TECHNISCHE
UNIVERSITÄT
DARMSTADT

COVERT CHANNELS IN IEEE 802.11 (WI-FI) ON MAC & PHY

HALIS ALTUG, STEPHAN PFISTNER, ATHIONA XHOGA

Secure Mobile Networking Lab Exercise / Project (SS 2014)

Secure Mobile Networking Lab
Department of Computer Science

Covert Channels in IEEE 802.11 (Wi-Fi) on MAC & PHY

Submitted by Halis Altug, Stephan Pfistner, Athiona Xhoga

Tutor: Matthias Schulz, Jiska Classen

Technische Universität Darmstadt
Department of Computer Science
Secure Mobile Networking Lab

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

ACK	Acknowledgement
AP	Access Point
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CP	Cyclic Prefix
CRC	Cyclic Redundancy Check
DCF	Distributed Coordination Function
FCF	Frame Control Field
FCS	Frame Check Sequence
FER	Frame Error Rate
FFT	Fast Fourier Transformation
FPGA	Field Programmable Gate Array
HICCUPS	Hidden Communication System for Corrupted Networks
HTTPS	HyperText Transfer Protocol Secure
ICI	Inter-Carrier Interference
IFFT	Inverse Fast Fourier Transformation
ISI	Inter-Symbol Interference
LTS	Long Training Symbols

MAC	Media Access Control
OFDM	Orthogonal Frequency-Division Multiplexing
PHY	Physical Layer
PRNG	Pseudo-Random Number Generator
PSDU	Physical Layer Service Data Unit
PSDU	Physical layer Service Data Unit
QPSK	Quadrature Phase Shift Keying
RSS	Received Signal Strength
SNAP	Subnetwork Access Protocol
STA	Station
STS	Short Training Symbols
WARP	Wireless Open-Access Research Platform
WARPLab	Wireless Open-Access Research Platform Lab
WEP	Wired Equivalent Privacy
WiPad	Wireless Padding
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

1 PROJECT DEFINITION

In the past few years, we have experienced a rapid change in network security. One demanded aspect involves covert channels in networks. In this lab, we will focus on wireless network-based hidden information transportation using steganography. The idea of hiding information is not new, different solutions have been investigated by researchers. Most of them were based in the upper layers of the network, while the lower layers were neglected. On that account, the goal of this lab is the exploration of approaches of hidden transportation of data in the Media Access Control (MAC) and Physical Layer (PHY) layers.

In this report, the term covert channel, hidden channel and steganographic channel have the same definition.

1.1 PROJECT OVERVIEW

The goal that we aim to achieve is to investigate approaches on hidden communications channels between stations in IEEE 802.11 based network infrastructure, that are based on MAC and PHY layers. We will asses a subset of these by ourselves using various means including a implementation on the WARP software-defined-radio platform in a practical environment.

The evaluation of our implementations is performed by comparing them using metrics, which have to be chosen and discussed. These should include the steganographic channel itself as well as the host channel carrying the covert channel and communication of other clients using the same network. In particular, we will have to consider the impact of the covert channel on the performance of standard-conforming IEEE 802.11g communications. The assessment will be discussed further in chapter 4.

1.2 SCENARIO

The practical evaluation is based on the scenario illustrated in figure 1.1. It describes the following situation: two participants A and B try to perform a convert communication in a infrastructure mode 802.11 wireless network. Therefore A communicates over an Access Point (AP)[1] whereas B eavesdrops the communication between A and and the AP to establish a half-duplex covert channel from A to B. A hidden channel is constructed between the participants A and B in such a way that the hosting connection between A and its partner and in consequence A and the AP is not inhibited. A attacker may be located anywhere in this network.

The goal of this approach is to hidden transport information between A and B while retaining anonymity of B and the possibility to repudiate any information flow from A to B towards the attackers perspective. Thus, not only the data but also the receiver is

1 It is irrelevant if the other partner of this communication is part of this wireless network or not, as any communication in a infrastructure mode 802.11 network is forwarded through the AP on the first hop nevertheless. The partner could be for example a server in the internet as shown in the figure.

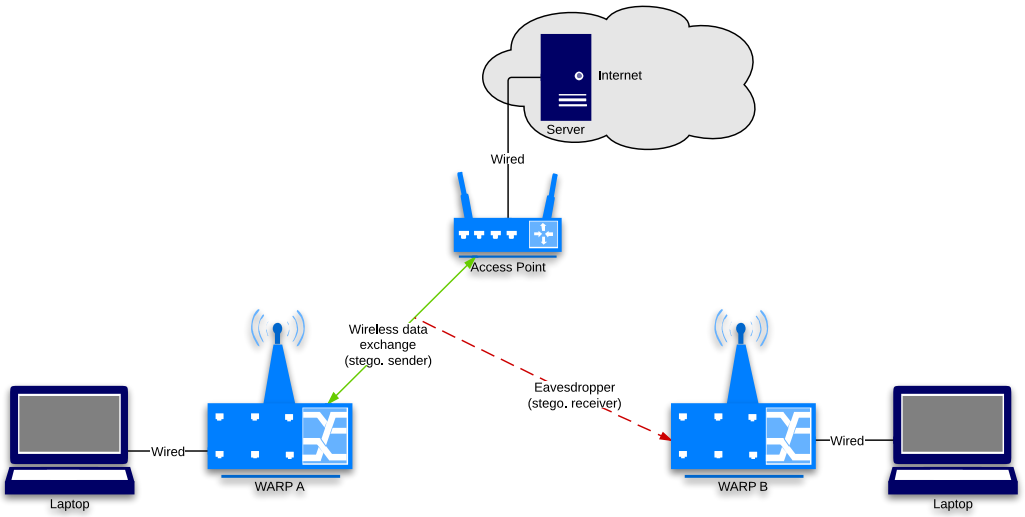


Figure 1.1: Covert channel deployment scenario [source: own source.]

hidden. For simplicity this channel is only half-duplex, but it can be applied two times in parallel in the same network. Then, B would be communicating to the AP while A is listening to this communication resulting in an additional covert channel from B to A.

In our implementation nodes A and B will be WARP nodes as they will contain the necessary modifications to establish the covert channel. Each has a laptop connected for controlling during runtime. The AP can be any kind of hardware, a of-the-shelf or WARP device will both work. It requires no modifications to the 802.11 stack at all and can validate that network works despite the covert channel augmented communication.

1.3 ATTACKER MODEL

Basically we can assume the attacker to have at least the capabilities of a regular network station. This means he can overhear the network and thus read unencrypted data of messages not intended for him, if not protected and accessible. He may even look at data of lower layers apart from the normal protocols operation, like specifics of PHY modulation or padding data. He may also inject messages into the network or try to forge messages of other nodes. Unlike a regular node these messages can be maliciously malformed. It is safe to assume the attacker can also jam, and thus destroy, messages over the air. This corresponds to the capabilities of an attacker in the Dolev-Yao model.

As covert channel do not employ cryptography per se – which is an important method to protect against a Dolev-Yao attacker in a regular network – schemes have to use different measures to protect themselves. Depending on the intent of the attacker he may use his capabilities to reach certain typical goals. Will we assume the attacker to have to following goals:

DETECTION OF PRESENCE - A first step for an attacker is to detect the presence of a covert channel in a host channel (here: IEEE 802.11 transmission). To achieve this, the attacker will mostly overhear the network and look out for atypical changes in

the behavior of the other nodes, e.g. changes in timings, transmission or error rates. Although less relevant, active capabilities can also be used, e.g. to provoke errors.

READ CHANNEL - After establishing a specific certainty that a covert channel is currently being used the attacker may change into actively overhearing the channel. Depending on the covert channel this may be easy, it its main defense is to not be detected in the first place. At this point cryptography can help to confidentiality protect the data sent over the covert channel.

PREVENT COMMUNICATION - Alternatively to overhearing the channel[2] the attacker may wan to prevent the communication altogether. For this, he needs to use his active capabilities, that is injection/jamming. Depending on the covert data it may be possible to destroy the covert data only, leaving the hosting channel intact, e.g. jamming certain parts of a IEEE 802.11 frame. Injecting malicious messages to attack the participants of a covert channel may also be possible. Jamming may not result in a successful attack in all cases as the attacker may jam his own and his neighbor's reception but the intended receiver of the covert message may be out of his transmission range and receive the message nonetheless or jamming signals are dampened by modulation. If the attacker controls the AP itself, dropping the covert message is not enough to prevent communication as the presented model does not rely on message forwarding but indirect communication by the recipient being overhearing the host channel.

IDENTIFY PARTICIPANTS - To end the communication altogether the attacker may resort to locate and/or identify the participants and prevent access to the network. As this is a wireless network this can be quite hard without touching encryption of the channel.[3] As the recipient is passive and anonymous it is not possible for the attacker to find him. The sender instead can be located by triangulation or similar means.

1.4 PLATFORM

In the context of the lab, we intend to implement and evaluate the approaches on the Wireless Open-Access Research Platform (WARP). This is a scalable and extensible programmable wireless platform. It is especially intended to prototype advanced wireless networks on a high-performance programmable hardware and even provides open-source reference designs and support material.

1.4.1 *Hardware*

We will use the latest revision of the WARP Field Programmable Gate Array (FPGA) Board (v3), as shown in figure 1.2. It provides the following components especially relevant to for the lab:

2 In most cases overhearing and prevention is mutually exclusive as prevention includes destroying the data, even for the attacker himself.

3 This is out of scope for this work, as access control in wireless networks brings it very own issues.

- The FPGA under the fan is the central processing system and can be programmed using hardware description languages (low-level but high-performance) or C on virtual CPU (slower but easier to handle).
- Two radio interfaces including the required converters and transceivers to be used at 2,4 and 5 GHz which fits quite well as we are working on 802.11 Wireless Local Area Network (WLAN).
- 2 GB SDRAM to store runtime data, e.g. logs.
- Ethernet ports to allow runtime configuration and management.
- The JTAG interface/SD card can be used to program the FPGA.

So, WARP provides all necessary means to host a 802.11 stack and thus fits the needs of the lab project quite well.

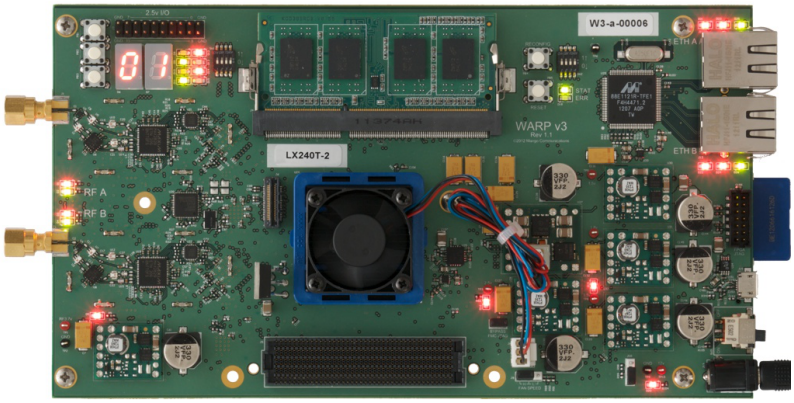


Figure 1.2: WARP FPGA Board v3 [source: http://warpproject.org]

1.4.2 *Designs*

As developing as implementing a custom IEEE 802.11 stack is a challenge even for professionals and thus out of scope of this lab, we rely on reference implementations provided by the WARP project. These *reference designs* allow us to focus on the challenges presented by the covert channels themselves and have to care less about implementation details.

1.4.2.1 *802.11 reference design*

The 802.11 reference design provides a real-time implementation of IEEE 802.11 Orthogonal Frequency-Division Multiplexing (OFDM) PHY and Distributed Coordination Function (DCF) MAC. It can communicate with commercial 802.11 devices and act as as AP and Station (STA) in the network. As it's open-source we can modify any layer if needed to implement a covert channel and directly test these modifications, even with real 802.11 devices.[4]

4 See http://warpproject.org/trac/wiki/802.11

Figure 1.3 provides an overview over the 802.11 reference design's architecture. The PHY layer directly builds on top of WARPs platform support cores which control the peripherals on the hardware board and is implemented using hardware descriptions generated by several Simulink cores. Most important this is logic implements the OFDM transmission and reception, like en-/decoding and (de-)modulation of PHY frames into and from the wireless spectrum. This done as close to the hardware as possible as it operates with extreme time constraints and on a huge bandwidth. These cores are used by the DCF core which handles timers and carrier sensing and acts as a interface between MAC software designs and the PHY cores. The remaining MAC layer is implemented using two MicroBlaze softcores[5] which run C code. The lower one executes more time-critical code of the DCF, like Acknowledgement (ACK)s, backoff scheduling and handling re-transmissions. The higher CPU handles non-critical high-level functions like construction of non-control packets, handling handshakes and integrating a wired network over one of the ethernet ports.

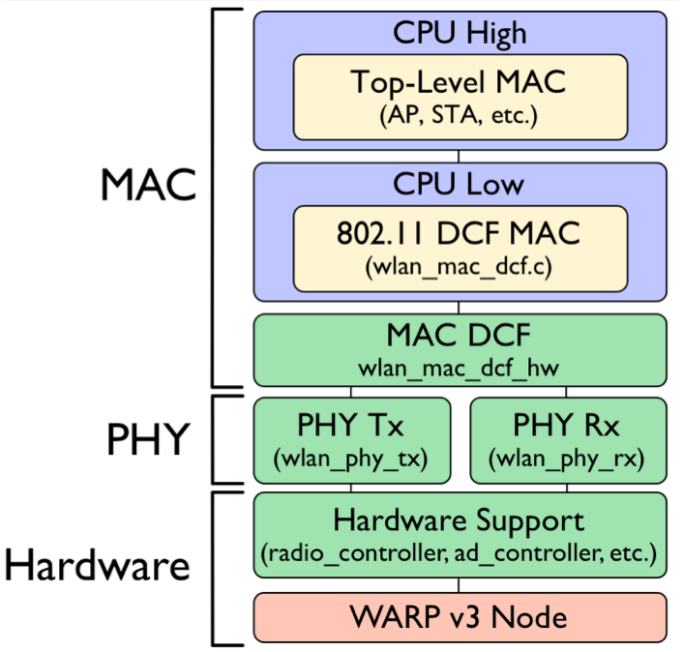


Figure 1.3: WARP project 802.11 reference design architecture overview [source: http://warpproject.org]

1.4.2.2 *WARPLab reference design*

The WARPLab reference design is a basic buffer-based implementation with no physical or MAC layer itself. Its purpose is to provide an endpoint for use by the WARPLab design flow. WARPLab is a framework for rapid prototyping of phyiscal layer implementations on arbitrary combinations of single and multi-antenna transmit and receive nodes. It uses MATLAB to control nodes and perform signal processing from a host pc, but allows to move time critical components to the FPGA of the WARP nodes involved[6]. Figure 1.4

5 Softcores execute code like real CPUs but are in most cases kept simpler as they are slower. They are used because they are allow for easier implementation of functionality than Simulink or hardware description language cores.

