DIPLOMA THESIS

Interference Analysis of LoRaWAN Systems

supervised by
Univ. Ass. Dipl.-Ing. Jure Soklic
and
Assoc. Prof. Dipl.-Ing. Dr. techn. Holger Arthaber

performed at the
Institute of Electrodynamics, Microwave and Circuit Engineering

by

Harald Eigner
Matr. Nr. 01225917
Währingerstraße 160/20, 1180 Wien

January 15, 2021

Abstract

The Internet of Things (IoT) gained an exponentially growing amount of devices in the last two decades. The concept of IoT has been adopted to an increasing number of applications in several sectors. There are unlimited possibilities for connecting smart "things", such as medical sensors, security systems, or even contact lenses or refrigerators. The key property of IoT devices is communication over a long distance with low power and low costs. For this purpose, many new technologies have evolved recently. LoRa and LoRaWAN are a physical and network layer standard, which gained increased interest in the last years in this regard. It is operated in license-free ISM frequency bands, simultaneously to many other technologies, which leads to much traffic in this frequency spectrum.

The focus of this thesis is to evaluate the resistance of LoRa against several sources interfering during the transmission. For the measurements of the evaluation of the resistance, a communication system was set up. In addition, we simulated the transmitter and receiver chain in Matlab to create specific signals for our measurement purposes and evaluated the results. To do this, we had to examine and implement the encoding and modulation process used by LoRa. In this thesis, we tested the system with two different interference sources. First, a continuous wave signal representing the carrier frequency of a general signal was used to examine the impact on the transmission. Our results show that a successful reception is possible with an interference power 10 to 20 dB higher than the LoRa signal, dependent on the spreading factor. The second is a collision of two LoRa signals. We tested the impact of power, spreading factor, and time when the second signal starts to interfere during the reception. It can be shown that the time at which the delayed signal starts to interfere has an impact on the reception of the examined signal. This finding further supports the notion that the spreading factor does not significantly affect the sensitivity at a collision with another LoRa signal.

Zusammenfassung

Das Internet of Things (IoT) hat in den letzten zwei Jahrzehnten eine exponentiell wachsende Anzahl an Geräten dazugewonnen. Seitdem wurde das Konzept von IoT in Anwendungen vieler verschiedener Sektoren übernommen. Es gibt unbegrenzte Möglichkeiten für die Verbindung "intelligenter Dinge" wie z.B. medizinische Sensoren, Sicherheitssysteme oder sogar Kontaktlinsen oder Kühlschränke. Die Schlüsseleigenschaft von IoT-Geräten besteht darin die Kommunikation über große Entfernungen mit geringem Stromverbrauch und niedrigen Kosten zu ermöglichen. Aus diesem Grund wurden in den letzten Jahren viele neue Technologien entwickelt. LoRa bzw. LoRaWAN ist ein Standard auf der physikalischen und der Netzwerkebene, der in den letzten Jahren in diesem Zusammenhang verstärkt an Interesse gewonnen hat. Dieser Standard wird in lizenzfreien ISM-Frequenzbändern neben vielen anderen Technologien betrieben, was zu viel Verkehr in diesem Frequenzspektrum führt.
Der Fokus dieser Arbeit ist die Widerstandsfähigkeit von LoRa gegen verschiedene Störquellen während der Übertragung zu untersuchen. Zu diesem Zweck wurde ein Kommunikationssystem aufgebaut und zusätzlich die Sender- und Empfängerkette in Matlab simuliert, um spezifische Signale für die projektbezogenen Messzwecke zu erzeugen und die Ergebnisse auszuwerten. Dazu mussten wir das von LoRa verwendete Kodierungs- und Modulationsverfahren untersucht und implementierten werden. Im Zuge dieser Arbeit wurde das System mit zwei unterschiedlichen Störquellen getestet. Zunächst wurden die Auswirkungen einer Interferenz mit einer Trägerfrequenz eines allgemeinen Signals in Form eines continuous wave (CW) Signals untersucht. Diese Ergebnisse zeigten, dass je nach Spreizfaktor ein erfolgreicher Empfang mit einer 10 bis 20 dB höheren Störleistung als das LoRa-Signal möglich ist. Weiters wurde die Empfangsqualität bei einer Kollision von zwei LoRa-Signalen getestet. Dabei wurde der Einfluss der Leistung, des Spreizfaktors und der Zeit, ab der das zweite Signal während des Empfangs zu stören beginnt, überprüft. Unsere Untersuchungen ergaben, dass der Zeitpunkt, an dem das verzögerte Signal zu stören begann, einen Einfluss auf den Empfang des untersuchten Signals hatte. Diese Erkenntnis unterstützt die Annahme, dass der Spreizfaktor die Empfindlichkeit bei einer Kollision mit einem anderen LoRa-Signal nicht signifikant beeinflusst.

Acknowledgements

I want to thank my supervisors Assoc. Prof. Dipl.-Ing. Dr. techn. Holger Arthaber, Univ. Ass. Dipl.-Ing. Jure Soklic, and Univ. Ass. Dipl.-Ing. Bernhard Pichler for guiding me through my thesis and giving me the support and technical expertise when needed. I would also like to thank all members of the Microwave Engineering Group at TU Wien. And special thanks go to my family and friends who supported me during my time at the university.

Parts of this work have been funded by EFRE Interreg project InterOP (ATCZ175).

Contents

Abbreviations

ABP Activation by Personalization

ADR Adaptive Data Rate

AppKey Application Key

AppSKey Application Session Key

AS Application Server

BER Bit Error Rate

BLE Bluetooth Low Energy

BW Bandwidth

CFlist Channel Frequency List

CR Code Rate

CRC Cyclic Redundancy Check

CSS Chirp Spread Spectrum Modulation

CW Continuous Wave

DevAddr Device Address

DevEUI Device Extended Unique Identifier

DevNonce Device Nonce

DSSS Direct Sequence Spread Spectrum

FCnt Frame Counter

FCntDown Downlink Frame Counter

FCntUp Uplink Frame Counter

FCtrl Frame Control

FD Frame Delimiter

FEC Forward Error Correction

FHDR Frame Header

FOpts Frame Header

FPort Frame Port

FRMPayload Frame Payload

FSK Frequency Shift Keying

IoT Internet of Things

JoinEUI Join Extended Unique Identifier

JS Join Server

LFSR Linear Feedback Shift Register

LoRa Long Range

LoRaWAN Long Range Wide Area Network

LPWAN Low-Power Wide-Area Network

MHDR MAC-header

MIC Message Integrity Code

NS Network Server

NwkSKey Network Session Key

OS Operating System

OTAA Over the Air Activation

TLS Transport Layer Security

RSSI Received Signal Strength Indicator

SF Spreading Factor

SIR Signal to Interference Ratio

SNR Signal to Noise Ratio

UNB Ultra Narrow Band

Chapter 1

Introduction

The concept of the so called Internet of Things (IoT) started to grow tremendously in the last two decades. After the internet was established, the number of devices connected to it grew exponentially ever since. The idea of IoT evolved almost simultaneously, covering other needs than the internet itself. While the main goal of the internet is to achieve faster data rates, IoT tries to connect devices with 'less of everything' [4]. In order to be able to connect a huge amount of devices, the goal is to establish a communication with less memory, less processing power, less bandwidth, and less energy.

For the needs of IoT, some applicable technologies have been developed and invented over the past years. Figure 1.1 shows a comparison of currently established wireless communication standards associated with IoT. Almost the whole world is covered with a cellular communication system, and Wifi can be found almost comprehensively. But a huge drawback of these communication systems is high power consumption, which is essential for a big part of the IoT-segment. Since many applications are located where no power supply is available, or mobility is needed, battery-driven devices are conventional. A communication standard facing this problem is Bluetooth Low Energy (BLE), which has evolved in the last decade from the standard Bluetooth technology. While the power consumption got distinctly lower, the range is very limited with BLE.

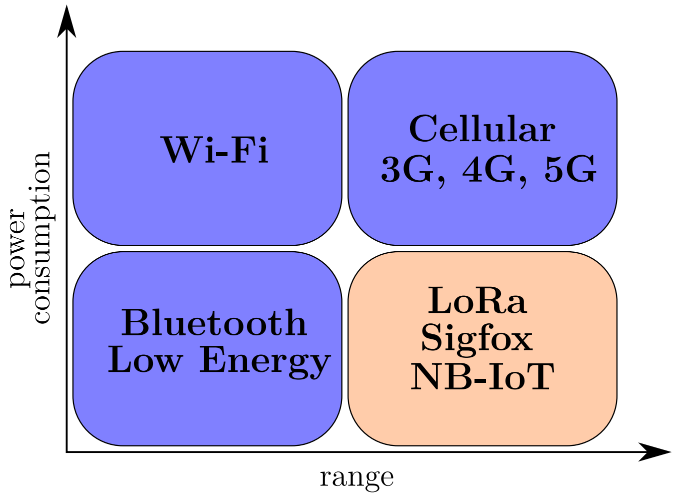


Figure 1.1: Comparison of different communication systems

Several so called Low-Power Wide-Area Network (LPWAN) standards have been developed to fulfil the IoT requirements as best as possible. The main attributes of LPWAN-technologies are:

1. **Long Range:** Coverage of areas over 10 km.
2. **Low Power:** Optimized for small power consumption and a long lifetime.
3. **Low Cost:** Reduced complexity of hardware and software design. Inexpensive infrastructure due to long range and star topology.

Two established LPWAN-standards are Sigfox and LoRa. Sigfox uses an Ultra Narrow Band (UNB) modulation, and operates on the 868 MHz ISM frequency band [10]. The spectrum is divided into 400 channels with a bandwidth of 100 MHz each. Therefore, the noise contribution is meager and the communication works with extremely low power signals. There is a limitation of 6 messages per hour and no acknowledgement is exchanged in the communication process. To ensure a correct reception, each message is sent multiple times on different channels, which combats fading. Because of all these properties, Sigfox is a very suitable choice for many IoT applications.

LoRa stands for Long Range and represents a long range wireless communication system created by Semtech and the LoRa Alliance. The name LoRa commonly refers to two different layers:

1. **LoRa:** LoRa is a physical layer standard, developed by Semtech and is based on a Chirp Spread Spectrum (CSS) modulation technique.
2. **LoRaWAN:** LoRaWAN stands for Long Range Wide Area Network and is a network layer protocol, specifically designed to work on top of LoRa by the LoRa Alliance.

LoRa can achieve data rates up to 50 kbps at a fixed bandwidth of 125 kHz, 250 kHz or 500 kHz. It also operates in the 868 MHz frequency band as Sigfox. A combination of the modulation scheme used, a trade off of data rate for energy consumption and the network architecture leads to the following key features of LoRa [12]:

- **Bandwidth Scaleability:** LoRa can be adapted in both narrowband frequency hopping and wideband direct sequence applications due to its scalability in frequency and bandwidth.
- **Constant Envelope / Low-Power:** LoRa is a constant envelope modulation scheme, similar to Frequency Shift Keying (FSK). Therefore, the same low-cost and low-power amplifier stages can be used for both modulation schemes without modifications.
- **High Robustness:** LoRa is very resistant to in-band and out-band interferences due to a large bandwidth and low data rate.
- **Multipath, Fading and Doppler Resistant:** The large bandwidth makes the communication very resistant against multipath and fading, which is very dominant in urban and suburban areas. Small frequency shifts due to the Doppler effect introduce a relatively small shift in the time axis of the baseband signal, which mitigates the requirement for tight tolerance reference clock sources.
- **Long Range Capability:** For a fixed output power and throughput, the link budget of LoRa exceeds that of conventional FSK. When taken into conjunction with the proven robustness to interference and fading mechanisms, this improvement in link budget translates to an even bigger range.

Many technologies fit for different needs for the IoT and other applications. Table 1.1 shows a comparison of the major technologies.

Table 1.1: Comparison of different communication standards

Technology	**max. Data Rate**	**Bandwidth**	**max. Range**	**Standard**
BLE	3 Mbps	1 MHz	100 m	propietary
Wi-Fi	54 Mbps	20/40/80 MHz	100 m	IEEE 802.11
3G	2 Mbps	5 MHz	10 km	3GPP
4G	1 Gbps	1.4-100 MHz	11 km	3GPP
5G	10 Gbps	10 - 400 MHz	11 km	3GPP
LoRa	50 kbps	125/250/500 kHz	20 km	LoRa Alliance
Sigfox	100 bps	100 Hz	50 km	Sigfox
NB-IoT	200 kbps	200 kHz	10 km	3GPP

Several standards operate parallel in the open source ISM bands, which leads to a lot of traffic. In this master thesis, we investigated the performance of a LoRaWAN system with several sources interfering the transmission.

In the beginning, this thesis gives an introduction to the theoretical basis of the theory of LoRa and LoRaWAN in chapter 2. Firstly, it provides an insight at the LoRa physical layer, starting with the principles of spread spectrum and the LoRa spread spectrum modulation. It then continues with an explanation of structure of the frames followed by the encoding process in the transmitter and receiver chain. Secondly, the network layer LoRaWAN is introduced. After basics about the protocol have been outlined, an explanation of the three different device classes and the activation methods follows.

Chapter 3 describes the setup we used to perform measurements and simulations with a LoRa communication system. It starts with details about hardware and software parts needed to build a working communication, followed by a sensitivity analysis. To validate the results of our measurements, we made a simulation of the system in Matlab, shown in the second part of this chapter.

With a working test setup, we started to investigate the performance of the communication system with several interferences. We started with a continuous wave signal as interference source in chapter 4. LoRaWAN operates in a license free frequency band parallel to several other communication technologies. With a continuous wave signal, we simulated carrier frequencies of signals from other systems and investigated the impact on the transmission of the LoRa. Lastly, we tested the performance of several LoRa signals interfering with each other in chapter 5 under different conditions regarding the signal parameters and receiving powers.

Chapter 2

Theory

This chapter provides the theoretical background of the communication process with LoRa-technology. Section 2.1 gives an introduction to the LoRa physical layer. It starts with an explanation of spread spectrum techniques in general and the Chirp Spread Spectrum (CSS) technique used in LoRa. It continues with a detailed description of the fundamental processing steps in the transmitter and receiver chain in section 2.1.4, including encoder, decoder, modulator, and demodulator, followed by insights into the packet structure of a LoRa message.
The second part of the chapter is an introduction to the LoRaWAN network layer in section 2.2, including the architectural structure of a LoRaWAN system and specifications of the frequency band, in which LoRaWAN operates on top of LoRa. Furthermore, the three different classes A, B, and C for end-device types are described, which define the timing of the communication process. The last section of the chapter describes the two available activation procedures: Over the Air Activation (OTAA) and Activation by Personalization (ABP).

2.1 LoRa

LoRa is based on a spread spectrum technique, which uses a fixed bandwidth and 'spreads' the information over time and frequency. Due to this, the receiver can achieve higher sensitivity and recover signals over a longer distance, even with a low Signal to Noise

Ratio (SNR).

2.1.1 Principles of Spread Spectrum

Spread-spectrum modulation spreads the information over a higher bandwidth (BW) as required. The advantage of this technique is the reduction of the Signal-to-Noise Ratio (SNR) as seen in the Shannon-Hartley Theorem in equation (2.1). The channel capacity C defines the maximum rate at which information can be transmitted over a communication channel. At a fixed and low SNR, the only variable to increase the capacity is the bandwidth.

$$C = BW\ log_2 \left(1 + \frac{S}{N}\right), \tag{2.1}$$

where

- C . . . Channel Capacity ($^{\text{bits}}/_{\text{s}}$)
- BW . . . Bandwidth (Hz)
- SNR . . . Signal to Noise ratio (1)

Direct Sequence Spread Spectrum (DSSS)

DSSS is the traditional spread spectrum technique. The original data signal is expanded by being multiplied with a pseudo-random sequence, the spreading code. A sequence of data bits, a spreading code and the resulting DSSS signal is depicted in figure 2.1. The bit rates of data and DSSS signal are defined as

$$R_{bit} = \frac{1}{T_{bit}} \tag{2.2}$$

and

$$R_{chip} = \frac{1}{T_{chip}}. \tag{2.3}$$

The chiprate of the DSSS signal is much higher than the bit rate of the data sequence. The resulting spectra of these signals are shown in figure 2.2. The DSSS signal has a much higher bandwidth, hence the information is 'spread' over frequency. The amount of spread is specified by the processing gain G_P, defined as

$$G_{P,\mathrm{dB}} = 10\ log_{10} \left(\frac{R_{chip}}{R_{bit}}\right). \tag{2.4}$$

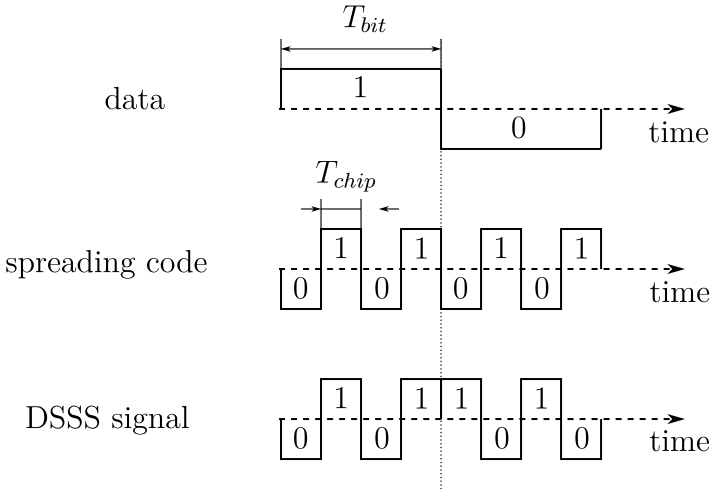


Figure 2.1: Data signal, spreading code, and DSSS signal

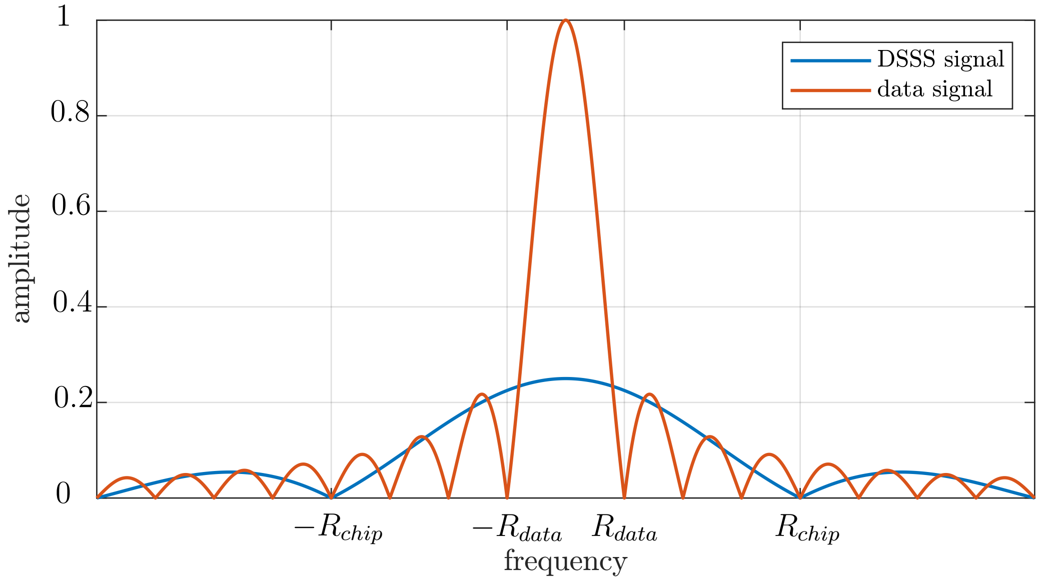


Figure 2.2: Frequency spectrum of the data and baseband signal

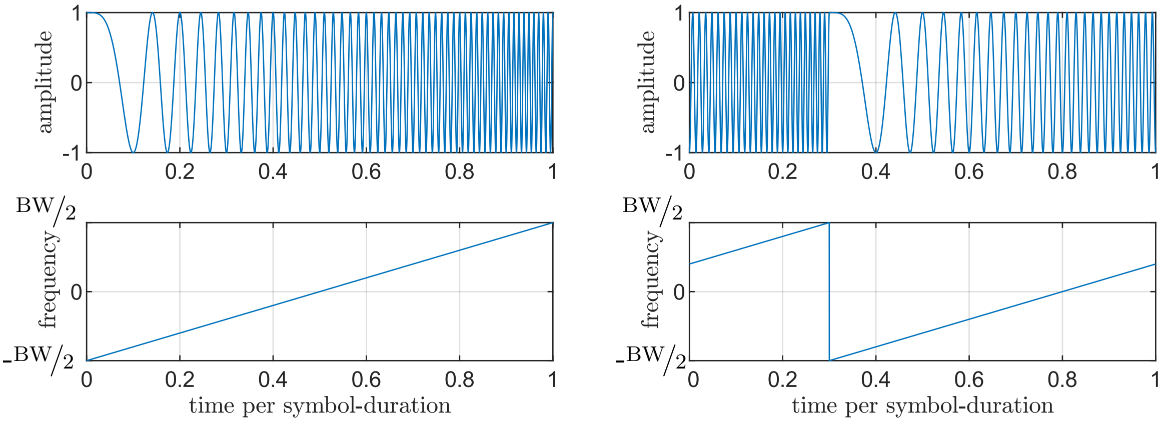
At the receiver end, the signal can be recovered by multiplying the received sequence with the same spreading code again.

Limiting factors of DSSS are, for one thing, the occurance of problems problems for low-cost or power-constrained devices occur, as they require a highly accurate and expensive reference clock source. Furthermore, it requires a longer spreading code, resulting in a longer time required to perform a correlation over the entire length of the code sequence at the receiver. This is problematic for devices that cannot always be active due to a limited energy source and need to be able to repeatedly and rapidly synchronize.

2.1.2 LoRa Spread Spectrum Modulation

LoRa Spread Spectrum Modulation was designed to address the issues of DSSS. The technique used in LoRa communication is based on the Chirp Spread Spectrum Modulation CSS. In this case, a spreading of the spectrum is achieved by generating a chirp signal that continuously changes its frequency. Figure 2.3a shows a raw chirp with BW in time and frequency-domain. A raw chirp is a complex signal with a frequency value starting at $f_{start} = {}^{-BW}/_{2}$ and continuously increasing up to the end value of $f_{stop} = {}^{BW}/_{2}$ after one symbol duration.

In order to transport data, each chirp is 'modulated' by cyclically shifting the chirp. Figure 2.3b shows a modulated chirp in time- and frequency domain. The starting value of the frequency is different and contains the data. The rate at which the frequency increases is always the same as in the case of a raw chirp. When the value of the frequency reaches f_{stop}, it restarts at f_{start} and continues with the same slope until the end of the symbol-duration.



(a) Raw chirp

(b) Cyclically shifted (modulated) chirp

Figure 2.3: LoRa chirps

Unlike in DSSS, the information is spread over time instead of frequency in LoRa, while the bandwidth is constant. The amount of spread is given by the spreading factor $SF = \{7, 8, 9, 10, 11, 12\}$, which defines the symbol duration and the number of bits N, contained in one symbol [12]. For higher spreading factors, a low data rate mode is available to increase the chances of a correct transmission. The number of bits contained in one symbol is defined as

$$N = \begin{cases} SF & \text{with disabled low data rate mode} \\ SF - 2 & \text{with enabled low data rate mode} \end{cases} \tag{2.5}$$

Figure 2.4 shows a comparison of chirps with a bandwidth of 125 kHz and different spreading factors. A comparison of chirps with different SF is depicted in figure 2.4.

