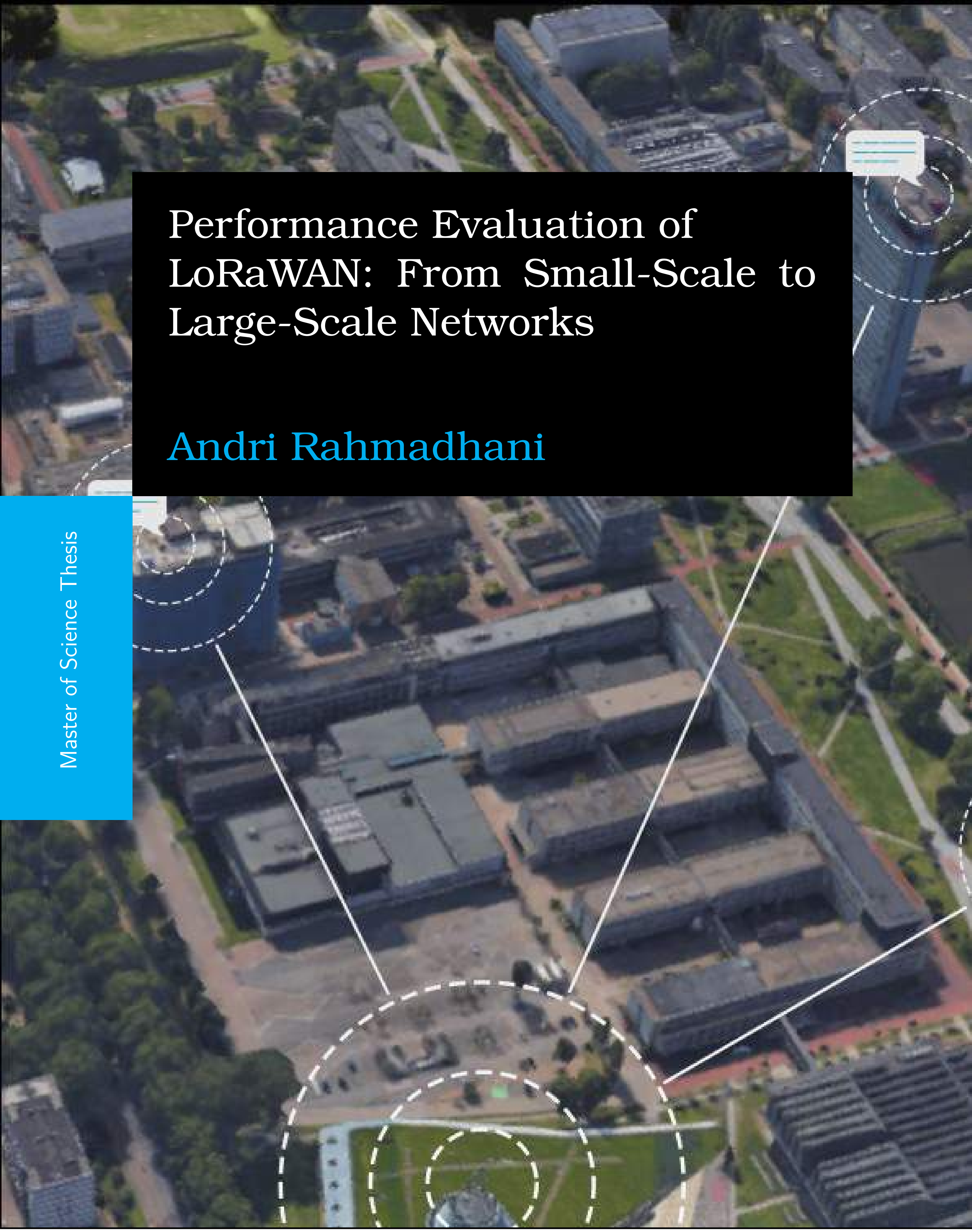
Performance Evaluation of LoRaWAN: From Small-Scale to Large-Scale Networks

Andri Rahmadhani

Master of Science Thesis



Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Embedded Systems

Performance Evaluation of LoRaWAN: From Small-Scale to Large-Scale Networks

Andri Rahmadhani
4505611

Committee members:

Supervisor: Dr.ir. Fernando Kuipers
Member: Dr.ir. Gerard Janssen
Member: Thomas Telkamp

November 15, 2017

Abstract

The uprising of Long Range (LoRa) technology for Low-Power Wide Area Network (LPWAN) has gained popularity in IoT applications. As LoRa utilizes the license-free ISM-bands, many network providers have deployed their LoRa networks that adhere to the LoRaWAN specification. With so many deployed networks, it is likely that devices belonging to neighboring networks will interfere and lead to increased packet loss due to collisions. However, not all colliding frames attribute to packet loss, because of the capture effect in LoRa.

In this thesis, we investigate the performance of LoRaWAN, particularly on frame collision and packet loss, through small-scale measurements, which is then followed by the evaluation in a large-scale setup involving multiple gateways. It is expected that adding more gateways can reduce packet loss. By estimating the Data Extraction Rate (DER) from the measurements and validating the result with simulations, we found that adding more gateways, to some extent, does improve the DER when the capture effect is considered. However, downlink traffic can significantly decrease DER of a single gateway as the currently available gateways operate in half-duplex mode. The impact of different spreading factor parameters to the network performance is also evaluated.

We also characterize and evaluate the two major LoRaWAN networks in the Netherlands, namely The Things Network (TTN) and KPN, which respectively represent the crowdsourced and commercial usage of LoRaWAN networks. The influence of traffic from end-devices belonging to other networks on the channel utilization is evaluated. The results can be used as an empirical basis for developing more accurate models and simulations to better understand the capabilities of LoRaWAN.

Table of Contents

Acronyms

ABP Activation by Personalization

ADR Adaptive Data Rate

AFA Adaptive Frequency Agility

BW Bandwidth

CAD Carrier Activity Detection

CF Carrier Frequency

CR Coding Rate

CRC Cyclic Redundancy Check

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CSS Chirp Spread Spectrum

DER Data Extraction Rate

ED End-device

FEC Forward Error Correction

FSK Frequency Shift Keying

GW Gateway

ISM Industrial, Scientific, and Medical

LBT Listen Before Talk

LoRa Long-Range

LoRaWAN LoRa Wide Area Network

LPWAN Low-Power Wide Area Network

MAC Media Access Control

NS Network server

OTAA Over the Air Activation

SF Spreading Factor

SIR Signal-to-Interference Ratio

SNR Signal-to-Noise Ratio

TP Transmission Power

TTN The Things Network

WSN Wireless Sensor Network

Acknowledegement

First, all praises to Allah for His blessing and for giving me the strengths, knowledge, and opportunity to undertake and complete this thesis. I would like to express my gratitude to my supervisor Dr. Fernando Kuipers for the guidance and support, especially for providing me access to the large-scale dataset used in this thesis. He always comes up with new ideas and challenges that make this thesis more colorful. It is such an honor for me to be guided and supervised by him. I would also like to thank Thomas Telkamp, who is associated with The Things Network (TTN), as well as Bart Hendriks and Ronald Rust from KPN, for the dataset. Not to forget Lembaga Pengelola Dana Pendidikan (LPDP) who gives me the opportunity to pursue a master's degree in Embedded Systems at TU Delft.

Special thanks to Dr. Vijay S. Rao, who recently has finished his Ph.D. (congratulations), for giving me support, ideas, suggestions, and hints related to LoRa and wireless network in general. Also, thanks to Marco for helping me compiling and preparing this thesis for scientific publication. Many thanks to my friends from Embedded Systems, Computer Engineering, and Control System Group (ES-CENDOL): Rizky, Bontor, Angga, Arif, Ridho, Priadi, and Ade who always were there giving me support and cheer.

Finally, I want to thank my family and friends, especially my beloved father who has passed away. His legacy in terms of knowledge and life experience is priceless, encouraging me to be a better person who always gives benefit to the most people.

Andri Rahmadhani
Delft, November 2017

Chapter 1

Introduction

This chapter highlights the research questions and contributions made by this thesis. This is done by first presenting the background and the importance of the work and then to define the research objective and question.

1-1 Research Background

The long-range technology for Low-Power Wide Area Network (LPWAN) has gained popularity in IoT applications. LPWAN enables devices to have long-range communication at a low bit rate using a minimum amount of energy. Recent LPWAN technologies that are available on the market include SigFox, LoRa, NB-IoT, LTE-M, and Weightless. In this thesis, we will evaluate LoRa as it is widely used in Europe, especially in the Netherlands.

LoRa, which stands for Long-Range, is a physical layer utilized to create a long-range communication link [1]. It is adapted from Chirp Spread Spectrum (CSS) modulation that maintains the same low power characteristics as FSK modulation in legacy wireless systems but significantly increases the communication range. CSS has been used for radar due to its broad coverage and robustness against multipath and Doppler shift. In the MAC layer, LoRaWAN defines the communication protocol and system architecture for the network [1].