6 See http://warpproject.org/trac/wiki/WARPLab

provides an overview over a typical setup when developing using WARPLab and the WARPLab reference design.

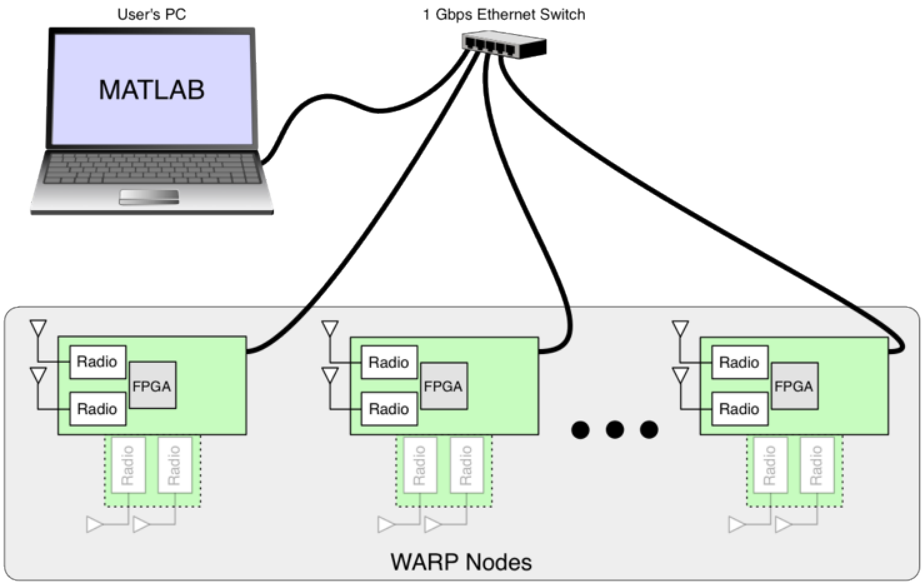


Figure 1.4: WARPLab development setup [source: http://warpproject.org]

2

BACKGROUND

2.1 IEEE PHYSICAL LAYER SPECIFICATION

The physical layer of the IEEE 802.11g is based on a OFDM system, that provides communication capabilities from 6 MBit/s up to 54 MBit/s in the 2.4 GHz frequency band. It defines a channel spacing of 20 MHz, 48 data subcarriers in conjunction with 4 pilot subcarriers and a cyclic prefix insertion for each OFDM block. A signal transmission follow a packet structure, that is depicted in Fig. 2.1. The packet structure is referred to as PPDU and is composed of a preamble (PLCP Preamble), an OFDM signal (SIGNAL) block and an OFDM data (DATA) block. Besides, the IEEE 802.11 specifies a scrambler, encoder and an interleaver for OFDM block processing.

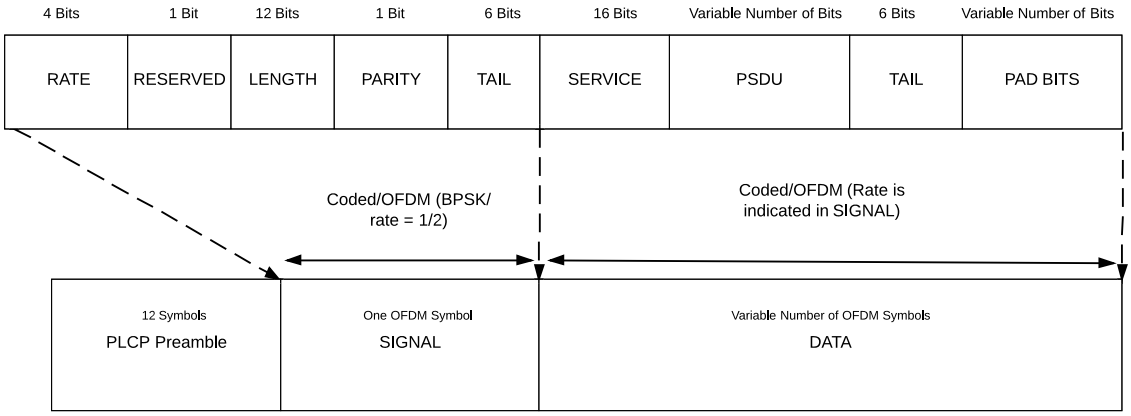


Figure 2.1: Packet structure of the PHY layer (also referred to as PPDU).

PLCP PREAMBLE The PLCP Preamble precedes every signal transmission and serves as synchronization. It consists of a short training sequence (STS) and long training sequence (LTS). The STS is used for signal detection and automatic gain control and is 8 µs in length. The LTS has the same length and serves as more accurate channel estimation.

SIGNAL The OFDM signal field contains the information to configure a receiver. It is composed of a rate and a length field and is 4 µs in length. The provided rates are given in Table 2.1. A parity field over the first 17 bits exists and a tail field is introduced to flush an encoder or decoder. The OFDM signal block is transmitted using BPSK modulation and coding rate of 1/2.

DATA The OFDM data is composed of a service field, data field (also referred to as PSDU), tail fiel and a padding field. The service field has the task to synchronize the scrambler. Similar to OFDM signal, the OFDM data is encoded and interleaved. In addition, it is scrambled to eliminate bit characteristics.

Modulation	Coding Rate	Coding Bit per sub-carrier	Coded Bit per OFDM Symbol	Data Bits per OFDM Symbol	Data Rate
BPSK	1/2	1	48	24	6
BPSK	3/4	1	48	36	9
QPSK	1/2	2	96	48	12
QPSK	3/4	2	96	72	18
16-QAM	1/2	4	192	96	24
16-QAM	3/4	4	192	144	36
64-QAM	1/2	6	288	192	48
64-QAM	3/4	6	288	216	54

Table 2.1: Modulation Parameters

2.2 IEEE LAYER 2 DATAGRAM SPECIFICATION

IEEE 802.11 MAC layer datagrams are called frames and are used to transmit data as well as management and control information for the wireless links. Figure 2.2 shows the standardized section and field structure. Each frame is a combination of a MAC header, payload data and a tailing Frame Check Sequence (FCS). The MAC header consists of several fields, some of which are needed to be explained in further detail as they are relevant for the covert channels examined later:

FRAME CONTROL The FRAME CONTROL field consists itself of various bit flags. The type fields allow to differentiate between the frame types. ToDS and FromDS indicate if the frame is sent into or received from a distributed system (in most cases a infrastructure mode AP). The *retry* flag is set if the frame is retransmitted and allows to eradicate duplicate frames.

ADDRESSES The MAC header contains four fields for addresses, although some of them are only relevant when forwarding frames. Most relevant are addresses 1 and 2 as they contain the transmitter and intended direct receiver of the frame.[1]

SEQUENCE CONTROL The SEQUENCE CONTROL field is used for frame numbering to detect duplicates and defragment transmissions.

NETWORK DATA Network Data or payload of a 802.11 frame can be up to 2304 bytes long. Some frames may not have any payload at all.

FCS The Frame Check Sequence (FCS) field is stored in the last four bytes of the frame and are the result of a Cyclic Redundancy Check (CRC) calculation over the complete frame including the header, excluding itself. It allows for integrity check of the received frame. A frame with a matching FCS is acknowledged by a corresponding control frame.

1 For further details on the address fields see [3].

There are several management frame types, most of them for association and authentication of a station to a network and thus maintenance of the communication network. Furthermore there are control frames for exchange between stations themselves, namely acknowledgement and collision avoidance frames (RTS, CTS).

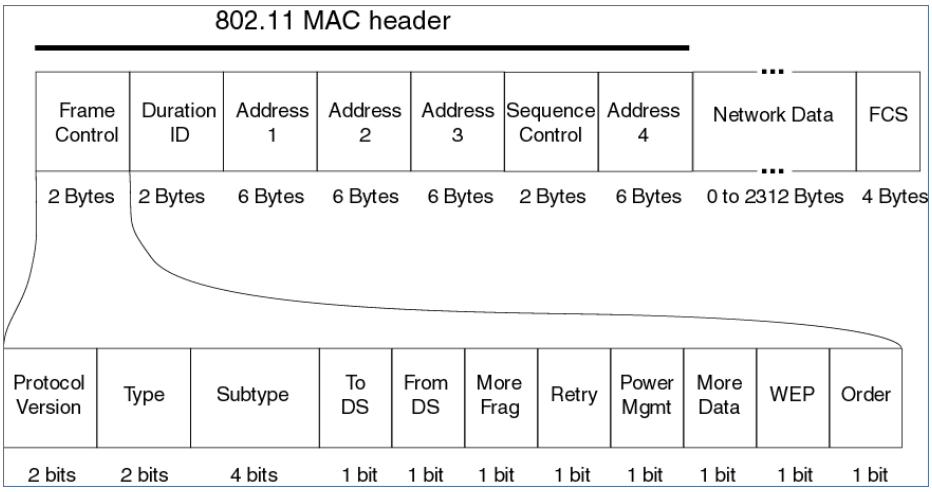


Figure 2.2: IEEE 802.11 layer 2 datagram format [source: http://wildpackets.com]

3 RELATED WORKS

As we focus on MAC and PHY layer, we have investigated different concepts of building a covert channel in both network layers. This chapter provides an overview over existing approaches.

Table 3.1 presents different methods for covert channels that either are implemented or simulated in IEEE 802.11. The following sections examine these methods in more detail and provide argumentation about the classifications and the importance for this work.

3.1 MAC LAYER

1. Wireless Padding (WiPad) [8] by Szczypiorski is a steganographic method, that is based on the insertion of hidden data into the padding fields at the physical layer of WLAN. The relevant fields for the implementation of this method are the SERVICE, Physical layer Service Data Unit (PSDU) and TAIL. The length of SERVICE and TAIL are constant, respectively 16 and 6 bits. PSDU is a MAC field and has a variable length according to the user data, ciphers and network operating mode. Since these fields are present in all frames, some frames like ACKs, that are frequently exchanged, are especially interesting for the WiPad method. The padding bits are usually set to zero, in this paper is assumed that all of them could be used for steganographic purposes. This method achieves a capacity of 1.1 Mbps for data frames and for ACK frames a capacity of 0.44 Mbps, which gives a total capacity of 1.54 Mbps. It was simulated in conditions chosen by the developers but not evaluated for a real working wireless network.

   To the best of our knowledge and according to the IEEE Specification for Wireless LAN Medium Access Control and Physical Layer [7] not all the flags used from WiPad are suitable for covert data transmission. The seven LSB of the field SERVICE shall always be set to zero to enable the estimation of the initial state of the scrambler at the receiver. This would reduce the throughput of the method. Another unclear point of this method is the detectability. The authors purpose the estimation of the detectability of the system as future work. Since the method does not influence the performance of the system a normal user does not percept the existence of the covert data. Nevertheless is this channel theoretically easy to detect, because no further obfuscation is applied. If the attacker of the covert channel knows the steganographic principle, it would not difficult for him to detect the hidden data.

2. Hidden Communication System for Corrupted Networks (HICCUPS) [9] is a steganographic system with bandwidth allocation for shared medium networks developed by Szczypiorski. HICCUPS uses three different methods to covertly transport data of which one constitutes a novel approach. It introduces the usage of frames with intentionally corrupted checksums as a way to provide on-demand bandwidth for the steganographic channel. The other two methods include the use of cipher initialization vectors and network address fields to transport data. The performance

of the corrupted frames approach of HICCUPS was done by the same author in [10]. The achieved bandwidth is quite high compared to other hidden channels. For a 54 Mbps WLAN with 10 stations the covert bandwidth is stated to be at a maximum of 1.27 Mbps for additional 5% Frame Error Rate (FER).

This approach is designed to be adaptable and allow weighing between throughput and detectability. The throughput depends entirely on the change of FER, but higher FER slow down the network and allow for easier detection. This covert channel degrades the performance of the normal channel resulting in a low detectability, since a high FER could raise suspicions in the participants of a network. The method is theoretically easy, since the attacker could receive the packets if he knows that covert data are sent in the network and accordingly changes the configuration of the WLAN card to forward the frames with a wrong checksum and not to throw them away. These values were only determined by calculation and simulated and not implemented, although this approach is referenced by multiple other works.

3. Two other approaches were proposed in [5] for constructing a steganographic channel on the MAC layer. One of the scenarios is based on a storage channel using specific header fields as transport and the second other is a covert timing channel using re-transmissions. Both scenarios were practical implemented and evaluated using consumer hardware.

   PACKET MODIFICATION Packet modification takes place in the FRAME CONTROL FIELD (FCF) and DURATION/ID fields of the 802.11 MAC header. These were chosen after conducting a real world packet analysis concerning practical occurrence. Further analysis of the FCF resulted in only the RETRY and MORE DATA fields being useful for implementing a covert channel. Others were either unsuitable for steganography or were used too seldom like PWRMGT and MORE FRAG. The actual implementation uses RETRY and MORE DATA fields for synchronization and the DURATION/ID field for transmission of the actual payload. The capacity of the approach were practically measured and resulted in 2.1 Bps.

   This steganographic method has a high detectability, since the attacker can easily detect the covert channel if he compares the normal and manipulated packets. Otherwise he can do some statistical analysis on the occurrence of selected fields in the WLAN MAC header.