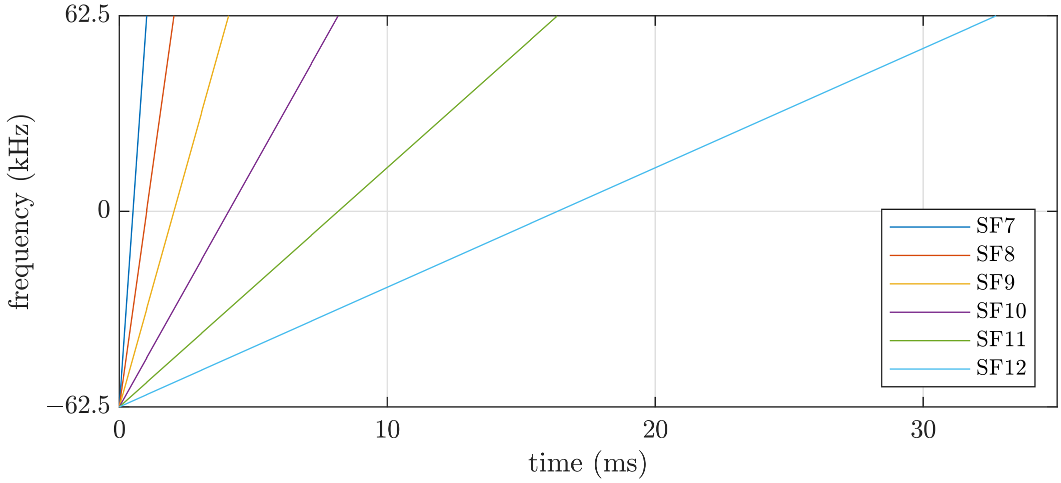


Figure 2.4: Lora Chirps with $BW = 125\,\text{kHz}$ and various spreading factors

The symbol duration $\mathrm{T_s}$ is defined as

$$T_s = \frac{2^{SF}}{BW}, \tag{2.6}$$

6 where:

- SF ... Spreading Factor (7 ... 12)
- BW ... Bandwidth (Hz)

The data is being encoded with the code rate $^4/_5$, $^4/_6$, $^4/_7$, or $^4/_8$, represented by $CR = 1 \ldots 4$. This leads to an expression of the bit rate defined as

$$R_b = N \frac{4}{4 + CR} \frac{BW}{2^{SF}}. \tag{2.7}$$

Table 2.1 shows different bit rates for specific spreading factors and bandwidths.

Table 2.1: Physical bit rates with lowest code rate $^4/_5$

SF	**BW** (kHz)	**bit rate** ($^{\mathbf{kbit}}/_{\mathbf{s}}$)
12	125	0.293
11	125	0.537
10	125	0.977
9	125	1.758
8	125	3.125
7	125	5.469
7	250	10.938
7	500	21.875

A mathematical description LoRa-chirps is given in [5]. LoRa uses mainly upchirps to transmit information. For synchronization purpose, also downchirps are used. The following derivation is a mathematical description only of an upchirp. Every chirp represents one symbol with a length of N bits and a corresponding symbol alphabet size of $M = 2^N$. $a[k]$ denotes the symbol at time kT_s with $a[k] \in \{0, \dots, M-1\}$. To distinguish between all symbols $a[k]$, M different chirps have to be defined. The signal frequency $f_{chirp}(\tau)$ of an upchirp is given by

$$f_{chirp}(\tau) = (f_{start} + \frac{BW}{T_s}\tau). \tag{2.8}$$

The raw upchirp starts at $f_{start} = -^{BW}/_2$ and the phase is determined by

$$\theta_{chirp}(\tau) = 2\pi \int_{\tau}^{T_s} f_{chirp}(\tau) d\tau = 2\pi \left[-\frac{BW}{2T_s}\tau^2 + \frac{BW}{2}\tau \right]. \tag{2.9}$$

The complex envelope $c_{chirp}(\tau)$ of the raw upchirp can be described with

$$c_{chirp}(\tau) = e^{j\theta_{chirp}(\tau)} = e^{j2\pi\left[-\frac{BW}{2T_s}\tau^2 + \frac{BW}{2}\tau\right]}. \tag{2.10}$$

For a modulated chirp, a delay $\tau_{a[k]}$ is introduced:

$$\tau_{a[k]} = \frac{a[k]}{M} T_s. \tag{2.11}$$

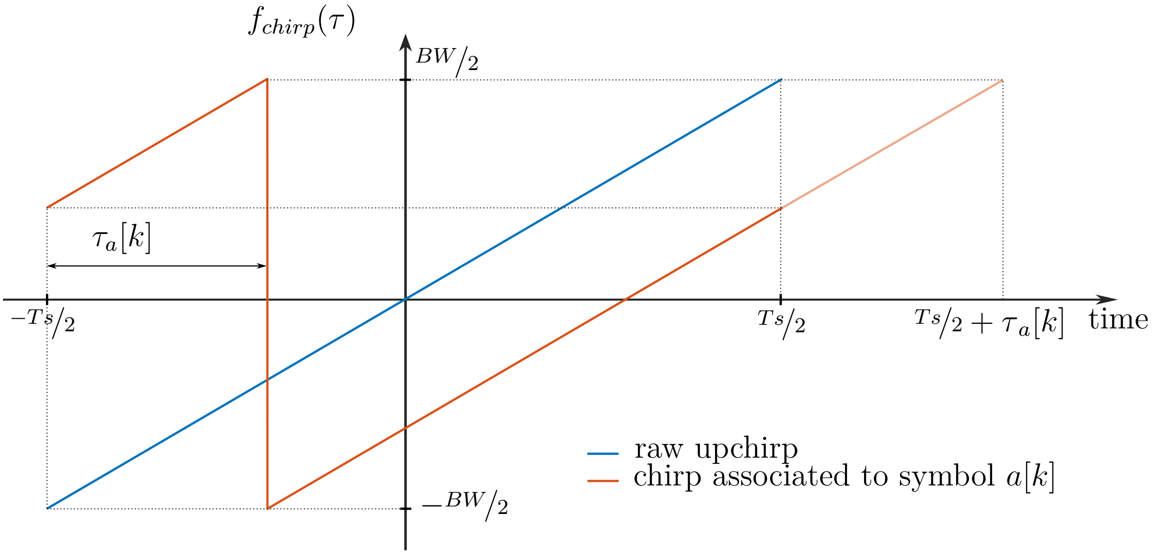


Figure 2.5: Raw and modulated chirp

Figure 2.5 shows the frequency over one symbol-period for a raw chirp and a delayed chirp, associated with the symbol $a[k]$. The modulated chirp is a cyclically shifted version of the raw chirp, where the section of the curve between $\frac{T_s}{2}$ and $\frac{T_s}{2} + \tau_k$ is shifted to the beginning of the symbol period. The frequency of a chirp associated with a symbol a[k] with $f_{start} = {}^{-BW}/2$ is described by

$$f_{chirp}(\tau, a[k]) = \left\{ -\frac{BW}{2} + \frac{BW}{T_s}\left[\left(\tau - \tau_{a[k]}\right) \bmod T_s\right]\right\}, \tag{2.12}$$

which leads after integration to an expression of the phase

$$\theta_{chirp}(\tau, a[k]) = 2\pi\left\{ -\frac{BW}{2T_s}\left[\left(\tau - \tau_{a[k]}\right) \bmod T_s\right]^2 + \frac{BW}{2}\left[\left(\tau - \tau_{a[k]}\right) \bmod T_s\right]\right\}. \tag{2.13}$$

With equation (2.10) and equation (2.13), the complex envelope $y(t)$ of the transmitted signal can be calculated with:

$$y(t) = \sum_{k\in\mathbb{N}} c_{chirp}(t - kT_s, a[k]) = \sum_{k\in\mathbb{N}} e^{j2\pi\theta_{chirp}(t-kT_s,a[k])}. \tag{2.14}$$

2.1.3 Physical Frame Format

The format of a LoRa-frame consists of up to four parts, depicted in figure 2.6. A preamble for the recognition of the frame at the receiver, an optional header which includes information about the message, the payload and an optional CRC to check the validity of the message.

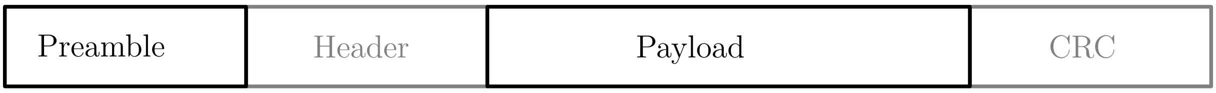


Figure 2.6: Physical Frame of a LoRa signal

Preamble

In figure 2.7, a preamble of a LoRa signal is depicted. It starts with a sequence of $N_{pre} = 6 \dots 65535$ unmodulated upchirps $c_1, c_2, \dots c_{N_{pre}}$. These chirps serve for the synchronization of the receiver to the signal.

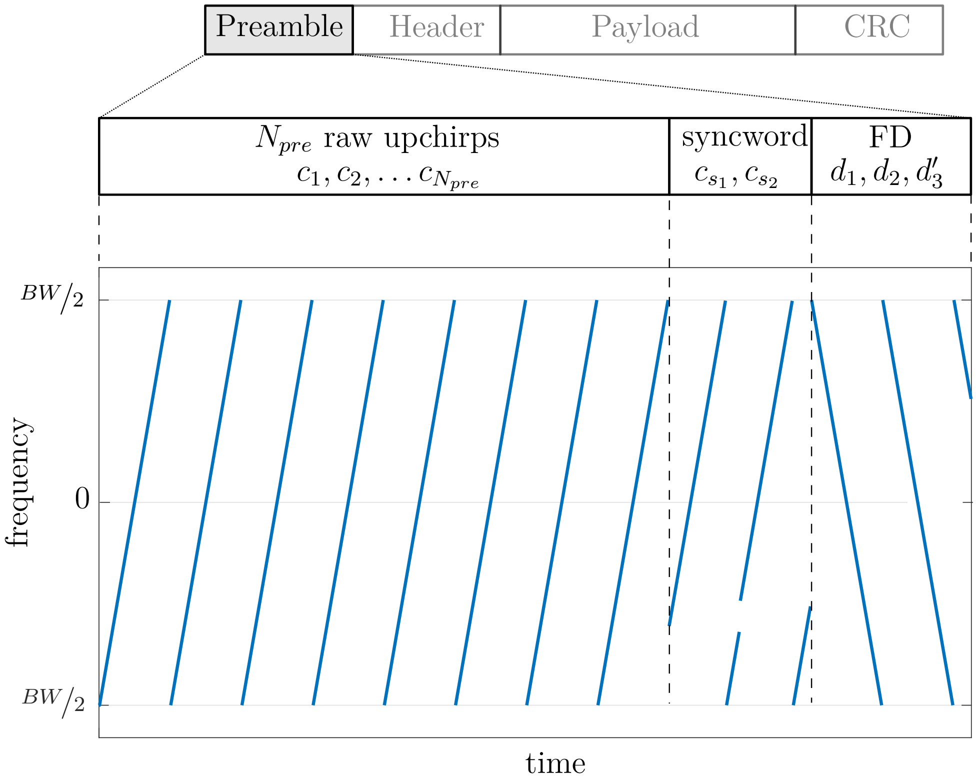


Figure 2.7: Preamble of a LoRa signal

The next two chirps c_{s_1}, c_{s_2} correspond to the syncword. The syncword is one byte long and defines the transmission type. Figure 2.8 shows how the syncword is modulated onto the two chirps c_{s_1} and c_{s_2}. For higher spreading factors, all additional bits are set to zero. The syncword defines the mode of the LoRaWAN network. Typically two different types of sync-words are common, 0x34 for public and 0x12 for private mode. Messages of a specific mode are received solely by gateways configured in the same mode.

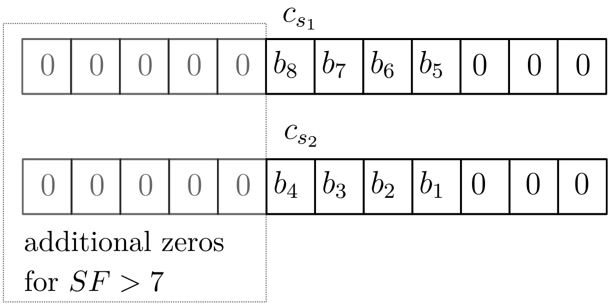

c_{s_1}
0 0 0 0 0 b_8 b_7 b_6 b_5 0 0 0
c_{s_2}
0 0 0 0 0 b_4 b_3 b_2 b_1 0 0 0
additional zeros for $SF > 7$

Figure 2.8: Modulation of a syncword

After the sync-word, the preamble ends with the Frame Delimiter (FD) which consists of 2.25 downchirps d_1, d_2, d'_3, where d'_3 corresponds to the chirp with a length of $0.25T_{chirp}$.

Header

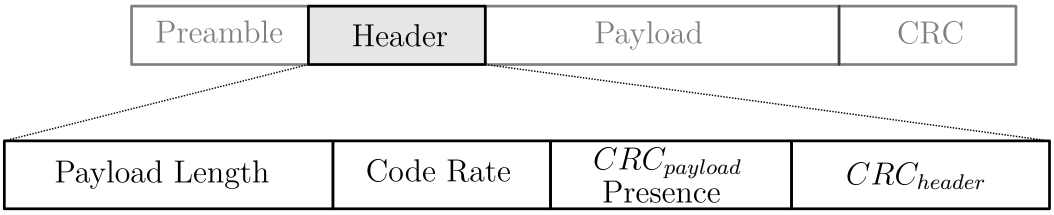


Figure 2.9: Header of a LoRa frame

The next part of a LoRa-frame is an optional header, depicted in figure 2.9. It contains information about the payload length, the code rate used for the payload, a bit indicating the presence of a payload-CRC, and ends with a header-CRC. It is always encoded with the maximum code rate of 4/8.

The option of including a header is determined by two different modes, *implicit* and *explicit* mode. In *implicit* mode, the code rate and the presence of a CRC must be configured by the transmitter and receiver beforehand, whereas in *explicit* mode, the header is included in the frame.

Payload

A LoRa frame consists either of a generic or a LoRaWAN payload. A detailed description of the LoRaWAN frame format can be found in section 2.2.1.

Time on Air

To be able to determine the Time on Air of a LoRa-message, we have to calculate the amount of symbols needed to transmit the information. The length of preamble and payload are determined by

$$N_{preamble} = (N_{pre} + 4.25) \tag{2.15}$$

and

$$N_{payload} = 8 + max\left(\left\lceil\frac{(8PL - 4SF + 28 + 16CRC - 20H)}{4N}\right\rceil\frac{4}{CR}, 0\right), \tag{2.16}$$

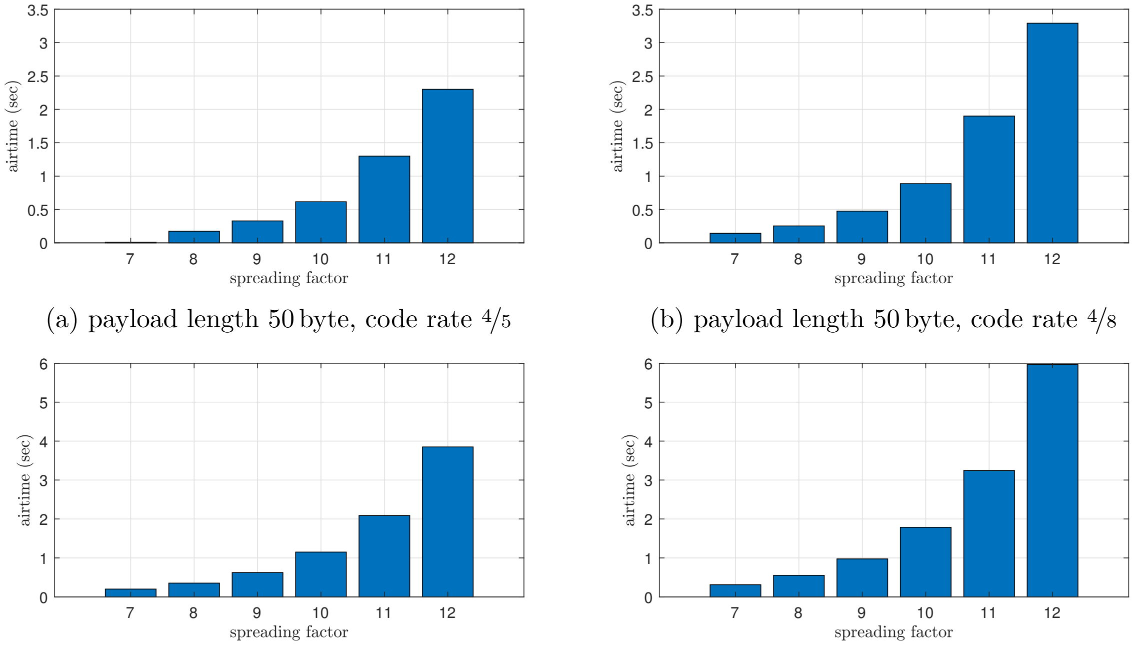
which corresponds to a total symbol length of

$$N_{LoRa} = N_{preamble} + N_{payload}. \tag{2.17}$$

The preamble length is determined by the amount of synchronization chirps N_{pre}, plus additional 4.25 chirps from sync-word and frame delimiter FD. The payload-length depends on various factors, but is at least 8 symbols long. PL is the number of payload bytes (0 . . . 255 Bytes), CRC indicates the presence of a CRC, and H represents the mode chosen, $H = 0$ for explicit and $H = 1$ for implicit mode.
With the symbol length N_{LoRa} from 2.17 and symbol duration T_s from 2.6, the total time on air of a LoRa-frame can be derived with

$$T_{LoRa} = N_{LoRa}\,T_s. \tag{2.18}$$

A comparison of signals lengths with different spreading factor, payload length, and code rate is depicted in figure 2.10. All plots show the big difference of the airtime between signals with different spreading factors. A signal with the highest code rate of 4/8 is about 70 % longer compared to signals with the lowest code rate of 4/5.



(a) payload length 50 byte, code rate 4/5

(b) payload length 50 byte, code rate 4/8

(c) payload length 255 byte, code rate 4/5

(d) payload length 255 byte, code rate 4/8

Figure 2.10: Airtime of LoRa signals with different payload lengths, code rates, and spreading factors. Preamble length 10 chirps, bandwidth 125 kHz, enabled header

2.1.4 Transmitter/Receiver Chain

This section gives an overview about all steps in the communication process. The steps of the transmitter chain are shown in figure 2.11. The raw payload-data p with a length of $L_p = 0 \dots 255$ Byte gets encoded and mapped to symbols $\underline{a} \mathrel{\widehat{=}} (a_0\ a_1 \dots a_{N_{LoRa}})$, with N_{LoRa} from equation (2.17). During the encoding steps, the header is included in the data. The symbols then get modulated onto chirps and the preamble is added to the front of the signal $s(t)$. $s(t)$ gets corrupted by the channel, which leads to the received signal $y(t)$.

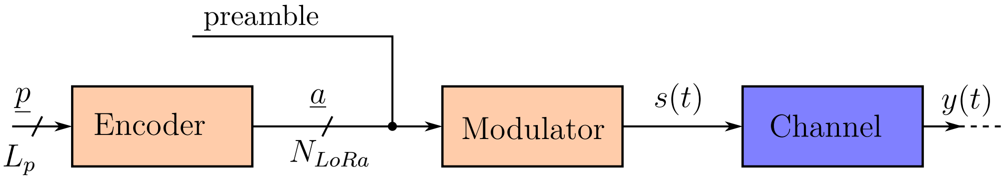


Figure 2.11: LoRa transmitter chain

The steps at the receiver side are depicted in figure 2.12. The receiver synchronizes to preamble of the signal $y(t)$ and demodulates it. If the message type, determined by the sync word included in the preamble, does not match with the type of the gateway, the packet gets emitted. The next step is the demodulation of the signal $y(t)$ to get the detected symbols $\underline{\hat{a}} \mathrel{\widehat{=}} (\hat{a}_0\ \hat{a}_1 \ldots \hat{a}_{N_{LoRa}})$.

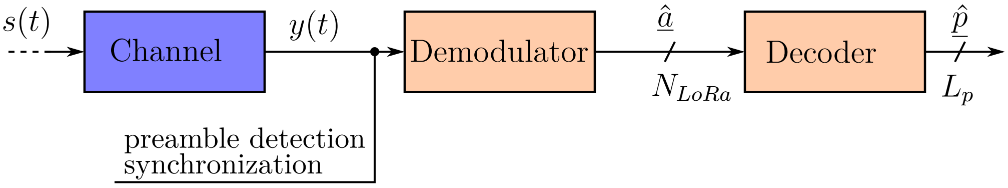


Figure 2.12: LoRa receiver chain

2.1.4.1 Encoder/Decoder

The data of a LoRa-signal is encoded to increase the resilience against interferences over the air [1] [8] [9]. Since the information contained in the header is important for the transmission, it gets a different treatment in the encoding process than the payload.

Payload

Figure 2.13 shows the encoding-steps of the payload for a LoRa-transmission. Before the data is modulated, it gets whitened, encoded, interleaved, and Gray-mapped. An additional step is the calculation of an optional 2 Byte CRC, which gets added to the whitened payload $\underline{p_w}$ and results in a length of $L_w = L_p + 2$ Byte.

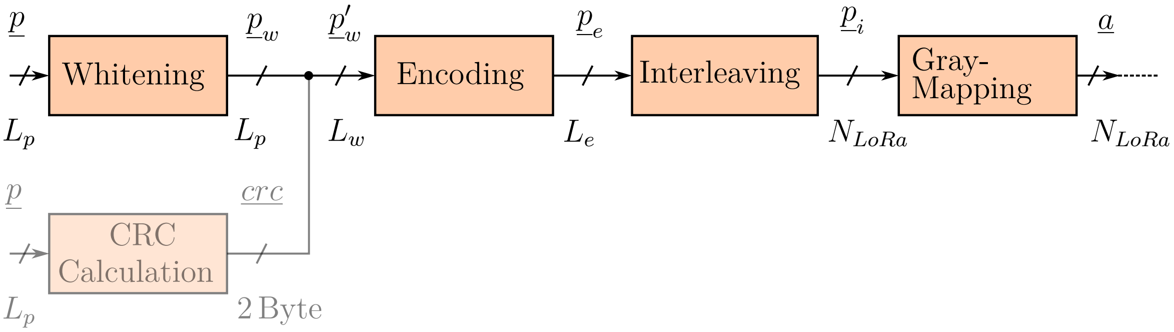


Figure 2.13: LoRa payload encoding scheme

- **Whitening:** The purpose of whitening is to remove some DC-bias in the data. This is done by an XOR operation by the data bits with a pseudo-random sequence.

$$\underline{p_w} = \underline{p} \oplus \underline{W} \tag{2.19}$$

The whitening-sequence is calculated with a linear feedback shift register (LFSR) with a starting value of 0x11. A graphical representation of the LFSR is depicted in equation (2.20). It can be described with the polynomial

$$p_w(x) = x^4 \oplus x^3 \oplus x^2 \oplus 1. \tag{2.20}$$

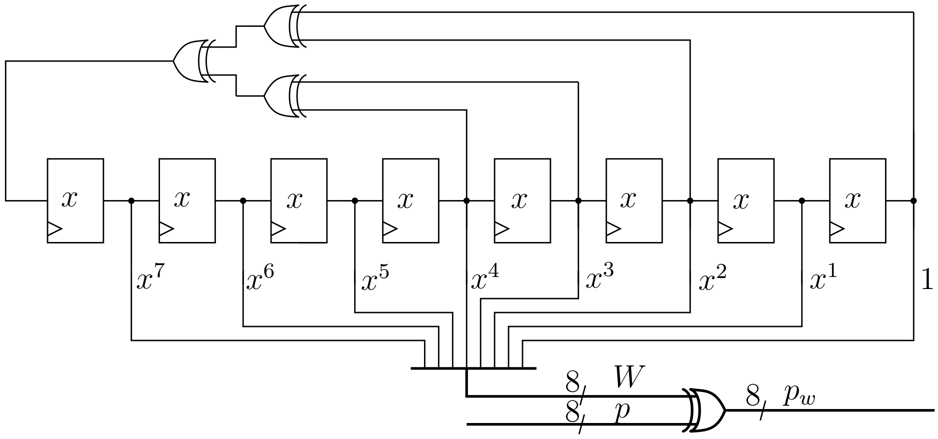


Figure 2.14: Calculation of whitening sequence with LFSR

- **Encoding:** In order to be able to correct corrupted data at the receiver, a Forward Error Correction (FEC) scheme is used. LoRa supports four code rates $4/5$, $4/6$, $4/7$, and $4/8$ represented by $CR = 1 \ldots 4$. The whitened payload is separated into 4-Bit nibbles which leads to a length of the encoded data $\underline{p_e}$ of $L_e = 2L'_p$.