The recent trend shows that the interest in developing LoRa technology is rapidly increasing. For instance, one of the largest network operators in the Netherlands, KPN, announced full coverage of LoRa networks across the country in June 2016. The Things Network (TTN), an open community which crowdsources the LoRaWAN network, is deploying more gateways to connect the network around the globe. This is supported by the fact that LoRa uses an unlicensed ISM band. It furthermore opens up the possibility for individuals to create private LoRaWAN networks by utilizing the TTN back-end [2] or by creating their own server using open-source software such as LoRa Server [3]. In [4], it can be seen how one can easily build a low-cost single-channel LoRa gateway, even though it does not properly comply with the LoRaWAN specifications.

1-2 Problem Description

The rising popularity of LoRaWAN attracts network operators or individuals to deploy their own LoRaWAN network. However, at some point, the networks may overlap. The overlapping LoRaWAN coverage leads to interference between the networks. For example, if the coverage of gateway A overlaps with the coverage of gateway B and two devices are placed inside the intersection area, a device that connects to gateway A may block the uplink transmission from another device that connects to gateway B and vice versa. The collision can also happen for downlink messages. For example, gateway A and gateway B that belong to different networks, send downlink messages to the corresponding end-device at the same time. These situations are illustrated in Figure 1-1.

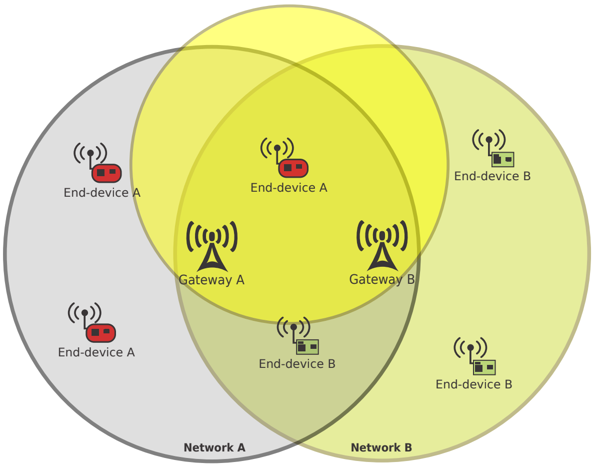


Figure 1-1: Overlapping coverage area may cause inter-network interference.

This problem occurs not only when gateways A and B belong to different network operators, but also when they belong to the same network operator. Intuitively, from the perspective of a gateway, it is similar as if we put new devices into a single network, whereas from the perspective of users, the results may vary, depending on the network settings. In the case of the same network operator, as all gateways in range receive packets from end-devices, there is a chance of getting the packet as long as one gateway receives the packet correctly. In the case of different network operators, most likely there is no coordination and link between the gateways, thereby increasing the probability of packet loss due to collisions. Frame collisions, therefore, becomes the main issue that should be studied better to improve the network performance and scalability.

In [5], research into inter-network interference of pure LoRa networks was evaluated. According to [5], using directional antennae and increasing the number of base stations can reduce packet loss. The result is obtained from a simulator called LoRaSim as described in [6]. LoRaSim is characterized on the basis of real measurements that cover the communication range, traffic generation, and collision behaviour. It was observed from the collision behaviour experiment that the capture effect appears in LoRa. The capture effect, also known as co-channel interference tolerance, is the ability of certain wireless systems to correctly receive a strong signal from one transmitter regardless of the significant interference from other transmitters [7]. Capture effect invalidates the notion of all-packet-loss during collisions. The capture effect occurs in FM receivers as described in [8], and it implies that LoRa encounters such effects due to the frequency modulated signal used in CSS. The capture effect should be taken into account when simulating and deploying LoRa networks as it can be beneficial in terms of reducing packet loss. However, how significant the influence of the capture effect is on the performance as the network scales up needs to be evaluated.

1-3 Research Objective

The aim of this thesis is to extensively evaluate the performance of LoRaWAN starting from small-scale to large-scale networks through practical measurements and focusing more on characterizing frame collisions. The outcome provides an overall performance report on the existing large-scale LoRaWAN networks, possible enhancements, and solutions on reducing packet loss to improve the performance of LoRaWAN.

1-4 Research Questions

Based on the research background and objective described, the main research question is:

To what extent do the number of devices and network densification affect the performance of LoRaWAN?

The following sub-questions will be answered:

1. What are the key factors that degrade the performance of LoRaWAN?
2. How significant does the addition of gateways affect the performance?
3. To what extent can the capture effect reduce packet loss?
4. What are the characteristics of existing real-world LoRaWAN networks?

1-5 Research Methodology

The methodology used in this thesis is depicted in Figure 1-2. The research starts with defining problems as well as collecting information from the literature. This information is then used as the basis for small-scale and large-scale measurements.

The small-scale experiments are performed to characterize features of LoRa technology and to observe frame collisions. Network infrastructure consisting of an application server, network servers, LoRaWAN gateways, and LoRaWAN end-devices are required for the small-scale experiments. The application server is used as a bridge to collect all measurement data from different network operators.

The large-scale measurement dataset comprises aggregated data obtained from the public TTN network. The dataset was collected in 2016 by Blenn and Kuipers [9]. At the end of this thesis work, we received two large-scale measurement datasets, one from KPN and one with respect to the E&A gathering event held at Jaarbeurs, Utrecht. The latter dataset was specifically designed for investigating frame collisions. We also performed simulations using R to compare the results.

The results from both small-scale and large-scale measurements will be discussed. Several possible improvements regarding the thesis work will also be described in the conclusion.

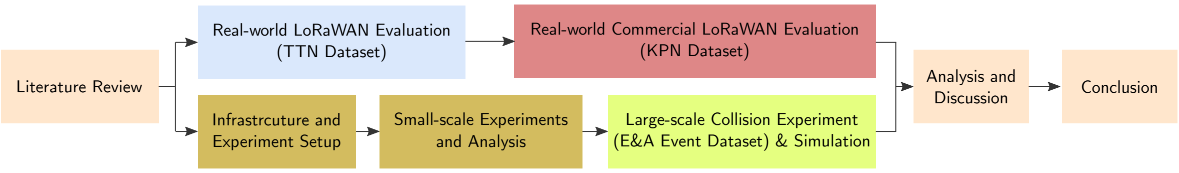

Literature Review
Real-world LoRaWAN Evaluation (TTN Dataset)
Real-world Commercial LoRaWAN Evaluation (KPN Dataset)
Infrastrcuture and Experiment Setup
Small-scale Experiments and Analysis
Large-scale Collision Experiment (E&A Event Dataset) & Simulation
Analysis and Discussion
Conclusion

Figure 1-2: Research methodology

1-6 Organization

This thesis is structured as follows. Chapter 2 describes an overview of LoRa and LoRaWAN, as well as several related works. Chapter 3 presents a literature study on the performance analysis of LoRaWAN. Chapter 4 explains small-scale experiments performed to characterize link and frame collision in LoRaWAN network. Chapter 5 describes measurements and analysis of frame collision in a large-scale setup. Chapter 6 discusses measurements and analysis of large-scale real-world LoRaWAN networks to gain some insight on the existing deployment of LoRaWAN, particularly in the public TTN crowdsourced network. In Chapter 7, an extensive evaluation of the commercial KPN LoRaWAN network will be discussed. Finally, Chapter 8 concludes the overall thesis and provides possible directions for future work.

Chapter 2

LoRa and LoRaWAN

In this chapter, an overview of LoRa and LoRaWAN will be given. The LoRa modulation technique will be explained first as it is used as an underlying PHY layer for LoRaWAN. In the last section, some existing research on LoRa and LoRaWAN will be explained to gain some insights into possible improvements.

2-1 LoRa

LoRa is a proprietary spread spectrum modulation scheme patented by Semtech. It employs a variable data rate by utilizing orthogonal spreading factors, which allows to trade data rate for range or power and to optimize network performance given a fixed bandwidth [10]. Since LoRa is patented, only few documentations are available [11, 12, 13, 10], thereby encouraging some researchers to reverse-engineer LoRa modulation, for example, in research done by Matt Knight [14].

2-1-1 LoRa Modulation

LoRa adopts Chirp Spread Spectrum (CSS) as its modulation scheme. LoRa CSS modulation encodes symbols into multiple signals of increasing (up-chirp) or decreasing frequencies (down-chirp) with a constant chirp rate. It utilizes Forward Error Correction (FEC) in combination with whitening and interleaving that gives longer range than typical Frequency Shift Keying (FSK) modulation [15]. It is said to be robust against Doppler shifts, multipath fading, and shadowing [10].

A LoRa frame is initialized by a long constant-chirp preamble. It is used for the receiver to lock the LoRa signal. The preamble consists of a variable number of symbols, which can be set depending on specifications. The preamble is followed by two reverse-chirp signals that indicate the end of the preamble. Since there is no distinction between the preamble from one LoRa transmitter and another transmitter, the receiver may listen to an unwanted LoRa signal. Therefore, the detection of a preamble is very crucial to the reception of LoRa frames. A waterfall plot of LoRa preamble and data modulation can be seen in Figure 2-1.