   PACKET DUPLICATION The second approach uses timed re-transmissions as a channel. In this approach a steganographic message is sent not by modifying the WLAN packet itself, but by duplicating it with the RETRY flag being set. It looks like a normal sender side re-transmission since the CRC/FCS is corrected accordingly. To decide which of the packets should be duplicated and to guarantee a reliable performance, the developers of this channel investigated the behavior of the WLAN network and decided to use Beacon Frames since this packet type can be duplicated without causing side effects. This scenario is asserted to have a very high level of transparency and thus being hard to detect. An attacker can use statistics of WLAN about occurrence of packet duplication to detect the hidden communication. Even thought something might look unusual while comparing the channel to a "normal" channel the attacker either

can identify which of the duplicated packets is associated to which meaning, nor can identify the channel used for communication. This is avoided by using one communication matrix K and one key matrix P. Without knowing this values the covert data cannot be discovered. Because of synchronization problems this approach was not implemented, but only simulated. The authors assume that the RETRY bit is set in such a way that the collision detection is avoided. Since this method is not implemented on a real working network but only simulated it may be assumed that the some kind of performance degradation may be introduced to the network. The simulation resulted in an estimation of about 0.2 Bps of capacity.

3.2 PHY LAYER

1. Szczypiorski et al. designed in [4] an approach that exploits the characteristics of OFDM communication systems. The IEEE 802.11g is based on the OFDM technique and, as such, cyclic prefix (CP) is a feature of the system. The CP is a repetition of some part of the signal and, therefore, conventionally not read or interpreted by radio receivers. The idea of the authors is to substitute CP with a secret message. An adversary or partner, respectively, extracts from the received signal the information. In order to prevent detection of the covert channel, the authors proposed a shared secret key. The secret key, that is referred to as T_{max}, is used to initialize a common Pseudo-Random Number Generator (PRNG). Therewith, an equal random sequence is produced to sign modified CPs. The detection is significantly complicated but the initial key exchange (of T_{max}) remains a problem. That could be done by means of cryptography e.g. based on DIFFIE HELLMAN. However, the exchange has to be done before the covert channel is ready to use – either setting $T_{max} = 0$ in the beginning or using a layer different from PHY. The approach is evaluated with a simulation in Simulink. The simulation reveals a high performance with 3.25 Mbps up to 19.5 Mbps but signal noise and channel response were neglected. This is the highest capacity in steganographic systems we encountered during our review of related work. However, Szczypiorski et al. do not detail into the propagation of CP modification. Typically, changing CP comes along with losing the properties of orthogonality. Besides, the substitution of the CP is not precisely declared. If it can be proved that inserting covert data in the CP do not influence the orthogonality of the sub-carriers, this method would have an advantage according performance degradation of the "normal" channel, since it will not be influenced. This approach introduces a theoretically hard method for implementing a covert channel.

2. Dutta et al. present in [2] a novel approach to build a covert channel, that is based on modification of signal modulation. The proposed technique is called Dirty Constellation and mimics noise commonly imposed by hardware imperfections and channel condition. The approach leverages the variability in wireless channels to set up a covert channel. For that reason, the encoding of signal in Binary Phase Shift Keying (BPSK) or Quadrature Phase Shift Keying (QPSK) are supplemented with additional information. So each symbol is decoded as an additional QPSK within the tolerance. Since the joint constellation is not aware to the ordinary receiver due to conventional noise in radio, a nearby partner is able to decode the hidden message.

To further diminish the detection of the covert channel by vector-signal analyzers, the authors introduce a rotation feature. The constellation points around a symbol are rotated monotonically and clockwise. Dutta et al. implemented the idea on a SDR platform and performed a rate of 9 Mbps with 3/4 encoding rate for the covert channel. Although higher modulation schemes would lead to higher throughput, the authors did not consider these configurations. The joint constellation of higher modulation such as 16-QAM and 64-QAM demand more precise signal mapping. We assume that the tolerance space of higher modulation is to small to achieve a reliable determination of radio noise and modulated signals. This method does not influence the performance of the "normal" channel.

Table 3.1: Comparison of Covert Channels Models

Steganographic System	Capacity	Degradation	Detection	Evaluation
WiPad [8]	1.54 Mbps	no	easy	simulated
HICCUPS[9]	1.27 Mbps	yes	easy	simulated
Packet modification [5]	2.1 Bps	no	easy	implemented
Packet duplication [5]	0.2 Bps	yes	hard	simulated
Cyclic Prefix [4]	19.5 Mbps	yes	difficult	simulated
Dirty Constellation [2]	9 Mbps	yes	average	implemented

3.3 MODEL COMPARISON

From Table 3.1 it follows that the steganographic systems implemented in IEEE 802.11 have different capabilities for the covert channel. The results are taken from the papers and present the statement from the authors. Since most models are based on simulation in Matlab, it is highly probably that the results are not conform to practical implementations. The assessment of the models brought out that a performance degradation from most models on the normal channel occurs. For instance, the Dirty Constellation can run the covert channel only with the modulation order QPSK. As an another example, HICCUPS has a direct impact on the normal channel due to corruption of the CRC frame. The detection level is assessed according the effort of an attacker to identify modifications on the normal channel, to detect the covert channel as well as to read hidden information. As an example, the Header Modification model has actual no impact on the normal channel but an attacker needs only a common-off-the-shelf laptop to identify modifications on the protocol. In contrast to this, Cyclic Prefix requires special equipment to be detected.

4 MEASUREMENT

This chapter describes the measurement of the performance of covert channels and the host channels. First the requirements are discussed and defined. Then, several possible approaches are examined and assessed, the implementation is designed and explained in detail.

4.1 REQUIREMENTS

As the goal of this work is to implement several convert channels and evaluate them in a real-world scenario instead of simulating them, requirements and metrics have to be chosen to really conduct the assessment and compare the different approaches against each other.

4.1.1 *Channels*

In this particular case the most metrics, which will be chosen have to be measured for multiple entities. A covert channel always needs a host channel in which the it hides its data. Both have to be measured to not only make a statement of the covert channel itself, but also about its impact on the original channel, as it is obviously modifying it. This has to be done by measuring a) the non-augmented host channel as well as b) the host channel carrying the additional covert channel using identical setups.

As we are trying to evaluate in scenarios which are based on real-world environments, other clients – which are not manipulated by the participants of the covert channel – may also use the network containing the host and covert channel. These may also be affected by adding a covert channel to the host stream. This may be the case if for example the covert channel does not strictly comply to the IEEE 802.11g WLAN standard and thus may trigger hardware or software stack incompatibilities on other clients. As for the host channel client performance has to be measured both in the non-augmented as well as in the covert-channel-augmented case, too.

All these tests should be performed several times or over a long time span, as short-time effects of the environment may be affect the channels. It is even easy to think of (covert) channels adapting to the current environment and thus may perform different over time.

4.1.2 *Metrics*

The first obvious metric for a network data channels is **throughput**, or how much bits a channel can carry in a given time span. This is quite easy to measure on the receiving party by just counting all relevant incoming data.

Another easy metric is **latency**, being defined as the time it takes for a specific amount of data from being transmitted by the sender to reception by the receiver. Measurement requires either very exact synchronized clocks and comparing the timestamp of transmis-

sion/reception against each other, or the calculation of half the round trip time, which is likely to be less accurate, as processing on the receiver's side adds to the value.

Our approaches are not solely based on the MAC layer and above and thus may also effect the low level transmission and thus the reliability of the channel. So, metrics for measuring the low level PHY channel quality have to be added. The common choice for such a metric is **Bit Error Rate (BER)**. It is defined as the ratio of the count of unmatched bits between the transmitter and receiver against the overall count of bits transmitted. As it is obviously not possible to determine this by just the received data, the transmission of the sender has to be taken into account, too. BER can then be determined by comparing the sender's output versus the receiver's input data stream bit by bit and the total count of bits sent.

IEEE 802.11g includes a FCS at the MAC layer which allows the receiver to detect some bit errors and thus invalidate the frames if it cannot be recovered. The count of such invalidated frames, causing retransmissions, versus the total frames transmitted can also be taken into account as a metric. This is called **FER** and can be computed in several ways: Based on the data also required for BER, one can compute FER based on the number of frames received with a good FCS versus frames sent at all (it is possible that some frames weren't received at all instead of being damaged). Another way is based on the following idea: As correctly received WLAN frames have to be acknowledged, comparing the number of sent frames versus the number of received ACK frames allows for an additional way to calculate FER. This may be easier and even allows the sender to adapt its channel based on FER if possible and/or needed.

Another important aspect is the assessment of the **detectability** of the covert channels. Covert channels are specifically made to carry data undetected (or else one could simply use the host channel itself), thus they should try to be as hard to detect as possible as an adversary. Assessment of this may be hard, and involve statistical models. Adding it to the metrics may be decided later. Also, resilience against an adversary trying to interrupt the covert channel may also be aspect worth taking into account.

4.2 APPROACHES

To measure the mentioned metrics on the WARP platform, several possible approaches can be taken. The first is rather high level and thus cannot measure all mentioned metrics, the second is more low level and works by using data provided by the 802.11 reference design itself.

4.2.1 *Custom host based solution*

This approach is a naive black box approach not concerning the internals of the WARP platform. It is based on external hosts being connected to the WARPs actually measuring the metrics themselves. Software like iperf allows easy measurement of throughput and packet loss between the hosts, but is not adapted to the specific needs of the WARP platform.

All the other metrics an obviously not measured, as a) bit errors already are corrected and b) invalid frames already has been dropped and re-transmission handled locally by the WARPs.

4.2.2 *802.11 reference design experiments framework*

The 802.11 reference design provides an elaborate experiments framework which allows remote controlling the behavior WARP nodes running the reference design and provides low-level visibility of both layers. This includes providing very detailed logs about various events and traffic generation. This functionality can later be used to test and validate implementations of covert channels using this reference design. It also contains an easy to use python3-based client library to provide a stable interface for client software to control the WARPs[1] and the framework parts running on the WARP and process log data.

Logs include, but are not limited to, even the low level transmission and reception events including every data bit, FCS and meta data like Received Signal Strength (RSS). WARPs include about a gigabyte of memory to store those data, for later retrieval and analysis. This allows us, using two WARP systems together, to easily calculate BER, FER and throughput based on the logs of both systems. The mentioned remote control also allows very exact synchronization of clocks (WARP platform even allows hardware synced clocks) and thus enabled us to measure very exact latency.

4.2.3 *WARPLab experiments*

Measuring metrics when using WARPLab and the corresponding reference design can not be standardized, as their calculation depends on the implemented scheme and its specifics.

4.3 EXPERIMENT IMPLEMENTATION

To simplify regular measurement and validation of our implementations we implemented additional python software using the provided bindings for 802.11 reference design experiments framework and even had to modify the framework itself. This resulted in an easy to use toolchain that can – after starting a 802.11 reference design-based implementation two WARP nodes – compute the specified metrics in a matter of seconds.

Modification of the framework was needed to add an additional log entry type for use by covert channel data events. This allows us to later differentiate between the host and the covert channel.

There are two main parts:

- Execute a measurement run. This involves several scripts which each employ different run profiles, depending on the current needs. Each selects looks for WARPs it needs on the network and then sets its relevant settings like transmission bit rate, enables logging of relevant information[2] and then starts the traffic generator. After a certain period of time – the measurement time span – traffic is again disabled and the logs are fetched from the WARPs and written to log files.
- Analysis is performed by another script. It is build quite modular and allows the calculation of the metrics on certain log event types. This enables us to calculate this metrics for different channels, like the hosting channel and the covert channel

1 Communication is based on the WARPnet commands.

2 The 802.11 reference design by default does not log faulty frames and only records the MAC header and not the payload.

and enable/disable metrics as needed. To quickly process the log data files (the file size averaged by round about 200 MiB for each node for each measurement run), the bindings provide helpers which use the numpy library to fasten the number crunching process.

5 COVERT CHANNELS

In this chapter we will discuss concepts of hidden channel that are considered realizable and relevant to be practical evaluated. The presented approaches in section 3 stated different solutions in both PHY and MAC layer. From these solutions we selected two concepts as potential candidates for implementation on the WARP platform. In addition, we redesigned an existing concept for the covert channel purpose and also extended the one approach in order to eliminate critical points of the design. In the following section these concepts are described elaborately. Moreover, critical parts are discussed in more detail as well as optional features are proposed. Each approach is considered according to the expected workload and necessary resources for implementation.

5.1 CODED CYLIC PREFIX

In 3.2 we summarized the proposal of Szczypiorski et al. [4] that focussed on exploiting CP in the OFDM procedure. We think that the authors do not declare in detail the impacts of replacing the entire CP. On the one hand, CP maintains the necessary orthogonality of OFDM subcarriers to mitigate Inter-Carrier Interference (ICI) from adjacent subcarriers that occurs due to multipath propagation [6]. As shown in Fig. 5.1, transmitted radio signals are affected by reflections and the overall received signal is the sum of all appearing radio signals. Moreover, CP serves as a guard interval between two successive OFDM symbols. The guard interval is typically used to prevent Inter-Symbol Interference (ISI) that occurs due to fading of various reflected radio signals from the previous OFDM symbol.

On the other hand, CP alleviates for the receiver the location of the starting point of an OFDM symbol. Since the CP repeats the ending part of the OFDM symbol, the time window for extracting the OFDM block can be shifted ahead. Consequently, the synchronization complexity is reduced. Apart from that, the authors did not state precisely the substitution of the CP. Because of these missing clarifications, it is necessary to reassess the approach.

Therefore, we aim to design and implement a more robust approach that replaces partially the cyclic prefix. Our goal is to build a steganography system that puts the hidden message at beginning of the cyclic prefix, while the ending part maintains several samples of the original CP. The design criteria are based on assessment on the OFDM system and are discussed in more detail in the following section.

