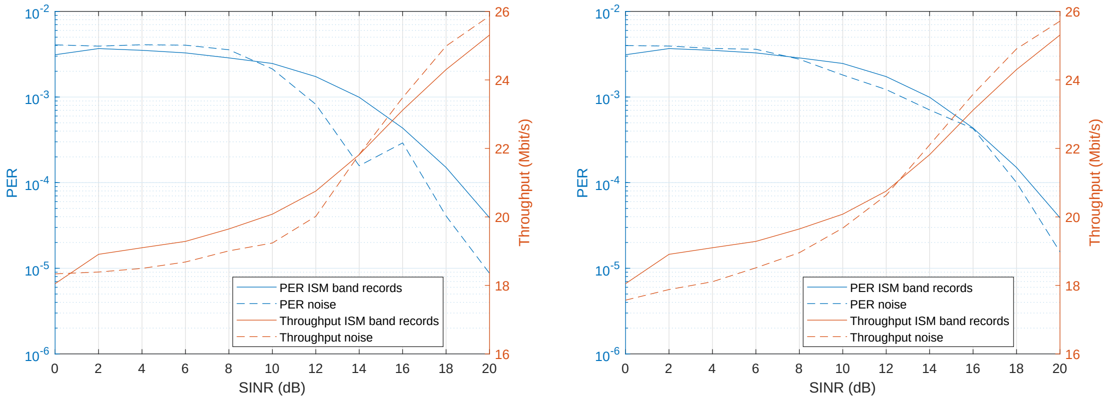


(a) $N = 500$

(b) $N = 2{,}000$

Figure 5.8: Comparing VSG with noise source: time-quantized noise, step size: 100 µs, HT-MCS-4, packet length 200 Byte, WLAN channel 1

In Figure 5.8, one can see UDP performance test results for different power level distribution resolutions (N). Obviously, an amount of $N = 500$ (one sequence) already yields similar performance curves, which can be found in Subfigure 5.8(a). Increasing the number of sequences fourfold (Subfigure 5.8(b)) fits the PER curve even better, but the throughput shows stronger deviations below an SINR of 10 dB.



(a) $N = 220$

(b) $N = 2{,}200$

Figure 5.9: Comparing VSG with noise source: modulated noise, HT-MCS-4, packet length 200 Byte, WLAN channel 1

Utilizing on- and off-times allocates more memory of the microcontroller. Hence, one sequence consists of $N = 220$ on-/off-times and power level values, i.e., 660 values in total. Due to varying two more parameters, compared to the time quantized method, the amount of parameter combinations increases. Hence, the memory depth is a serious issue concerning this modulation technique. The performance curves from Figure 5.9 indicate that a high amount of parameter combinations are necessary to gain similar results. Even though the approximation gets better for an increasing amount of samples, the PER curve still shows a nonsmooth behavior for $N = 2{,}200$.

Recapitulating this section, regarding the realization issues of the noise source and performance test results, indicates that it is possible to replace a VSG by a low-cost implementation. The most serious impact turned out to be the memory depth of the microcontroller. As the time-quantized method led to a satisfactory approximation even for low resolutions, the utilization of on- and off-times demands an on-board logic with more memory space.

Chapter 6

Conclusion

First, a measurement campaign was held to describe interference signals in the ISM band. Two methods have been found to analyze the recorded baseband data in terms of on-/off-times and respective power level distributions, i.e., time-quantized analysis and energy detection. Furthermore, the detected events were classified according to their communication standards. It turned out that more than 90 % of the investigated traffic was related to WLAN communications. Therefore, it was verified if it is possible to replace a real-time recording device (VSA) by a Linux PC system with an implemented WLAN module to describe such a scenario. Because of several issues this was not the case. Mainly, crosstalk and isolation related problems of the WLAN module led to ambiguous measurement results compared to the VSA.

In order to characterize interference effects in terms of throughputs and PERs, a WLAN based measurement setup has been created. The setup was perturbed by an interference source while traffic performance tests were performed. The reference interference source was a VSG replaying certain ISM band records from the measurement campaign. As the amount of communication standards is dominated by WLAN, an alternative interference source, consisting of a Linux PC system with a WLAN module implemented, was investigated. Due to limitations concerning the timing and the adaptation of the power level, it was not possible to obtain a useful interference source utilizing Linux. Therefore, another method was introduced. With a simple simulation model, it was successfully confirmed that noise causes a similar BER, such as WLAN modulations. Thus, a low-cost interference source, which is capable to generate noise with a digitally-controlled output power level, was presented.