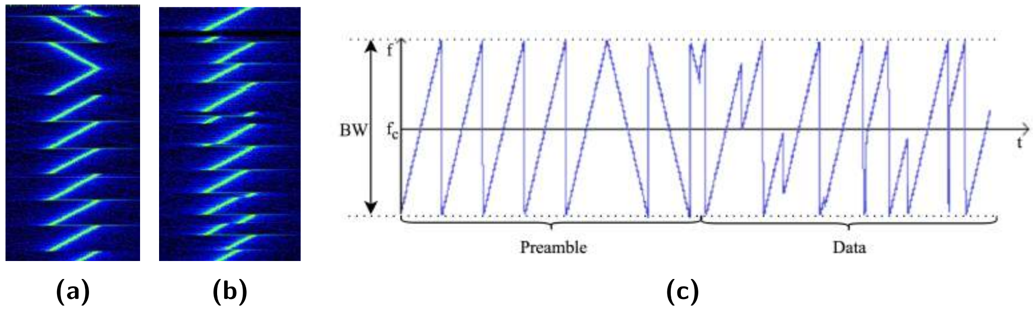


Figure 2-1: LoRa modulation in waterfall plot showing (a) preamble and (b) data modulation, taken from [16]. (c) A complete illustration of frequency variation over time in LoRa [17]. f_c and BW represent center frequency and bandwidth, respectively.

2-1-2 LoRa Configuration Parameters

Spreading Factor (SF)

LoRa receivers can demodulate several orthogonal signals at the same frequency, assuming they have different chirp rates [16]. This feature is supported by the Semtech SX1301 chip, which is commonly used in LoRaWAN gateways. LoRa defines chirp rates as Spreading Factors (SF) with lower SF denoting faster chirps. SF represents the ratio between the symbol rate and chip rate. One symbol is encoded into 2^{SF} chirps. A higher spreading factor improves the Signal-to-Noise Ratio (SNR), resulting in greater sensitivity and range, but also increases the airtime of the packet and power consumption [6]. LoRa supports SF6 to SF12, but only SF7 to SF12 are used in LoRaWAN.

Bandwidth (BW)

The width of radio frequencies being used in the transmission band is defined as bandwidth (BW). An increase in signal BW allows the use of a higher effective data rate, thereby reducing time on air at the cost of reduced receiver sensitivity due to accumulation of additional noise [18]. On the other hand, a lower BW requires more accurate crystals, which may increase the cost of LoRa chips, to achieve a higher sensitivity but sacrificing the data rate. LoRa modems, such as Semtech SX1272/SX1276, support BW from 7.8 kHz to 500 kHz, but in LoRaWAN, only three BW are commonly used, namely 125 kHz, 250 kHz, and 500 kHz.

Carrier Frequency (CF)

LoRa chips support different ranges of frequency. For example, SX1272 chip has frequency range 860 MHz to 1020 MHz [19] while SX1276 has 137 MHz to 1020 MHz [18]. The center frequency (CF) can be programmed with a resolution of 61.035 Hz.

Coding Rate (CR)

Coding Rate (CR) relates to the FEC rate used by the LoRa modem that provides protection against bursts of interference and can be set to either 4/5, 4/6, 4/7 or 4/8 [6]. The higher the

CR, the more protection can be offered, but it increases time on air. The CR value is stored in the LoRa frame header when the explicit header mode is enabled, allowing end-devices with different CR and the same SF, BW, and CF to communicate with each other [18].

Transmission Power (TP)

Transmission power (TP) that is supported by typical LoRa chips ranges from -4 dBm to 20 dBm. However, due to hardware limitations, transmission power higher than 17 dBm is limited to a duty cycle of 1% for end-devices [18]. For gateways, the transmission power can be up to 27 dBm, depending on the regional regulations.

2-1-3 LoRa Frame Format

The LoRa frame format is illustrated in Figure 2-2. LoRa supports explicit and implicit frame formats. In explicit format, the frame contains a header and its Cyclic Redundancy Check (CRC). The header contains information about the payload length in bytes, CR, and whether the 16-bit CRC for the payload is used or not. The header and its CRC are not included in the implicit format and must be manually configured by both sides of the LoRa modems, thereby reducing transmission time [18]. Note that the explicit frame format is used in LoRaWAN, both for uplink and downlink communications [20].

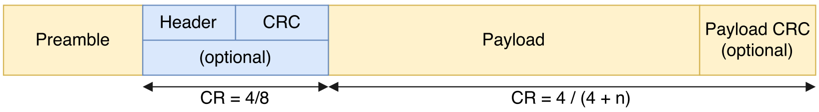


Figure 2-2: LoRa frame format with variable CR $n \in \{1..4\}$ [17]

The preamble can be configured from 6 to 65535 symbols. The preamble ends with additional symbols (2 up-chirps and 2.25 down-chirps) representing SyncWord and Start Frame Delimiter (SFD) as previously shown in Figure 2-1, and thus adding 4.25 symbols to the preamble length. The header has a fixed FEC rate of 4/8. The payload has a variable length ranging from 1 to 255 bytes.

2-1-4 Carrier Activity Detection

Carrier Activity Detection (CAD) is one of the unique features implemented on LoRa. CAD can detect the preamble of other LoRa signals that may exist below the noise floor of the receiver, which is difficult to achieve and relatively slow when using RSSI readings [18]. However, it may be useful to implement a combination of RSSI and CAD detection to select the best possible signal path on the receiver and avoiding a false CAD detection [21].

The CAD detection takes around 2 symbol periods in which the radio is operated in receiving mode. The power required for the CAD process is around 11 mA in full-receiver mode and followed by reduction of power consumption down to 6 mA, depending on SF or BW. The receiver is in full-receiver mode for around half of the total CAD duration. If the CAD yields positive detection, the receiver goes back to standby mode and can be switched to RX mode to receive the on-going transmission afterwards [22].

2-2 LoRaWAN

LoRaWAN is a specification of a Media Access Control (MAC) protocol for controlling low-powered devices in wide area networks. The first specification was released by the LoRa Alliance in January 2015. LoRaWAN is implemented on top of LoRa or FSK modulation in unlicensed Industrial, Scientific, and Medical (ISM) radio bands, which have different regulation depending on region or country. The LoRaWAN stack is illustrated in Figure 2-3.

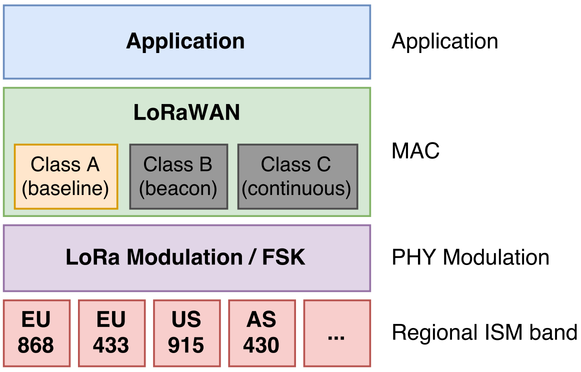


Figure 2-3: LoRaWAN operating classes

2-2-1 Network Architecture

A LoRaWAN network utilizes a star-shaped network topology, which is different from a mesh network used typically in wireless sensor networks. A typical LoRaWAN network architecture is illustrated in Figure 2-4 and it comprises:

- **Node/mote/end-device (ED)**, a low-powered module (sometimes equipped with sensors) that communicates with gateways using LoRa modulation scheme or FSK as explained in the LoRaWAN specification.
- **Base station/gateway (GW)**, an intermediate device that receives messages from end-devices and forwards the messages to the network server over an IP-based backhaul, such as Ethernet, 3G, and 4G.
- **Network server (NS)**, a centralized server or distributed servers that handle messages, including message duplication removal, decoding, and link configuration adjustment of end-devices.

An end-device connects to the network directly through gateways in a one-hop manner, thereby reducing the complexity of the protocol. Unlike in cellular mobile networks, the LoRaWAN star-shaped topology does not have a complex and power-hungry handover procedure between network cells [23]. A message transmitted from end-devices is received by all gateways in range, and each gateway forwards the message to a network server, resulting in duplicated messages, which can be handled by the network server. Transmission from a network server to an end-device is established through a gateway that is selected based on certain algorithms.

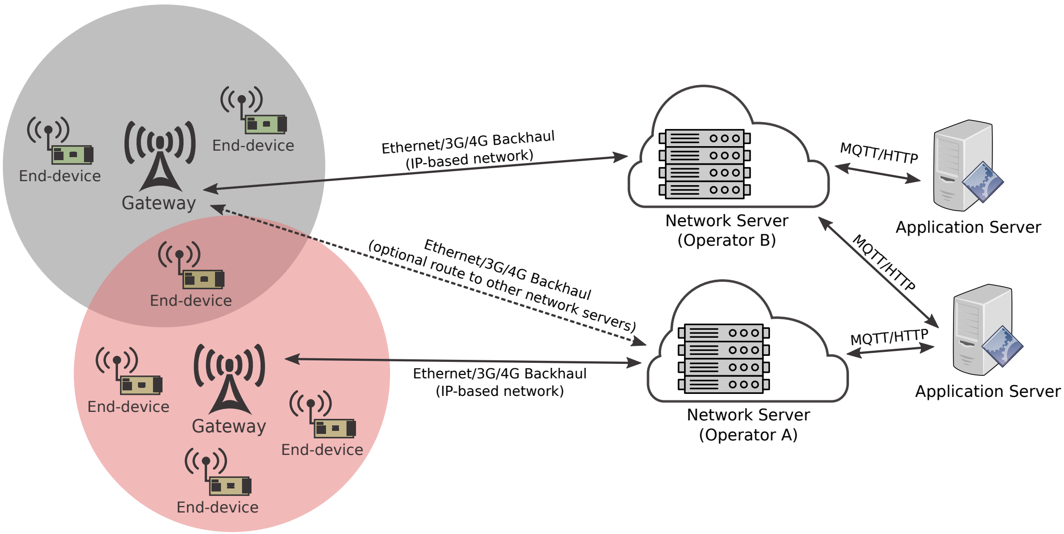


Figure 2-4: Typical LoRaWAN network architecture

2-2-2 Operating Classes

In LoRaWAN, three different operating classes of end-devices are introduced to address various applications. As can be seen in Figure 2-3, the three classes are:

- **Class A**: End-devices trigger the communication channel, and they transmit a message immediately based on their needs. The transmission is scheduled with a small variation based on a random time basis, similar to ALOHA-type protocols. Each uplink transmission is followed by two receive windows, namely RX1 and RX2. As a result, a downlink message from the server at any other time needs to wait for the next uplink message. This Class A operation is mandatory for every LoRaWAN-compatible device and it provides the lowest power consumption.
- **Class B**: End-devices open more receive windows at scheduled times in addition to Class A. To synchronize the timing, the gateway transmits beacon frames that may be transmitted at regular intervals.
- **Class C**: End-devices open receive windows almost continuously. This Class C operation, in fact, has most power consumption, but has the lowest latency.