5.1.1 *Design Challenges*

The message insertion inside the CP can be done in several ways but it is necessary for an efficient and robust design to leverage the features of the IEEE 802.11g communication system. As we have learned, the IEEE 802.11g specifies at the beginning of each transmission a *OFDM preamble* with 16 µs in length, that is composed of a short training sequence (STS) for signal detection and a long training sequnce (LTS) for channel estimation and fine synchronization. The *OFDM preamble* is followed by the *OFDM signal* symbol, that consists of the rate and length and is transmitted with the most robust combination of

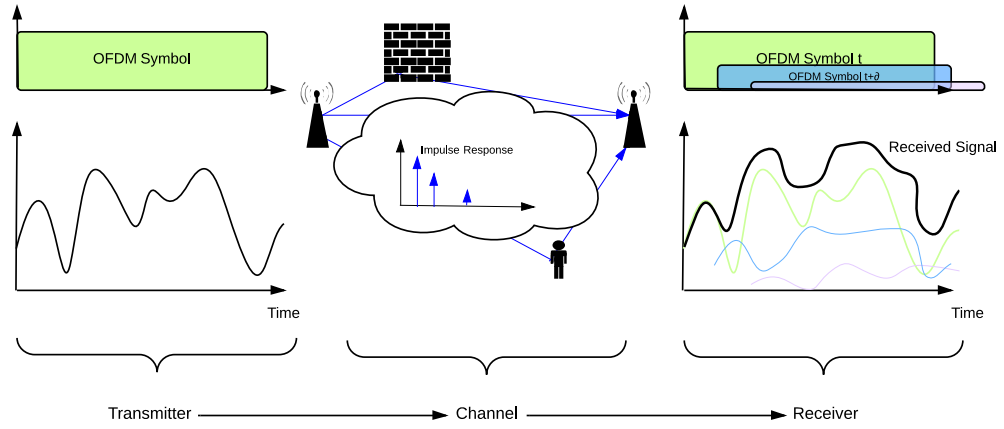


Figure 5.1: Multipath signal transmission of one OFDM symbol with overlapping at receiver. [Own source]

modulation and coding rate (BPSK and 1/2). It is worth to note that a receiver will ignore the data signal if the *OFDM signal* symbol cannot be parsed. The remainder part of the overall transmitted signal is utilized by *OFDM data* symbols, that are transmitted with the rate defined in the *OFDM signal* symbol. One period of an OFDM symbol is 4 µs in length and is composed of 80 samples. The CP has a duration of 0.8 µs and allocates 16 samples of that. The composition of all OFDM fields is shown in Fig. 5.2.

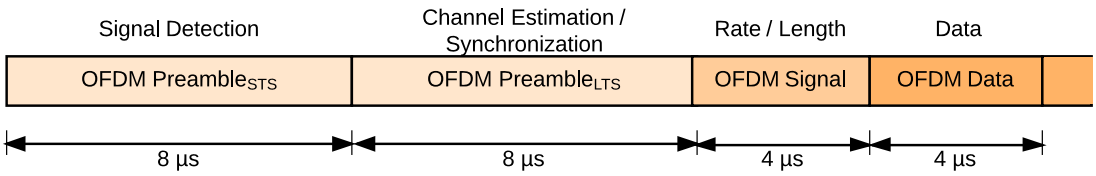


Figure 5.2: Structure of the OFDM signal. [Own source]

The 0.8 µs period of the CP implies that one hidden OFDM symbol might allocate the space. Provided that, we can determine the number of subcarriers with the following equation:

$$N_{Subcarrier} = t_{Interval} \cdot f_{ChannelSpacing} = 0.8\mu s \cdot 20MHz = 16$$

So, one OFDM block consists of 16 samples according to 20 MHz channel spacing. Given that, the subcarrier frequency spacing is calculated with the following equation:

$$\Delta_{Subcarrier} = f_{channelSpacing}/N_{Subcarrier} = 20MHz/16 = 1.25MHz$$

As a result, one symbol on a subcarrier has a frequency spacing of 1.25 MHz. This is more than four times higher than the subcarrier frequency spacing of the IEEE 802.11 specification with 0.3125 MHz, that is also shown in Fig. 5.11.

We implemented at beginning of the lab this approach with 8 symbols and an intervall length of 0.4 µs and set the remainder part of the CP to null. We tested this model of

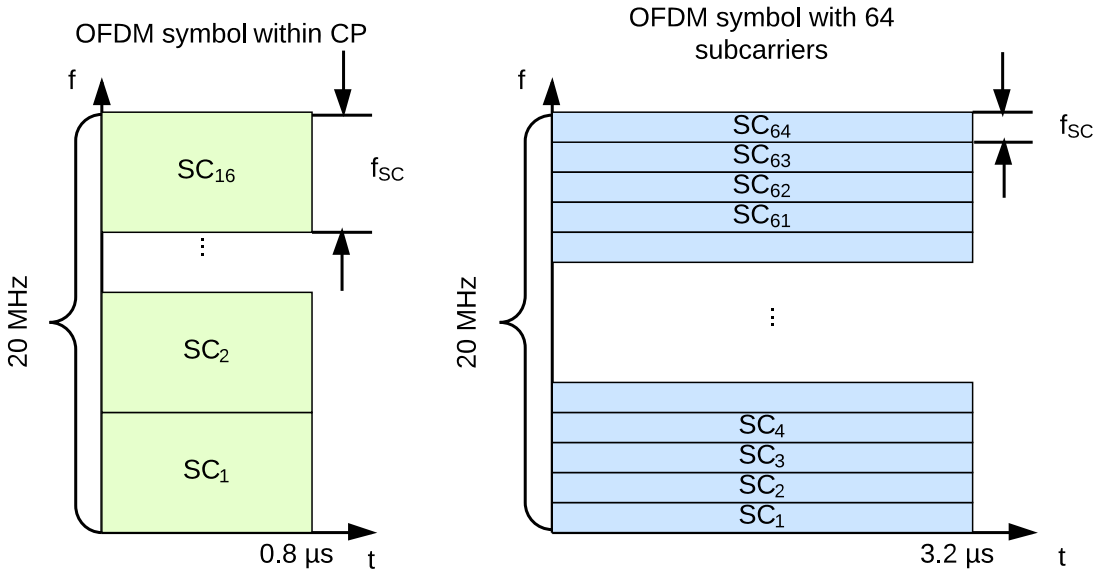
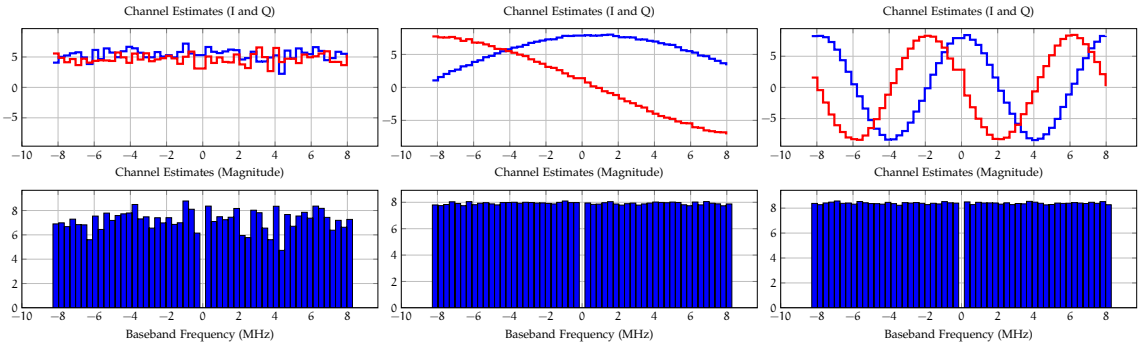


Figure 5.3: Subcarrier allocation within CP space compared to normal OFDM block. [Own source]

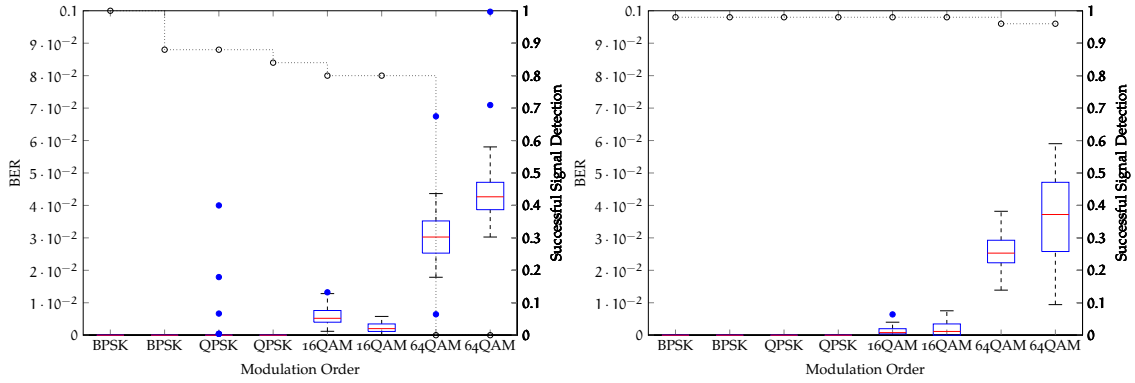
covert channel in our test environment with different modulation orders but we were most often not able to decode the transmitted signal. Upon extensive examination, we assessed the channel estimation procedure as the cause of bad signal detection. Channel estimation is a general approach in wireless communication systems to determine the impact of interferences on the signal in the used medium. The IEEE 802.11 uses the LTS over 52 subcarriers to detect the timing and frequency offset. In order to estimate the covert signal, we used the mean of the corresponding subcarrier frequencies from the overall channel estimation to the subcarriers of the covert channel. The results of several channel estimations are depicted in Fig. 5.4. It follows from the second and third channel estimation that a great variety between adjacent subcarriers exists. The mean over the corresponding subcarrier frequencies causes a wrong estimation and result in wrong symbol detection in the covert channel. Alternatively, we used the median but it did not improve the quality. Hence, this approach is unsuited for building a covert channel.



(a) Estimation of 1st transmission. (b) Estimation of 2nd transmission. (c) Estimation of 3rd transmission.

Figure 5.4: Channel estimation of various signal propagation.

We investigated the impact of completely replacing the cyclic prefix with hidden data. For that we used various scenarios with different modulation orders and two different receivers. Our goal was to declare the necessary offset from the cyclic prefix, that has to be kept in order to decode a signal. Moreover we aim to study the synchronization capability of common-off-the-shelf receiver by sending continously beacon frames on a channel with several APs. Therefore we set the cyclic prefix to null. The scenarios start with ascending modulation orders and descending offset and are shown in Fig. 5.5 and Table 5.1. The results of importance are successful transmission and detection and the BER (Bit Error Rate).



(a) Completely removed CP. $\mathrm{Offset}_{\mathrm{CP}}$ 0.

(b) First twelve samples of CP removed. $\mathrm{Offset}_{\mathrm{CP}}$ 4.

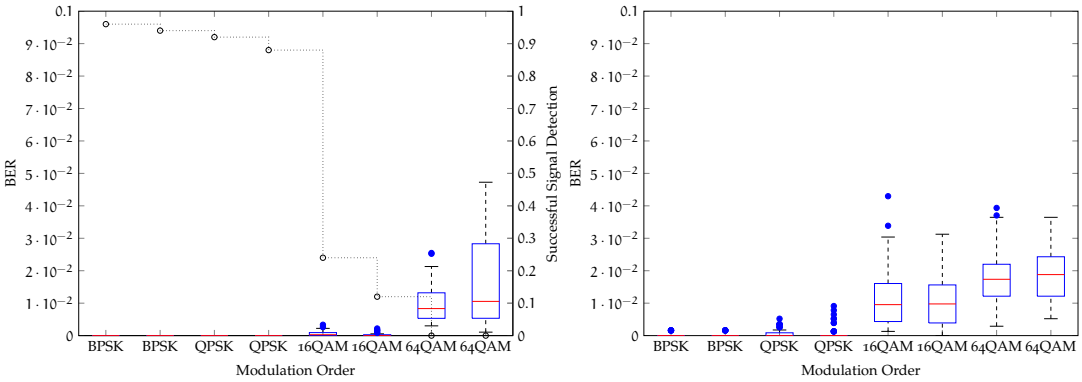
Figure 5.5: Signal detection and BER with completely and partially removed cyclic prefix. Values of BER are presented in normalized form. Dotted lines (along circles) represent the probality of successful detected signal transmission. Filled circles demonstrates outliners of the measurement.

Table 5.1: BER of signal transmission according to Fig. 5.5. [BER] = Bits.

$\mathrm{Offset}_{\mathrm{CP}}$ 0	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	5132	3504	5136	3504	5232	3696	4080	3504
σ	0.0	0.0	14.863	8.563	7.600	2.312	26.626	22.198
Variance	0.0	0.0	22.912	6.626	57.766	5.346	708.991	492.751
Mean	0.0	0.0	3.160	2.961	14.780	3.800	65.220	90.060
$\mathrm{Offset}_{\mathrm{CP}}$ 4	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
σ	0.0	0.0	0.0	0.0	3.363	3.393	10.612	26.774
Variance	0.0	0.0	0.0	0.0	11.314	11.516	112.615	716.852
Mean	0.0	0.0	0.0	0.0	3.540	3.440	52.420	74.380

The results of the Fig. 5.5 implies that a common-off-the-shelf receiver encounters difficulties in detecting transmitted signals if the CP is completely removed. In particular, the receiver cannot decode the signal if the transmission is encoded with higher rates such as 64-QAM. It is obviously that this effect has an enormous impact on the probability of detection. It is highly probable that the reason for this effect is caused by synchronization

issues. This becomes obvious if the results are compared with the second scenario, where the CP were partially removed and the ending part with 4 samples was maintained. Here, we declared a successful signal decoding of 95 %, which is high enough to ensure a realiable processing and low detectability. It is interesting to point out that the BER is slightly reduced and the outliners could nearly prevented. The Table 5.1 highlights the improvement between both scenarios.



(a) Normal channel.

(b) Covert channel.

Figure 5.6: Diagram shows the influence of a covert channel with 16 samples in length on the normal channel. Results are presented in normalized BER and successful signal detections (in dotted lines).