With the gained knowledge from the measurement campaign, random variables of on-/off-times and power levels have been created. These variables were used to create modulated noise sequences and perturb the WLAN test setup with the VSG. The utilized techniques (time-quantized noise bursts and modulated noise varying on-/off-times) were compared to ISM band traffic. It turned out that the time-quantized method yields the best approach to approximate PER curves. Due to several effects, such as a missing preamble and individual CCA adjustments, it was not possible to fit the throughput. For some cases modulating on- and off-times came to similar throughput results. Unfortunately, this behavior was not reproducible for every modulation and frequency band. Furthermore, using modulated noise for fitting the PER curve led to blocking effects of the WLAN modules. In conclusion, it was not possible to approximate the PER and throughput curve simultaneously for both techniques.

At last, the low-cost interference source was utilized to emulate the same scenarios as the VSG. Considering the time-quantized method, already a low amount of data sequences ($N = 500$) were sufficient to yield similar performance results compared to the VSG. The memory depth of the noise source became a limiting factor for modulating on- and off-times. As the amount of parameter combinations increases drastically using this technique, a microcontroller with a larger storage is mandatory to gain satisfactory results.

Because of device dependent properties regarding WLAN modules, the performance results caused by modulated noise and ISM band traffic are different. In order to further improve this concept, for instance, hidden Markov chains could be utilized to create an incremental learning system, which is capable to approximate each operating point (SINR) by weighting power level distributions or varying step sizes.

References

[1] Northstream, White Paper, 2017. *Massive IoT: different technologies for different needs.* http://northstream.se/insights/white-papers/massive-iot-different-technologies-for-different-needs 1, 2

[2] IEEE Standards Association, IEEE Std 802.11, 2016. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* 4, 11, 14

[3] Matthew Gast. *802.11 Wireless Networks: The Definitive Guide.* O'Reilly, Sebastopol, Canada, 2002. 5, 6, 7, 8, 9, 10

[4] Travis F. Collins, Robin Getz, Di Pu, Alexander M. Wyglinski. *Software-Defined Radio for Engineers.* Artech House, 2018. 11, 50

[5] Keysight Technologies http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/wlan-ofdm/Content/ofdm_80211-overview.htm 13

[6] Timothy M. Schmidl, Donald C. Cox. *Robust Frequency and Timing Synchronization for OFDM.* IEEE Trans. Commun., Vol 45, No. 12, December 1997. 13, 48

[7] Robin Heydon. *Bluetooth Low Energy: The Developers Handbook.* Prentice Hall, 2013. 18

[8] IMST GmbH, 2012. *Channel Access Rules for SRDs.* https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Koexistenzstudie_EN.pdf?__blob=publicationFile&v=2 20

[9] Youping Zhao, Brian G. Agee, Jeffrey H. Reed. *Simulation and Measurement of Microwave Oven Leakage for 802.11 WLAN Interference Management.* IEEE Int. Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings, 2005. 21

[10] Harmonised European Standard. *ETSI EN 301 893 V2.1.1, 5 GHz RLAN; Harmonised Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU*, 2017. 22

[11] S. Atapattu, C. Tellambura, H. Jiang. *Energy Detection for Spectrum Sensing in Cognitive Radio.* Springer, 2014 29

[12] Andrea Mariani, Andrea Giorgetti, Marco Chiani. *Effects of Noise Power Estimation on Energy Detection for Cognitive Radio applications.* IEEE Trans. Commun., Vol 59, No. 12, December 2011. 29, 30

[13] Tektronix. *Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements*, 2013. `https://www.cnrood.com/en/media/solutions/Wi-Fi_Overview_of_the_802.11_Physical_Layer.pdf` 32

[14] Michael Galarnyk. *Understanding Boxplots*, 2018. `https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51` 34

[15] Adrian W. Bowman and Adelchi Azzalini. *Applied Smoothing Techniques for Data Analysis.* Clarendon press, Oxford, 1997. 35

[16] David Freedman, Persi Diaconis. *On the Histogram as a Density Estimator: L_2 Theory.* Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, Springer, 1981. 37

[17] Jon Dugan, Seth Elliott, Bruce A. Mah, Jeff Poskanzer, Kaustubh Prabhu. *iPerf - The ultimate speed test tool for TCP, UDP and SCTP.* `https://iperf.fr/` 44

[18] Candela Technologies. `https://www.candelatech.com/` 45, 70, 75

[19] Philippe Biondi. *Scapy documentation.* `https://scapy.readthedocs.io/en/latest/` 46

[20] Christian Spindelberger BSc. *Noise source with a digitally-stepped output power level*, 2018. `http://www.interreg-interop.eu/results/interference_emulator/` 52, 53

Appendix A

WLAN Setup Instruction Guide

This configuration guide provides information about installing the required WLAN test setup, introduced in Section 4.1. As this manual is based on a Linux operating system, basic knowledge about such systems and respective programming skills are mandatory. In the following, utilization of the required kernel and further instructions, such as firmware installation and adaptation of transmission parameters, will be discussed.