In this thesis, we focus only on Class A devices as Class A is mandatory for every LoRaWAN device.

2-2-3 Device Activation

End-devices can join a LoRaWAN network by using two methods:

- **Over the Air Activation (OTAA)**: End-devices initiate a join-procedure with the network by sending a join request message, which contains a globally unique end-device identifier (DevEUI), the application identifier (AppEUI), and a 128-bit AES encrypted application key (AppKey). The network then sends a join-accept message that contains a device address (DevAddr), a network session key (NwkSKey), and an application session key (AppSKey). These session data are then stored in the end-device's memory. Whenever an end-device loses its network session info, it has to do a new join-procedure. OTAA offers a fairly secure way to join a network as the network session info is dynamically assigned by the network.

- **Activation by Personalization (ABP)**: End-devices have a pre-programmed network session info (DevAddr, NwkSKey, AppSKey), and thus do not need to initiate a join-procedure. DevEUI and AppEUI are not required for ABP. However, it is less secure than OTAA. If attackers have access to the end-device, they can extract the keys and use them to clone the end-device.

2-2-4 Frequency Regulation and Access Policy

LoRaWAN uses unlicensed ISM radio bands that have certain regulations depending on the region [24]. In Europe (EU), for example, LoRaWAN can use frequency bands 433 MHz (EU433) and 868 MHz (EU863-870) as described in Figure 2-3. This thesis will be focused only on the EU863-870 frequency band given that most of LoRaWAN devices that are available in the market today are targeted for Europe region with this frequency band.

The LoRaWAN specification defines three different 125 kHz channels for the 868 MHz band, as listed in Table 2-1. These channels must be supported by all devices and networks as they are used for the join-procedure. After the devices join the network, the network can instruct the devices to add extra channels to their channel set, up to 16 channels for EU863-870. These channels are used for uplink and downlink communications. In the downlink, the first receive window RX1 uses the same frequency and data rate as the uplink. For the second receive window RX2, LoRaWAN uses a fixed parameter that can be pre-configured. The default RX2 parameter uses frequency of 869.525 MHz, SF12, and 125 kHz bandwidth.

Table 2-1: Default channels for EU863-870 [24]

Modulation	Bandwidth (kHz)	Frequency (MHz)	Duty cycle
LoRa	125	868.10 868.30 868.50	$< 1\%$

The European frequency regulation (ETSI) imposes the use of either duty cycle or Listen Before Talk in combination with Adaptive Frequency Agility (LBT-AFA) as the access policy. LBT-AFA avoids interference by enabling clear-channel assessment (CCA) under which an end-device or gateway listens to the channel first before transmitting a packet, and then switches to another frequency channel if the current channel is being occupied. The long run history of the occupied channels can be used by the network to avoid crowded channels for the next transmissions. The current LoRaWAN specification exclusively uses the duty cycle access policy [24]. As can be seen from Table 2-1, LoRaWAN defines a maximum duty cycle of less than 1%. Duty cycle defines the percentage of time an end-device occupies a specific channel. For example, an end-device that use a channel for 1 second should wait 99 seconds for the next transmission, when 1% duty cycle is used. The waiting time or off-time can be expressed as:

$$T_{\text{off}} = \frac{\text{TimeOnAir}}{\text{DutyCycle}_{\text{subband}}} - \text{TimeOnAir} \tag{2-1}$$

ETSI specifies two bands and five sub-bands for EU863-870, as listed in Table 2-2 [25]. These duty cycles should be respected by both end-devices and gateways, therefore making downlink messages relatively expensive. One solution to tackle this problem is by implementing a smart message scheduler in which downlink messages are transmitted through less busy gateways or higher duty-cycled channels [26].

Table 2-2: Bands and sub-bands for 868 MHz band operation regulated by ETSI

Band	Edge Frequency		Field / Power	Spectrum Access	Bandwidth
	Fe-	Fe+			
g(Note 7)	865 MHz	868 MHz	+6.2 dBm / 100 kHz	1% or LBT AFA	3 MHz
g(Note 7)	865 MHz	870 MHz	-0.8 dBm / 100 kHz	0.1% or LBT AFA	5 MHz
g1	868 MHz	868.6 MHz	14 dBm	1% or LBT AFA	600 kHz
g2	868.7 MHz	869.2 MHz	14 dBm	0.1% or LBT AFA	500 Khz
g3	869.4 MHz	869.65 MHz	27 dBm	10% or LBT AFA	250 kHz
g4	869.7 MHz	870 MHz	7 dBm	No requirement	300 kHz
g4	869.7 MHz	870 MHz	14 dBm	1 % or LBT AFA	300 kHz

LoRaWAN defines five transmission power options, as listed in Table 2-3. The default transmission power of end-devices for EU863-870 is 14 dBm and is limited to this value, except for g3 sub-band, that is, up to 27 dBm. The g3 sub-band is typically used for downlink, which is related to the default RX2 configuration of end-devices.

Table 2-3: Transmission power for EU863-870 [24]

TXPower	Configuration (ERP)
0	20 dBm
1	14 dBm
2	11 dBm
3	8 dBm
4	5 dBm
5	2 dBm

2-2-5 LoRaWAN Frame Format

As described in Section 2-1-3, LoRaWAN uses explicit frame format in which the header and its CRC are included in the frame. The payload CRC is included in the uplink frame, but not in the downlink frame as it is intended for keeping the message as short as possible and minimizing the impact on duty cycle. LoRaWAN uses fixed preamble length of 8 symbols for all regions.

The LoRaWAN message format is illustrated in Figure 2-5. **MACPayload** can be filled up from 1 byte until M bytes. **FRMPayload** can be left empty or used up to N bytes. The maximum value of M and N depend on the data rate listed in Table 2-4 in Section 2-2-6. **MType** contains information of message type: confirmed uplink, confirmed downlink, unconfirmed uplink, unconfirmed downlink, join-request, and join-accept messages. **MIC** is a message integrity check that is used to identify corrupted messages in relation to security. **FPort** is commonly used to distinguish FRMPayload of specific applications. FPort 0 means no FRMPayload, but only MACCommand present in the message. **MACCommand** contains control messages that can be sent from end-devices or gateways. For example, *LinkADRReq* command (CID: 0x03) sent by a gateway to an end-device is used to request the end-device to change its data rate, transmission power, repetition rate or channel. These data are carried on the MACCommand arguments. **FCnt** counts either the number of uplink (**FCntUp**) or downlink (**FCntDown**) messages. Both gateways and end-devices have their FCnt so that they can be compared to prevent replay attacks. **FCtrl** contains frame control messages, such as:

- **ADR** bit indicates whether the network should attempt to optimize data rate or not. ADR bit can be set by the end-device or the network whenever required.
- **ADRAckReq** bit forces the network to send a downlink to respond to the ADR request.
- **ACK** indicates whether the message contains an acknowledgement or not.

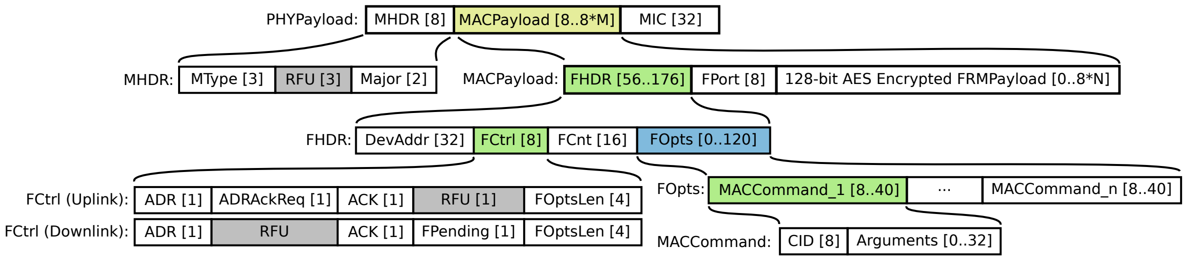


Figure 2-5: LoRaWAN frame format [17]. The square brackets represent the size in bits.