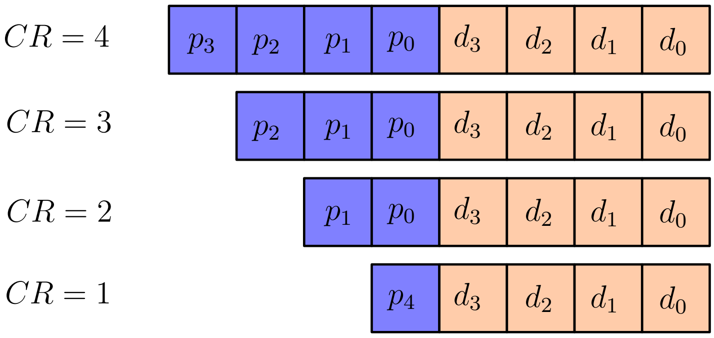

$CR = 4$
p_3 p_2 p_1 p_0 d_3 d_2 d_1 d_0
$CR = 3$
p_2 p_1 p_0 d_3 d_2 d_1 d_0
$CR = 2$
p_1 p_0 d_3 d_2 d_1 d_0
$CR = 1$
p_4 d_3 d_2 d_1 d_0

Figure 2.15: Parity bits after encoding

Figure 2.15 shows the parity bits added to the data bits for each CR. The parity bits p_0 to p_4 are calculated as follows:

$$\begin{aligned} p_0 &= d_0 \oplus d_1 \oplus d_2 \\ p_1 &= d_1 \oplus d_2 \oplus d_3 \\ p_2 &= d_0 \oplus d_1 \oplus d_3 \\ p_3 &= d_0 \oplus d_2 \oplus d_3 \\ p_4 &= d_0 \oplus d_1 \oplus d_2 \oplus d_3 \end{aligned} \tag{2.21}$$

The set of codewords with $CR = 3$ is a Hamming code (k, n) with data length $k = 4$ and codeword length $n \in \{5, 6, 7, 8\}$. The codes for $CR = 2$ and $CR = 4$ are shortened and respectively extended versions of the $(4, 7)$ Hamming code. The single parity bit p_4 for $CR = 1$ is a checksum of the data bits d_0 to d_3. With $CR = 1$ and $CR = 2$, errors can only detected at the receiver, while a codeword with $CR = 3$ and $CR = 4$ can be corrected if one bit error occurs.

- **Interleaving:** The signal is corrupted by noise, fading, and other interferences, which leads to symbol and bit errors. Multiple bit errors, caused by one symbol are highly correlated. In the LoRa receiver/transmitter-chain, a diagonal interleaver is used to spread the biterrors over multiple codewords to decorrelate them. The scheme of a diagonal interleaver for $SF = 7$ and a code rate of $^4/_5$ is depicted in figure 2.16. A block of SF codewords with a dimension of $SF \times (CR + 4)$ is transformed into a block with a dimension of $(CR + 4) \times SF$. The vector of codewords $\underline{p_e}$ have a length L_e which is a multiple of SF, for the interleaver to process the whole data. If this is not the case, random codewords get added to the vector $\underline{p_e}$ to achieve this. The resulting length of $\underline{p_i}$ is equal to the symbol length N_{LoRa} from equation (2.16).

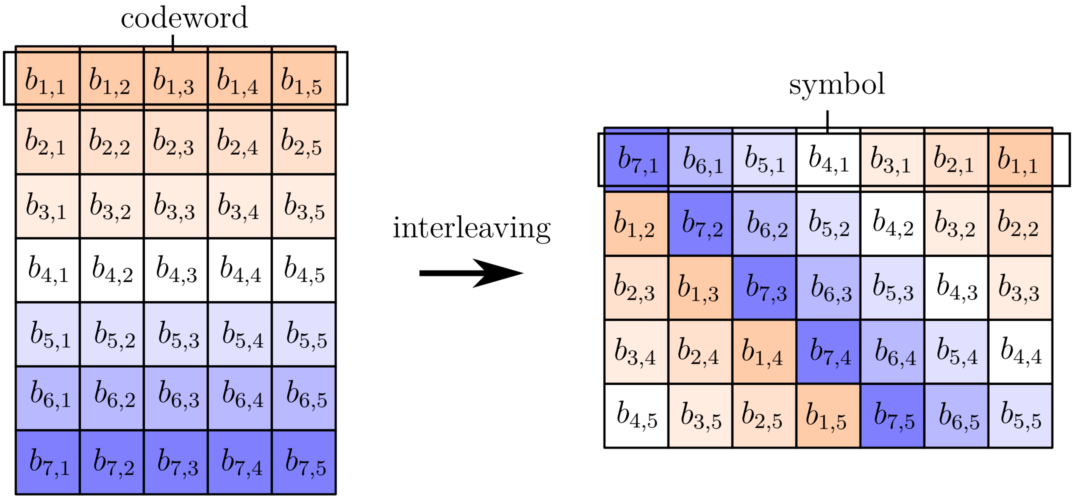

codeword
$b_{1,1}$ $b_{1,2}$ $b_{1,3}$ $b_{1,4}$ $b_{1,5}$
$b_{2,1}$ $b_{2,2}$ $b_{2,3}$ $b_{2,4}$ $b_{2,5}$
$b_{3,1}$ $b_{3,2}$ $b_{3,3}$ $b_{3,4}$ $b_{3,5}$
$b_{4,1}$ $b_{4,2}$ $b_{4,3}$ $b_{4,4}$ $b_{4,5}$
$b_{5,1}$ $b_{5,2}$ $b_{5,3}$ $b_{5,4}$ $b_{5,5}$
$b_{6,1}$ $b_{6,2}$ $b_{6,3}$ $b_{6,4}$ $b_{6,5}$
$b_{7,1}$ $b_{7,2}$ $b_{7,3}$ $b_{7,4}$ $b_{7,5}$
interleaving
symbol
$b_{7,1}$ $b_{6,1}$ $b_{5,1}$ $b_{4,1}$ $b_{3,1}$ $b_{2,1}$ $b_{1,1}$
$b_{1,2}$ $b_{7,2}$ $b_{6,2}$ $b_{5,2}$ $b_{4,2}$ $b_{3,2}$ $b_{2,2}$
$b_{2,3}$ $b_{1,3}$ $b_{7,3}$ $b_{6,3}$ $b_{5,3}$ $b_{4,3}$ $b_{3,3}$
$b_{3,4}$ $b_{2,4}$ $b_{1,4}$ $b_{7,4}$ $b_{6,4}$ $b_{5,4}$ $b_{4,4}$
$b_{4,5}$ $b_{3,5}$ $b_{2,5}$ $b_{1,5}$ $b_{7,5}$ $b_{6,5}$ $b_{5,5}$

Figure 2.16: Interleaving of a block of bits with $SF = 7$ and code rate $4/5$

- **Gray Mapping:** Interferences during the transmission, or an inaccurate synchronization at the receiver end can lead to corrupted signals. When an error occurs, it is most likely that the receiver mistakes a symbol for one of its adjacent symbols. The Gray code maps a bit sequence to the symbol, so that two successive values differ by just one bit. With a code rate of $4/7$ or $4/8$, such an error can always be corrected. LoRa uses the standard Gray code which is calculated with

$$c_i = \begin{cases} d_i & \text{for } i = 0 \\ d_i \oplus c_{i-1} & \text{else} \end{cases} \tag{2.22}$$

and the respective decoding at the receiver

$$d_i = \begin{cases} c_i & \text{for } i = 0. \\ c_i \oplus c_{i-1} & \text{else.} \end{cases} \tag{2.23}$$

Header

Since the header contains essential information of the message, it gets treated with a different encoding process, as depicted in figure 2.9. The header does not get whitened and is independent of the SF, it is always represented by the first eight symbols after the preamble. The header symbols are always transmitted in low data rate mode, while leaving the symbol duration at $T_s = 2^{SF}/BW$. It also gets encoded with the highest code rate of $^4\!/_8$ independently of the rest of the signal to ensure the best error correction capabilities at the receiver. The data contained in the header has a length of 20 bits, which increases to 40 bits after encoding. Any other bits contained in the first eight symbols belong to the payload, which leads to a length of $L_h = \mathrm{SF} - 2$ for the header $\underline{h}$.

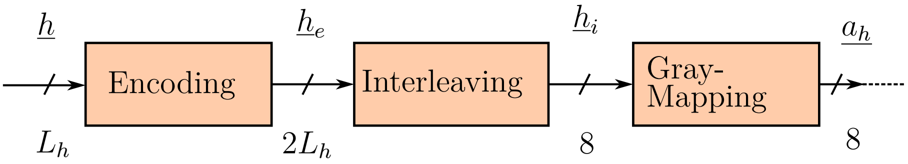


Figure 2.17: LoRa header encoding scheme

Example: *Calculation of the symbols for a LoRa signal with the parameters*

- $SF = 8$
- $CR = {}^4\!/_5$
- $L_p = 2\mathrm{Byte}$
- $\underline{p} = [0\,x\,00\,00]$
- CRC deactivated

with "$0\,x \ldots$" denoting hexadecimal value and "$0\,b \ldots$" denoting binary values.

Whitening: With the a whitening sequence $\underline{W} = [0\,x\,\mathrm{FF\,FE}]$*with length* L_p*, we get the whitened payload*

$$\underline{p_w} = \underline{p} \oplus \underline{w} = \left[0\,x\,\mathrm{FF\,FE}\right]$$

Encoding: Before the data is encoded, we have to determine the header data as described in section 2.1.3:

- Payload Length = $[0b\,0000\,0010]$
- Code Rate = $[0b\,001]$
- CRC disabled = $[0b\,0]$
- Header CRC = $[0b\,0000\,1110]$

With $\mathrm{SF} = 8$*, we get a length of* $L_h = 8 - 2 = 6\mathrm{Byte}$ *for the header. In this case, there is space for one Byte of the payload in the header, and therefore we get*

$$\underline{h} = [0\,x\,02\,20\,\mathrm{EF}]$$

and

$$\underline{p'_w} = [0\,x\,\mathrm{FEF}]$$

After encoding the header with code rate 4/8 and the payload with 4/5, we get

$$\underline{h_e} = [0\,x\,00\,27\,27\,00\,\mathrm{E2}\,\mathrm{FF}]$$

and

$$\underline{p_e} = [0\,x\,\mathrm{F0}\,\mathrm{F1}\,\mathrm{F0}]$$

Interleaving: In order to be able to interleave the payload, we have to fill the payload with random data, to get full blocks of bits with dimension SF × $4\frac{1}{CR}$*, in our case* $8x5$*. The results after interleaving are*

$$\underline{h_i} = [0\,x\,01\,36\,\mathrm{0C}\,18\,16\,\mathrm{2D}\,19\,02]$$

and

$$\underline{p_i} = [0\,x\,\mathrm{A0}\,\mathrm{C1}\,38\,07\,04]$$

Gray Mapping: In the last step, the data gets mapped with a Gray code, which leads to the header symbols

$$\underline{a_h} = [0\,x\,\mathrm{3F}\,12\,04\,08\,32\,\mathrm{1B}\,39\,\mathrm{3E}]$$

and the payload symbols

$$\underline{a_p} = [0\,x\,60\,\mathrm{DF}\,81\,\mathrm{FD}\,\mathrm{FC}]$$

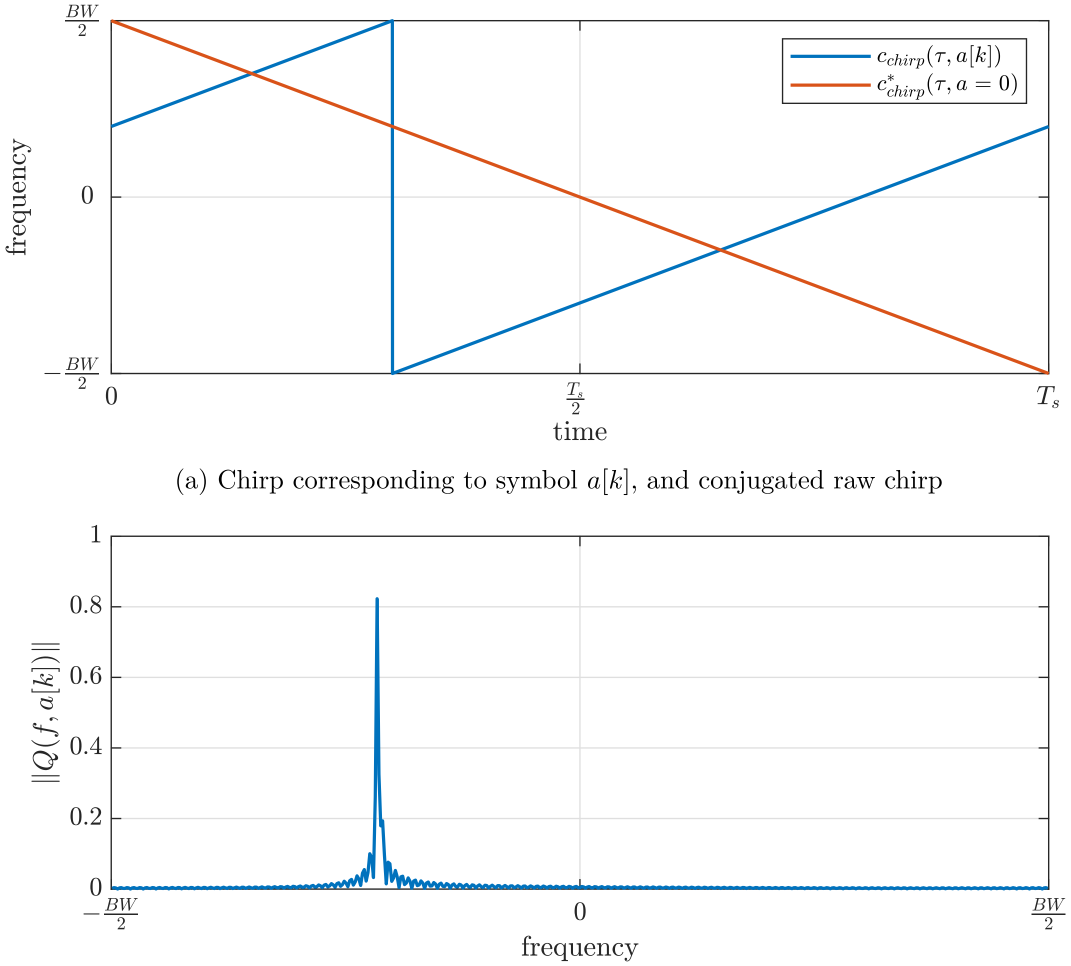
2.1.4.2 Modulator/Demodulator

The simulation of a modulator is a relatively easy task. By feeding equation (2.14) with the symbols $\underline{a}$ from the output of the encoder we get the modulated data.
There are several ways to demodulate and extract the information out of a LoRa chirp. The most convenient way is to multiply the chirp with a conjugate version of a raw chirp and perform a fast Fourier transformation of the resulting signal. Figure 2.18a shows a chirp corresponding to the symbol $a[k]$ and a conjugate raw chirp over one symbol period. The two signals get multiplied, which results in

$$q(\tau, a[k]) = c_{chirp}(\tau, a[k]) \cdot c^*_{chirp}(\tau, a = 0). \tag{2.24}$$

After performing a fast Fourier transformation, we get

$$Q(\tau, a[k]) = FFT\Big[q(\tau, a[k])\Big]. \tag{2.25}$$



(a) Chirp corresponding to symbol $a[k]$, and conjugated raw chirp

(b) Spectrum $Q(\tau, a[k])$

Figure 2.18: Demodulation process of a chirp with $SF = 9$ and bandwidth BW

An example of a spectrum $Q(\tau, a[k])$ for the chirp $c_{chirp}(\tau, a[k])$ in figure 2.18a is depicted in figure 2.18b. The result shows a peak at a specific frequency, which is linearly proportional to the delay $\tau_{a[k]}$ from equation (2.11) and consequently the symbol $a[k]$. The spectrum is divided into $M = 2^N$ parts, representing the symbols $a[k] \in \{0, \dots, M-1\}$.
The resolution, described by the ratio of symbol alphabet size to symbol duration, is relatively high. Therefore, good synchronization is essential to build a working demodulator.

We succeeded in building a demodulator with relatively low effort that works with signals in low data rate mode. This fulfilled our needs for this project, since the focus was on generating signals at the transmitter end, which works perfectly for all different settings of the signal. The resolution is decreased by the factor four, which allows the demodulator to work with imperfect synchronization.

2.2 LoRaWAN

The Long Range Wide Area Network (LoRaWAN) protocol is specially designed by the LoRa Alliance for the use on top of LoRa. The implementation of the network is based on a star topology. This approach reduces the complexity of the system and the energy consumption of the end-devices compared to a mesh network [3]. LoRaWAN is the network layer in the communication system, depicted in figure 2.19. It operates in region specific frequency bands using LoRa modulation as a physical layer. LoRaWAN offers different options for uplink and downlink management, defined by the three different classes A, B, and C, further described in section 2.2.2.

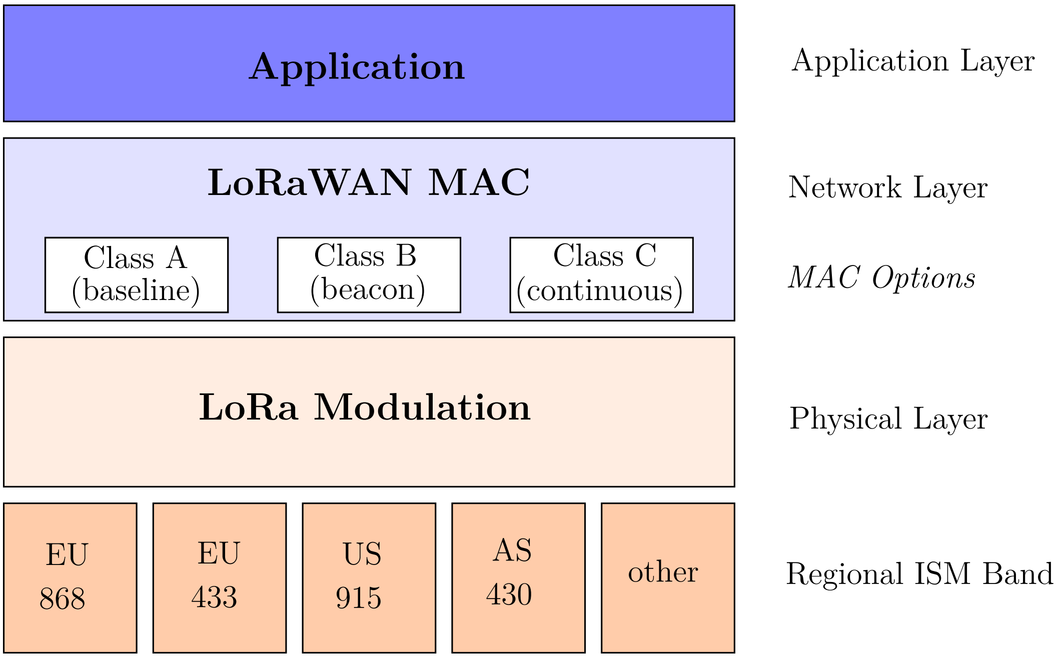


Figure 2.19: Layers of LoRa [3]

The architecture of a LoRaWAN-network is depicted in figure 2.20. Messages sent by an end-node get received by all gateways in its range. The received frames then get forwarded to the network server, where duplicate messages are filtered. Afterwards, the network server sends the frame to the application server where they are processed further.

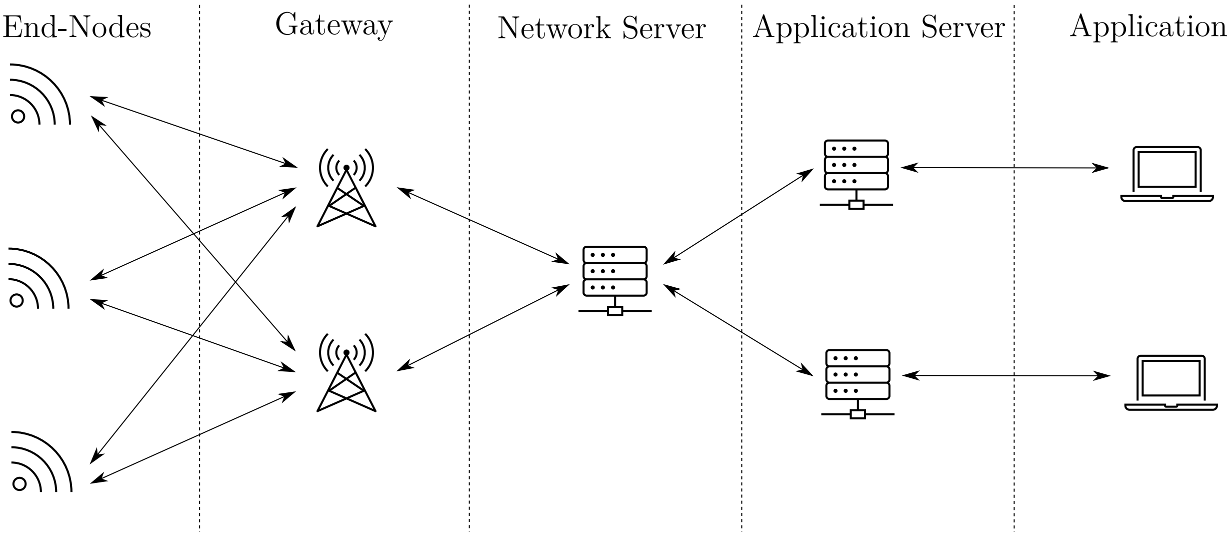


Figure 2.20: Architecture of a LoRaWAN-system

The frequency band at which LoRaWAN operates differs geographically. In Europe, the two ISM bands at 433 MHz or 868 MHz are allocated for LoRa communication, whereas the 433 MHz band is not in use yet. No license fee is required and everyone is allowed to use it, with the drawback of limited transmission power and a resulting low data rate regulated by the ETSI [EN300.220] standard. In the European ISM bands, the maximum transmission power is 14 dBm. Figure 2.21 shows the channel plan of the 868 MHz band. Three mandatory channels at the frequency carriers of 868.1MHz, 868.3MHz, and 868.5MHz have to be implemented by every device with the specifications listed in figure 2.21 [2]. Further channels can be activated with the frequency channel list CFlist exchanged between gateway and end device, further described in section 2.2.3.
The ETSI regulation allows a transmission with a duty-cycle of maximum 1% in the 868 MHz ISM band. This requirement can be bypassed by implementing the so called Listen before Talk Adaptive Frequency Agility (LBT AFG) transmission management. In this case, the device senses a channel to determine if there is activity by measuring the RSSI for five seconds. If the value is below a certain threshold, the device starts to transmit on this channel.

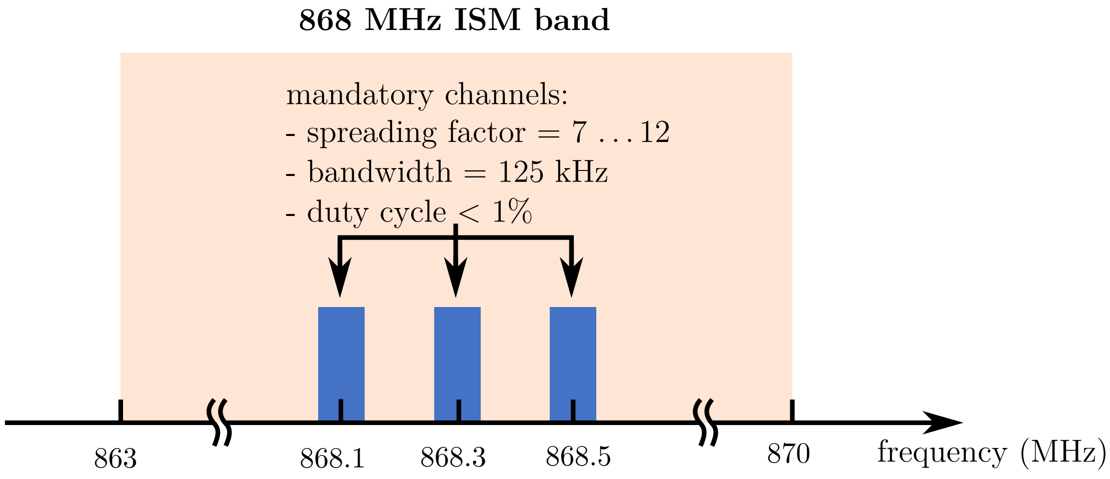


Figure 2.21: Frequency channels in European 868 MHz ISM band

To optimize the energy consumption of an end-device, LoRaWAN offers the option of an adaptive data rate (ADR). With ADR activated, the network analyzes the reception statistics of messages for each node individually and adapts data rate, spreading factor, frequency channel, and output power, to ensure the best battery saving operation of the device.
There are end-devices for several applications, which have different requirements. In order to optimize the individual needs, the LoRaWAN network layer offers three different device classes.