Table 5.2: BER and EVM of normal channel according to Fig. 5.6. [BER] = Bits, [EVM] = %

Ch_{Normal}	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	5132	3504	5136	3504	5232	3696	4080	3504
BER_{σ}	0.0	0.0	0.0	0.0	4.392	1.749	24.395	53.108
BER_{Var}	0.0	0.0	0.0	0.0	19.290	3.059	595.113	2820.539
BER_{Mean}	0.0	0.0	0.0	0.0	3.340	0.960	41.780	62.460
EVM_{σ}	2.231	2.231	2.196	1.441	2.215	1.905	1.424	2.868
EVM_{Var}	4.980	4.980	4.825	2.076	4.908	3.630	2.027	8.228
EVM_{Mean}	10.461	10.461	11.249	10.462	11.038	10.468	9.380	10.870

Since the importance of the CP is determined, another critical point from the designers point of view is the impact from substituted CP on the normal channel. Therefore, we constructed a scenario in which the CP is completely replaced and measured the BER of the normal channel as well as the covert channel. The results of the measurement is depicted in Fig. 5.6 as well as Table 5.2 and 5.3. As a consequence of Fig. 5.6, the signal detection of the receiver is heavily interfered if the signal is transmitted in the higher rate modes. This follows on the one hand from the high BER and on the other hand from the low signal detection of the receiver. This effect contradicts the statement of Szczypiorski et al. [4] and strengthen our assumption, that ICI lead to degradation of the signal. For reason of detectability, this model of covert channel is not suited in practical implementations.

Table 5.3: BER and EVM of covert channel according to Fig. 5.6. [BER] = Bits, [EVM] = %.

Ch_{Normal}	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	5132	3504	5136	3504	5232	3696	4080	3504
BER_{σ}	0.328	0.328	1.482	1.675	21.299	12.530	14.825	13.990
BER_{Var}	0.107	0.107	2.197	2.808	453.669	157.020	219.806	195.724
BER_{Mean}	0.120	0.120	0.920	0.740	27.380	16.600	31.500	32.480
EVM_{σ}	6.089	6.089	5.276	6.179	7.683	6.642	7.379	6.885
EVM_{Var}	37.081	37.081	27.836	38.188	59.042	44.117	54.454	47.412
EVM_{Mean}	27.311	27.311	27.189	27.352	27.878	26.413	24.897	25.139

5.1.2 *Concept*

For the above mentioned reasons, we decided to design an novel approach. The idea is to generate a covert OFDM block with the same size of a normal OFDM procedure, and to split the entire OFDM block in equal smaller signal blocks at the transmitter. Each signal block has 8 samples in length and replaces the beginning of the CP, such that the ending part of the CP is kept. The complete substitution of the covert channel has a length of 12 samples, that consists of 2 samples from the previous signal block, 8 samples of the actual signal block, and 2 samples of the next signal block. The structure of the covert channel is depicted in Fig. 5.7. Once the receiver has received all parts of an entire OFDM block, the parts are composed and processed analogously to normal OFDM blocks.

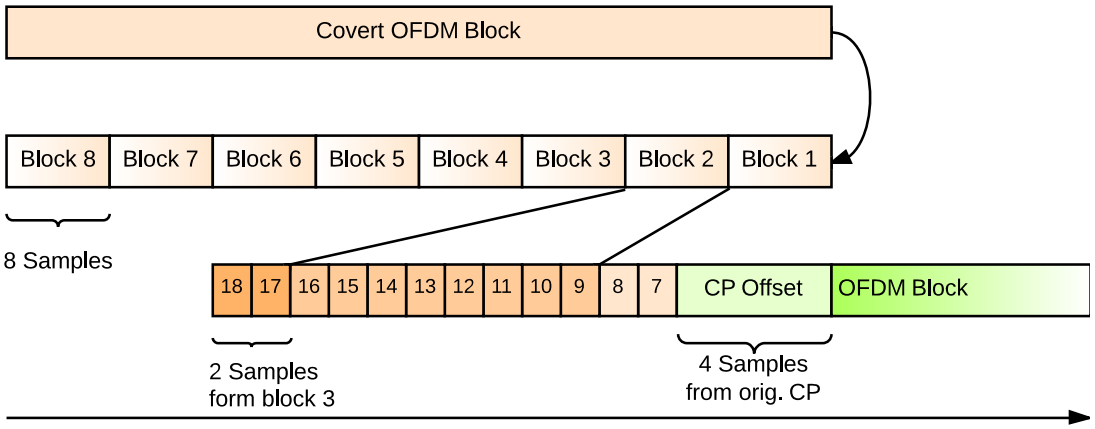


Figure 5.7: Coded Cylic Prefix: Structure of the covert channel. [Own source]

The characteristics of one covert OFDM block are similar to the normal OFDM block, such that it is composed of 48 data subcarriers and 4 pilot subcarriers. Thus, we can apply the channel estimation from the LTS and also correct the phase error by means of pilots. To enhance the mitigation of ICI and ISI as well as synchronization complexity, the last 4 samples from the actual CP are maintained. This provides a low detectability, since the ordinary user does not perceive any performance degradation. We did not utilize the entire 12 samples from the cyclic prefix because of receiver complexity. Instead, we

extended both block sides with samples from the previous and the next block to improve the signal quality. We found out that the wave signal between adjacent signal blocks shows a significant difference. It can be supposed that the processing of samples on the hardware is affected by subsequent samples. Since we are generating and processing similar to normal OFDM blocks, we can utilize for higher robustness the same procedure such as encoding and interleaving without to develop new system components. In so doing, we can determine the achieveable rate of the covert channel by means of the following equation:

$$R = N_{Subcarrier} \cdot N_{BlocksPerSec} \cdot N_{BitsPerSubcarrier} \cdot N_{CodingRate}$$

$$R = 48 \cdot \frac{1}{T_{Symbol}} \cdot \frac{1}{8} \cdot 6 \cdot 1 = 9\,Mbit/sec$$

The covert channel starts with the first OFDM block of the PSDU. In doing so, we ensure that the OFDM signal is not corrupted in any form from the covert channel. Once all parts of one OFDM block are received, the samples are extended cyclic at the receiver, as it is done typically from the transmitter. Thus, we are able to utilize the channel estimation, even though the channel estimation is shifted in time domain (shift in time domain results in phase shift). Pilots in the covert OFDM block allows the correction of phase similar to the normal OFDM block. The evaluation of this model is done in chapter 6.1.

5.1.3 *Requirements and Implementation*

As we aim to build and test a covert channel, we have to analyze the WARP platform. The covert channel model is implemented upon the WARPLab Reference Design, which is provided with several implementation of components in Matlab. More precisely, the WARPLab Reference Design has basic components of an OFDM communication system and is designed for offline generation and processing of samples in Matlab. The framework of WARPLab comes along with a random bit generator, a symbol mapper, IFFT/FFT component, a block for channel estimation and CFO correction and a phase correction block. The structure is shown in Fig. 5.8. Moreover, it uses a predefined PLCP preamble and a procedure for processing the data according an OFDM communication system with 64 subcarriers. Since the data is generated and processed offline, WARPLab uses a buffer on the WARP hardware and triggers all attached WARP boards for processing a signal transmission between two boards. Therefore, it is actually not designed for establishing bidirectional communication. Nevertheless, the processing of samples are done similar to a wireless communication system such as the IEEE 802.11.

Therefore, our goal was to implement the missing components of an IEEE 802.11g platform. The IEEE 802.11 specification states several units blocks, that were introduced in section 2.1. Furthermore, it is necessary to create a MAC frame that is conform to the specification. From the MAC frame, we create the OFDM signal with the corresponding length and define a rate for payload. The MAC frame needs to be scrambled according the specification, which has demanded a scrambler. The specification intends for adding redundancy an encoder and an interleaver, that were not provided by the WARPLab. Since the modulation block of WARPLab were only designed for modulation orders of BPSK, QPSK and 16QAM, we have added a 64QAM mapper. According to the specification, we define the PLCP preamble and pilots. From the receivers point of view, we just match

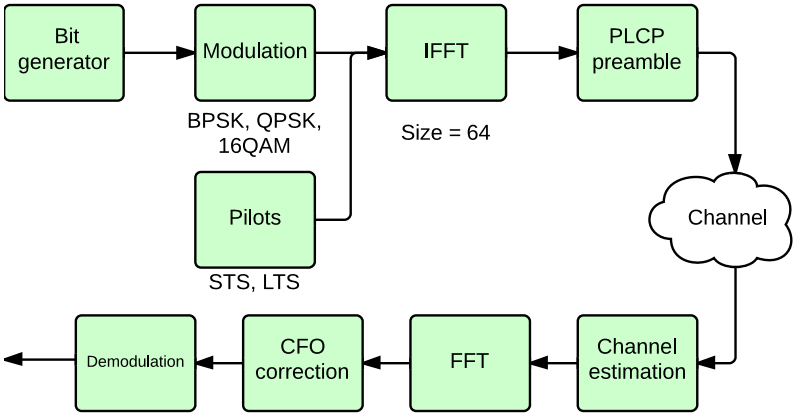


Figure 5.8: Block diagram of WARPLab. [Own source]

the demodulation block as the channel estimation and CFO correction conform the specification. The block diagram in Fig. 5.9 illustrates the structure of the implemented transmitter in WARPLab.

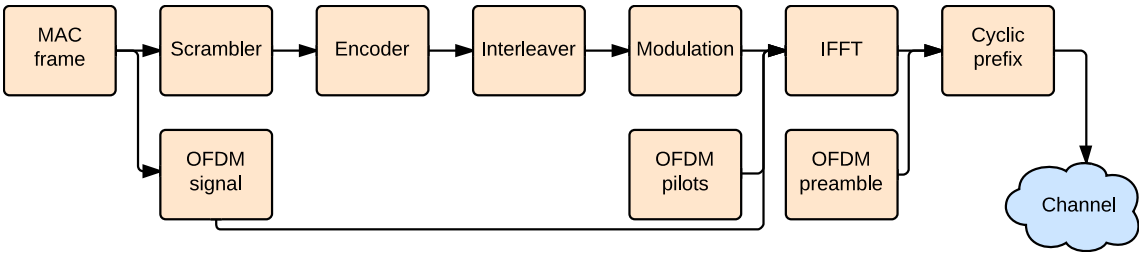


Figure 5.9: Block diagram of IEEE 802.11g transmitter. [Own source]

As shown in Fig. 5.10, the implementation of the covert channel starts at the modulation block because the number of transmitting OFDM blocks must be known. The reason for that is that covert OFDM blocks are splitted and equally allocated to the CP of normal OFDM blocks. As a result, the number of OFDM blocks must be at least a divisible by 8. As we intend to transmit the covert signal in different rate modes, we use an own modulation block as well an own IFFT block. The parser block has the task to create the samples for the covert channel and to replace corresponding to the CP of the normal OFDM block.

5.2 CODED SUBCARRIERS

In this section we present the potential of integrating a covert channel within an OFDM waveform. In OFDM a high rate data stream is split up into a set of low rate substreams, each of which is modulated on a separate subchannel. Most of OFDM standards have unused subchannels for channel spacing, synchronization of transmitter and receiver and to mitigate poor channel response. Here we aim to study the effect of inserting a narrow band signal that will be used for covert communication in the unused subcarrier locations. We further give a description of the method we aim to use for realizing it.

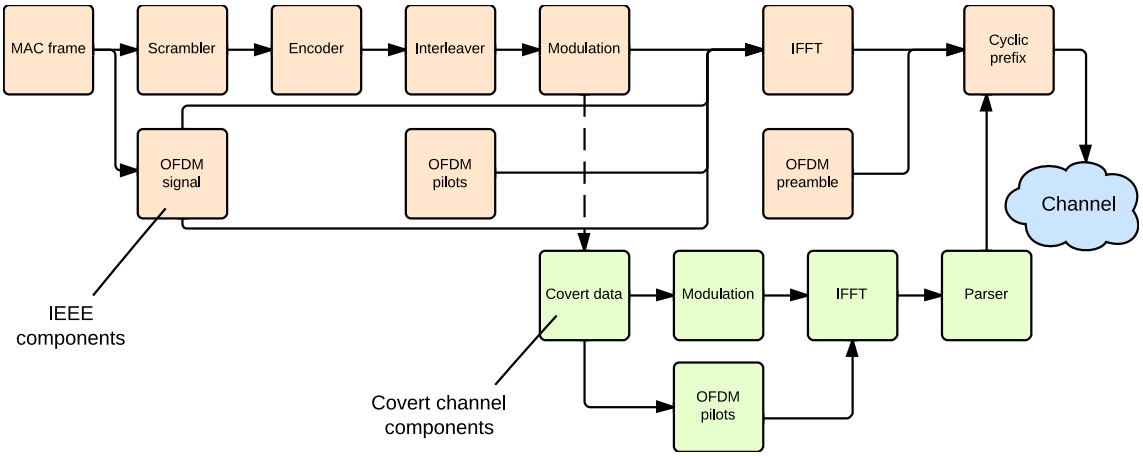


Figure 5.10: Block diagram of the Coded Cyclic Prefix transmitter. [Own source]

5.2.1 *Design Challenges*

The 802.11g physical layer builds upon OFDM structure. The channel is 20MHz and a N=64 point Fast Fourier Transformation (FFT)/Inverse Fast Fourier Transformation (IFFT) is used. From the 64 subcarriers in 802.11g, there are 52 (4 pilots) populated subcarriers and the rest are null subcarriers. The lowest six subcarriers, the DC subcarrier and the highest five subcarriers are not used [6]. The reason for the unused subcarriers are the lowpass filters required for the analog to digital and digital to analog conversion of the transmitted and received signals. Because of the lowpass filter not all the N subcarriers can be used for data transmission 5.11. The subcarriers located close to the Nyquist frequency f_s/2 will be attenuated by these filters and cannot be used.

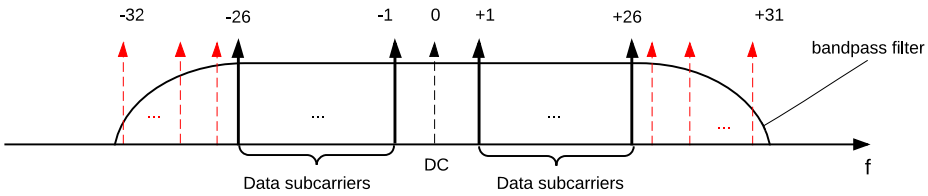


Figure 5.11: OFDM subcarriers under the impact of the transfer function of the transmitter/receiver

The first design decision to be taken, was to decide how many subcarriers to substitute. The available number is 11, and the first thought was to use two pilots in the covert data. This would reduce the throughput of the covert channel, since only 9 subcarriers could be used for hidden data transmission. The first implementations of the Coded Subcarriers channel included two pilots in the covert data. The pilots had predefined values and after they were received the per-symbol phase error was calculated. This phase error was then

applied to the covert data. Since no considerably improvement could be achieved, we decided to not include extra pilots in the covert data. The per-symbol phase error could be done in this case by using the pilots of the normal OFDM symbol.