2-2-6 Adaptive Data Rate

Adaptive Data Rate (ADR) is a method in which the actual data rate is adjusted to ensure reliable packet delivery, optimal network performance, and scale for capacity [27]. For example, nodes close to gateway will use a higher data rate, leading to a shorter time on air, and lower output power. The list of supported data rates in LoRaWAN is described in Table 2-4.

ADR can be initiated either by the network or by the end-device. The end-device sends uplink messages with ADR bit set. These ADR-enabled messages are collected in the network and are calculated based on certain algorithms, yielding an optimal configuration, which consists of data rate, transmission power, usable channels, and the number of retransmissions. These settings are carried on *LinkADRReq* command sent by the network. If the new parameters are applied successfully, the end-device will acknowledge them by sending *LinkADRAns*.

Table 2-4: Transmission data rate for EU863-870 [24]

DataRate	Modulation	Spreading Factor	Bandwidth (kHz)	Bit rate (bit/s)	Maximum MACPayload size [M] (byte)	Maximum FRMPayload size [N] (byte)
0	LoRa	SF12	125	250	59	51
1	LoRa	SF11	125	440	59	51
2	LoRa	SF10	125	980	59	51
3	LoRa	SF9	125	1760	123	115
4	LoRa	SF8	125	3125	250	242
5	LoRa	SF7	125	5470	250	242
6	LoRa	SF7	250	11000	250	242
7	FSK	-	-	50000	250	242

The algorithm for ADR calculation is not explained in the LoRaWAN specification. It depends on the network operators to implement their own algorithms. For example, TTN adopts a simple rate adaptation algorithm provided by Semtech, which can be seen in [28]. As ADR is calculated and analyzed on the network back-end, it drastically reduces the complexity on the end-devices. However, it requires downlink messages, which is very expensive due to the duty cycle limitations of gateway. Research toward developing efficient ADR algorithms is still open, for example, utilizing machine learning to obtain the optimum ADR parameters.

2-3 Related Work

This thesis is closely related to the research done by Bor et al. [5] who have investigated the use of directional antennae and the addition of new LoRa base stations on the reduction of packet loss due to inter-network interference. LoRaSim was used to evaluate the performance in terms of Data Extraction Rate (DER), which represents the ratio of received messages to transmitted messages over a period of time. DER measures the overall network performance instead of per individual node. Capture effect was also included in LoRaSim to achieve more realistic results. They concluded that deploying multiple base stations gives better result than employing directional antennae on end-devices. Note that the simulation was performed using the same settings for all nodes, no ADR, no channel hopping, and only uplink messages were considered. It was not tied to a specific protocol, such as LoRaWAN.

This thesis is also related to the work done by Blenn and Kuipers [9], which extensively measured and analyzed a real-world, large-scale LoRaWAN network (TTN) within a time-frame of 8 months. They analyzed frame payloads, signal quality, and spatio-temporal aspects, to estimate the performance of LoRaWAN. They also simulated packet loss, as well as the effect of confirmed and downlink packets, based on the empirical results. The results show that not all channels provided by the network are used evenly as the end-devices in the dataset use ABP, leading to more packet loss. Also, increasing the number of confirmed and downlink packets induces more packet loss due to the half-duplex mode and duty cycle limitation of gateways. Implementing ADR can reduce packet loss but is not recommended for mobile scenarios as it may increase communication overhead. In addition, using a lower SF and limiting the transmission power can reduce packet loss, but it is preferable to use the lowest SF instead of limiting the transmission power as it can reduce time on air and prevent collisions.

Scalability and Interference

Li et al. [29] proposed the use of stochastic geometry to model interference in the time-frequency domain for LPWAN (SigFox and LoRaWAN). By characterizing the outage probability and throughput, the performance of LPWAN for narrowband (SigFox) and wideband (LoRaWAN) can be evaluated. In the case of LoRaWAN, only time domain is considered as every packet is spread in the available bandwidth, e.g., 125 kHz, meaning that there is no partially overlapping frequency in LoRaWAN. Georgiou et al. [30] also investigated the scalability of LoRaWAN using stochastic geometry to model the uplink coverage and the effect of interference on a single LoRaWAN gateway. The results show that the end-device coverage drops exponentially as the number of end-devices increases.

Reynders et al. [31] compared the robustness against interference between LoRaWAN and SigFox by using simulations. Only uplink messages were considered in the simulations. The physical layer propagation was modelled using Hata model. For LoRaWAN, three different interferers were simulated, namely continuous wave (CW), Additive white Gaussian noise (AWGN), and SF12 interferer. The simulations used the assumption that different SFs are orthogonal, and thus messages with different SFs can be decoded simultaneously. Other SFs are regarded as noise from the viewpoint of the receiver. Nevertheless, the results show that at least Signal-to-Interference Ratio (SIR) of -18 dB is needed to decode the message correctly while receiving other messages. From these results, one can see the effect of intra-network and inter-network interferences on the network performance. Devices in the network cell edge will suffer most because of the lower signal and the higher probability of collisions, as well as the increased noise level.

Practical measurements on interference level in the European 868 MHz for LoRa and SigFox were performed by [32]. The measurements used Rohde & Schwarz TSMW radio network scanner and took place in the city of Aalborg, Denmark in five distinct areas, namely industrial, residential, hospital, business park, and shopping mall. The results show that for the default LoRa channels, 22-33% of signals exceeding -105 dBm was observed in the shopping area and business park, which can deteriorate LoRa signals. The other places have less than 3% probability of interfering signals.

In [22], Bor et al. investigated performance and capability of LoRa for IoT. The research includes evaluation of SF orthogonality, concurrent transmissions, and CAD. An energy-efficient multi-hop protocol called LoRaBlink was also proposed and evaluated. In [6], Bor et al. developed LoRaSim to evaluate LoRaWAN scalability. The results showed that the current LoRaWAN deployments do not scale well. However, implementing dynamic transmission parameter selection or adding more gateways could make LoRaWAN scale quite well. Bankov et al. [33] did a similar research regarding the capacity limit of a LoRaWAN network using simulation, which utilized the Poisson process to generate messages and the Okumura-Hata model to calculate path loss. The transmissions are simulated using various data rates with different probabilities. The result shows that a single LoRaWAN gateway can be easily congested. Adding more gateways can solve the problem, but may contribute to inter-network interference.

Recent research done by Petäjäjärvi et al. investigates the robustness of LoRa against Doppler shift [34] and its susceptibility to inter-network interference using practical measurements [35]. It was observed that using SF12 with relative speed exceeding 40 km/h, the performance of

LoRaWAN degrades, whereas when the relative speed is below 25 km/h, the performance is relatively stable. Using lower SF values may improve performance due to the less Doppler shift influence. In real-world deployments, it was found that concurrent transmissions using different data rate with the same frequency can influence each other. Lower data rates are more susceptible to interference. Payload size can also affect the probability of receiving correct messages, depending on data rate and power level of the transmitter and its interferers.

A comprehensive study on LoRa and LoRaWAN was accomplished by Augustin et al. in [17]. The study opens up possibilities for future research regarding the feasibility of implementing collision-aware protocol by utilizing Carrier Sense Multiple Access (CSMA) or Listen Before Talk in combination with Adaptive Frequency Agility (LBT-AFA). Adelantado et al. [36] suggested future study regarding collision prevention by exploring more channel hopping methods and enabling network coordination.

Coverage

Petäjäjärvi et al. [37] also published articles related to coverage [38], scalability and capacity [39], as well as its use case specifically in indoor environment for health and well-being monitoring [37]. The outdoor coverage reported in [38] was up to 5 km in an urban environment, 15 km in open space, and 30 km on water. The experiments were performed using a node operating in the frequency of 868 MHz, SF12, and 14 dBm transmission power. The indoor coverage was evaluated in [37] using a sensor node positioned close to a human body. It was observed that, on average, 96.7 % of the unconfirmed packets were successfully delivered from almost all indoor locations that span for over 570 meters North to South and over 320 meters East to West. Scalability and capacity were evaluated in [39]. Uplink throughput and data transmission time were used as metrics to analyze the maximum number of end-devices that can be handled by a single LoRaWAN gateway. It was observed that the capacity of the uplink channel did not exceed 2 kbit/second, depending on the distance between the end-devices and the gateway. This implies that even though thousands of end-devices sending small payload per day can be handled by a single gateway, only small portion of end-devices can be located far away from the gateway. In terms of reliability, LoRaWAN is not suitable for low-latency and high-reliability applications due to its poor performance on downlink, but may be used for non-critical or environment monitoring applications.

Iova et al. [40] conducted experiments on LoRa coverage in a mountain to observe the effect of environmental factors and also the effect of different end-device antennas. The results show that the coverage decreases when end-devices are moved to a dense forest. The high temperature also degrades the signal quality. However, transmission power is not the predominant factor affecting the coverage. Instead, it is more dependent on BW, SF, and CR. This makes LoRa different from typical Wireless Sensor Network (WSN) technologies.

Downlink Traffic

Most research on LoRa only discusses uplink communication. Recently, Pop et al. [41] investigated the impact of downlink traffic on LoRa by extending the LoRaSim simulator. Several findings were observed from the simulation results. First, unreliable downlink channel is caused by gateway duty cycle limitations and collisions. Second, network goodput drops significantly as the number of retransmissions increases. Third, failure to obtain an ACK does not mean that the link quality is poor. Fourth, different scenarios may need a different number of retransmission attempts. Finally, LoRaWAN networks do not scale if many nodes request ACKs.