2.2.1 LoRaWAN Frame Format

The structure of a LoRaWAN frame, depicted in Figure 2.22, is defined in the LoRaWAN specification by the LoRa Alliance [3].

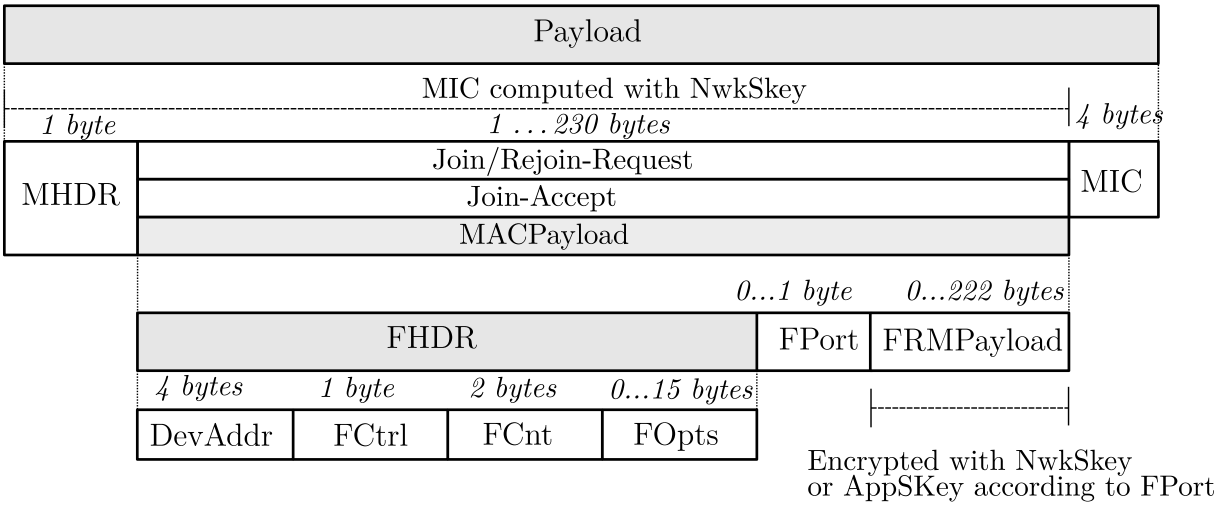


Figure 2.22: Structure of the payload

The payload starts with a 1-byte MAC-header MHDR. It consists of a 3-bit MType-field specifying the message type of the frame, a 2-bit major field, specifying the messages exchanged in the join procedure and 3 bits for future use. A list of the different options for the message type is listed in table 2.2.

Table 2.2: Message Formats

MType	**Description**
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Up
100	Confirmed Data Up
101	Confirmed Data Up
110	RFU
111	Proprietary

The join process and the structure of these messages are described in section 2.2.3. In case of a data message, a detailed description is depicted in figure 2.9. The payload of a data message (*MACPayload*) starts with the frame-header *FHDR*. It consists of the device address *DevAddr* of the end-device followed by a frame control byte *FCtrl*. The *FCtrl* stores information about the adaptive data rate, message acknowledgments for confirmed data messages and pending data to be sent at the downlink communication. The payload ends with a frame counter *FCnt* and a frame options field *FOpts*, where MAC commands

are transported. The frame counter keeps track of the amount of messages transmitted by the end-device. There are two different counters transmitted using this field, the uplink counter FCntUp and downlink counter FCntDown.
The frame port value *FPort* contains information about the actual payload *FRMPayload*. If it is set to zero, the payload contains MAC commands only. Otherwise, *FPort* can be used application-specific. The *FRMPayload* contains the actual payload or MAC-commands.

2.2.2 Device Classes

LoRaWAN adopts an ALOHA-type random access, which leads to an energy efficient communication and keeps the network complexity low. The specification defines three classes for different power consumption strategies.

2.2.2.1 Class A

Class A is the basic transmission strategy which has to be implemented by all LoRaWAN nodes. All transactions start with an uplink transmission by the node based on its own needs. After the transmission, the node opens two receiving slots for the reception of messages by the gateway. By default, the first receive window RX1 uses the same channel as the uplink and a spreading factor depending on the offset of the spreading factor between uplink and the first downlink [3]. As a standard, the downlink spreading factor is set to $SF = 12$. The second receive window RX2 uses a fixed channel and spreading factor. If a frame was detected by the node during the first slot, and this frame was intended for this specific end-device, the node does not open the second receive window. Class A allows for bi-directional communication, which is initiated by the end-device. A downlink transmission is just possible after a successful uplink transmission. This strategy shows the lowest power consumption. Applications for class A communication are, for example, battery powered sensors requiring downlink only shortly after uplink or no downlink at all.

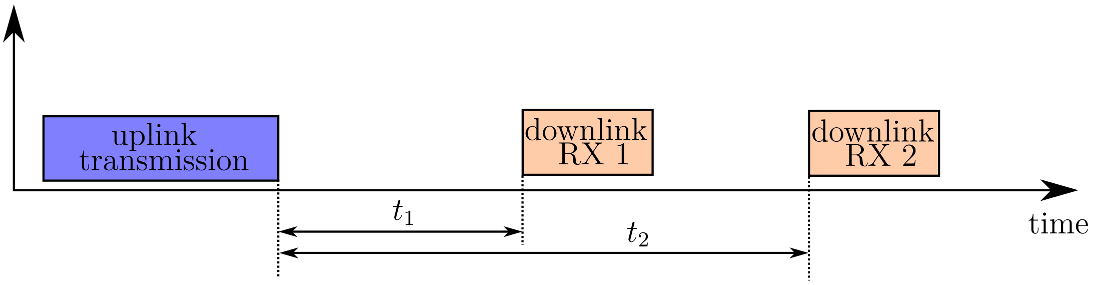


Figure 2.23: Class A Communication

2.2.2.2 Class B

Class B upgrades the basic class A communication by adding additional synchronized reception slots. The gateway sends a broadcast message (beacon) in a specific time interval (beacon period) as a timing reference. Between the beacons, the beacon window defines the time, when the end-device opens reception slots in a fixed interval (ping period). Downlink is possible only during the ping slots, while a class A uplink, including the two delayed downlink slots, is possible at any time. Due to the extra receive windows, class B communication is less energy efficient than class A communication but has a lower latency. This strategy is suited for applications like battery powered actuators or sensors, where downlink is needed regularly.

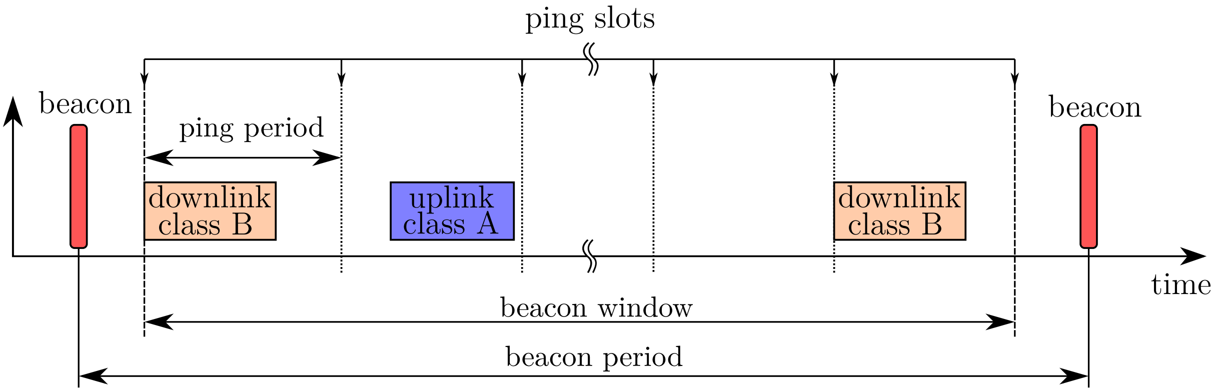


Figure 2.24: Class B Communication

2.2.2.3 Class C

A class C device has nearly continuously open reception windows. Only during an uplink transmission, these windows are closed and the class A standard has to be supported. Class C has a high power consumption compared to the other classes, but has the advantage of no latency. This makes it suitable for all kinds of applications with a power source available.

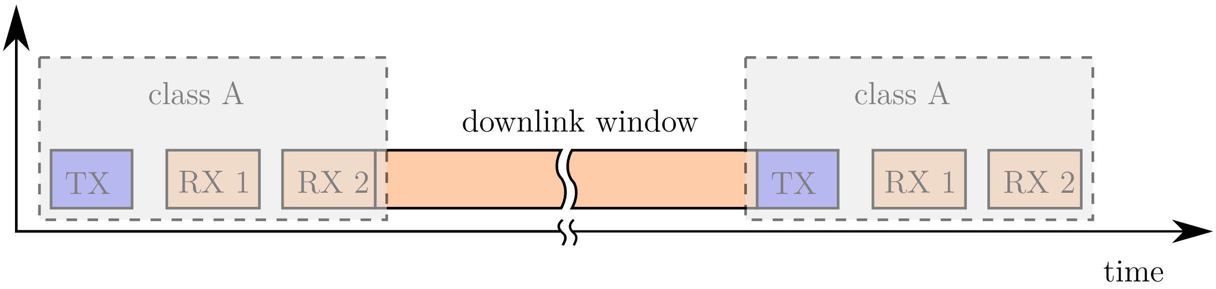


Figure 2.25: Class C Communication

Figure 2.26 shows a comparison of the three different classes available.

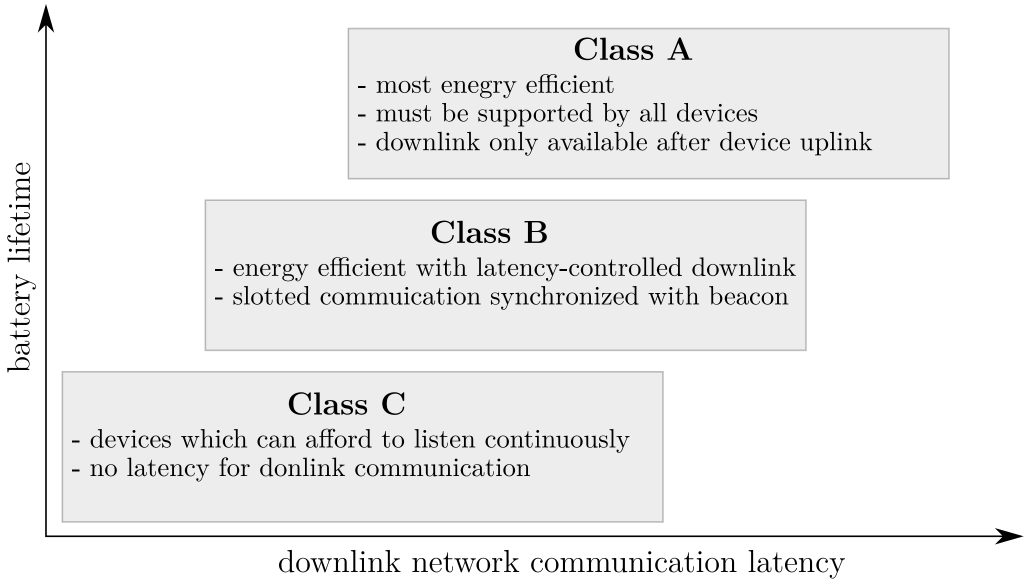

Class A
- most enegry efficient
- must be supported by all devices
- downlink only available after device uplink
Class B
- energy efficient with latency-controlled downlink
- slotted commuication synchronized with beacon
Class C
- devices which can afford to listen continuously
- no latency for donlink communication
battery lifetime
downlink network communication latency

Figure 2.26: Classes

2.2.3 Activation of an End-Device

In LoRaWAN-network, each end-device needs to join the network by performing a join process. There are two different procedures for this activation, Over the Air Activation (*OTAA*) and Activation by Personalization (*ABP*). This section gives a brief introduction into both activation types which are defined in the LoRaWAN specification [3] and LoRaWAN Backend Interfaces Specification [14] by the LoRa Alliance.

2.2.3.1 Security in LoRaWAN

LoRaWAN applies a two-layer encryption to the data being transmitted, as depicted in figure 2.27. The network connection between end-device and network server is secured by the Network Session Key (NwkSKey) and the communication at application level between end-device and application server is secured by the Application Session Key (AppSKey).

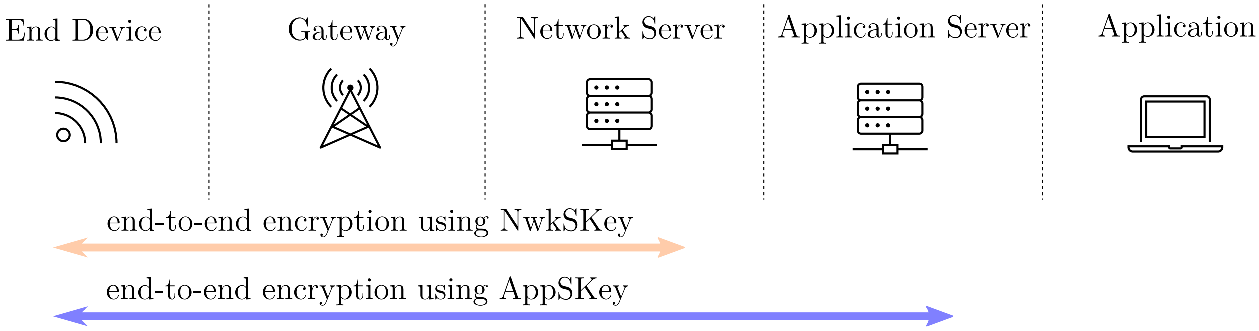

End Device
Gateway
Network Server
Application Server
Application
end-to-end encryption using NwkSKey
end-to-end encryption using AppSKey

Figure 2.27: Security in a LoRaWAN system

2.2.3.2 Over the Air Activation (*OTAA*)

OTAA is a procedure, where end-device and network server NS authenticate each other mutually and exchange keys for the communication. The process is depicted in figure 2.28 and includes 4 steps.

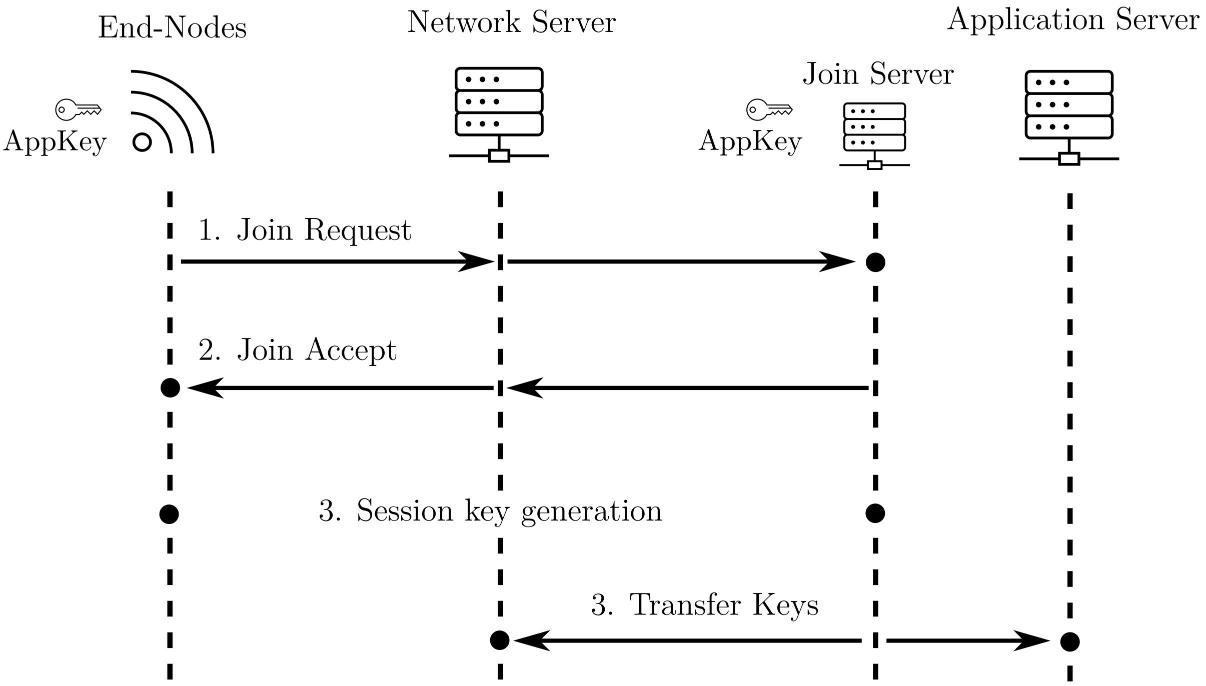
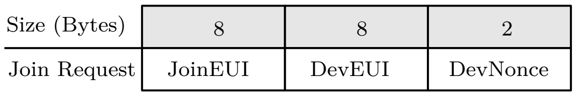


Figure 2.28: OTAA process

1. **Join Request Message:** The join procedure starts with a request message by the end-device to the Join Server (JS) via Network Server (NS). There is no encryption used and the transmission is possible with any spreading factor at the mandatory channles, depicted in figure 2.21. The request message is depicted in figure 2.29 and includes the global end-device identifier DevEUI, a join identifier JoinEUI, and the DevNonce. The DevNonce is a counter starting at zero and incrementing with every join request.[1] The NS keeps track of the values for an end-device and ignores any Join Requests with the ones already used. The message integrity code MIC is calculated and validated by the NS. The Join-request is not encrypted and can be transmitted using any spreading factor and bandwidth. The NS determines if it is the home NS of the end-device. If this is the case, the NS looks up the IP address corresponding JS, based on its JoinEUI. The JS manages the entire OTAA process and has stored the AppKey of the device, required for the derivation of the NwkSKey and AppSKey. The NS forwards the request to the JS, including some downlink parameters and an optional channel frequency list CFlist defined in the

[1] The DevNonce is a counter according to the LoRaWAN Specification 1.1 [3]. In previous versions (LoRaWAN specfication 1.0.3 or lower), the DevNonce was a random number, changing its value after every request.

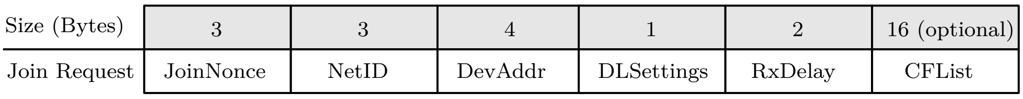
regional parameters [2] by the LoRa Alliance.



Size (Bytes)	8	8	2
Join Request	JoinEUI	DevEUI	DevNonce

Figure 2.29: Join Request message

2. **Join Accept Message:** In the next step, the JS validates the request and returns a Join Accept message back to the end-device via the NS. There is no response if the end-device is not accepted. The Join Accept message is depicted in figure 2.30 and contains information essential for the end-device to derive theNwkSKey and AppSKey. Included are two identifiers, the JoinNonce for the current join process and the NetID of the network the device is trying to join. It also contains the network address of the device (DevAddr), settings for downlink messages (DLSettings), the delay between up- and downlink messages (RxDelay), and network parameters and frequency channels (CFlist).



Size (Bytes)	3	3	4	1	2	16 (optional)
Join Request	JoinNonce	NetID	DevAddr	DLSettings	RxDelay	CFList

Figure 2.30: Join Accept message

3. **Session Key Generation:** The next step in the OTAA procedure is the derivation of the session keys NwkSKey and AppSKey by the end-device and join server with the information contained in the join request, join accept, and the AppKey. The keys are derived by encrypting with the 128 bit advanced encryption standard (AES) as follows:

   - NwkSKey = encrypt(AppKey, 0x01 | JoinNonce | NetID | DevNonce | pad16)
   - AppSKey = encrypt(AppKey, 0x02 | JoinNonce | NetID | DevNonce | pad16)

   Pad16 extends the exchanged data wit octets of zeros so that the length is a multiple of 16.

4. **Transfer Keys:** After the end-device has been accepted, the JS transmits the AppSKey to the AS and the NwkSKey to the NS. After completing the last step, a secure communication as depicted in figure 2.27 is established.

2.2.3.3 Activation by Personalization (*ABP*)

In ABP, all information needed for a secured communication is known by end-device, network server and application server beforehand. The DevAddr, NwkSKey, and AppSKey are stored by end-device, NS, and AS. The device is activated after the first uplink message is transmitted.

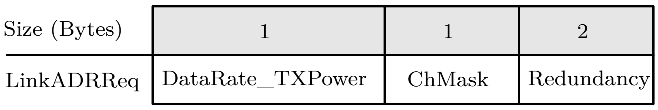
Since there is no join procedure, network settings have to be exchanged during payload transmission. The channel list, delivered with OTAA, has to be exchanged beforehand, or by requesting new channels during the first messages. It also has to be ensured that NwkSKey and AppSKey are unique to secure the communication. Another disadvantage comes with the frame counter, which increments its value after every transmission. By restarting the end device, the counter is set to zero and messages are neglected by the gateway until the device is re-registered in the backend. OTAA bypasses this problem by performing a rejoin procedure. Network settings specified during the join procedure in OTAA have to be defined with the first up- and downlink messages.

2.2.4 Adaptive Data Rate

LoRaWAN protocol defines the Adaptive Data Rate (ADR) scheme to control the uplink transmission parameters of LoRa devices.

- Data Rate
- Bandwidth
- Transmission Power

Adaptive data rate should only be used in stable RF situations where end devices do not move. The ADR settings are transmitted as MAC commands either piggybacked in the frame header or as a separate message in the *FRMPayload*. Requests by the network server to the end-device to change data rate, bandwidth, or transmission power are made by the *LinkADRReq* command. The structure of the request is depicted in figure 2.31. The *DataRate_TXPower* byte contains information about the maximum transmission power and data rate the end-device is allowed to use. The channel mask *ChMask* encodes the channels usable for uplink access.



Size (Bytes)	1	1	2
LinkADRReq	DataRate_TXPower	ChMask	Redundancy

Figure 2.31: *LinkADRReq* command

If an ADR request is received by the end-device, it answers with an LinkADRAns command, depicted in figure 2.32. Bit representing the acknowledged parameters power, data rate, and channel mask are ste to one.

Bit Nr.	7 . . . 3	2	1	0
LinkADRAns	RFU	Power ACK	Data Rate ACK	Channel Mask ACK

Figure 2.32: *LinkADRAns* command

A complete progress of an ADR message flow is depicted in figure 2.33. The receiver checks the quality of messages by an end device by collecting data from the n most recent uplink messages. Based on the signal strength RSSI and the signal to noise ratio SNR, the network server determines the minimum data rate and link budget that can be supported by the end device. If needed, the network server send a request command with the determined parameters to be changed. The end device sends back a *LinkADRAns* command with acknowledgments of the new settings.