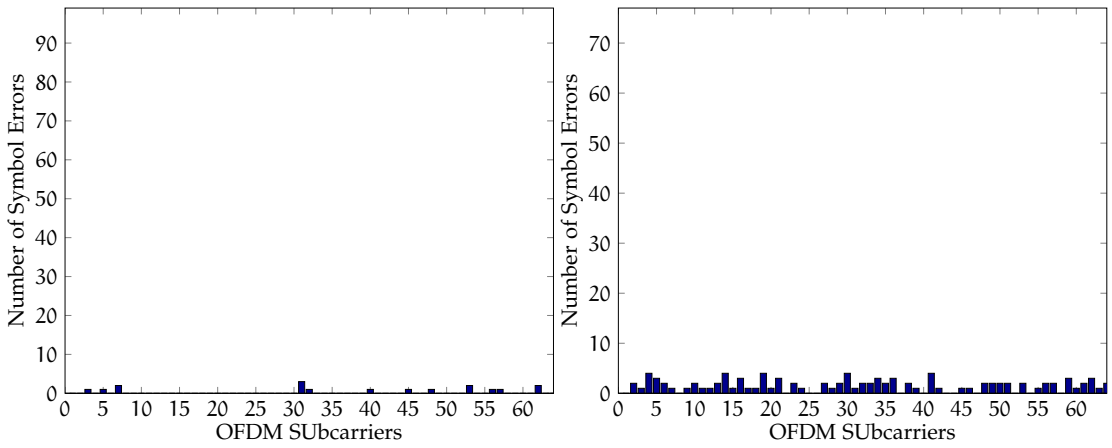
Table 5.4: BER and EVM of Coded Subcarrier covert channel. [BER] = Bits, [EVM] = %

Ch_{Covert}	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	539	363	539	363	539	363	409	409
BER_{σ}	4.165	3.717	2.460	3.707	9.244	7.051	16.367	9.437
BER_{Var}	17.352	13.819	6.052	13.7467	85.452	49.718	267.894	89.059
BER_{Mean}	1.647	1.313	1.549	1.333	15.784	9.039	45.509	38.019
EVM_{σ}	3.5538	4.2771	2.2378	19.304	4.793	4.803	2.938	2.251
EVM_{Var}	12.625	18.293	5.008	372.643	22.978	23.077	8.6358	5.067
EVM_{Mean}	7.983	7.957	7.920	10.298	9.364	8.130	8.934	8.328

Table 5.5: BER and EVM of normal channel for Coded Subcarrier channel [BER] = Bits, [EVM] = %

Ch_{Normal}	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	2016	3504	2400	1632	2496	1728	2016	2016
BER_{σ}	0.0	0.0	0.0	0.0	2.775	3.158	35.711	29.1523
BER_{Var}	0.0	0.0	0.0	0.0	7.699	9.97	1275.31	849.85
BER_{Mean}	2.910	2.350	3.167	2.749	2.595	2.648	4.810	3.187
EVM_{σ}	2.910	2.350	3.167	2.749	2.595	2.648	4.810	3.187
EVM_{Var}	8.4705	5.522	10.031	7.560	6.737	7.016	23.143	10.166
EVM_{Mean}	13.785	11.206	14.737	13.791	13.719	13.905	17.293	15.208

The second design decision concerns the spectral position of the covert subcarriers. The null subcarriers located at the boundaries of the OFDM Symbol could be attenuated by the low-pass filter at the receiver. In such a case other implementation steps are required at the receiver to change the low-pass filter from 20MHz to 40MHz or to change the interpolation filter. To confirm this assumption the number of symbol errors pro subcarrier is plotted in figure 5.12. The same packet was send with different modulations and coding rates. The two most critical situations are presented by 16-QAM encoded with coding rate 3/4 and 64-QAM encoded with coding rate 3/4. The symbols errors located in the covert subcarriers are not suspiciously high compared with the symbol errors in the normal channel. Out of this measurements can derived that the effect of the spectral position of the covert data does not have a high influence in the quality of the received data.



(a) 16-QAM R=3/4 (b) 64-QAM R=3/4

Figure 5.12: Symbol Error pro Subcarrier

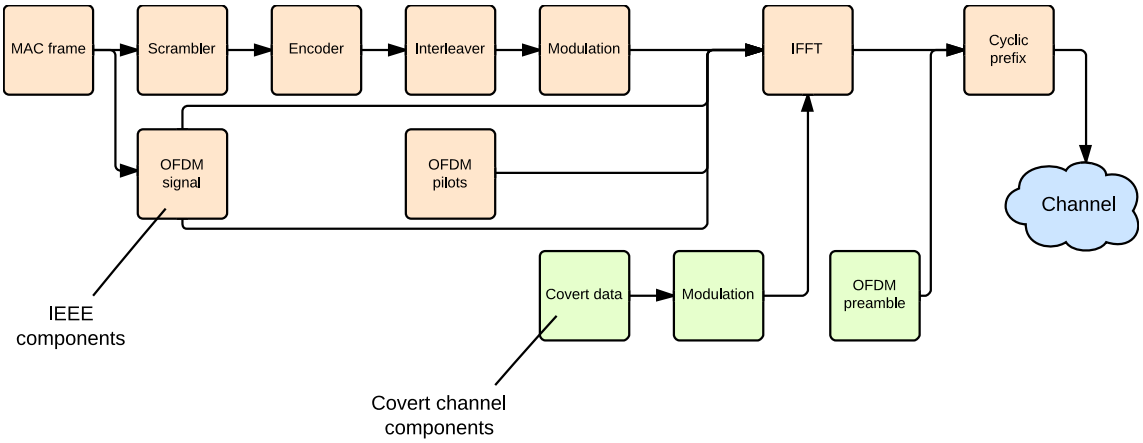


Figure 5.13: Block diagram of the coded subcarrier transmitter. [Own source]

5.2.1.1 *Concept and Implementation*

After deciding how many subcarriers should be substituted or their spectral position the next step is the construction of the covert channel and the integration with the normal channel.

Parallel to the creation of the random signal data, the random covert data are created **??**. There are 11 bits that can be transmitted per OFDM symbol. The parallel to serial conversion transforms the vectors containing random data in two matrices. The data matrix has a row number of 48 and the matrix containing the covert data with a row number of 11. The matrices are separately modulated. According to the IEEE Specification of 802.11 the first OFDM symbol is modulated with BPSK and coding rate R=1/2. The covert data attached to the first OFDM symbol are also modulated with BPSK and coding rate of 1/2. According to the measurement results for the covert and normal channel no direct influence could be observed between the presence of the covert channel and the error rate for the normal channel. In order to exploit the best possible throughput data are sent to the SIGNAL OFDM symbol too. The following covert data are modulated with the same modulation and coding rate as the normal channel, which is specified in the

RATE field of the SIGNAL symbol. The structure of the implementation in WARPLab is depicted in Fig. 5.13.

During the insertion of the pilots in the OFDM symbol, the covert data are inserted in the subcarriers with indexes 28 to 38. The next change is made during the preamble insertion. The preamble consists of 10 identical Short Training Symbols (STS) each containing 16 bits and 2 Long Training Symbols (LTS) each containing 64 bits divided by a Guard Interval of 32 bits. The LTS have null at subcarriers at the position 28 to 38. Due to this values the hidden subcarriers cannot be read at the receiver. For this purpose the values of the subcarriers at these indexes have random values, either 1 or -1. At the receiver the values out of the subcarriers with indexes 28 to 38 are read by using the same LTS as at the sender. Changing the preamble makes the Coded Subcarriers covert channel even robuster to attacks. The channel is characterized by a lower detectability. A normal user would use the preamble specified at the IEEE 802.11 specification and thus overwrite the values of the null subcarriers with ∞. If an attacker would try to read the values out of the covert channel, he needs to know the values of the LTS.

5.3 HICCUPS

We find the Hidden Communication System for Corrupted Networks (HICCUPS) [9] approach quite interesting as it states to build a channel with quite a huge bandwidth and has not yet been analyzed in practice.

The mentioned covert data channels are used in a larger scheme consisting of three modes as follows:

SYSTEM INITIALIZATION All systems included in hidden group which will later exchange covert data using the HICCUPS method establish a key for use in the optional cryptographic parts of the latter modes. The author does not describe any group recruitment, key agreement or distribution protocol or cipher.

BASIC MODE In this mode covert data is exchanged. As transports cipher's initialization vectors and MAC network addresses are used. As these hidden communication channels have a very low bandwidth, they are only meant be used to exchange control information between the members of the group. Also a – not specified – sequence is used to transition into the next mode.

CORRUPTED FRAME MODE This mode has far more bandwidth. Data is transported as payload of IEEE 802.11 frames with intentionally created bad checksums (FCS). This modes lives directly off the bandwidth of the host channel and can consume up to 100% of it, if needed. This certainly does interrupt the normal channel operation. The key point to covert operation is the fact that stations, which do not belong to the hidden group, automatically discard frames with bad FCS.

5.3.1 *Design Challenges*

The definition and specification of the author leaves miscellaneous key decisions up to the implementing party. Thus, we have to decide about the following features:

- There are two main data-exchanging *operation modes*, of which one is no new contribution by the author (IEEE 802.11 header data has been used as a covert channel before, see [5]). It may be worthwhile to not implement this mode.
- Cryptography and key exchange are not defined by the author at all. Analysis and assessment of these may be done theoretically anyway.
- There are no explicit operation procedures defined for any of the modes. For example the author does not specify if – using the corrupted frame mode – covert data should be replaced into existing frames or new frames shall be created. This can be a huge difference.
- There has to be a clear structured implementation and measurement setup.

Before implementation all these challenges have to be clarified in a overall approach concept.

5.3.2 *Concept*

Basic mode

For several reasons the basic mode contributes quite some problems. First of all it is no new contribution and has been assessed using a implementation before. Furthermore its capacity is quite low as only some header fields are manipulated and HICCUPS is interesting just because of its quite large proposed bandwidth, which is mainly offered by the corrupted frame mode. Another relevant point is its obvious detectability problem: Even an attacker with a sketchy approach can easily detect this method.

Because of the mentioned arguments we decided against implementing the basic mode.

Cryptography

One major problem with the optional cryptographic parts of the scheme lies in its literally non-definition by the author. To implement these parts we would have to design and implement a whole key management and communication system, a problem whereby far more experienced researched have already been failing. Also, the WARP platform's 802.11 reference design does not support encrypted communications over WLAN[1] at all as of now, thus we would have to implement that, too. Both points would require far more time than available. As we decided against implementing the basic mode there is another advantage in not implementing cryptography: The system initialization mode has not to be implemented, leaving only one mode left, which simplifies the implementation considerably.

As a result, we do not apply cryptography at all, but consider it during theoretical parts of the assessment.

Operation procedures

This operation procedures problem also concerns the overall mode if measurement. The differences between replacing a packets contents and setting a bad checksum or

1 The scheme is originally using Wired Equivalent Privacy (WEP), but it is applicable to Wi-Fi Protected Access (WPA)/WPA2, too.

just corrupting the checksum has one major difference: Replacing the package contents instead of sending a fresh corrupted frame would have the effect that it has to be retried at the next try. Sending the new one would leave the old intact and no impact on the frames itself could be observed.

To make the covert channel as stealthy as possible we decided to combine both: We willy copy the headers of an existing frame and add our covert data.

Measurement setup

The measurement is a key part of this implementation as this will be the core of the assessment. For this, we will use the existing WARP 802.11 reference design experiment framework as it provides all necessary means to trace the metrics. It can record all transmission and reception events including the full payload and meta data, thus allowing us to compare contents for a solid BER calculation. Furthermore it provides data for FER and bandwidth calculations. As proposed, but not explicitly defined or reasoned, we will inject corrupted frames to pretend FER to be at a certain percentage (obviously above the FER level of the channel without HICCUPS operations).

5.3.3 *Implementation*

Implementation of this covert channel seemed easier on first sight then it ended up being. The core feature to realize the HICCUPS covert channel is to modify the 802.11 reference design to allow modification of the FCS. This looked quite easy to implement: As (nearly) all of layer 2 operations are handled by the CPULOW and CPUHIGH, we assumed the FCS calculation to be also located there, somewhere. After quite deep dive into the innards of the 802.11 reference design it turned out the FCS calculation was made on the *FPGA hardware design* level, that means in Simulink structures. Figure 5.14 shows the default unmodified calculation of the FCS. The frame is clocked through bit-wise and the FCS is updated accordingly on each step. When the *PhyTX* state registers show, that the whole frame has been clocked through and the FCS should be sent, mux 2 switches and the calculated FCS is sent to the output instead of the input data.

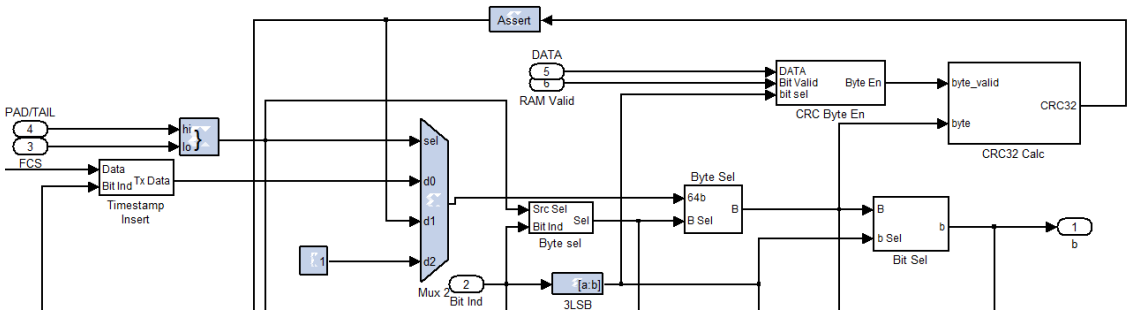


Figure 5.14: 802.11 reference design FCS calculation and output. [Own source]

As this level was far below what we expected we resorted to implementing a easy variant to invalidate the FCS: Setting it to ca constant (zero). This can be easily accomplished by adding a flag that indicates if a frame should be smashed (obviously not all should)

and using it in the output mode to switch between the real FCS – which is still correctly calculated even for HICCUPS frames – and the static corrupted one. The updated scheme can be seen in figure 5.15.