Chapter 3

Performance Evaluation of LoRaWAN: A Literature Study

In this chapter, a literature review on the performance evaluation of LoRaWAN networks will be presented. Several findings and methods from existing research will be discussed in detail, extending the related work section in Chapter 2.

3-1 Performance Evaluation Metrics

3-1-1 Packet Delivery Ratio

Packet delivery ratio (PDR) measures the number of packets successfully received over the total number of transmitted packets. Due to the nature of a LoRaWAN transmission where every gateway in the vicinity can receive the packet and is aggregated in the network server, we classify PDR based on two different perspectives:

- **Device to gateway PDR**, measures the number of packets successfully received with correct CRC by individual gateways over the total number of transmitted packets.
- **End-to-end PDR**, measures the number of packets successfully received by the application layer without any duplicates. Only one packet will be considered regardless of the number of duplicate packets received from multiple gateways.

3-1-2 Data Extraction Rate

Data extraction rate (DER) is used as a metric for measuring network performance, particularly on the reliability of the network from the network perspective. DER represents the ratio of received messages to transmitted messages over a period of time [5]. DER is similar to the end-to-end PDR. $DER = 1$ when all transmitted packets are successfully received by the application layer. DER measures the network performance as a whole, not per individual node as end-to-end PDR does. To avoid confusion, we will use DER for evaluating the network performance, particularly for the large-scale networks.

3-2 Coverage Estimation

The coverage of LoRaWAN gateways needs to be characterized to check whether the transmissions interfere with other networks. Several factors such as link budget, receiver sensitivity, as well as path loss and shadowing need to be considered to estimate the coverage.

3-2-1 Link Budget

Link budget analysis is commonly used to determine the amount of received power, usually measured in dBm, at the receiver side. It includes gains and losses along the transmission channel. Assuming no cable loss at the transmitter and receiver, the link budget formula can be expressed as [42]:

$$P_{RX(dBm)} = P_{TX(dBm)} + G_{TX(dBi)} - L_{PL(dB)} + G_{RX(dBi)}\,, \tag{3-1}$$

where P_{RX} is the received power at the receiver, P_{TX} is the transmission power at the transmitter, G_{TX} is the transmitter antenna gain, G_{RX} is the receiver antenna gain, and L_{PL} is the path loss of the channel, which includes shadow fading. As LoRa signals can travel over long distances, the loss is predominantly originated from the free-space path loss L_{FS}. It implies that the receiver requires a high receiver sensitivity, that is around -140 dBm.

A LoRaWAN gateway receives packets from end-devices and forwards information of signal quality in the form of RSSI (dBm) and SNR (dB) to the network server. Some gateways may report RSSI with noise and some others may not. For instance, we figured out from the KPN developer portal that the gateway provides two RSSI values, with noise correction (`LrrESP` or `rssi_sig`) and without noise correction (`LrrRSSI` or `rssi_chan`). It is important therefore to check RSSI reading mechanism in the gateways and network servers. The RSSI value without noise correction can be expressed as the sum of actual received power P_{RX} and noise N in linear scale:

$$RSSI_{(Watt)} = P_{RX(Watt)} + N_{(Watt)}\,. \tag{3-2}$$

Since LoRa signals can be decoded below the noise floor, we can substitute the noise factor from $SNR_{(Watt)} = \frac{P_{RX(Watt)}}{N_{(Watt)}}$, leaving only the actual received power. Thus, P_{RX} (`LrrESP`) can be expressed as:

$$P_{RX(dBm)} = RSSI_{(dBm)} + SNR_{(dB)} - 10\log_{10}\left(1 + 10^{0.1SNR_{(dB)}}\right)\,. \tag{3-3}$$

This correction significantly alters the received power values for SNR below zero. Figure 3-1 shows the comparison between RSSI and ESP for different SNR values. The figure is illustrated for $RSSI = -110$ dBm.

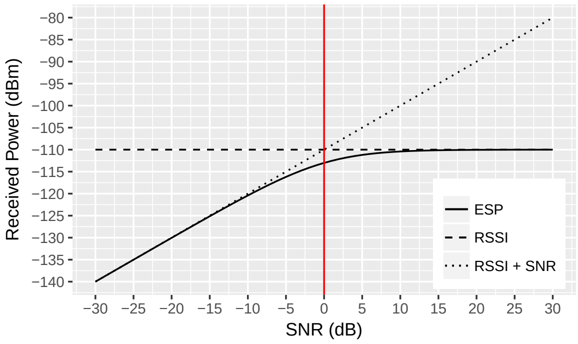


Figure 3-1: Comparison of RSSI and ESP over SNR

3-2-2 Receiver Sensitivity

The LoRa receiver has a minimum power threshold to detect LoRa signals, known as receiver sensitivity, which is denoted by S_{RX}. The calculated link budget P_{RX} needs to be larger or equal to the S_{RX}. For LoRa transceivers, the receiver sensitivity is expressed as [11]:

$$S_{R(dBm)} = -174 + 10\log_{10}(BW_{(Hz)}) + SNR_{(dB)} + NF_{(dB)}\,, \tag{3-4}$$

where SNR represents the minimum signal-to-noise ratio that is required to demodulate LoRa signals, BW stands for receiver bandwidth, and NF stands for receiver noise figure. The required SNR depends on the SF used. As can be seen in Table 3-1, the higher the SF, the lower the required SNR. Increasing SF improves the sensitivity by 2.5 dB.

Table 3-1: SNR required at the receiver input for different SFs [18]

SF	Chips/Symbol	Demodulator SNR
7	128	-7.5
8	256	-10
9	512	-12.5
10	1024	-15
11	2048	-17.5
12	4096	-20

Using Equation 3-4, SNR values in Table 3-1, and taking into account the typical 6 dB noise figure of the receiver, the receiver sensitivity can be calculated for different data rates. Note that the NF and SNR can be slightly different for different LoRa chipsets. The results are listed in Table 3-2. The SX1301 chipset is commonly used for LoRaWAN gateways as it supports multiple channels and multiple SFs and gives a higher sensitivity level, up to -139.5 dBm for SF12BW125. Note that doubling the bandwidth decreases the sensitivity by approximately 3 dB, which is contrary to increasing the SF.

3-2-3 Path Loss and Shadowing

Path loss and shadowing attenuate signals and give variation of received power over distance. Path loss is caused by dissipation of power radiated by the transmitter and by effects of

Table 3-2: Receiver sensitivity of different LoRa chipset [11, 18, 19]

SF	SX1301			SX1276			SX1272		
	Bandwidth (kHz)			Bandwidth (kHz)			Bandwidth (kHz)		
	125	250	500	125	250	500	125	250	500
7	-126.5	-123.5	-120.5	-123	-120	-116	-124	-122	-116
8	-129.0	-126.0	-123.0	-126	-123	-119	-127	-125	-119
9	-131.5	-128.5	-125.5	-129	-125	-122	-130	-128	-122
10	-134.0	-131.0	-128.0	-132	-128	-125	-133	-130	-125
11	-136.5	-133.5	-130.5	-133	-130	-128	-135	-132	-128
12	-139.5	-136.5	-133.5	-136	-133	-130	-137	-135	-129

the propagation channel [43]. Shadowing gives variation to the path loss, which is caused by obstacles between transmitter and receiver. Path loss and shadowing are also known as large-scale propagation effects as they occur over relatively large distances. They can be used to estimate outage probability or coverage area of gateway.

To calculate path loss from measurement data, we can combine Equation 3-1 and Equation 3-3. Thus, the path loss can be described as:

$$\begin{aligned} L_{PL(dB)} = P_{TX(dB)} &- RSSI_{(dBm)} \\ &- SNR_{(dB)} + 10\log_{10}\left(1 + 10^{0.1 \cdot SNR_{(dB)}}\right) \\ &+ G_{TX(dB)} + G_{RX(dB)} \end{aligned} \tag{3-5}$$

In this thesis, a simplified path-loss model, known as log-distance path-loss model [43], was used for estimating path loss and its dependency on landscape type, such as urban, suburban, and rural areas. The log-distance path-loss model in combination with shadow fading can be expressed as [6]:

$$L_{PL}(d)_{[dB]} = \overline{L_{PL}}(d_0)_{[dB]} + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma[dB]}\,. \tag{3-6}$$

where d denotes the distance from the transmitter for which path loss is measured, d_0 denotes the reference distance (typically 10-100 m for outdoors), $\overline{L_{PL}}(d_0)$ denotes the average path loss at reference distance d_0, γ denotes path loss exponent, and X_σ represents shadow fading contribution, which has a normal distribution with zero mean and σ standard deviation. Note that the log-distance model is valid only at transmission distance $d > d_0$ due to scattering phenomena [43].