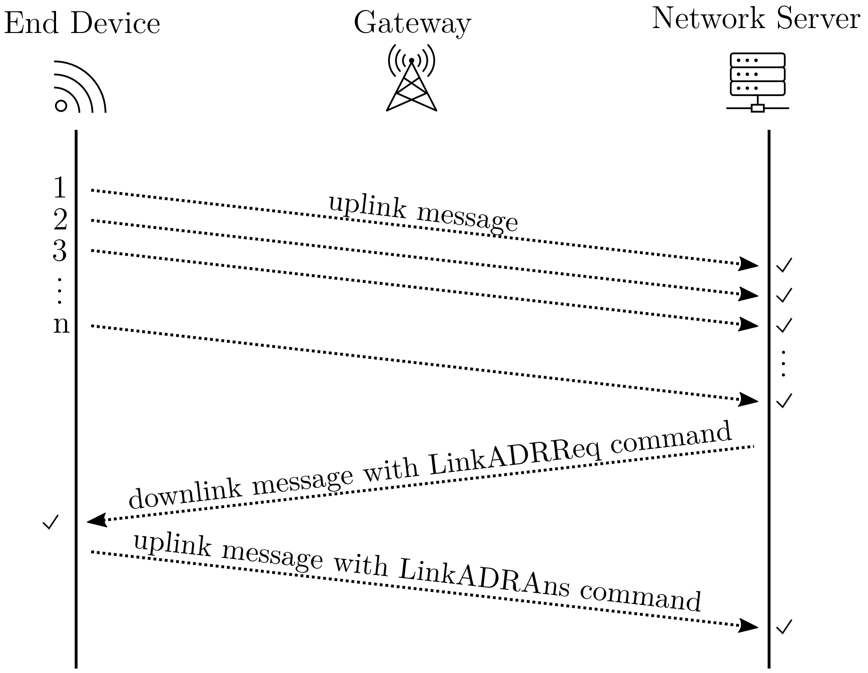
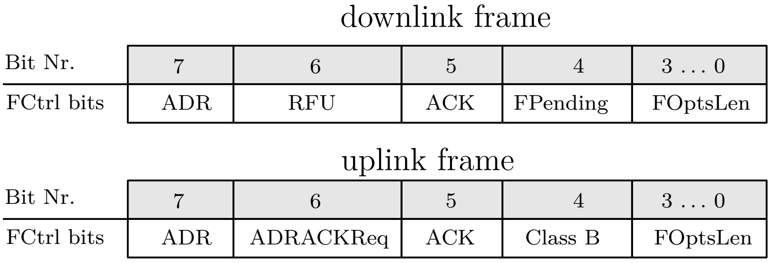

End Device
Gateway
Network Server
1
2
3
⋮
n
uplink message
✓
✓
✓
⋮
✓
downlink message with LinkADRReq command
✓
uplink message with LinkADRAns command
✓

Figure 2.33: ADR message flow

Whether ADR functionality will be used is requested by the end device by setting the ADR bit to one in the frame control *FCtrl* part of the frame header, depicted in figure 2.34.

If the bit is set to one, the network server can control the end devices transmission parameters.



downlink frame

Bit Nr.	7	6	5	4	3 . . . 0
FCtrl bits	ADR	RFU	ACK	FPending	FOptsLen

uplink frame

Bit Nr.	7	6	5	4	3 . . . 0
FCtrl bits	ADR	ADRACKReq	ACK	Class B	FOptsLen

Figure 2.34: Frame control *FCtrl* command

If the link between end device and gateway gets lost, the end device runs a specific procedure to regain the communication. The message flow of this procedure is depicted in figure 2.34. Each time the uplink frame counter is incremented, the ADR acknowledgment counter *ADR_ACK_CNT* is incremented as well. After *ADR_ACK_LIMIT* uplink messages without any downlink response, the ADR acknowledgment request *ADRACKReq* bit in the frame control *FCtrl* is set to one. The network is required to respond with a downlink frame within the next *ADR_ACK_DELAY* frames. Any downlink response indicates that the gateway still receives messages by the gateway without the need to set the *ACK* bit in the frame control. If no reply is received within *ADR_ACK_LIMIT* + *ADR_ACK_DELAY* uplink messages, the end device switches to the next lowest data rate to regain connectivity. If still no reply is received after another *ADR_ACK_DELAY* uplink messages, the data rate is changed again. If a downlink message is received, the *ADRACKReq* bit and the *ADR_ACK_CNT* counter are set back to zero.

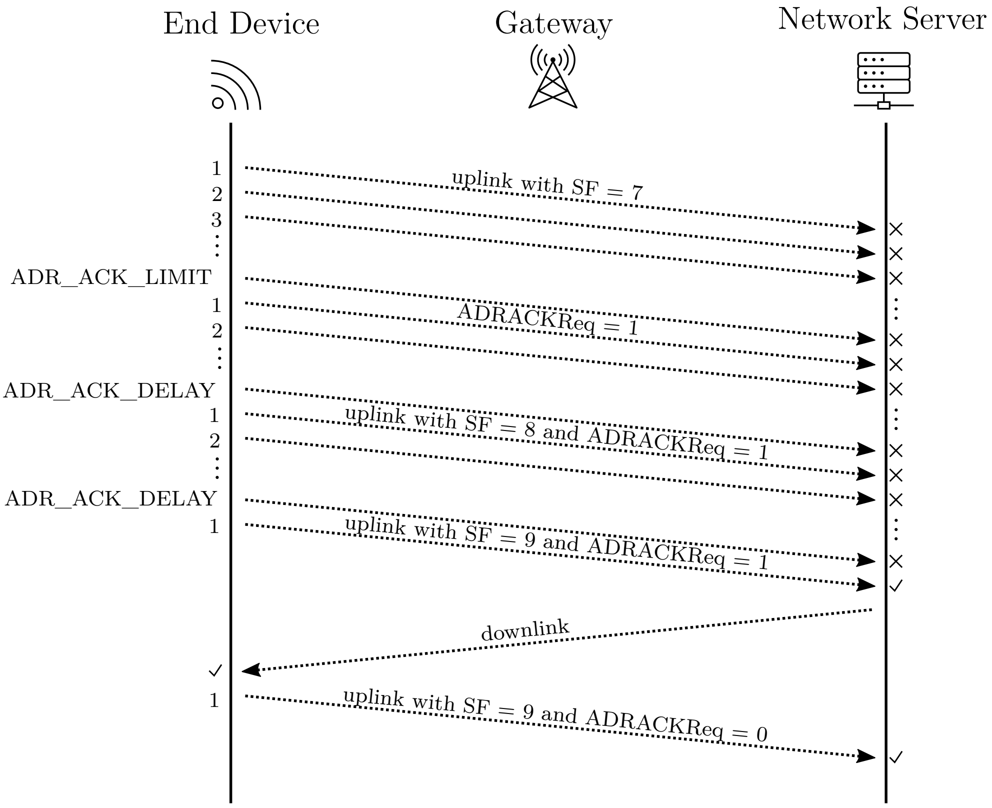


Figure 2.35: ADR message flow in case of a lost link between end device and gateway

Chapter 3

LoRa Test Setup

In order to test the limits of LoRa, we built and simulated a communication system, which allows us to perform some measurements.
Section 3.1 gives an overview about the components we used in our setup, including the gateway, the end-devices, and the whole protocol stack. Many different options are available on the market, including some prebuilt LoRa gateways or seperate parts to construct one on top of a Raspberry Pi for example. Subsequently, section 3.1.1 shows some measurements with the system without any interferences. The last chapter described the communication system which built and already tested the sensitivity with no interference disturbing the transmission. In order to validate the results of measurements we perform, we simulated a LoRa communication system in Matlab. To do this, we have to understand the whole process of a transmission from transmitter to receiver.
Simulation of the transmission chain gives us the opportunity to construct LoRa signals for replay with the signal generator.
Besides the encoding step, LoRa includes several methods to detect and correct errors in the signal, like a cyclic redundancy check and a message integrity code. This has an impact on the reception of signals which we have to include in the simulations, in order to validate our measurements correctly. In section 3.2.1, we tested the communication system with erroneous signals to find out more about the error correction mechanism.
With a working simulation, we are now able to compare the results with our system built in the previous chapter.

3.1 Measurement Setup

To establish a LoRaWAN communication, essential things are a gateway for reception of a packet, a LoRaWAN network which processes the data, and an end node to transmit data.

Gateway

We used a Raspberry Pi 3 with the IMST iC880A-SPI-LoRaWAN concentrator which operates in the 868 MHz frequency band. Figure 3.1 shows the parts of the gateway, assembled in a box, suitable for outdoor conditions.

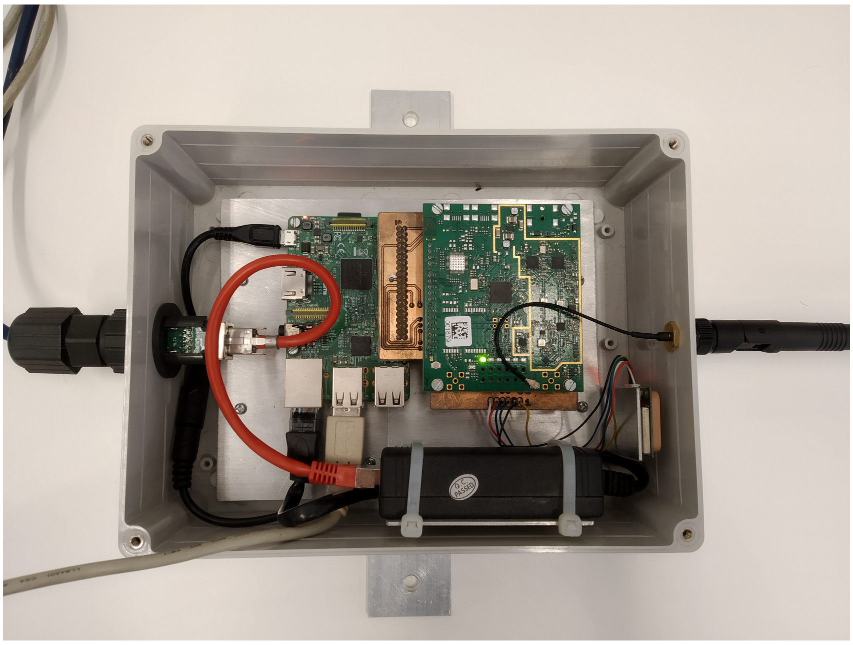


Figure 3.1: Gateway: Raspberry Pi 3 with IMST iC880A-SPI-LoRaWAN concentrator

The IMST concentrator is a transceiver module designed to receive up to 8 LoRa-signals simultaneously using different spreading factors and channels [7]. The block diagram of the IC880A concentrator is depicted in figure 3.2. It features two SX1257 Tx/Rx front-ends and a SX1301 baseband processor, both by Semtech. The SX1301 receives digitized bit streams by the front ends and demodulates them using several demodulators. The data then gets stored and further processed by the raspberry pi. The channels IF0 to IF7 have a bandwidth of $BW = 125\,\text{kHz}$ which cannot be modified. Eight packets with different

spreading factors at different frequency channels can be demodulated simultaneously. The channel IF8 can be configured to a bandwidth of 125 kHz, 250 kHz, or 500 kHz. It can also be configured to demodulate one spreading factor only and is used as a backhaul link to other gateways or infrastructure equipment. The IF9 Channel is used for the demodulation of FSK packets.
In the transmitter path, the packets get modulated using a (G)FSK or LoRa modulator and transmitted via the SX1257 front ends.

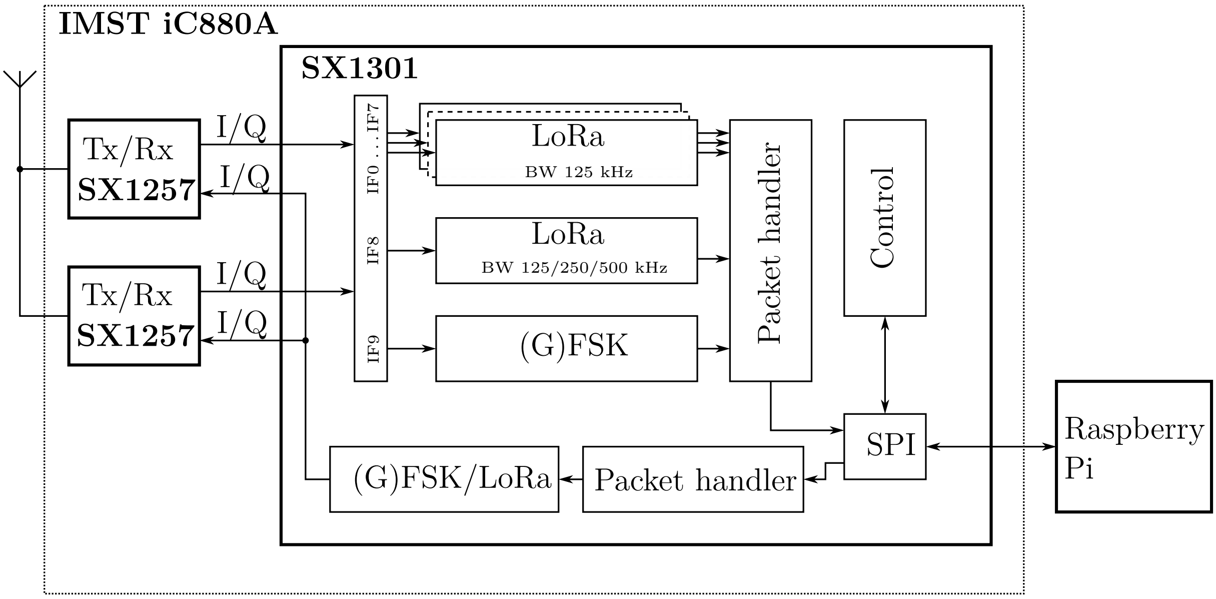


Figure 3.2: Block diagramm of the IMST iC880A-SPI LoRaWAN concentrator [7]

ChirpStack-LoRaWAN Network

There are many open source LoRaWAN network bundles available. We used the LoRaWAN Network Server stack called ChirpStack[1], formerly known as Loraserver. It provides software to run a complete, standalone LoRaWAN network, including a software bundle to run a gateway. The architecture of the server stack is depicted in figure 3.3.

[1] https://www.chirpstack.io/

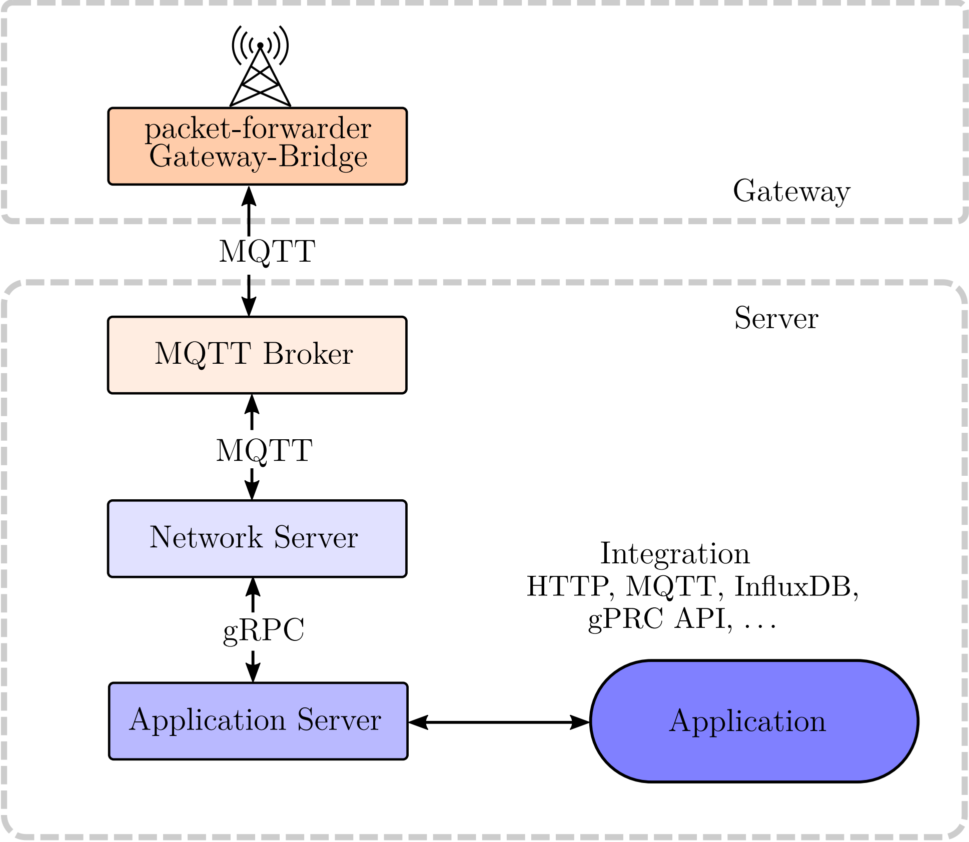


Figure 3.3: ChirpStack architecture

ChirpStack provides the following open-source components for LoRaWAN networks:

ChirpStack Gateway Bridge:

The Gateway Bridge handles the communication with one or multiple LoRaWAN gateways connected to the network. LoRa messages are received by the gateways within reach of the end-device, demodulated, and forwarded to the Gateway-Bridge. It uses a backend called Packet Forwarder, which passes the demodulated packets on to the Gateway Bridge. Two backends are commonly used by Chirpstack, the Semtech UDP Packet Forwarter and Basic Station Packet Forwarder.
The Semtech UDP Packet Forwarder was the first implementation, which the Semtech UDP Protocol, specifically designed for LoRaWAN. Messages are exchanged in a pseudo-JSON format through UDP between gateway and network server. ChirpStack also provides an alternative, the LoRa Basic Station. Basic Station uses two different protocols, the LNS (LoRaWAN Network Server) protocol to exchange data and the CUPS (Centralized

Update and Configuration Management) protocol, where a separate server gets contacted for configuration and software updates. The LNS protocol uses TCP based communication using web sockets, and is able to use TLS or token based authentication.
The Gateway Bridge uses MQTT to communicate with the network server via a MQTT broker. We used the open source Mosquito™ MQTT broker by Eclipse.

ChirpStack Network Server:

The ChirpStack network server NS is the key element of the communication system. It manages the gateways in the network and is responsible for data routing, security, and energy management. The NS deduplicates messages, received by multiple gateways and checks their authenticity and integrity. It also selects the gateway used for downlinks and sends ADR commands to optimize data rate and power for a device, to ensure the best energy profile.

ChirpStack Application Server:

The LoRaWAN Application Server implementation manages the infrastructure of the devices in a LoRaWAN network. It is responsible for processing the data and generate downlink payloads. Various integrations are available to connect the application with the Application Server.

ChirpStack Gateway OS:

The Gateway operating system (OS) is an embedded Linux-based OS to run the full ChirpStack stack on various gateway models, including a combination of a Raspberry Pi with the IMST concentrator we are using. Two types of OS are available, a base version and a full version. The base version provides software to run the concentrator including a Gateway Bridge and an interface for the configuration. The full version includes a full version of the network and application server environment by ChirpStack on top of the base version, which allows to run a whole LoRaWAN network on the gateway.

End-Device, RN2483 Transceiver:

As an end-device, we used a RN2483 Transceiver Module by Microchip. It can operate in the 868 MHz or the 433 MHz frequency band and comes with an On-Board LoRaWAN

Protocol Stack. To perform measurements effectively, we recorded several messages from the RN2483 to replay with a signal generator SMBV100A by Rhode & Schwarz.

Measurement Setup:

Figure 3.4 shows the scheme of our measurement setup. Via a MQTT subscription to the MQTT broker, we are able to capture the traffic at the gateway and implement it into Matlab. We were also operating the signal generator SMBV100A with Matlab, which completed the whole communication system.

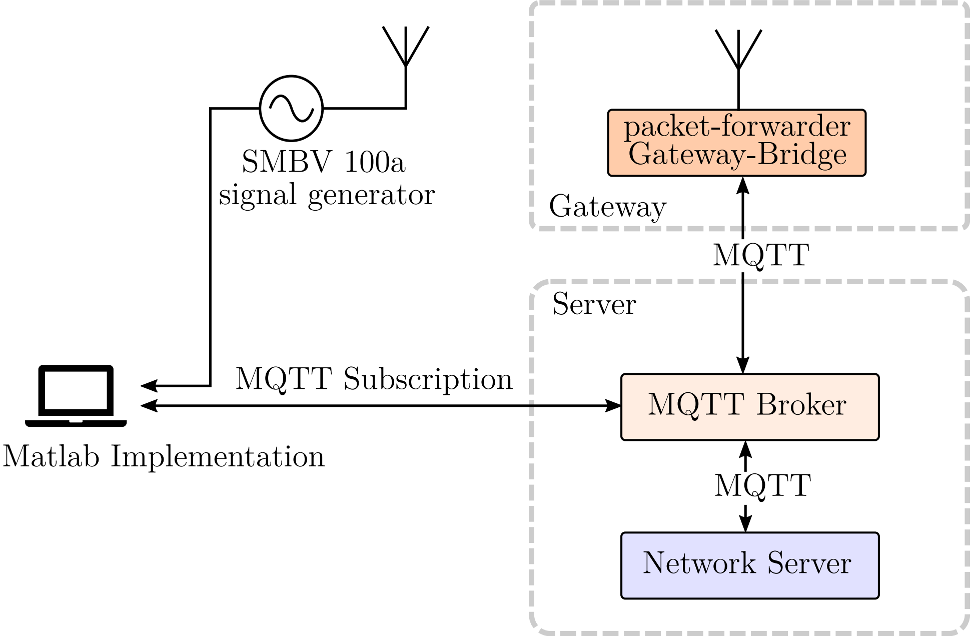


Figure 3.4: Measurement setup

3.1.1 Sensitivity Analysis

Before testing the resistance of the communication system against several different interference sources, we performed a sensitivity analysis to check the limits of a LoRa communication. The receiver sensitivity shows the minimum receiving power, at which a reception of a message is ensured.

As a trade off for a relatively low data rate, signal reception is possible with extremely low receiving power with LoRa modulation. In figure 3.5, the sensitivity analysis for a signal with a bandwidth of 125 kHz, a code rate of $^4/_5$, and different spreading factors is depicted. The sensitivity level is lowest with a high spreading factor. The difference between the highest values $SF = 11$ and $SF = 12$ is very small. Especially with high spreading factors, the time and frequency synchronization has to be precise for a correct decoding of the signal. The gain of sensitivity with a higher spreading factor is reduced by a harder decoding. To compensate this, the low data rate mode is used for $SF = 11$ and $SF = 12$. Table 3.1 shows a comparison of our measured sensitivity levels for different LoRa signals.

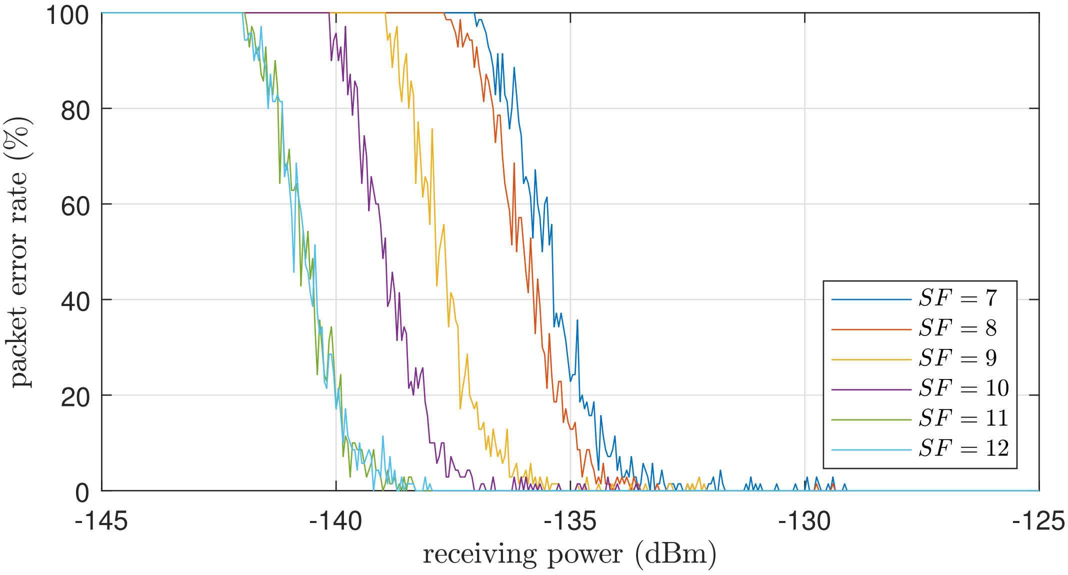


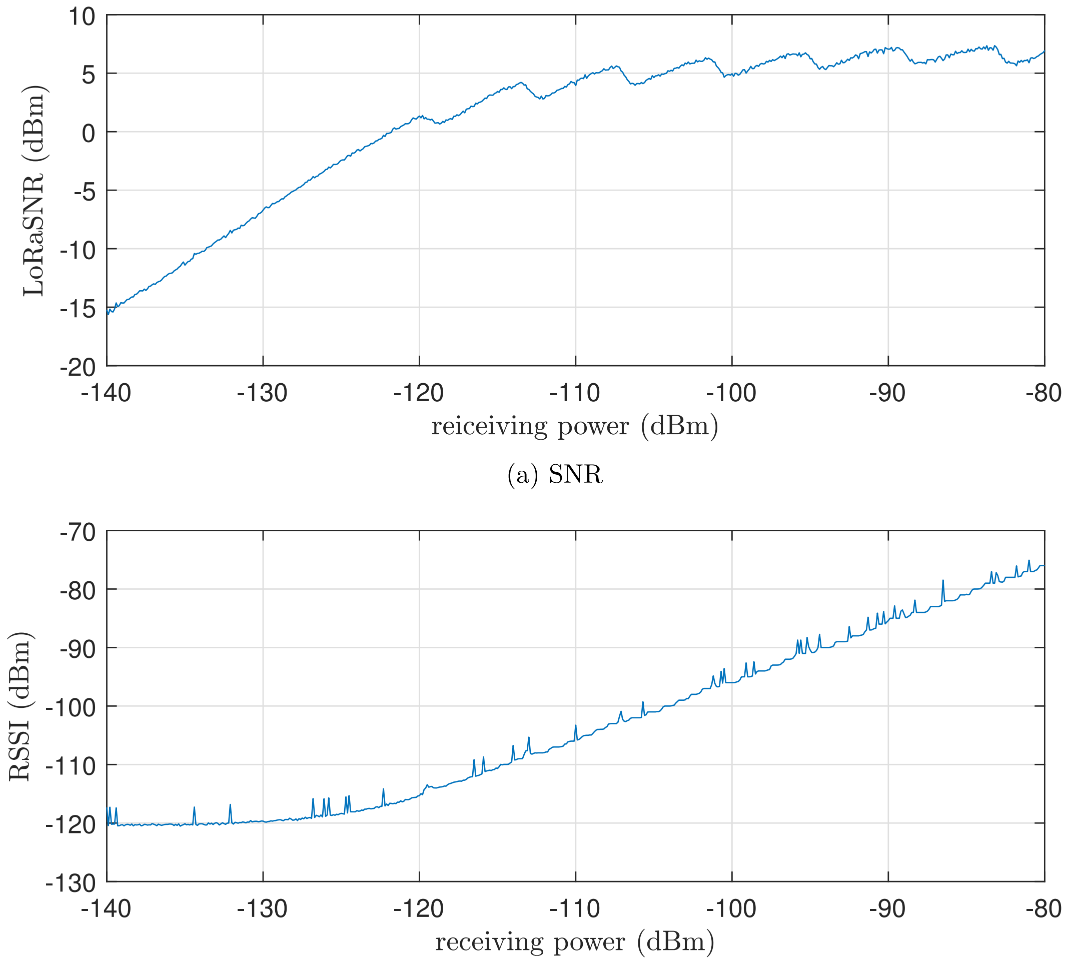
Figure 3.5: Sensitivity analysis; LoRa signal with bandwidth 125 kHz, code rate 4/5, and different spreading factors; low data rate mode is enabled for signals with $SF = 11$ and $SF = 12$.