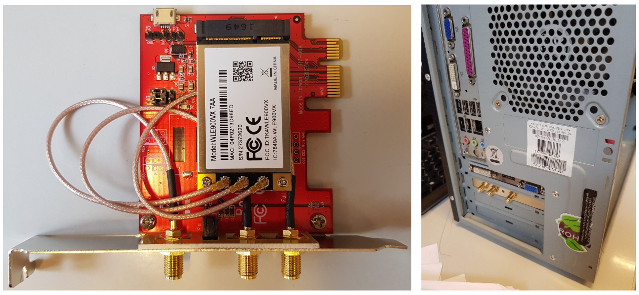


Figure A.1: WLAN module (left), installed in PC (right)

The measurement setup mainly consists of two PCs with the distribution Ubuntu LTS 18.04, two *Atheros* WLAN cards (*WLE900VX*) of 10$^{\text{th}}$ generation (*ath10k*) with a proper mini-PCIe to PCIe adapter and additional RF equipment. As already mentioned, only single channel scenarios are investigated. The respective WLAN cards are therefore connected over one port via coaxial cables, a coupler, and a variable attenuator. The remaining ports are terminated with 50 Ω loads. Figure A.1 depicts the WLAN module with the according PCIe adapter. Furthermore, three UFL-to-SMA cables are utilized to connect further components, such as the coupler and attenuator.

Precompiled Kernel

In order to characterize different interference sources, it is necessary to have full control of PHY parameters. *Candela Technologies* (CT) offers a precompiled Linux kernel, which enables specifying, for instance, utilized bandwidth or transmit power. Typically, Linux offers commands out of the box to change most of these settings. Unfortunately, not all devices are supported by current drivers. Hence, CTs kernel, which can be downloaded on their homepage (`www.candelatech.com`), must be considered.

In order to install the required kernel, download the compressed file and save it to the home directory. Subsequently, untar the file and configure the bootloader *GRUB* appropriately.

Firmware Installation

Current firmware versions for *ath10k* devices suffer from several software errors. CT implemented a further developed version, which can be downloaded for the respective WLAN module category. Mainly, two different types, called **wave-1** and **wave-2**, exist. The first type describes single user MIMO systems, which are capable of transmitting data to one single station at a time. Furthermore, such devices are limited to three spatial streams, resulting in a 3x3 MIMO system and a maximum bandwidth of 80 MHz. Improving this category led to **wave-2** enabling multi-user MIMO. Simply put, it is possible to transmit to several stations at the same time. This is achieved by beamforming and broadening the bandwidth up to 160 MHz, yielding a higher throughput compared to **wave-1**. Considering the WLAN test setup from Section 4.1, a **wave-1** module was chosen, as no multi-user scenario is investigated.

In this setup, **wave-1** *WLE900VX* WLAN cards with a *QCA988X* chipset are utilized. In the following, all instructions are based on this device category. After choosing the right firmware version from CT homepage, the following instructions implement the desired software [18]:

```
mkdir -p /lib/firmware/ath10k/QCA988X/hw2.0/orig
mv /lib/firmware/ath10k/QCA988X/hw2.0/firmware-[3456].
    bin /lib/firmware/ath10k/QCA988X/hw2.0/orig/
cp firmware-2-ct-full-community.bin /lib/firmware/
    ath10k/QCA988X/hw2.0/firmware-2.bin
```

Furthermore, the actual **board.bin** file from the official firmware site of *Atheros* has to be saved into the directory mentioned above. After rebooting the system, one can check with **ethtool** if the right firmware version has been loaded. If the output contains a **"-ct"**, such as:

```
ethtool -i wlan0
10.1-ct-8x-_xtH-019-ddf2a35
```

the installation was successful.