Substituting Equation 3-6 without shadowing variations into Equation 3-1 gives the gateway coverage radius estimation R, where $P_{RX}(d = R) = S_{RX}$. The coverage radius can be expressed as:

$$R = d_0 \cdot 10^{\left(P_{TX} + G_{TX} + G_{RX} - S_{RX} - \overline{L_{PL}}(d_0)\right)/10\gamma}\,. \tag{3-7}$$

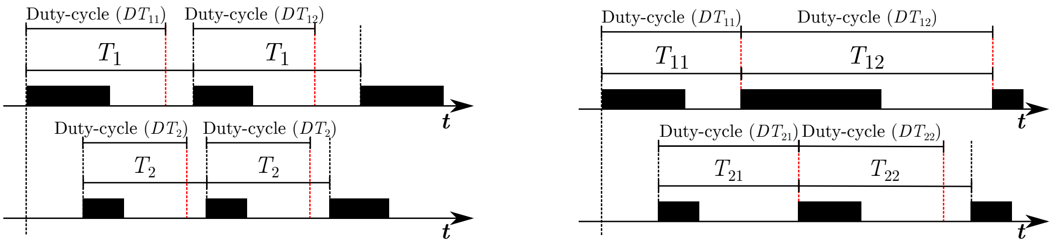
Bor et al. [6] performed some measurements and obtained $\overline{L_{pl}}(d_0) = 127.47$ dB, $\gamma = 2.08$ and $\sigma_{SF} = 3.57$ with $d_0 = 40$ m. They used the SX1272-based platform as transceivers and placed them indoor. These parameters yield a coverage radius of 1.1 km for SX1301, 754 m for SX1276, and 842 m for SX1272 using SF12 and BW 125 kHz. They used these parameters in LoRaSIM, representing worst-case coverage for urban areas.

3-3 Traffic Characteristic

The essence of characterizing LoRaWAN traffic is to determine the probability of frame collisions in relation to packet loss. Such collisions depend on several factors, such as periodicity, traffic directions, and channel occupancy.

3-3-1 Periodic and Irregular Traffic

Traffic periodicity defines the regularity of packet transmission intervals per node. As LoRaWAN use cases are mostly intended for low power and low data rate machine to machine (M2M) communications, the traffic is different from traditional human-type communication (HTC) [44]. Depending on the applications, such machine-to-machine communications (MTC) can be classified into three patterns: **periodic update (PU)**, **event-driven (ED)**, and **payload exchange (PE)** [44]. In terms of interval regularity, those three patterns can be generalized into two basic categories: **regular** and **irregular**. Figure 3-2 illustrates regular and irregular periodicity of MTC traffic in a LoRaWAN network.



(a) Regular period

(b) Irregular period

Figure 3-2: Traffic periodicity in a LoRaWAN network

The regular period or periodic traffic includes PU traffic in which an end-device transmits status reports to a network server on a regular basis. The transmission can be triggered by an end-device in which subsequent transmissions are sent at regular intervals. In LoRaWAN, as the end-device is tied to duty cycle limitation, the period should be equal or larger than the duty cycle. Several examples of PU traffic in LoRaWAN include smart meter reading, water level monitoring, health monitoring and location reporting.

The irregular traffic includes ED, which is sporadic and is triggered by an event after which the corresponding data needs to be transmitted immediately. In the case of bursty ED traffic, the transmission needs to wait before transmitting the next packet due to the duty cycle limitation. Unlike the periodic traffic, the end-device sends a packet immediately after the duty cycle waiting time has expired. The duty cycle time depends on LoRaWAN parameters so that the transmission interval may not be the same, especially when using ADR. Several examples of ED traffic in LoRaWAN are smart lighting controller, waste management, and early-warning disaster alert.

The PE traffic is triggered after an event and is followed by either PU or ED traffic. It consists of all cases in which a significant amount of data is exchanged between the end-devices and the network server, e.g., data streaming caused by an alarm. This kind of traffic forms bursty traffic, but it is still hindered by LoRaWAN duty cycle limitation, thus limiting the use case of LoRaWAN, especially for data streaming.

3-3-2 Uplink and Downlink Traffic

The traffic direction, either from an end-device to a network server (uplink) or from a network server to an end-device (downlink), can influence the performance of a LoRaWAN network. Uplink and downlink traffic comprise different message types:

- **Uplink**: join-accept, unconfirmed data up, confirmed data up
- **Downlink**: join-request, unconfirmed data down, confirmed data down

A confirmed uplink message requires an ACK from the network server, which can be sent in the form of a confirmed or unconfirmed downlink message. Similarly, a confirmed downlink message requires an ACK from the end-device, which can be sent using a confirmed or unconfirmed uplink message. The join-accept and join-request messages are used for OTAA.

Apart from the end-device, a LoRaWAN gateway has to respect the duty cycle which limits the number of downlink messages. For LoRaWAN class A, downlink messages are transmitted by means of unicast, meaning that one gateway sends downlink messages to a particular end-device. Figure 3-3 depicts uplink and downlink transmissions for LoRaWAN Class A.

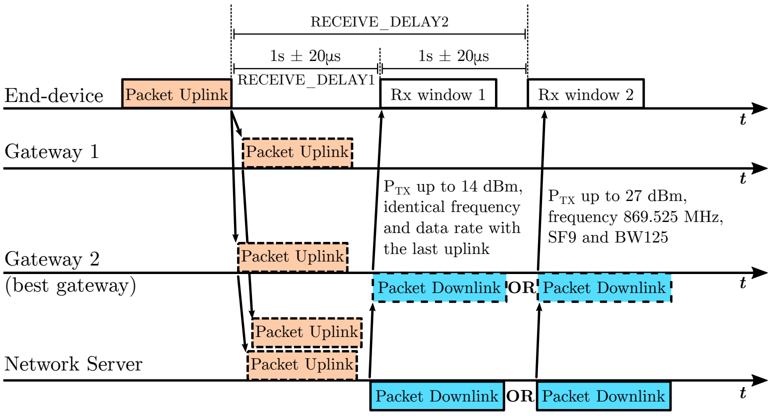


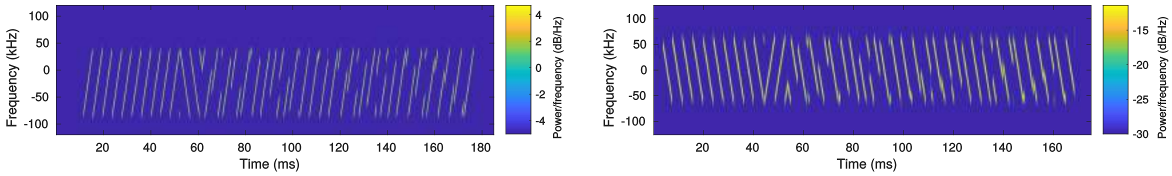
Figure 3-3: Uplink and downlink transmission of LoRaWAN Class A

In LoRaWAN class A, the end-device opens two receive windows, RX1 and RX2, after sending an uplink packet. RX1 is opened approximately 1 second, depending on the `RECEIVE_DELAY1` value in the corresponding region, after the end of uplink transmission. RX1 will remain open until timeout, which can be expressed as:

$$RxTimeout = SymbTimeout \cdot T_S \,, \tag{3-8}$$

where $SymbTimeout$ represents the timeout values (symbols), and T_S denotes the symbol time. According to [45], the default value of $SymbTimeout$ is 8 symbols for $SF <= 9$ and 5 symbols for $SF >= 10$. If data are received and validated before RX1 timeout, the end-device will not open RX2 window. Otherwise, the end-device will open RX2 window approximately 2 seconds after the end of uplink, depending on `RECEIVE_DELAY2` value. RX1 frequency and data rate are identical with the parameters used in the last uplink transmission, and the downlink packet can be transmitted with P_{TX} up to 14 dBm. RX2 opens using a fixed setting, i.e., frequency 869.525 MHz, SF12BW125 with P_{TX} up to 27 dBm. This configuration can be modified by the network server. RX2 is typically used to compensate end-devices at the cell-edge in which they tend to transmit packets using the highest SF (SF12), thereby requiring longer time on air and processing time.

One should notice that a gateway cannot listen to all channels while transmitting a downlink message, and thus dropping all incoming uplink messages. It implies that in the case of a single gateway, a high number of downlink messages may incur more uplink packet loss. Such packet loss depends solely on the gateway busy time. As uplink and downlink messages are transmitted with different polarization by using I/Q inversion, both messages will not collide even though they overlap in time and have the same frequency and data rate [41]. Figure 3-4 shows the spectrogram of LoRaWAN uplink and downlink frames received by a Software Defined Radio device (RTL-SDR).



(a) Uplink

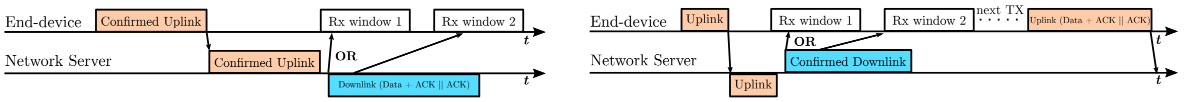
(b) Downlink

Figure 3-4: Spectrogram of uplink and downlink LoRaWAN frames received by RTL-SDR

An end-device will only receive messages from gateways, not from other end-devices. Similarly, a gateway will only receive messages from end-devices, preventing interference from other gateways. However, a downlink message can be interfered by downlink messages sent from gateways in different networks since in a single network, only the best selected gateway will send the downlink message to the corresponding end-device.