SF	7	8	9	10	11	12
Sensitivity (dBm)	−131	−132.5	−133.5	−136	−138	−138

Table 3.1: Comparison of the sensitivity for a reception rate higher than 99 % between different spreading factors. $BW = 125$ kHz with a payload length of 9 Byte.

The LoRa gateway adds internal measured values of the signal quality to each received frame. LoRaSNR describes the ratio of wanted signal power to noise that can be demodulated and RSSI is the received signal strength indicator at the receiver. The resulting curves of this data for different receiving power is depicted in figure 3.6. The resulting SNR in figure 3.6a shows a value of −15 dBm at the lowest receiving power of about −140 dBm. It grows linearly until the receiver internally attenuates the signal to level it off between

5 and 10 dBm. This indicates a gain control by the receiver when a sufficient SNR of the signal is available. The RSSI decreases linearly with decreasing the receiving power. At a power lower than −120 dBm the RSSI stays constant at about −120 dBm.



(a) SNR

(b) RSSI

Figure 3.6: SNR and RSSI over receiving power for a message with $SF = 12$, $BW = 125$ kHz, and $CR = 1$ at the frequency channel with $f_c = 868.1$ MHz.

3.2 Simulation of a LoRa Test Setup

In order to validate the results we get from our measurements, we simulated a LoRa communication system. We implemented the steps of the transmitter and receiver chain from section 2.1.4 into Matlab, which makes it possible to create, encode/decode, and modulate/demodulate LoRa signals.

3.2.1 Error Correction Mechanism

To get a simulation which provides comparable results to our measurements, we have to find out more about the error correction mechanism. Using signals with code rates of $^4/_6$, $^4/_7$, or $^4/_8$, it is possible to detect and correct some bit errors in a codeword. At a code rate of $^4/_5$, a simple parity check is used, which allows to detect an odd amount of bit errors. To check the error correcting mechanism in this case, we performed some measurements with erroneous signals.

Figure 3.7 shows the packet delivery ratio of a signal with one bit error at different positions in the message. The signal was created by flipping a bit in the first seven nibbles of the encoded payload $\underline{p_e}$. The position of the bit error in a codeword tends not to be correlated with a successful error correction. It also has no impact if data or a parity bit is affected by the error. The signals were received with a power of −110dBm, measurements with different powers also showed no specific correlation with the reception rate.

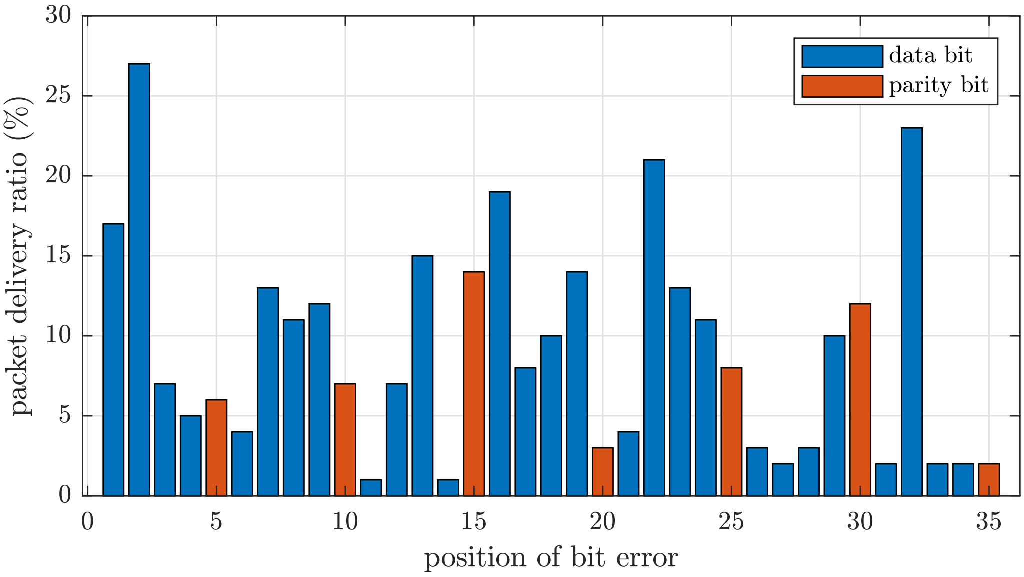


Figure 3.7: Reception of a erroneous signal with bit error at different positions after encoding step in section 2.1.4.1; signal with receiving power of −110dBm, $BW = 125\,\text{kHz}$, $SF = 7$, payload length of 20 Byte, enabled CRC, and disabled low data rate mode.

Since the position of bit errors after the encoding step showed no specific correlation with the delivery ratio, we tested the impact of an error after the encoding step by introducing different symbol errors to the signal. The impact of changing a single symbol on the reception rate is depicted in Figure 3.8. It shows the packet delivery ratio for a signal with $SF = 7$ and the symbol $a[12] = 0 \dots 127$, swept over the whole symbol alphabet. While the transmission with the correct symbol $a[k] = 55$ shows a successful reception all the time, the signals with a symbol error show a reception rate between 0% and 20%.

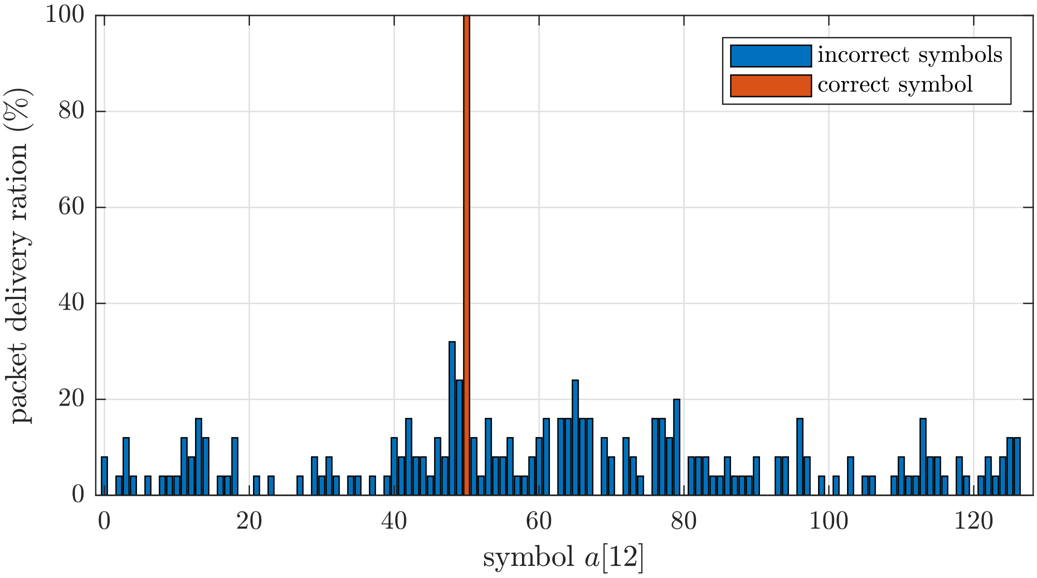


Figure 3.8: Packet Delivery Ratio over variation of symbol $a[12]$; signal with receiving power of -110dBm, $BW = 125\,$kHz, $SF = 7$, payload length of 20 Byte, enabled CRC, and disabled low data rate mode.

While reception with a single symbol error was still possible, introducing two or more symbol errors to the signal led to a non working communication.
These results show a random, partly successful error correction, with a code rate of $^4/_5$ and a single symbol error.

3.2.2 BER vs SNR

With working Matlab simulations of all parts of a LoRa communication system, we are able to perform some simulations to validate the results we get from our measurements. Figure 3.9 shows the bit error rate BER over signal to noise ratio SNR for a frame with $SF = 12$, $CR = 1$, and a bandwidth of 125 kHz. As expected, a higher spreading factor leads to a lower bit error ratio and a higher noise resistance. The curves also show, that the low data rate mode (LDR) has no impact on the bit error rate over signal to noise ratio. The long duration of LoRa packets with $SF = 11$ and $SF = 12$ can cause issues if drift of the crystal reference oscillator happens, due to either temperature change or motion. For this case, the low data rate mode is used to add a small overhead to increase the robustness against reference frequency variations.

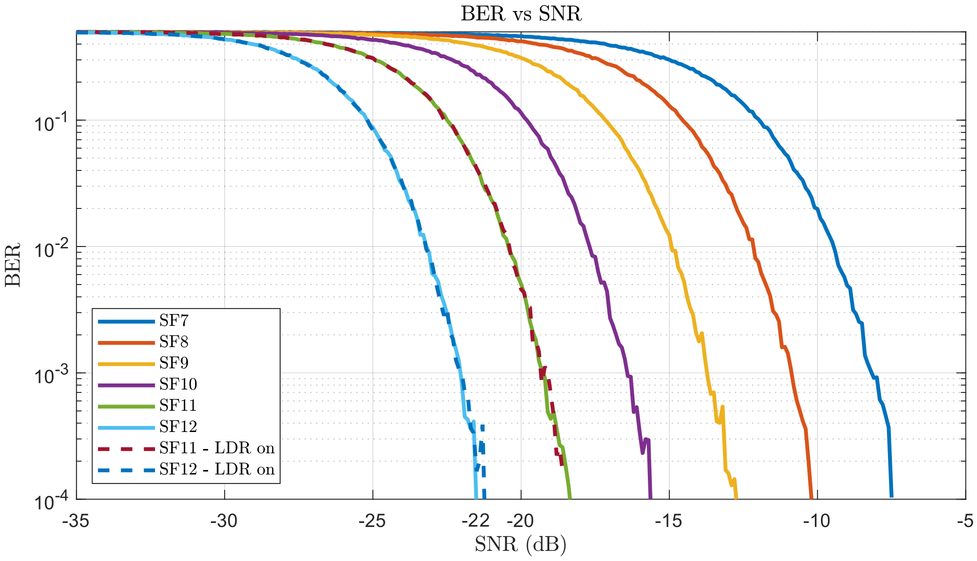


Figure 3.9: Simulated bit error rate BER over signal to noise ratio SNR fo a LoRa signal with $SF = 7 \ldots 12$, $BW = 125\,\text{kHz}$, and low data rate mode (LDR) for $SF = 11$ and $SF = 12$

Receiver Sensitivity and Link Budget

The receiver sensitivity describes the at which a certain maximum bit error ratio is achieved. The sensitivity is derived by

$$S = -174\,\text{dBm} + 10log_{10}BW + NF + SNR \tag{3.1}$$

with

- S . . . Receiver Sensitivity (dBm)
- BW . . . Bandwidth (Hz)
- NF . . . Noise Figure (dB)
- SNR . . . Signal to Noise ratio (dB)

With receiver sensitivity and the transmit power we can calculate the link budget with

$$L = P - S \tag{3.2}$$

with

- L . . . Link Budget (dBm)
- P . . . Transmit Power (dBm)

Example: *Calculation of sensitivity and link budget for a maximum BER for a LoRa signal with the following parameters:*

- $SF = 8$
- $BW = 125\,\mathrm{kHz}$
- $NF = 6\,\mathrm{dB}$
- $P = 15\,\mathrm{dBm}$

From figure 3.9 we get a SNR of $-22\,\mathrm{dB}$ *for the given parameters. Using equation* (3.1) *we get a sensitivity of*

$$S = -174\,\mathrm{dBm} + 10log_{10}125000 + 6\,\mathrm{dB} - 22\,\mathrm{dB} = -139\,\mathrm{dBm}. \tag{3.3}$$

With the sensitivity and equation (3.2) *we get a link budget of*

$$L = 15\mathrm{dBm} - (-139\,\mathrm{dBm}) = 154\,\mathrm{dBm} \tag{3.4}$$

Chapter 4

Interference Immunity against a Continuous Wave Signal

With a working communication system, we are now able to test the performance with interferences during the transmission. The shared use of the spectrum in unlicensed ISM bands causes a rise of noise with increasing devices and technologies operating in the spectrum. We started to test the communication system‘s performance against general interferences in form of a continuous wave (CW) signal. The signal should simulate the carrier frequency of a signals from other devices, operating in the same spectrum This chapter starts with an introduction in the measurement setup we used. The second part, we performed a sensitivity analysis with the communication system and a CW interference with different power and carrier frequencies.

4.1 Measurement Setup

As CW-source, we used a Rohde & Schwarz SME03 signal generator. Figure 4.1 shows the setup of the measurement. The CW signal $u(t)$ gets added to the LoRa signal $s(t)$, resulting in the signal $y(t) = s(t) + u(t)$ at the receiver end.

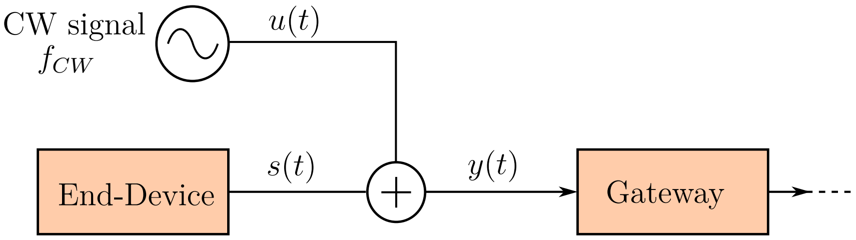


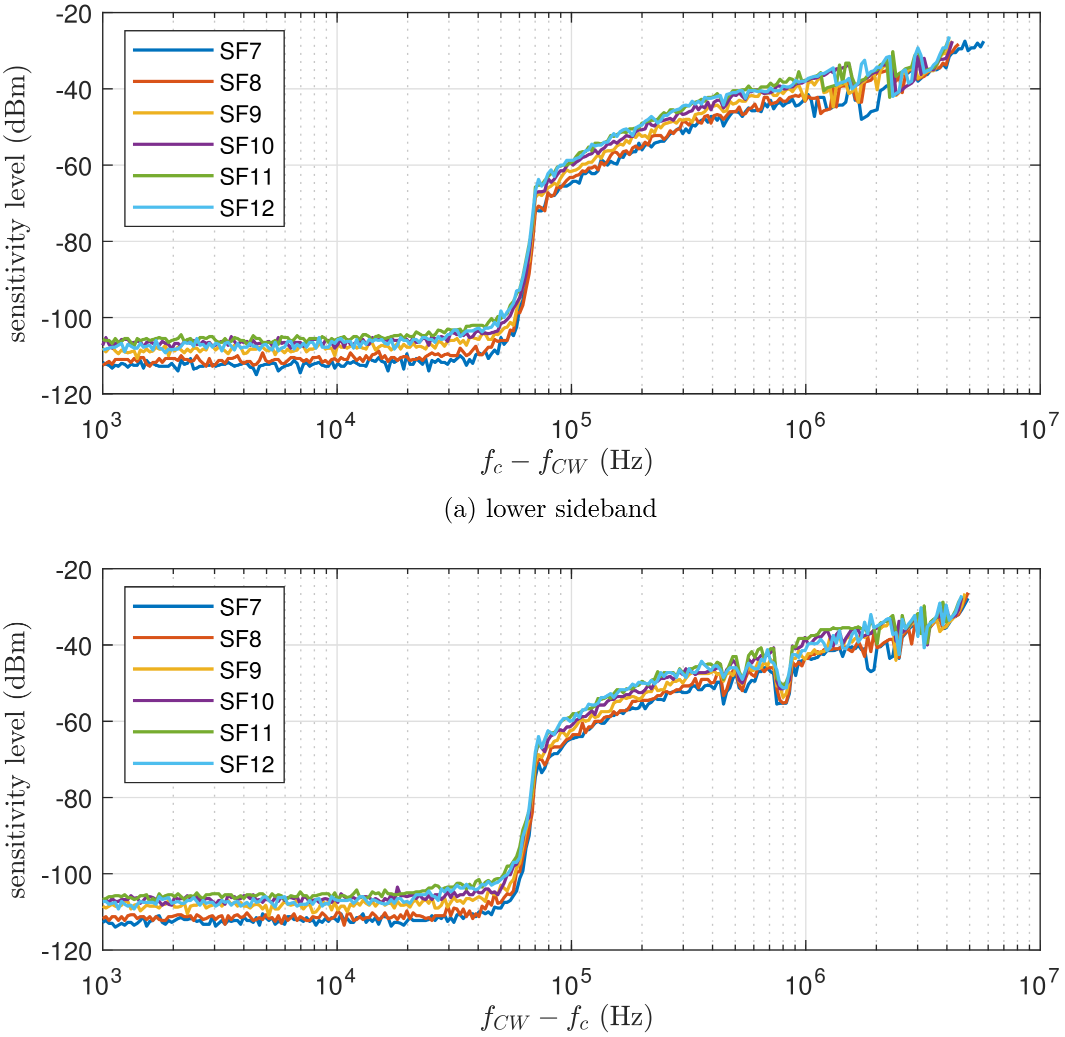
Figure 4.1: Scheme of measurement setup with a continuous wave interference

4.2 Sensitivity Analysis

The sensitivity of the LoRa reception is measured, while the frequency f_{CW} of the added CW signal is swept around the carrier frequency of the LoRa signal f_c. The sensitivity level shows the maximum interference power, at which the gateway can correctly receive and decode the LoRa signal with a packet error rate lower than 5%.

The impact of the interference is depicted in figure 4.2. We performed a measurement with a LoRa signal with bandwidth of 125 kHz, receiving power −125 dBm, and different spreading factors. Figure 4.2a shows the lower sideband and figure 4.2b the upper sideband of the spectrum around the LoRa carrier frequency f_c. Both curves show similar behavior. Within a CW frequency of $f_{CW} = f_c - {}^{BW}\!/_2 < f_c < f_c + {}^{BW}\!/_2$, the sensitivity level is between −115 dBm and −105 dBm for the different spreading factors. Outside the LoRa signal bandwidth, the sensitivity level rises rapidly. Due to a maximum input power at the gateway of −15 dBm, we stopped the measurements at the frequency f_{CW}, where an input power of −25 dBm was reached.

This behavior of the sensitivity level is determined by the input filter at the receiver. In the passband of the filter, the CW signal gets added to the LoRa signal and affects the demodulation process. It is possible to demodulate correctly up to a interference power 10 to 15 dB higher than the power of the LoRa signal, dependent on the spreading factor. A longer chirp duration, and therefore a higher spreading factor, is more resistant against the interference. In the stopband, the CW signal gets attenuated by the filter and therefore the reception gets affected at a higher interference power.



(a) lower sideband

(b) upper sideband

Figure 4.2: Frequency spectrum around the LoRa carrier frequency 868.1 MHz with a CW interference; LoRa signal with a receiving power of -125 dBm for a) lower sideband and b) upper sideband

A noticeable notch for every spreading factor can be seen in the upper sideband, about 800 kHz higher than the carrier frequency. To find out what causes this, we have to look into the receiver chain of the gateway. A simplified block diagram of the SX1257 Tx/Rx front-end is depicted in figure 4.3. The receiver is based upon a zero-IF architecture and converts the RF signal directly down to the baseband. The input gain is controlled by an low noise amplifier LNA, split into I and Q channel, and converted down by the mixer. Before the signal is fed into the analog to digital converter ADC, it is pre-filtered with a low-pass with a bandwidth up to 750 kHz and amplified. The digital bit-stream is further processed by the SX1301 baseband processor.

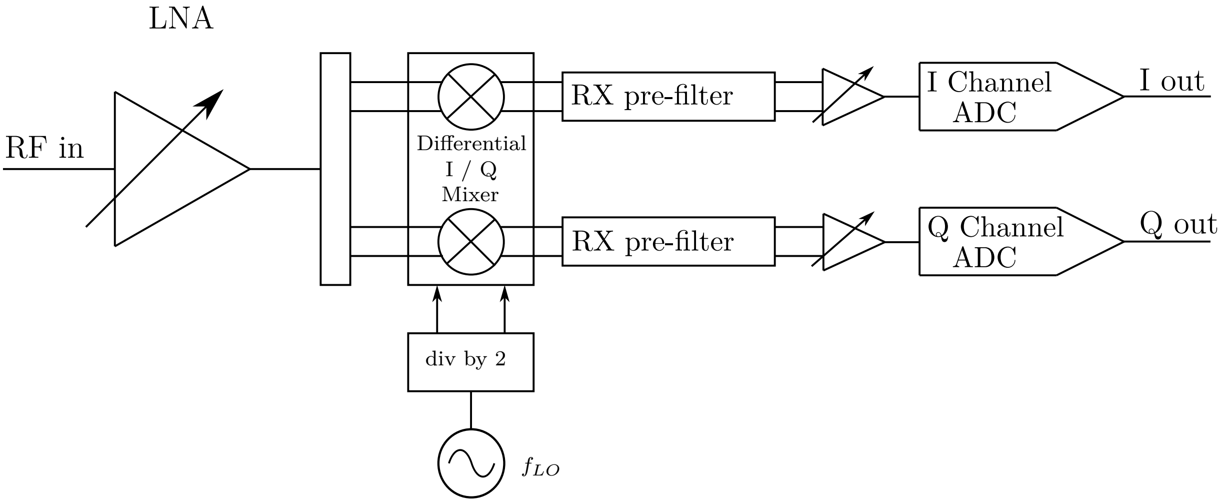


Figure 4.3: Simplified block diagram of the SX1257 front-end receiver path [13]

In our measurement setup, the local oscillator frequency f_{LO} of one of the SX1257 transceiver modules was set to $f_{LO} = 868.5\,\mathrm{MHz}$. Figure 4.4 shows the downconversion of the three mandatory channels. Channel one, with a carrier frequency of 868.1 MHz, is located at $-400\,\mathrm{kHz}$ in the baseband. When the frequency of the CW signal is about 800 kHz higher compared to the LoRa signal, image frequencies due to the mixer can interfere with the reception, which explains the notch in figure 4.2b.