Network Configuration

Connecting the two WLAN cards requires a static IP assignment for both sides (server and client). Notice that the cards can communicate only if they share the same **netmask** (255.255.255.0). In this setup, the server station has the IP address **10.0.0.1** and the client **10.0.0.2**. In order to set the network settings permanently, the **/etc/network/interfaces** file must be changed accordingly:

```
# network file /etc/network/interfaces

auto lo
iface lo inet loopback

auto wlan0
iface wlan0 inet static
address 10.0.0.1
netmask 255.255.255.0
```

With the **ifconfig** command, one can check the current IP address of the respective WLAN module after a reboot. The assigned IP address can be found next to excerpt *inet*.

```
ifconfig wlan0
wlan0:  flags=4163<....>  mtu 1500
inet 10.0.0.1  netmask 255.255.255.0
ether ff:ff:ff:ff:ff:ff
txqueuelen 1000  (Ethernet)
```

Access Point Configuration

One PC will act as an access point (server), which is configured in the hostapd file: **/etc/hostapd/hostapd.conf**. This file has a huge amount of configurable settings which define access point capabilities. For instance, it is possible to choose drivers, frequency bands, cyclic prefix lengths, and many more. The following hostapd file configures an access point called "*my_ap*" for WLAN channel 1 in the 2.4 GHz band. It enables the IEEE 802.11n standard up to 40 MHz. Key managements and a short CP of 0.4 µs have been disabled. For detailed information, please refer to the hostapd configuration file description:

https://w1.fi/cgit/hostap/plain/hostapd/hostapd.conf

```
# hostapd file /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=my_ap
channel=1
ignore_broadcast_ssid=0
country_code=US
ieee80211d=1
hw_mode=g
macaddr_acl=0
auth_algs=1
ieee80211n=1
ieee80211h=1
ht_capab=[HT40+]
```

To start the **hostapd daemon** per default, please uncomment and change the following line in file **/etc/default/hostapd**:

```
DAEMON_CONF = /etc/hostapd/hostapd.conf
```

After rebooting the system, one can check with **iwconfig wlan0** if the changed settings have been configured successfully. The output should then state something like this:

```
iwconfig wlan0
wlan0    IEEE 802.11  Mode:Master
Tx-Power=30 dBm
Retry short limit:7   RTS thr:off
Fragment thr:off
Power Management:on
```

WPA Supplicant

To ensure a stable connection between the two WLAN cards (server and client) the **wpa supplicant** has to be configured. This daemon will force the client to associate with the specified access point automatically. First of all, a **wpa_supplicant.conf** file has to be created in the following directory: **/etc/wpa_supplicant**.

```
# wpa_supplicant file
# /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=/var/run/wpa_supplicant
eapol_version=1
ap_scan=1

network={
ssid = "my_ap"
key_mgmt=NONE
}
```

According to the **hostapd** file mentioned above, no key encryption is utilized. For a more detailed description of the **wpa_supplicant.conf** file, refer to the following link: `https://w1.fi/cgit/hostap/plain/wpa_supplicant/wpa_supplicant.conf`.

To activate the **wpa supplicant**, the following line must be added to the **/etc/network/interfaces** file below the desired interface:

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

The applied configurations can be tested with the following line:

```
wpa_supplicant -i wlan0 -D wext -c /etc/wpa_supplicant/
   wpa_supplicant.conf -d
```

Afterwards, **iwconfig** shows if connecting to the desired access point was successful:

```
iwconfig wlan0
wlan0      IEEE 802.11  ESSID:"my_ap"
Mode:Managed  Frequency:2.412 GHz
Access Point: ff:ff:ff:ff:ff:ff
Bit Rate=1 Mb/s   Tx-Power=30 dBm
Retry short limit:7   RTS thr:off
Fragment thr:off
Power Management:on
Link Quality=26/70  Signal level=-84 dBm
```

The extended service set identifier (ESSID) mentions the associated access point. In this example, the ESSID is “*my_ap*”, indicating a successful connection with the desired server station.

WLAN PHY Configuration

After successful implementation of the CT kernel and applying the settings mentioned above, full control of PHY parameters is available. Basically, all PHY settings like bandwidth, modulation or transmit power are adjusted and explained by the **ct_special** file, located in the **/sys/kernel/debug/ieee80211/ph0/ath10k/** directory. In order to change PHY characteristics, the **echo** command can be utilized. All parameter settings are encoded by an ID and a 64 bits long codeword.

Set Transmit Power

One of the most important settings is the transmit power. As already mentioned in Section 3.3, WLAN modules suffer from insufficient isolation. Hence, it might happen that transmitted data packets between server and client are also received over the air. Concentrating the established traffic onto the desired wired measurement setup (Figure 4.1) requires a transmit power as low as possible. The default level is 20 dBm for 2.4 GHz and 23 dBm for 5 GHz per spatial stream. The following command sets the transmit power to a minimum of 0 dBm. The ID of this command is 6, followed by 8 hexadecimal numbers defining the transmit power:

```
echo 0x600000000 > /sys/kernel/debug/ieee80211/phy0/
  ath10k/ct_special
```

It is not possible to set these parameters multiple times. Thus, it is recommended to minimize the transmit power once before proceeding with performance tests using **iperf**. In addition to this, one must be careful with this parameter, too large values may damage the device.