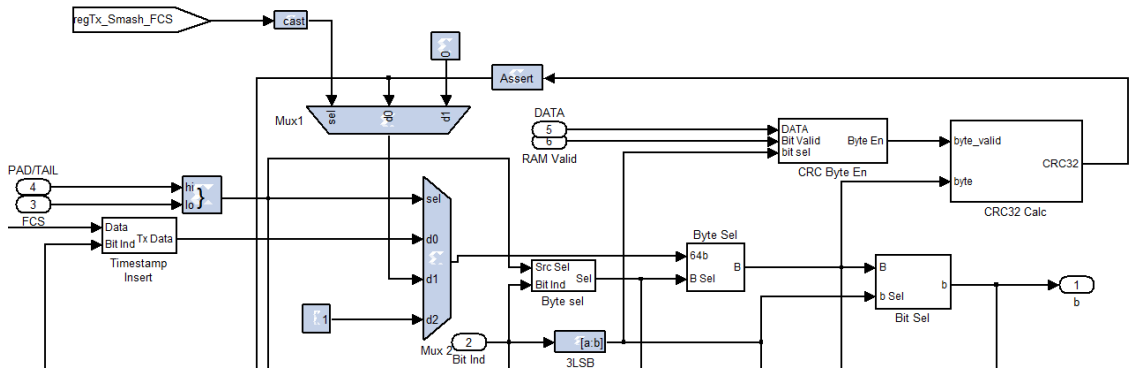


Figure 5.15: HICCUPS FCS calculation and output. [Own source]

As we want to log the full payload of the covert channel like all other normal events too, we had to get frame generation up to the CPUHIGH as this is the only place where logs can be written. The frame has then be handed down to CPULOW with a flag indicating a request for frame corruption, which can then itself set the flag for the Simulink change to become active. To could be achieved quite easily as there were already several flags passed around. We just added another, and all worked fine.

Creation of frames has to be triggered by a mechanism. We implemented it to be triggered on successful frame transmission, as this is the very place where we can adjust our sending behavior to the measured FER and keep the described level. A simple mechanismn can be used to determine the current level of FER and thus the amount of corrupted frames which can be added. The key is to calculate the FER of the own channel by counting ACKs and timeouts of transmitted *all* frames. The difference between our current and the higher target FER level can completely be used for the covert channel.

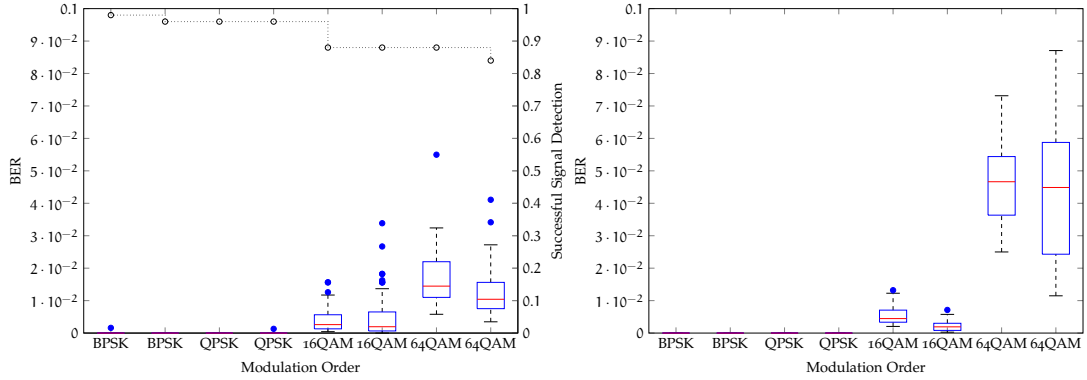
6 EVALUATION

As the performance is an important fact, this chapter demonstrates the capability of the above presented covert channel models. It highlights the throughput as well as BER of each concept, and points out the detectability of the covert channel. For this purpose, the results of the measurements are illustrated in various figures with corresponding tables.

6.1 CODED CYCLIC PREFIX PREFORMANCE

To assess the performance of the Coded Cyclic Prefix model, we use a scenario, in which 2 WARP boards and 1 common-off-the-shelf laptop (EeePC) with an integrated 802.11g radio chip are used. The measurements are taken in the laboratory of the Seemoo department, where the devices are placed on tables, each at a distance of around 5 meters. The WARP boards are flashed with the WARPLab Reference Design version 7.40 and attached via switch to a Macbook Air'13 laptop, that is configured with System Generator in version 14.4 from Xilinx. The Coded Cyclic Prefix model is implemented and executed within the environment of System Generator.

We send from one WARP board to the common-off-the-shelf laptop and to the other WARP board beacon frames at regular intervals. Although this represents a unidirectional communication, this setup proves the functionality of the covert channel. The laptop is used to present the ability of receiving the coded signals. The second WARP serves as measuring device. Channel 14 is chosen, which is shared with other APs but presents a low traffic. The WARPLab interface allows the configuration of the baseband gain and RF gain, that is set to -5 dB and -30 dB respectively. The SSI signal at the receivers are around -60 dBm. From section 5.1.2, we know that we transmit at a rate of 9 Mbit/s. So the interesting part is the BER of the normal channel and covert channel.



(a) Normal channel.

(b) Covert channel.

Figure 6.1: Performance of the Coded Cyclic Prefix in BER and signal detection.

It follows from the Fig. 6.1 that the normal channel is marginally affected. The average transmission of successful beacon are around 90 %. The slight lost can be seen

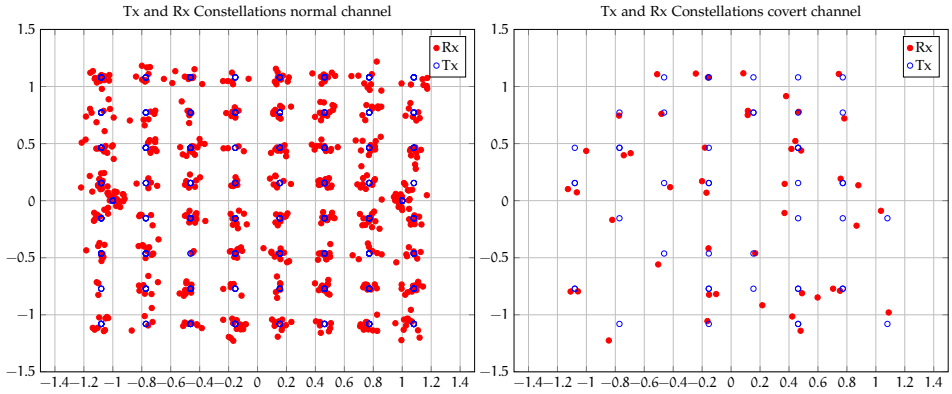
Table 6.1: BER and EVM of normal channel according to Fig. 6.1. [BER] = Bits, [EVM] = %

Ch_{Normal}	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	5132	3504	5136	3504	5232	3696	4080	3504
BER_{σ}	0.0	0.0	0.0	0.141	9.132	11.615	15.764	13.174
BER_{Var}	0.0	0.0	0.0	0.0	19.290	3.059	595.11	2820.5
BER_{Mean}	0.0	0.0	0.0	0.020	9.760	8.320	29.840	21.980
EVM_{σ}	6.789	6.791	4.983	4.805	4.610	7.613	9.924	6.812
EVM_{Var}	46.095	46.120	24.833	23.094	21.255	57.964	98.486	46.404
EVM_{Mean}	17.817	20.027	18.304	17.173	20.600	21.001	26.688	20.523

Table 6.2: BER and EVM of covert channel according to Fig. 6.1. [BER] = Bits, [EVM] = %.

Ch_{Covert}	BPSK	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Size	5132	3504	5136	3504	5232	3696	4080	3504
BER_{σ}	0.0	0.0	0.0	0.0	15.668	5.658	48.710	73.846
BER_{Var}	0.0	0.0	0.0	0.0	245.510	32.017	2372.6	5453.2
BER_{Mean}	0.0	0.0	0.0	0.0	29.600	8.060	198.540	163.300
EVM_{σ}	1.916	2.073	1.935	1.806	2.281	2.180	2.939	3.566
EVM_{Var}	3.674	4.297	3.746	3.262	5.204	4.754	8.641	12.716
EVM_{Mean}	10.714	11.236	11.159	10.631	15.296	12.647	17.868	16.312

as collision of signals from multiple transmitters. For that reason, we can assume that the probability of detection for the covert channel is very low. If the BER of the normal channel is considered, it can be stated that the degradation is relatively small, such that the expected influence from the covert channel can be ascribed to normal traffic lost in radio communication systems. The assumptions are strengthened if we take the results from the Table 6.1 into consideration. The 64-QAM modulation with 3/4 coding rate has BER of around 0.6 %, that is near to the average occuring in most radio system scenarios. The BER of the covert channel is 4 % and is around six times higher then the normal channel. Although this seems to be a high value we assume that features such as encoding in combination with interleaving can compensate the degradation of the covert channel. The EVM of both channels are depicted in Fig. 6.2. It shows that symbols of the normal channel are mapped according the provided target. The EVM of the covert channel presents lesser symbols that is a result of the short PSDU. The spread of the symbols are marginally, such that the fragmentation and composition process does not affect the corresponding signal.



(a) Normal channel. (b) Covert channel.

Figure 6.2: EVM of the Coded Cyclic Prefix model.

6.2 CODED SUBCARRIER PERFORMANCE

To evaluate the performance of the Coded Subcarrier covert channel the same scenario presented in the previous section 6.1 is used. Several beacon frames are sent in regular intervals from on WARP board to the common-off-the-shelf laptop. The other WARP eavesdrops the packet sent to the laptop and reads out of the received OFDM packets the covert data. The achieved throughput is calculated with the following equation:

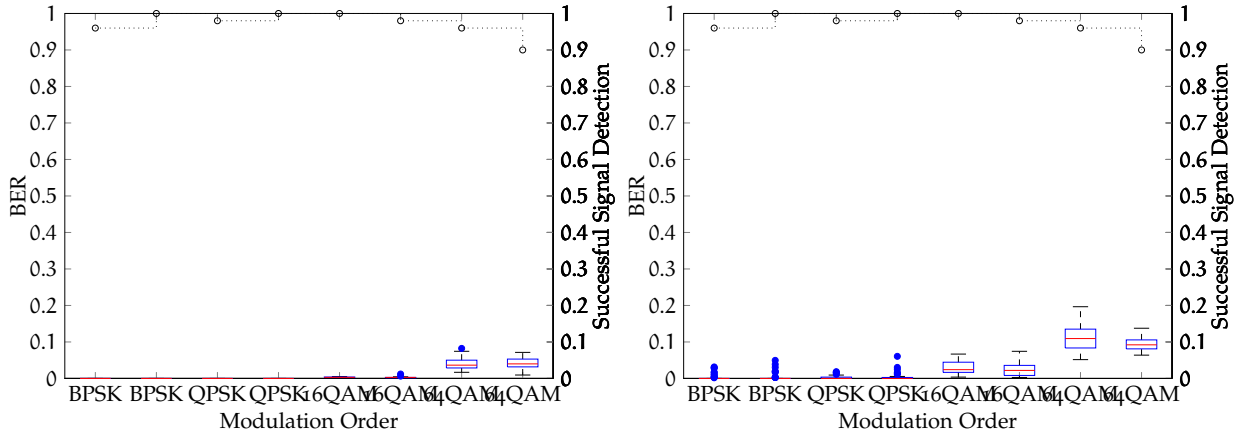
$$R = 11 \cdot \frac{1}{T_{Symbol}} \cdot 6 \cdot 1 = 16{,}5\,\text{Mbit/sec}$$

The maximal throughput is achievable with 64-QAM modulation. The performance of the covert and normal channel can be evaluated comparing the values of BER. In the following graphs in figure 6.3 is depicted the BER and the number of received beacon frames pro modulations art. For modulations such as BPSK or QPSK is the BER very low. Modulation arts such as 16-QAM or 64-QAM have higher BER. In this case the measurements show values up to 9% for the BER of the covert channel.

A certain affection is detected in the normal channel too. The BER has values up to 1% for 64-QAM modulation. In the case of lower modulation is the BER unaffected, remaining at very low values. Further shows the graph the successful transmission of beacons frames for different modulations rates. For 64-QAM amounts the successful transmission to 90%. Other modulation arts have a better transmission behavior. These behavior can be also caused by the collision with other transmissions on the same channel.

In the figure 6.4 is depicted the baseband frequency and the constellations of transmitted and received covert symbols. Out of the magnitude of the channel estimation can be seen that the phase of the subcarriers at the covert channel positions is manipulated. The points at the TX-RX constellations diagram depicted with red are the received symbols. Because of these manipulation the symbols experience phase shifting, which is also shown by the Error Vector Magnitude(EVM) at the table 5.4. The influence of the EVM over the signals is not two high, since they can be mapped to the right symbols. Changing the interpolation filter should improve the channel estimation and thus the mapping position of the symbols.