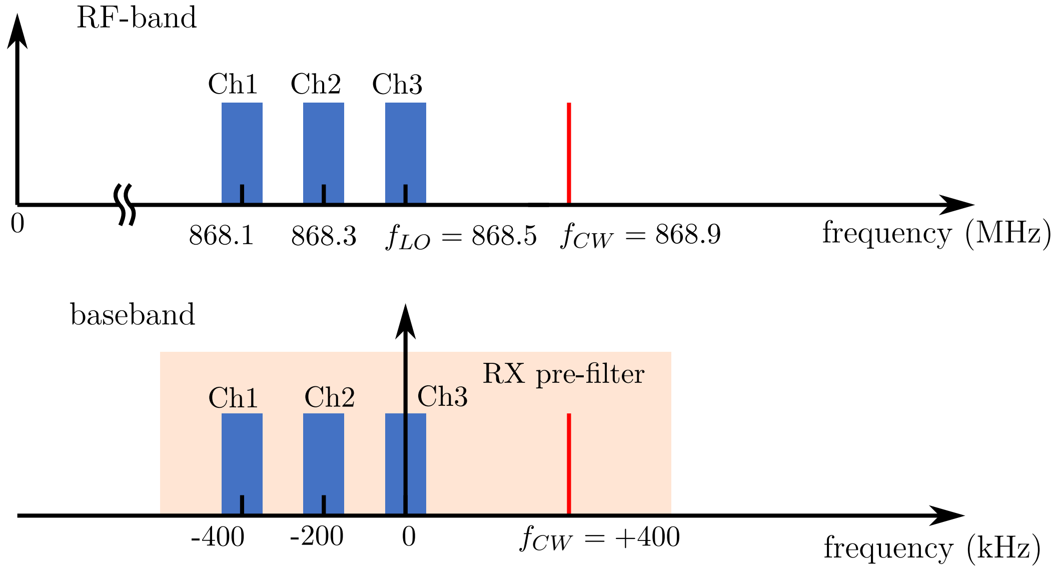
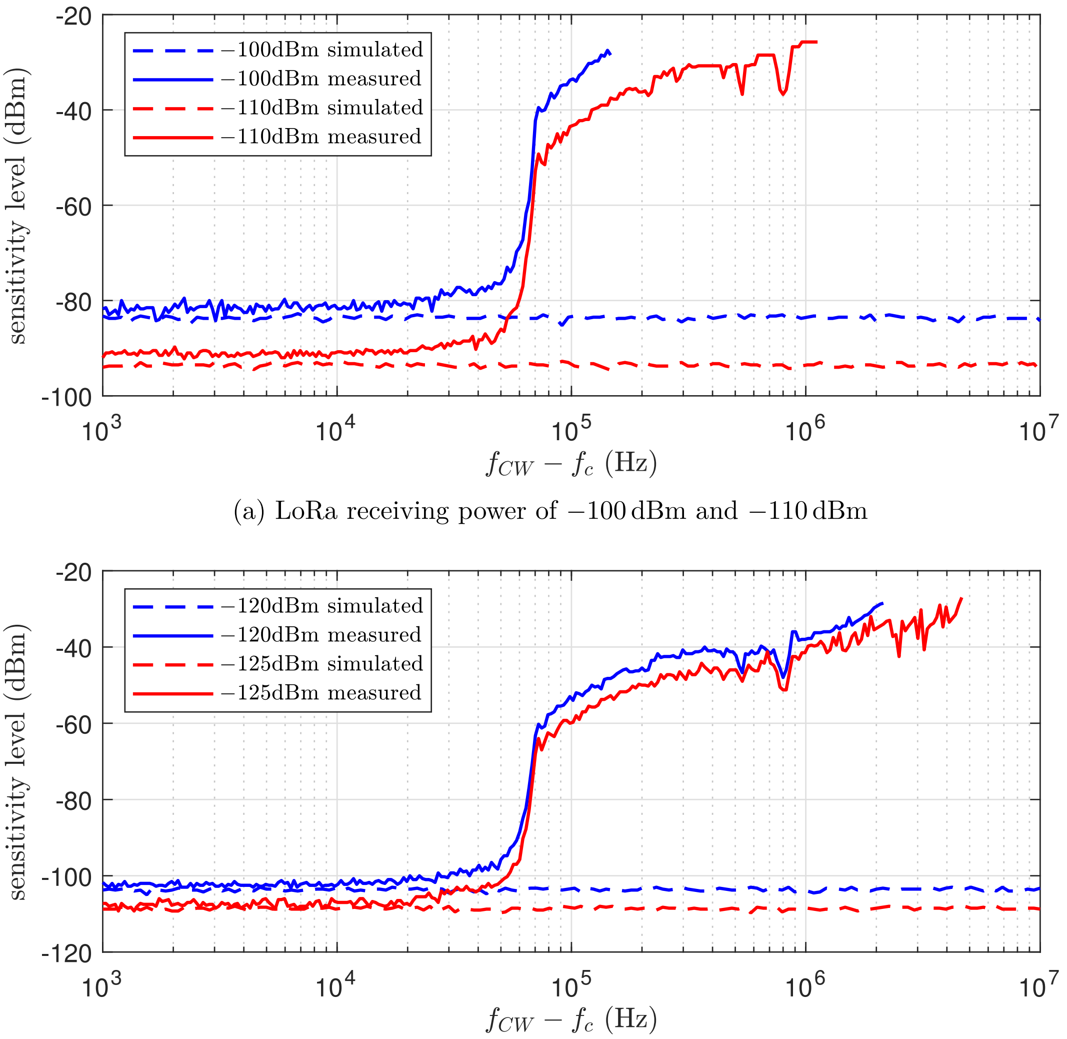


Figure 4.4: Spectrum in RF-band and baseband of the three mandatory channels with a CW interference

Figure 4.5 shows the results for different receiving powers of the LoRa frame compared with our simulations. While the simulation results inside the bandwidth are equal to the results of the measurements, the CW interference effects the reception of the packet differently outside the bandwidth. We did not implement a input filter in the simulation and therefore the effect of the interference in the demodulator is the same outside the bandwidth.



(a) LoRa receiving power of −100 dBm and −110 dBm

(b) LoRa receiving power of −120 dBm and −125 dBm

Figure 4.5: Frequency spectrum around the LoRa-carrier-frequency 868.1 MHz with a CW interference; LoRa signal with $SF = 12$, $BW = 125$kHz, and different LoRa receiving powers

Figure 4.5 shows a slight difference between the measured and simulated results for higher LoRa receiving powers. To show this difference in more detail, we performed a sweep over the LoRa receiving power with the CW interference having a frequency of $f_{CW} = f_c$. The results are depicted in figure 4.6. As seen in the previous figures, there is a slight difference between the measured and simulated values for a higher receiving power.

Figure 4.6 shows a linear correlation between the LoRa receiving power and the sensitivity against a CW interference. It is possible to receive the signal correctly with a packet delivery ration 95 % and a CW interference power about 15dB higher than the power of the LoRa signal.

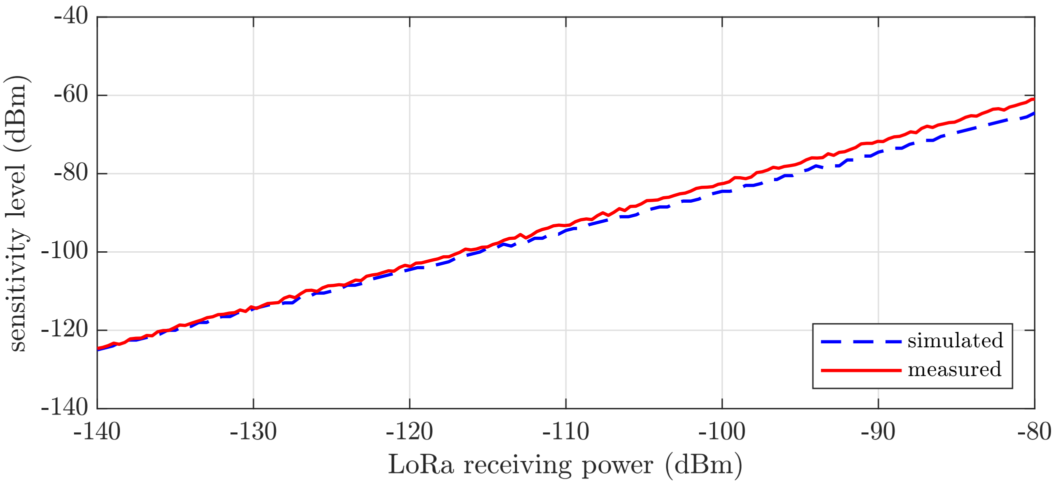


Figure 4.6: Sensitivity level of a LoRa signal with a CW interference. $f_{CW} = f_c = 868.1$MHz; LoRa signal with $SF = 12$, $BW = 125$kHz

Chapter 5

Interference Immunity against LoRa-Signals

In the previous chapter, the communication system's resistance was tested against general interferences in the form of a continuous wave signal, representing a carrier signal of another device operating in the spectrum. The next step was to investigate the performance with other LoRa sources interfering with the communication.
This chapter starts with the measurement setup we used in section 5.1. It then continues in 5.2 with an investigation of a collision of two LoRa signals with the same spreading factor. We tested the impact of the power difference between both signals and the time at which the second signal starts to interfere with the first one. Afterwards, we defined an expression for the packet error probability (PEP) for the case of same receiving power. At last, we investigated the impact on the communication with LoRa signals with different spreading factors as an interference source.

5.1 Measurement Setup

Figure 5.1 shows the measurement setup with two LoRa signals $s_1(t)$ and $s_2(t)$ added, resulting in the signal $y(t) = s_1(t) + s_2(t)$ at the receiver end. For emulating end-devices, we used two SMBV100A, which replayed recorded signals from an RN2483.

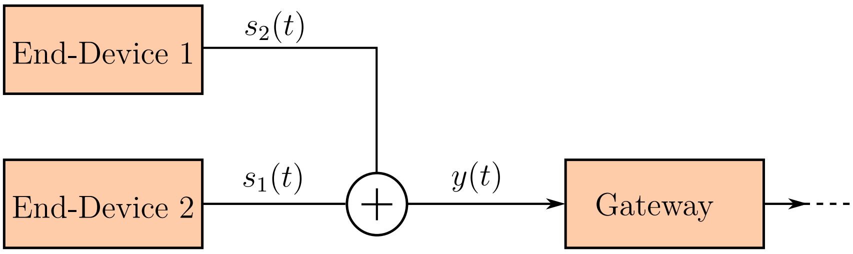


Figure 5.1: Scheme of the measurement setup with multiple LoRa signals

The first signal $s_1(t)$ depicted in figure 5.2, is disturbed by the signal $s_2(t)$ starting with a delay of t_d, relative to the beginning of $s_1(t)$. Goal of this approach is to see the impact of the interference, beginning at different stages of the reception of a LoRa packet.

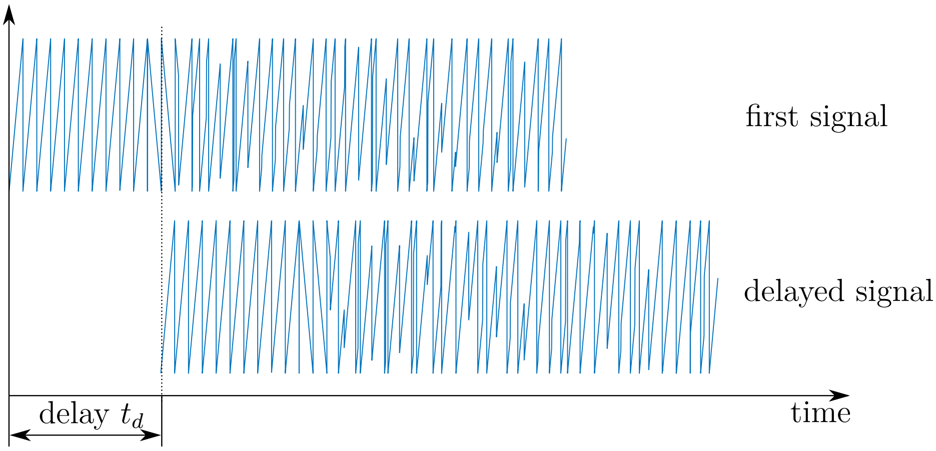


Figure 5.2: Summation of two LoRa signals with a delay t_d

5.2 Same Spreading Factor

The first approach is to test the interference with a LoRa signal of the same spreading factor. Figure 5.3 shows the packet delivery ratio over the Signal to Interference Ratio (SIR) of a frame with $SF = 12$ and an interfering signal with the same spreading factor. The simulation is about 0.5 dB off compared to the measurement. The interference shows an impact at the reception of the first signal at a receiving power of about 2 dB lower, compared to the first signal.

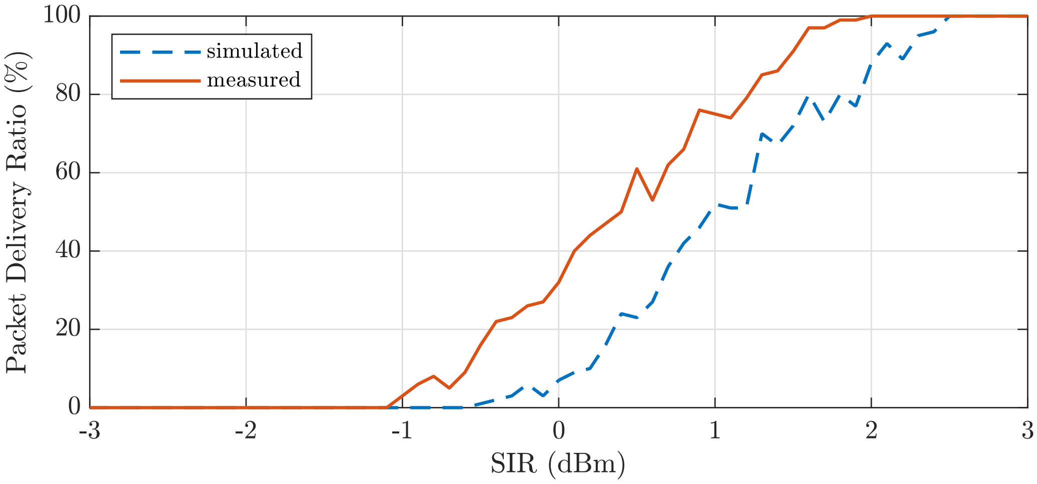


Figure 5.3: Packet delivery ratio over *SIR* of the first signal. Both LoRa signals with $SF = 12$, $BW = 125\,\text{kHz}$, low data rate mode, payload length of 20 Bytes, and random delay uniformly distributed over signal length of the first signal.

Table 5.1: measured SIR threshold for a correct reception of the first signal with packet delivery rate higher than 99 %

Spreading Factor	7	8	9	10	11	12
SIR threshold measured (dB)	0	0	0.4	1.3	2	2

The thresholds, measured for all spreading factors, are depicted in table 5.1.

As expected, reception of the signal is not possible when the interference source has a slightly higher receiving power. The demodulation process has to deal with multiple overlapping chirps, which leads to multiple peaks in the spectrum of the FFT, as described in section 2.1.4.2. In this case, no clear symbol can be assigned to the chirp, which leads to errors and the frame gets dropped.

With these results in mind, we were further investigating the impact of the timing, when the delayed signal starts to interfere with the reception.

5.2.1 Interference with Signals of Same Receiving Power

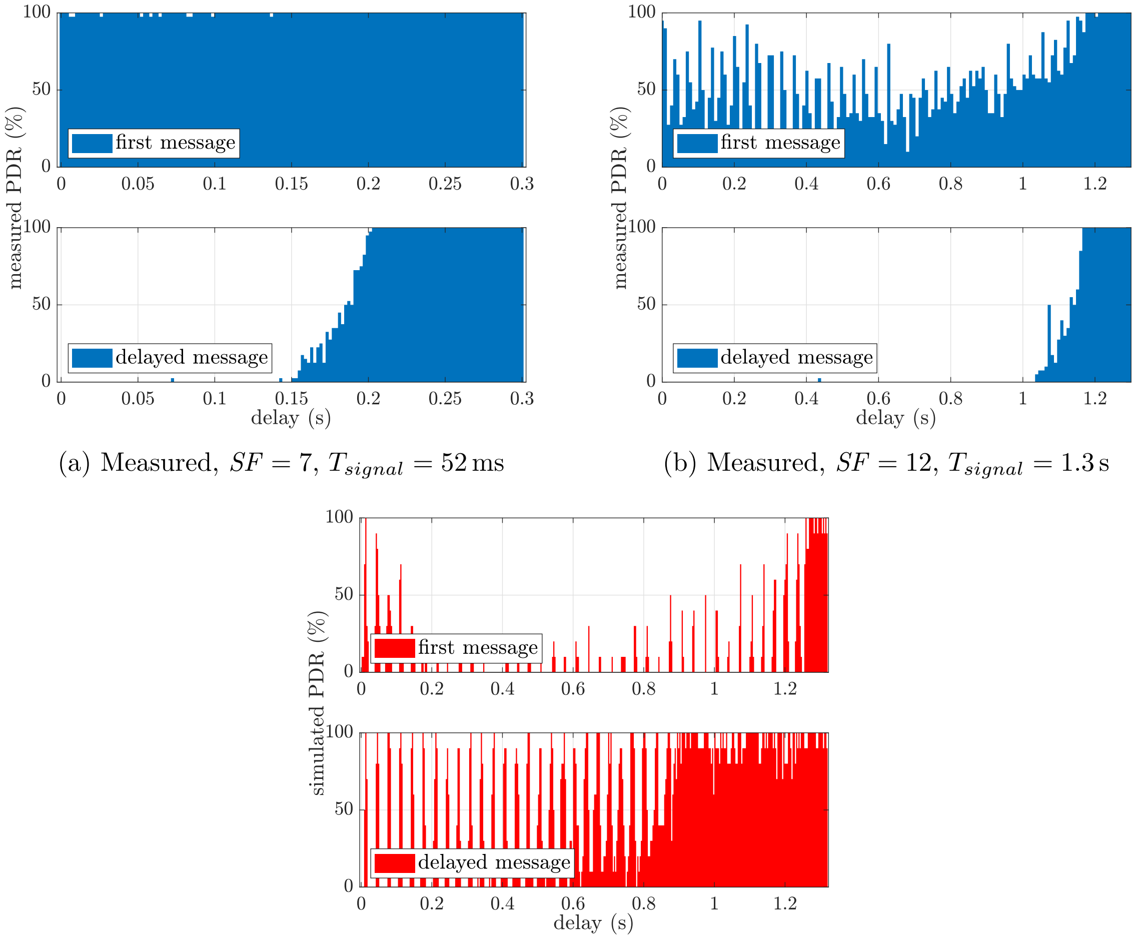
Figure 5.3 shows a successful reception of about 40 %, with an interfering signal with same receiving power ($SIR = 0\,\text{dB}$), same spreading factor, and a uniformly distributed random delay. The impact of the delay is depicted in figure 5.4.

Figure 5.4a shows the results of our measurements for the case of two messages with

$SF = 7$. The first signal is received with almost no impact on the delayed message. The delayed signal also has a relatively high reception rate. Starting with a reception rate of about 20 % with a small delay, it increases constantly. At a delay higher than ~ 0.007s, both signals have a delivery ratio of 100%. At this point, the second signal does not interfere with the synchronization with the preamble of the first message.

In the case with two signals with $SF = 12$, depicted in figure 5.4b, the delayed message interferes with the first one and decreases its packet delivery ratio. The measurements show peaks at a period of about one symbol duration T_s from equation (2.6). This shows that the probability of a correct reception is higher when the delayed signal is synchronized with the first one, and the chirps overlap. Unlike in the case with $SF = 7$, the delayed signal is not recognized by the receiver at a delay smaller than about one second. In this case, the only part overlapping with the first signal are some synchronization chirps of the preamble. At about 1.2 seconds, the receiver can synchronize to the delayed signal and receives it correctly.

A simulation of the setup with two signals with $SF = 12$ is depicted in figure 5.4c. While the reception rate is lower in general for the first message, the periodic peaks are also present in the simulation. It shows a reception of the delayed message, also with periodic peaks with a period of the symbol duration. The demodulator in our simulation has no atomized synchronization integrated. The simulated receiver is always perfectly synchronized with both signals, while the receiver in our test setup struggles with that.



(a) Measured, $SF = 7$, $T_{signal} = 52\,\mathrm{ms}$

(b) Measured, $SF = 12$, $T_{signal} = 1.3\,\mathrm{s}$

(c) Simulated, $SF = 12$, $T_{signal} = 1.3\,\mathrm{s}$

Figure 5.4: Packet delivery ratio of two delayed signals a) measured with both $SF = 7$, b) measured with both $SF = 12$, and c) simulated with both $SF = 12$; Each signal with $-100\,\mathrm{dBm}$ receiving power, $BW = 125\,\mathrm{kHz}$, and a payload length of 20 Bytes

5.2.2 Interference with Signals of Different Receiving Power

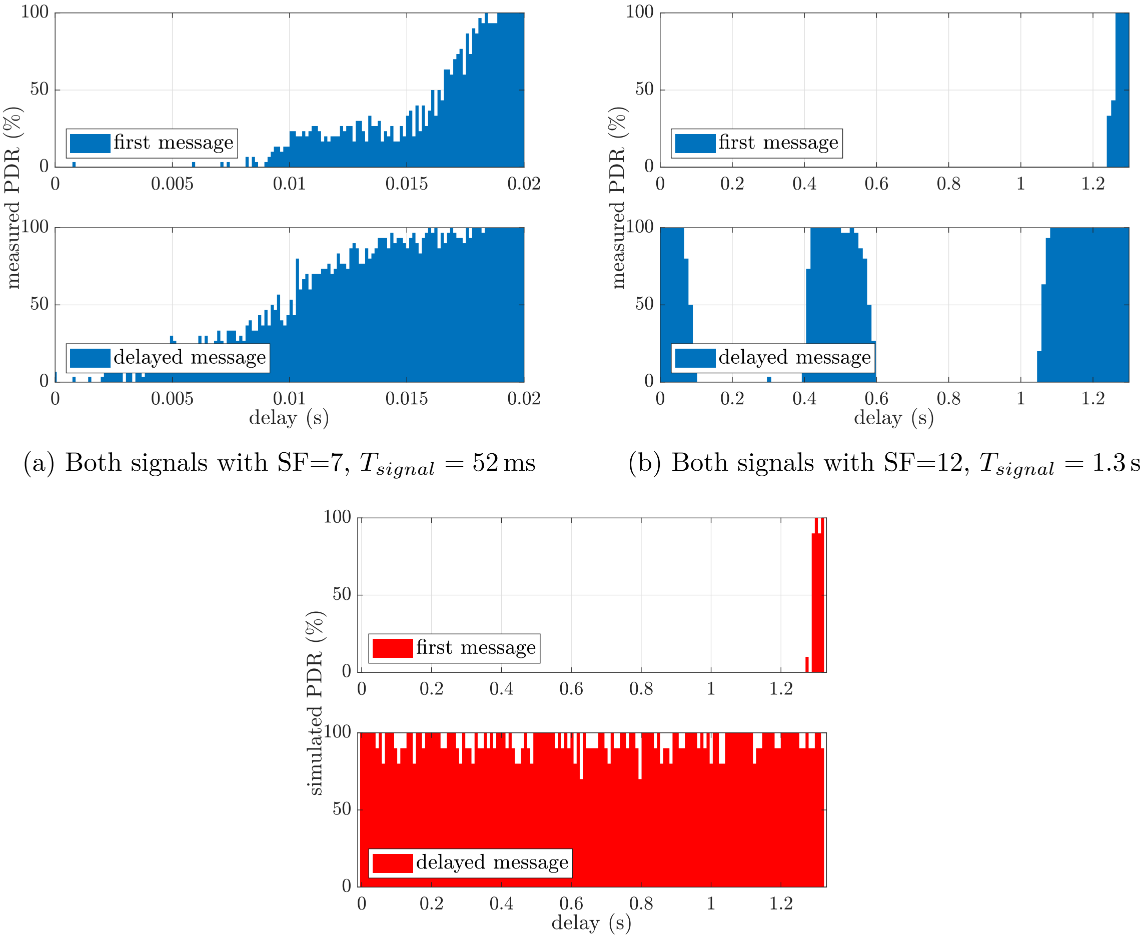
There are two different cases we have to investigate regarding the receiving powers of the first signal P_{signal} and the delayed interference signal P_{int},

- $P_{signal} > P_{int}$, or

- $P_{signal} < P_{int}$.