Set Modulation Parameters

Utilizing the **iw** command offers the opportunity to set certain modulation parameters, yielding different throughputs and PERs. The IEEE 802.11n standard, also called high throughput (HT) modus, defines modulations utilizing indices (MCS). Table A.1 quotes the MCS parameters for single spatial streams. The following example defines the high throughput (HT) modulation index 0 for 2.4 GHz:

```
iw dev wlan0 set bitrates legacy-2.4 ht-mcs-2.4 0 vht-
  mcs-2.4
```

Furthermore, the legacy rate (WLAN DSSS for 2.4 GHz) and very high throughput values (IEEE 802.11ac) can be set. As the main focus in this work lies on IEEE 802.11n, another example for setting MCS-3 in the 5 GHz band is given:

```
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 3 vht-mcs-5
```

HT-MCS	**Modulation &** **Coding**	**Data rate** **BW** = 20 MHz **CP** = 0.8 µs	**Data rate** **BW** = 40 MHz **CP** = 0.8 µs
0	BPSK-1/2	6.5 Mbit/s	13.5 Mbit/s
1	QPSK-1/2	13 Mbit/s	27 Mbit/s
2	QPSK-3/4	19.5 Mbit/s	40.5 Mbit/s
3	16-QAM-1/2	26 Mbit/s	54 Mbit/s
4	16-QAM-3/4	39 Mbit/s	81 Mbit/s
5	64-QAM-2/3	52 Mbit/s	108 Mbit/s
6	64-QAM-3/4	58.5 Mbit/s	121.5 Mbit/s
7	64-QAM-5/6	65 Mbit/s	135 Mbit/s

Table A.1: IEEE 802.11n modulation indices

Defining Certain Transmit Bandwidths

According to Table A.1, a lot of different modulation types exist. Therefore, the respective bandwidth has to be adapted to ensure that the desired modulation is utilized. The definition of three possible bandwidths is stated in the following example:

```
# 20MHz
echo 0xE00000006 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
# 40MHz
echo 0xE00000005 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
# 80MHz
echo 0xE00000003 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
```

Further encoding parameters and their definitions can be found in the **ct_special** file. It must be mentioned that the applied transmit bandwidth will only take effect until the modulation parameters have been set [18].

Performance Tests

Now, all necessary settings for a successful performance test using **iperf3** have been introduced. After connecting the devices mentioned in Table 4.1, according to Figure 4.1, an initialization sequence of the test setup will be explained. The following adjustments must be applied on both sides, server and client.

1. Set the transmit power to 0 dBm:

```
echo 0x600000000 > /sys/kernel/debug/ieee80211/phy0
   /ath10k/ct_special
```

2. Disable fragmentation- and RTS thresholds:

```
iwconfig wlan0 rts off
```

3. Set the desired transmit bandwidth (20 MHz):

```
echo 0xE00000006 > /sys/kernel/debug/ieee80211/phy0
   /ath10k/ct_special
```

4. Setting desired modulation index (HT-MCS-0):

```
iw dev wlan0 set bitrates legacy-2.4 ht-mcs-2.4 0
   vht-mcs-2.4
```

After initializing both systems, the **iperf3** client can be utilized for performance measurements. Mainly, two performance tests, based on UDP and TCP, are available. Independent of the utilized protocol, the **iperf3** server must be started on the access point, listening to a specified port. If the server is ready for data reception, the client can start **iperf3** as well. The following sequence extends the initialization process mentioned above.

5. Start **iperf3** server (IP: 10.0.0.1) on the access point:

```
iperf3 -s
```

6. Start the performance test (UDP, packet size 100 Byte):

```
iperf3 -c 10.0.0.1 -u -b 10G -l 100B -f m
```

According to the listing above, the server station starts listening to incoming packets. Subsequently, the client starts a UDP test with the respective destination IP address (10.0.0.1), a maximum target bandwidth of 10 Gbits/s, and a packet size of 100 Byte. The target bandwidth must always be set to higher values than the desired modulation is capable to achieve. Otherwise, the **iperf3** performance results will not yield maximum throughput values. For further informations about adjusting **iperf3**, refer to `https://iperf.fr/iperf-doc.php`.

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct – Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, July 17, 2019 *Christian Spindelberger*