One of the main aims of this work was the low detectability of the covert channel. Since the BER of the normal channel does not get affected in suspicious values, a normal



(a) Normal Channel (b) Covert Chhannel

Figure 6.3: Performance of Coded Subcarriers in BER and signal detection

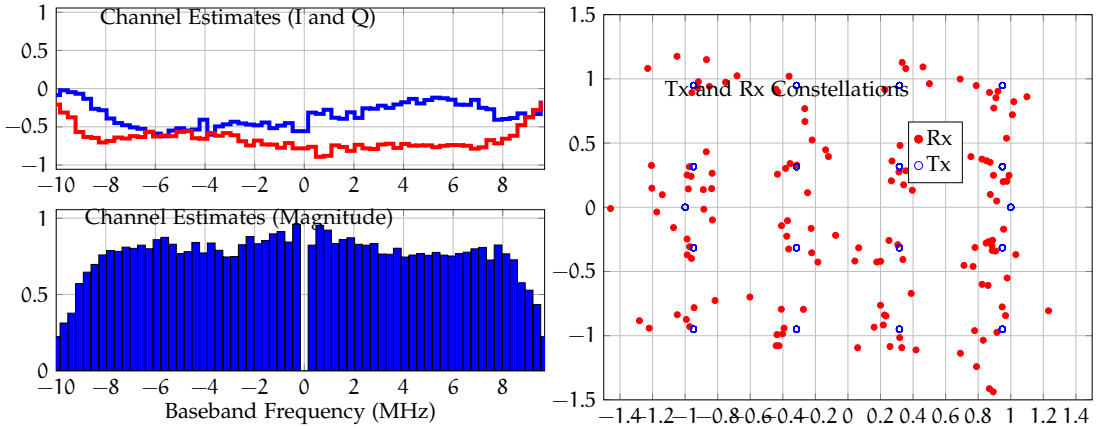


Figure 6.4: Baseband frequency and TX-RX Constellations for 16-QAM Modulation

user does not sense the presence of a covert channel. The low packet loss also supports the covert channel. Even if the attacker makes further measurements of the channel it would be difficult for him to detect the presence of the covert data in the null subcarriers, since he would need to change the preamble to see if there are hidden any data. The next challenge for him would be the decoding the covert data, since he would need the right sequence of the bits at the LTS.

6.3 HICCUPS PERFORMANCE

To asses the HICCUPS covert channel we use the metrics as defined in chapter 4.

Throughput

Measurements using the described implementation ended with a quite constant average of 1.2 Mbps for the covert channel for a two node WLAN network (sender, receiver, both WARPs) using a change in FER (ΔFER) of 5%. This value has been chosen to compare the

results with the calculations of the author of the original paper. He stated 1.27 Mbps in a calculation for an equivalent channel in [10, sec. 4]. Thus, it seems this is quite on point.

Latency

Latency of overall transmissions are staying the same. This was to be expected as no changes were made on the transmission itself. Nevertheless, latency for packets added to the sending queue are indeed expected to increase are they are kept back to covert channel inject frames into the process. We were not able to measure such a latency, as the WARP measurement framework does not include any means to check timing on the sending queue.

BER

We expected BER to stay the same – again – because the transmission itself was not changed. This could be verified throughout the measurements.

FER

We most obvious impact of the HICCUPS algorithm is FER as its whole channel lives of corrupted frames. Thus, FER is expected to rise at the same percentage that is used for the HICCUPS covert channel. This assumption could be verified by measurements.

Detectability

At first glance the protocol seems to rather easy to detect. Its implementation is based on layer 2, so any sniffing attacker can possibly the protocol with simple tools like a WLAn interface in promiscuous mode and a capturing tool like wireshark. Looking deeper in to the protocol structure there are some fixes for this and also some more elaborate detection methods, which an attacker as specified in section 1.3, can use.

The first quite obvious detection method against the described implementation consists of monitoring FCS values. The implementation inserts a constant for FCS for ease of implementation. This is not the smartest choice, an attacker could just watch for nodes with a high amount of recurring FCS values. The fix also very straight forward: use random FCS values instead of static ones. An attacker is not able to detect if the CRC is random or the frames got corrupted quite bad. Even if he would, taking the original FCS and changing it by a small value looks for an attacker as if the frame was just lightly corrupted.

Another approach for detection can be the inspection of frame contents. A regular frame contains several headers inside its payload. These are typically (in order) IEEE 802.2 header and Subnetwork Access Protocol (SNAP) header followed by layer 3 protocols and upwards. If no such headers are included in a data frame, this frame might be tampered/corrupted as there is no other legitimate use for such a packet. Encrypted HICCUPS data look suspicious in this context. A solution to this method can be the insertion of such headers. This means, the practical payload for the HICCUPS covert channel of such data frames is reduced to include headers that make the frame look like legitimate traffic. This can also be extended by inserting protocols up to higher layers that use cryptography (like HyperText Transfer Protocol Secure (HTTPS)) into the frame and and replacing their payload with the HICCUPS covert channel data. If done right,

this can truly make HICCUPS frame payloads indistinguishable from legitimate corrupted ones.

As the main effect of HICCUPS is the increase of FER this can also be used to try to detect the usage of HICCUPS. If an attacker observes a stable channel and recognizes a sudden increase of FER this *may* be a clue, that HICCUPS is being used. Unlike the previous detection methods this one is quite fuzzy: The sudden increase could also be the result of a station gone mobile. This is known to increase the FER as the author of the HICCUPS paper states himself in [9, sec. 6] ("increase up to 30%"). To even prevent the detection of this clue a HICCUPS implementation could itself observe the FER and try to keep within the observed error derivation. This may shrink the covert channels throughput drastically.

A last detection method includes measuring the RSS at the attackers position. In a static situation (relatively between the attacker and the observed node), the observed derivation RSS is low. This implicates that the observed FER is also nearly constant. Any larger changes indicate a manual intrusion, e.g. by HICCUPS. This mechanism can be thought of as an extension of the technique described before, which is now able to kind of estimate the uncertainty factor *mobility* and thus reduces the false negative rate considerably. A practical measurement of this coherence can be seen in figure 6.5. Over two hours 802.11 frames were received by a station and the average FER of each seen sending MAC address was plottet against the average reception power of all this events over the whole time. The size of the circle indicates how much frames were seen. This was a quite static environment with a big AP (circle in the middle) and several stations sensing a small amount of frames (smaller circles). A correlation between these two metrics can be seen. We currently cannot see any reliable countermeasure against this detection method.

When applying these countermeasures HICCUPS resilience against detectability can be assumed to be quite good. This does also prevent further attacks by an adversary, namely prevention of communication and identification of participants as described in section 1.3. In fact, HICCUPS can be attacked very easy by these attacks as all HICCUPS data is contained in isolated frames that can be touched attacked without preventing any other legitimate communication.

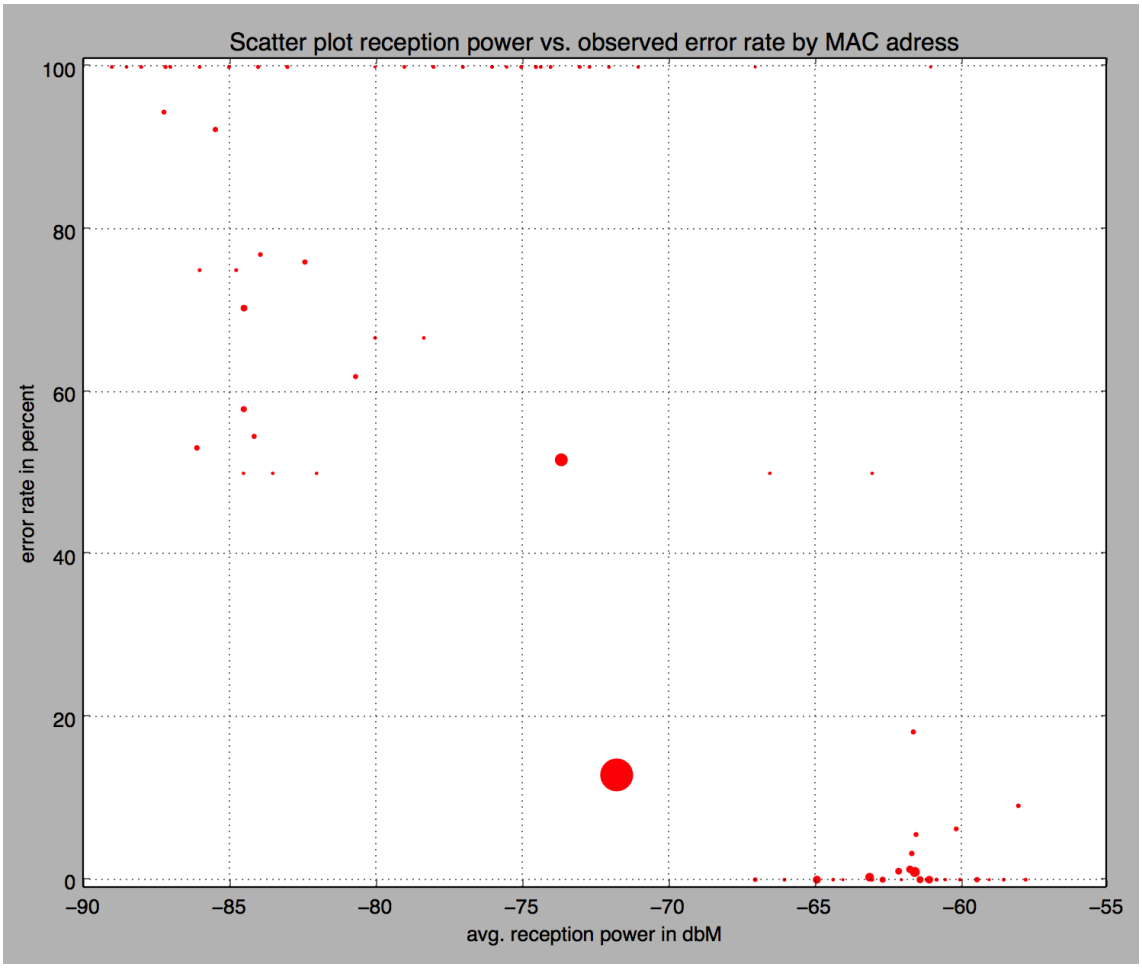


Figure 6.5: Measured FER vs reception power. [Own source]

7 CONCLUSION

Within the scope of the lab project, we designed and implemented various models of covert channel in the physical and data link layer of the IEEE 802.11 communication system. We analyzed different approaches on this topic and assessed the proposals and our implementations through the metrics throughput,BER, FER and detectability. Comparing the proposed theoretical and our practical results revealed discrepancies especially in respect to feasibility in practice. The implementation and evaluation of the covert channels in practice are realized on the WARP platform and base on the provided reference platform designs. We used a typical scenario to prove the functioning ability of the covert channel system and to perform measurements and tests, which demanded to implement own measurement tools. We achieve a throughput of around 1.2 MBit/s in the covert channel model of HICCUPS on the MAC layer and 16.5 MBit/s and 9 MBit/s Coded Cyclic Prefix and Coded Subcarriers on the PHY layer.

Part I

APPENDIX

A

SCHEDULE AND WORKLOAD

As shown in table A.1, our schedule mainly consists of the given deadlines. As of handing in the project definition we generally concluded our literature research. After that, we are parallel working together on getting to know the WARP platform in detail and dive into the mentioned performance measurement. For the alpha stage we are each going to chose a covert channel and start working on implementing it. This should have resulted in some basic tests when handing in the alpha documentation. After that we are going to complete and stabilize the implementations. For the final report we are working on polished and fine tuned implementations and full documentation.

Table A.1: Time schedule of lab project. Team members: Halis by [H], Stephan by [S], Athiona by [A]

Date due	Deliverable	Assignment	Status
Fri 09.05.2014	Literature Research	H, S, A	done
Fri 09.05.2014	Project definition, schedule, workload	H, S, A	done
Wed 21.05.2014	Mock-Up, focus and priorities	H, S, A	done
Wed 04.06.2014	WARP practice	H, S, A	done
Wed 04.06.2014	Structure of draft, deliverables, milestones	H, S, A	done
Wed 18.06.2014	Implementation of performance measuring	H, S, A	done
Wed 18.06.2014	First covert channel tests	H, S, A each	done
Wed 18.06.2014	Alpha stage with intermediate version of text, project def. and demo	H, S, A	done
Wed 16.07.2014	Covert channel demos	H, S, A each	done
Wed 16.07.2014	Beta stage: more elaborated version	H, S, A	done
Wed 23.07.2014	Stable covert channels with basic documentation	H, S, A each	done
Wed 23.07.2014	Hand in of pre final report including doc. of controlling	H, S, A	done
Wed 30.07.2014	Pre final slides	H, S, A	done
Wed 06.08.2014	Fine tuning of covert channel implementations	H, S, A each	done
Wed 06.08.2014	Final report	H, S, A	done
Wed 11.08.2014	Final slides	H, S, A	done
Wed 12.08.2014	Final presentation	H, S, A	done

BIBLIOGRAPHY

[1] Telvis E. Calhoun, Xiaojun Cao, Yingshu Li, and Raheem Beyah. An 802.11 mac layer covert channel. *Wireless Communications and Mobile Computing*, 12(5):393–405, 2012.

[2] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. Secret agent radio: Covert communication through dirty constellations. In Matthias Kirchner and Dipak Ghosal, editors, *Information Hiding*, volume 7692 of *Lecture Notes in Computer Science*, pages 160–175. Springer Berlin Heidelberg, 2013.

[3] Darwin Engwer. "WDS" Clarification, July 2005. Available online at: http://www.ieee802.org/1/files/public/802_architecture_group/802-11/4-address-format.doc; visited on 3.8.2014.

[4] S. Grabski and K. Szczypiorski. Steganography in OFDM Symbols of Fast IEEE 802.11n Networks. In *Security and Privacy Workshops (SPW), 2013 IEEE*, pages 158–164, May 2013.

[5] Christian Krätzer, Jana Dittmann, Andreas Lang, and Tobias Kühne. WLAN steganography: A First Practical Review. In *Proceedings of the 8th workshop on Multimedia and security*, pages 17–22. ACM, 2006.

[6] Ramjee Prasad. *OFDM for wireless communications systems*. Artech House, 2004.

[7] IEEE Computer Society. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[8] K. Szczypiorski and W. Mazurczyk. Hiding Data in OFDM Symbols of IEEE 802.11 Networks. In *Multimedia Information Networking and Security (MINES), 2010 International Conference on*, pages 835–840, Nov 2010.

[9] Krzysztof Szczypiorski. HICCUPS: Hidden communication system for corrupted networks. In *International Multi-Conference on Advanced Computer Systems*, pages 31–40, 2003.

[10] Krzysztof Szczypiorski. A performance analysis of HICCUPS – a steganographic system for WLAN. *Telecommunication Systems*, 49(2):255–259, 2012.

[11] WARP Project. Wireless Open Access Research Platform. Available online at: http://warpproject.org/trac; visited on 3.8.2014.

[12] WARPLab. Wireless Open Access Research Platform Lab. Available online at: http://warpproject.org/trac/wiki/WARPLab; visited on 3.8.2014.