In case of $P_{signal} > P_{int}$, figure 5.3 shows, that at a certain threshold, the interfering signal has no impact on the reception of the first signal. The measurement has also shown that

the interfering signal does not get recognized by the receiver due to the higher powered first signal.
The case with $P_{signal} < P_{int}$ shows different results. The impact of a collision of two signals with different power is depicted in figure 5.5. The result with $SF = 7$ for both signals in figure 5.5a) shows that in this case, the stronger signal, arriving during the reception of the first one, leads to a packet loss for the first frame. Up to a delay of about 10 ms, the first signal is not recognized by the receiver. The preamble length of the first signal is about 12 ms long. If the delayed signal starts to interfere after the reception of the preamble, the reception rate starts to increase. At a delay of about 20 ms or higher, the second signal does not interfere with the header symbols leading to an undisturbed reception of the first frame. The delayed signal has a similar behavior as the first signal. Even though its receiving power is 35 dBm higher than that of the first one, the reception rate is very low when it starts to interfere at the synchronization process at the beginning of the first signal.



(a) Both signals with SF=7, $T_{signal} = 52\,\mathrm{ms}$

(b) Both signals with SF=12, $T_{signal} = 1.3\,\mathrm{s}$

(c) Both signals with SF=12, $T_{signal} = 1.3\,\mathrm{s}$

Figure 5.5: Packet delivery ratio of two delayed signals a) measured with both $SF = 7$, b) measured with both $SF = 12$, and c) simulated with both $SF = 12$; First message $-125\,\mathrm{dBm}$ and delayed message $-90\,\mathrm{dBm}$ receiving power, $BW = 125\,\mathrm{kHz}$, and a payload length of 20 Bytes

The interference of two signals with $SF = 12$ in figure 5.5b) shows a different result. The first signal with lower power is unrecognized when a collision with a signal with higher power appears. The higher-power signal is received when it arrives at specific times during the reception of the lower-power signal. The simulation of this case is depicted in figure 5.5c). It shows a similar result for the reception rate of the first message. The delayed signal with higher power gets received successfully, contrary to the results of the measurements.

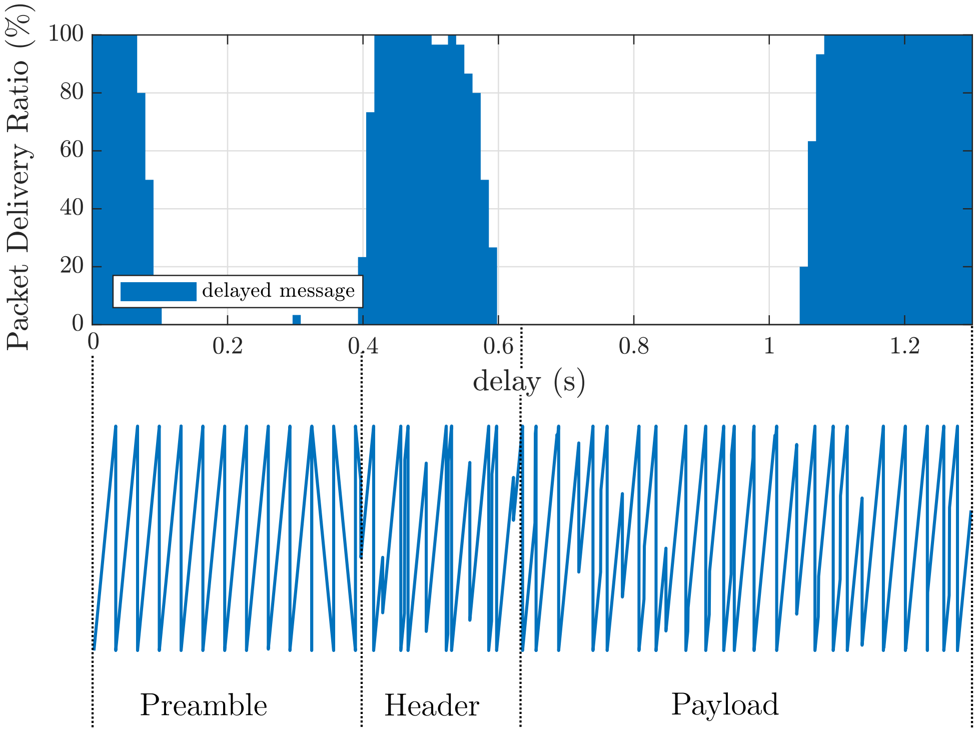


Figure 5.6: Packet delivery ratio of delayed signal; impact of delay relative to first signal

The results of the measurements from figure 5.5b) are depicted in figure 5.6 again, including the waveform of the first signal. This gives an insight of the moment the delayed signal starts to interfere with the first message.

The stronger frame survives the collision, when it arrives during the reception of the first three upchirps of the preamble of the first frame. Between the rest of the preamble and the header, the receiver is locked to the weaker signal and no frame gets received. The receiver tries to synchronize to the first signal and the interference leads to the loss of both packets. It seems that it needs about three chirps of the preamble for the receiver to be synchronized and at this point it is locked to this frame.

The interference starting during the reception of the header leads to releasing the lock on the weaker frame by the receiver. After being synchronized to the first signal, the receiver tries to decode and validate the header. The interference of the second signal leads to an incorrect header and the receiver stops receiving the first frame and a correct reception of the delayed signal is possible.

If the interference starts at a point where the header of the first frame has already been decoded, the receiver is aware of the information contained in the header including the payload length, the presence of a *CRC*, and the code rate. The receiver is now able to determine the signal length and tries to receive and decode it. Due to the interference, correct decoding is not possible.

5.2.3 Packet Error Probability

With the results in 5.2 and section 5.3, it is now possible to define a packet error probability. To derive the probability we have to make some assumptions. There are two different kinds of uplink messages in LoRa communication:

- *unconfirmed message:* When a message is received by the gateway, no acknowledgment (ACK) is returned.
- *confirmed message:* In this case, a successful transmission has to be acknowledged by the gateway by sending an ACK message back to the node.

In the following calculations we assume, that only unconfirmed messages get received, and no downlink messages are returned by the receiver after a successful reception. This simplifies the model, since the receive window is always on and deactivation during downlink can be neglected. We also assume perfect orthogonality between signals with different spreading factors, which therefore have no impact on each other. We also assume that there is no interference between the M frequency channels, and uniform distribution of the traffic load over the channels and the six different spreading factors. With the packet generation rate λ in packets/s, the traffic load $u^{(n)}$ for the frequency channel and spreading factor $n \in 1 \ldots 6M$ is derived by

$$u^{(n)} = \frac{\lambda}{6M}. \tag{5.1}$$

The packets are generated following a poisson process. For a node k with the packet generation rate λ_k and signal duration T_k we get the following expression for the receiving probability:

$$P_k^{(n)}(\text{reception}) = \exp^{-2u_k^{(n)}T_k}. \tag{5.2}$$

Figure 5.7 shows the case of two packets, arriving at the receiver. If packet 2 arrives during the interval $T_1 + T_2$ as shown in the figure, the packets collide. Using 5.2, we can derive the probability of a collision for packet 1 with

$$P_1^{(n)}(\text{collision}) = 1 - \exp^{-2u_1^{(n)}T_1} \exp^{-u_2^{(n)}(T_1+T_2)}. \tag{5.3}$$

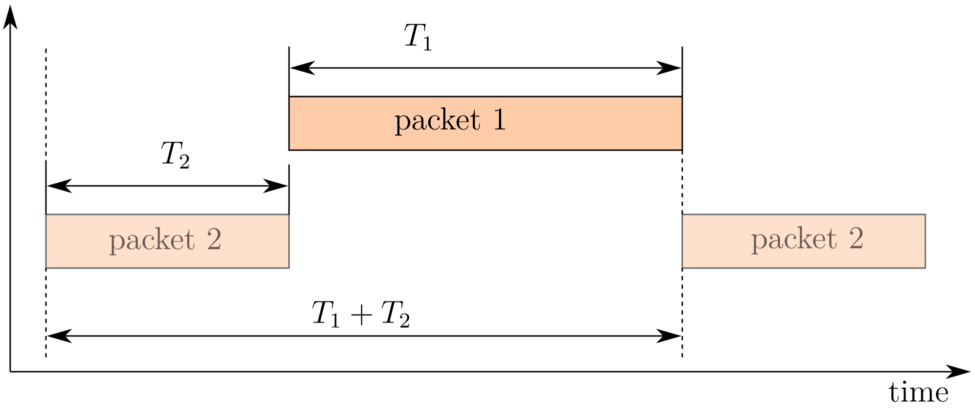


Figure 5.7: Collision of two LoRa packets at the receiver

Extending this case to a system with K different nodes of signal length T_k and arrival rate λ_k, leads to the expression

$$P_k^{(n)}(\text{collision}) = 1 - \prod_{\substack{i=1 \\ i \neq k}}^{K} \exp^{-2u_k^{(n)}(T_i + T_k)} \tag{5.4}$$

for the probability of the signal from node k to collide with another frame. Furthermore, the probability of a collision happening in the channel k is derived by

$$P^{(n)}(\text{collision}) = \frac{1}{K} \sum_{k=1}^{K} P_k^{(n)}(\text{collision}). \tag{5.5}$$

The collision probability over the arrival rate λ is depicted in figure 5.8. It shows the curves for $K = [2, 10, 50, 200]$ and 8 channels ($M = 8$), assuming that every node k has the same arrival rate $\lambda_k = \lambda$ for $k = 1 \dots 200$ at every point in the figure. As expected, the graphs show, that more nodes in the system lead to a higher collision rate with a constant packet generation rate over all nodes.

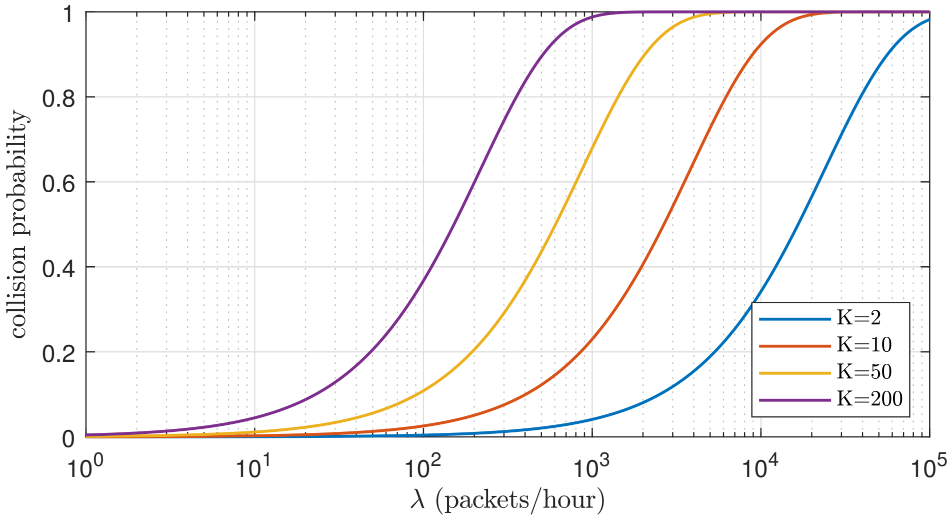


Figure 5.8: Collision probability with $M = 8$ channels for different amount of nodes and a signal length of 1.3 seconds, corresponding to a signal with $SF = 12$ and a payload length of 20 Bytes.

We have seen in section section 5.2.1 and 5.2.2, that a collision does not mean that both packets are lost. To consider this in our calculations, we introduce the probability of frame getting dropped during a collision $P_{k,i}(\text{dropped})$. By including this in 5.4, we get the following expression for the packet error probability PEP for node k:

$$PEP_k^{(n)} = 1 - \prod_{\substack{i=1 \\ i \neq k}}^{K} \exp^{-2a_k(T_i+T_k)P_{k,i}(\text{dropped})} . \tag{5.6}$$

The resulting packet error probability at the receiver is calculated by

$$PEP^{(n)} = \frac{1}{K} \sum_{k=1}^{L} PEP_k^{(n)} . \tag{5.7}$$

The probability of a frame getting dropped during a collision $P_{k,i}(\text{dropped})$ depends on the signal power of the frames k and i. We now look at the case with two signals with the same receiving power in section 5.2.1. In this case we have $P_i, k(\text{dropped}) = P(\text{dropped})$ for every i and k. If we look at the packet delivery ratio in case of a collision with $L = 2$ in figure 5.3, we can see a ratio of 32 % for the measurement and 7 % for the simulation at $SIR = 0\,\text{dBm}$, which leads to probability of $P(\text{dropped}) = 0.68$ for the measurement and $P(\text{dropped}) = 0.93$ for the simulation. Figure 5.3 shows the resulting packet error probability over arrival rate.

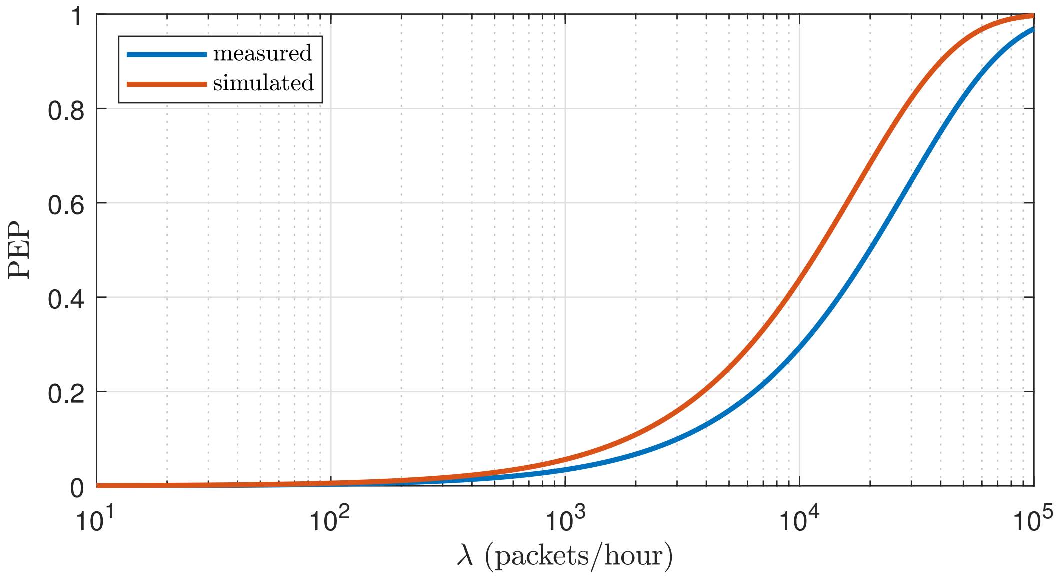


Figure 5.9: Packet error rate over arrival rate with $K = 2$ nodes with same receiving power and $P(\text{dropped}) = 0.68$

5.3 Different Spreading Factor

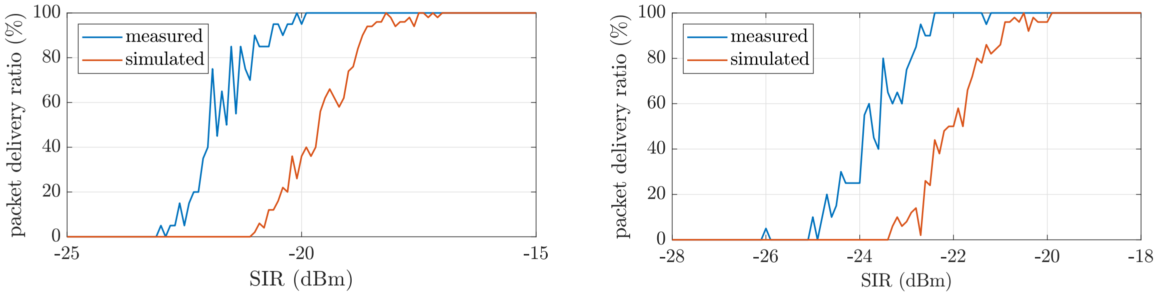
Next step was to investigate the impact of a collision of two LoRa signals with a different spreading factor. Chirps with different spreading factors are quasi-orthogonal, as derived in [11]. In [6] it is shown, that a signal with a higher spreading factor shows a significantly higher co-channel rejection than a signal with a lower one. Thus, signals further away from the gateway are preferably transmitted with a higher spreading factor, since they are more robust against co-channel interference.

Table 5.2: *SIR* thresholds in dB for two interfering LoRa signals with different spreading factors. Both signals with $BW = 125\,\text{kHz}$ and payload length 20 Bytes, interfering signal with spreading factor SF_{int} is received with a random delay t_d during reception of first signal with spreading factor SF.

$\mathbf{SF_{int}}$ **SF**	**7**	**8**	**9**	**10**	**11**	**12**
7	-	-8.5	-9	-9	-9.5	-9.5
8	-11	-	-11	-11.5	-12	-12.5
9	-14.5	-14	-	-14	-14	-14.5
10	-19	-16.5	-16	-	-17	-17
11	-21.5	-22	-21	-20.5	-	-20
12	-24.5	-24.5	-24	-23	-22.5	-

A sensitivity analysis with our measurement setup showed different results. We have computed and reported in table 5.1, the thresholds of the signal to interference ratio (SIR) for all combinations of spreading factors, at which a packet delivery ratio higher than 99% is achieved. The results show a relatively small difference of the co-channel rejection for different spreading factors. Based on these numbers, the spreading factor does not have a big impact on the reception in near-far conditions.

The results of the measurement of the packet delivery ratio over signal to interference ratio (*SIR*), compared to the simulation, is depicted in figure 5.10. The case of a signal with $SF = 11$ and an interference source with $SF_{int} = 12$ is shown in figure 5.10a, and the case with $SF = 12$ and an interfering signal with $SF_{int} = 11$ in figure 5.10b. It shows similar behavior between measurement and simulation, with a deviation of about 2 dB. This is caused by an imperfect implementation of the receiver in our simulation.



(a) Signal with $SF = 11$, interfering signal with $SF_{int} = 12$

(b) Signal with $SF = 12$, interfering signal with $SF_{int} = 11$

Figure 5.10: Measurement and simulation of the packet delivery ratio over *SIR* for two colliding LoRa signals.

Chapter 6

Conclusion

The focus of this thesis was to investigate the performance of a LoRaWAN system with several different sources interfering the communication. To do this, we established a LoRa communication system to perform some measurements. Additionally, we created a simulation of the system to validate the results we got from our measurements. To simulate a LoRa communication system, we had to figure out the encoding steps in the transmission chain. Before modulating the signal, the data gets whitened, encoded, interleaved, and Gray mapped, whereas header and payload get treated differently in this process. After implementing the encoding process and modulator in Matlab, we were able to create LoRa signals. With the respective decoder and demodulator steps, a simulation of a complete communication system was established in Matlab. To get a simulation comparable with the measurements, we had to investigate the error correction process in a LoRa communication system. With the simulated transmitter chain, we were able to create configurable LoRa signals for replay with the signal generator. With intentionally placed errors in the signal, we tested the error correction performance of the receiver and adopted this in our simulations.

With a working communication system and simulation, we performed some measurements with different interference sources. We started with a continuous wave signal source to test the resistance against inter-channel interferences by traffic from other technologies operating in the same frequency spectrum. The results showed that the CW signal with a frequency inside the bandwidth of the LoRa signal disrupts the communication at a power difference of about 10 to 20 dB, depending on the spreading factor. Signals with a higher spreading factor are more resilient than those with a lower spreading factor, whereas there is just a minor difference between SF = $9\ldots12$. A continuous wave interference with a frequency slightly outside the bandwidth of the LoRa signal has less impact, as the results have shown. A CW frequency of 100kHz higher or lower than the carrier frequency of the LoRa signal, the interference disrupted the communication at a power difference of about 65dB. The input filter of the receiver attenuates unwanted signals interfering with the LoRa signal and therefore improves the sensitivity of the reception.

Furthermore, we performed measurements with other LoRa signals interfering with the communication. We started by testing the systems‘ reception capabilities with two signals with the same spreading factor colliding during the transmission. First results showed, that the signal disrupted by a delayed signal does not get received correctly if the power of the interfering signal is about 0 to 2dB higher than the first one, depending on the spreading factor. We further investigated the impact of the delay, at which the second signal starts to interfere. The results showed, that signals with a lower spreading factor are more resistant and show a higher reception rate than signals with higher spreading factors, while colliding with a signal with same properties. An investigation of signals with SF = 10 or higher with different receiving powers showed, that the time at which they collide impacts the performance of the higher powered signal. If the stronger signal starts to interfere during the synchronization process, the receiver is not able to decode either of those frames. A collision during the first eight payload symbols, representing the header, the higher powered signal prevents a correct decoding of the weaker signals‘header. In this case the receiver drops the first packet and receives the interfering signal. If the interference starts after the header of the lower powered signal is already decoded correctly, the receiver sticks to the weaker signal and both packets get lost. We measured and simulated the probability of a frame getting dropped during a collision and calculated the packet error probability based on the results. The more nodes are present in a system, the higher is the chance of a frame getting fropped. With duty cycle and listen before talk, LoRa has a limited packet generation rate and is therefore keeping the packet error due to a collision low.
At last, we tested the impact of a collision of two packets with different spreading factor. The results of a sensitivity analysis showed, that the spreading factor of the interfering signal does not have a major impact on the performance of the reception. The approach

of devices at a further distance transmitting with a higher spreading factor and those closer to the receiver with a lower spreading factor does not have a major impact on the receiving capabilities.

Bibliography

[1] Orion Afisiadis, Andreas Burg, and Alexios Balatsoukas-Stimming. *Coded LoRa Frame Error Rate Analysis*. Nov. 2019.

[2] LoRa Alliance. “LoRaWAN 1.1 Regional Parameters”. In: (2017). URL: `https://lora-alliance.org/sites/default/files/2018-04/lorawantm_regional_parameters_v1.1rb_-_final.pdf`.

[3] LoRa Alliance. “LoRaWAN 1.1 Specification”. In: (2017). URL: `https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf`.

[4] Aloys Augustin et al. “A Study of LoRa: Long Range & Low Power Networks for the Internet of Things”. In: *Sensors* 16 (Oct. 2016), p. 1466. DOI: `10.3390/s16091466`.

[5] G. Ferré and A. Giremus. “LoRa Physical Layer Principle and Performance Analysis”. In: *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2018, pp. 65–68.

[6] Claire Goursaud and Jean-Marie Gorce. “Dedicated networks for IoT: PHY / MAC state of the art and challenges”. In: *EAI Endorsed Transactions on Internet of Things* 1 (Oct. 2015), p. 150597. DOI: `10.4108/eai.26-10-2015.150597`.

[7] *iC880A-SPI LoRa Concentrator*. IMST. 2018. URL: `https://wireless-solutions.de/products/lora/radio-modules/ic880a-spi/`.

[8] *Complete Reverse Engineering of LoRa PHY*. Tech. rep. 2019. URL: `https://www.epfl.ch/labs/tcl/wp-content/uploads/2020/02/Reverse_Eng_Report.pdf`.

[9] M. Knight and B. Seeber, eds. *Decoding LoRa: Realizing a modern LPWAN with SDR*.

[10] G. Margelis et al. “Low Throughput Networks for the IoT: Lessons learned from industrial implementations”. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. 2015, pp. 181–186.

[11] B. Reynders and S. Pollin. “Chirp spread spectrum as a modulation technique for long range communication”. In: *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. 2016, pp. 1–5.

[12] Semtech. “LoRa Modulation Basics AN1200.22”. In: (2015). URL: `http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf`.

[13] Semtech. *Semtech SX1257 Rx/Tx Module*. URL: `https://www.semtech.com/products/wireless-rf/lora-gateways/sx1257`.

[14] Ilsun You et al. “An Enhanced LoRaWAN Security Protocol for Privacy Preservation in IoT with a Case Study on a Smart Factory-Enabled Parking System”. In: *Sensors* 18 (June 2018). DOI: `10.3390/s18061888`.

List of Figures

Hiermit erkläre ich, dass die vorliegende Arbeit gemäßdem Code of Conduct, insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, Datum *Harald Eigner*