Acknowledgement Procedure

A confirmed message needs to be acknowledged by the receiver, which can be either the end-device or the network server. The acknowledgement is sent by enabling the acknowledgment bit (ACK) in the message data frame. For a confirmed uplink message, the acknowledgement will be sent once one of the receive windows is opened by the end-device, whilst for a confirmed downlink message, it is up to the end-device to transmit the acknowledgement to the network server. It should be noted that acknowledgements correspond only to the latest received message, and they are not retransmitted [20]. Figure 3-5 illustrates the acknowledgement procedure for LoRaWAN confirmed messages.



(a) Confirmed uplink

(b) Confirmed downlink

Figure 3-5: Acknowledgement procedure for LoRaWAN confirmed messages

Retransmission

Packet retransmission may be enabled so that whenever an end-device does not receive an acknowledgement after sending a confirmed uplink message, the end-device will retransmit the packet through another channel while still obeying the duty cycle regulation. Packet retransmission follows a confirmed downlink message when the network server does not receive the acknowledgement from the end-device. A retransmission message is transmitted after a random timeout (`ACK_TIMEOUT`) with an interval between 1 second to 3 seconds started from the end of RX2 window. According to the LoRaWAN specification, the recommended number of retransmissions is up to 8 times. If after eight consecutive trials the packet is not yet acknowledged, the MAC layer will send an error status to the application layer after which the packet will be retransmitted or discarded. Note that retransmission may use lower data rate after two consecutive attempts in order to regain connectivity.

3-3-3 Time on Air

Time on air denotes the time needed for transmitting a frame from node to gateway or vice versa. It represents the occupancy of a channel, which is related to duty cycle. The time on air can be derived from symbol time T_S by solving the following equations [18]:

$$T_S = \frac{2^{SF}}{BW} \tag{3-9}$$

$$n_{\texttt{payload}} = 8 + max\left(\left\lceil\frac{8PL - 4SF + 28 + 16CRC - 20IH}{4 \times (SF - 2DE)}\right\rceil \times (CR + 4)\,, 0\right) \tag{3-10}$$

$$T_{\texttt{payload}} = n_{\texttt{payload}} \times T_S \tag{3-11}$$

$$T_{\texttt{preamble}} = \left(n_{\texttt{preamble}} + 4.25\right) \times T_S \tag{3-12}$$

$$T_{\texttt{packet}} = T_{\texttt{preamble}} + T_{\texttt{payload}} \tag{3-13}$$

where $n_{\texttt{payload}}$ denotes the number of symbols used for transmitting the PHY payload. $n_{\texttt{preamble}}$ denotes the programmed symbol length for the preamble, which is typically 8 symbols length for LoRaWAN. If $n_{\texttt{preamble}}$ is set to zero, two up-chirps and two down-chirps with an addition of 1/4 up-chirp can still be observed [15]. Those two up-chirps denote the sync-word while the two down-chirps represent the sync-pattern or start frame delimiter (SFD), and thus adding 4.25 symbols to the preamble time. PL denotes the size of the PHY payload in bytes (1 to 255), and SF denotes spreading factor. It should be noted that PL is the size of PHY payload, not the actual data or FRMPayload. At least 13 bytes of LoRaWAN header should be added to the calculation if we know only the size of FRMPayload, assuming no FOpts are present in the payload. IH denotes implicit header in which $IH = 0$ when the header is enabled and $IH = 1$ when no header is present. As a consequence, the length of explicit header can be obtained, i.e., 20 bits. DE represents low data rate optimization in which $DE = 1$ when it is enabled. DE is enabled when the symbol time is larger than 16 ms, e.g., for SF11 and SF12 with 125 kHz bandwidth. CR denotes coding rate in which value 1 corresponds to coding rate 4/5 and 4 to 4/8.

3-4 Frame Collision Characteristics

3-4-1 Interference Source

Interference in LoRaWAN networks may be originated from other LoRaWAN end-devices transmitting LoRa signals and from other non-LoRa devices in which they occupy the same frequency bands. This interference is widely known as in-band interference. Out-of-band interference, on the other hand, may be originated from a transmitter or device that operates beyond the desired frequency band but still emits interfering signals in the desired frequency band. Out-of-band interference can result in adjacent channel interference (ACI) between LoRaWAN channels. In this thesis, we focus on the interference caused by the presence of other LoRa signals. The effect of interference from coexisting technologies as well as the presence of out-of-band interference may be interesting to study in future work.

Spreading Factor Noise

As LoRaWAN employs orthogonal SF, messages with different SF can be decoded simultaneously as long as they have SNR above a certain threshold value, even when the messages are transmitted using the same CF and BW. All data rates used in LoRaWAN specification are orthogonal as all the SF and BW combinations have different slope or chirp rate R_{chirp}, which is expressed as:

$$R_{\text{chirp}} = \frac{BW}{T_S} = \frac{BW \cdot BW}{2^{SF}} \tag{3-14}$$

The number of messages that can be decoded simultaneously depends on the receiver characteristic, particularly on the number of receive paths. An SX1301-based gateway has 8 receive paths, each listens to all SFs and can be programmed to different frequencies with a fixed 125 kHz bandwidth. The gateway also supports 2 additional receive paths, one for LoRa and another one for FSK modulation. The additional LoRa receive path can only listen to a particular SF and BW, e.g., SF7 with 250 kHz bandwidth. Multiple LoRa signals received by this gateway will be decoded by multiplying the signal with their inverse chirp generated using the corresponding SFs, thus treating other signals as noise.

According to [18], a LoRa receiver typically can handle interference from another LoRa signal having the same SF, BW, and CF with the desired signal as long as the interferer is received 6 dB below the desired signal. This threshold value forms a co-channel rejection (CCR), which measures the ability of a receiver to reject an unwanted signal that occupies the same channel. CCR is defined as the received power ratio between the interferer and the desired signal, which can also be seen as the inverse of SNR margin. Table 3-3 shows computed co-channel rejection values in dB for all SF combinations as reported in [46]. The negative value means that the desired signal can be decoded if the received power is x dB higher than its interferer, whilst the positive value denotes that the desired signal can still be decoded even when its interferer is up to x dB higher than the desired signal.

Table 3-3: Co-channel rejection (dB) for all SF combinations for the desired signal (S) and interferring signal (I) [46]

S \ I	7	8	9	10	11	12
7	-6	16	18	19	19	20
8	24	-6	20	22	22	22
9	27	27	-6	23	25	25
10	30	30	30	-6	26	28
11	33	33	33	33	-6	29
12	36	36	36	36	36	-6

Uplink-Downlink Frames Interference

As mentioned in Section 3-3-2, a downlink message sent by a LoRaWAN gateway uses I/Q inversion for transmission so that whenever it collides with uplink messages that are transmitted using the same channel and data rate, it will be regarded as noise by the other gateways. Hypothetically, if we assume the noise possesses additive property, the SNR will be affected, causing packet loss when the measured SNR is lower than SNR threshold. In this thesis, we do not take into account such interferences due to the time constraints and limitation of hardware.

3-4-2 Capture Effect

The capture effect, also known as co-channel interference tolerance, is the ability of certain wireless systems to correctly receive a strong signal from one transmitter regardless of the significant interference from other transmitters [7]. Due to the nature of capture effect, the stronger signal suppresses the weaker signal after which the stronger signal may be survived and decoded successfully. The survivability of the stronger signal depends on the arrival time and received power of both signals as well as the receiver characteristic, such as CCR and signal synchronization mechanism. As mentioned in the previous section, LoRa receivers typically need the stronger signal to be 6 dB higher than the weaker signal and the stronger signal should arrive no later than 4 symbols time after the arrival of the weaker node.

Bor et al. [6] conducted a three-nodes experiment to verify the capture effect property in a LoRa transceiver. Two transmitters and one receiver were used. One transmitter was set to transmit with 3 dBm power and another one with 2 dBm. Both transmitters sent packets with 32-byte length. The stronger transmitter sent packets with a particular time offset relative to the weak transmitter in order to achieve stronger-first and stronger-last effects. The experiment was repeated for all different combinations of SF, BW, and time offsets. The result can be seen in Figure 3-6.

From Figure 3-6, we can deduce several characteristics in the collision of two frames:

- Both frames are dropped when the stronger transmission arrives later than 4 symbols time relative to the weaker transmission.
- The stronger transmission can be decoded as long as it arrives no later than 4 symbols time relative to the weaker transmission.

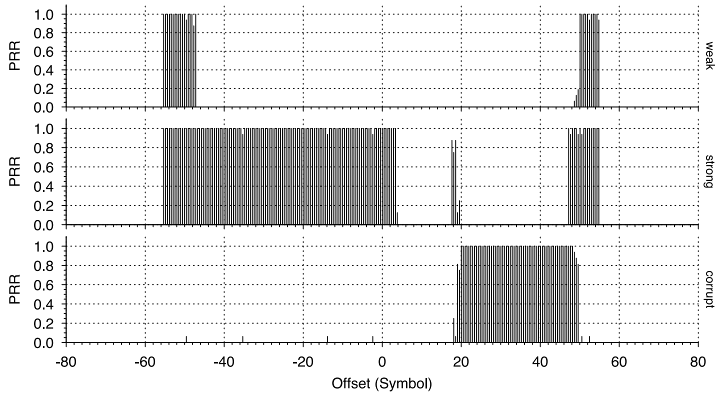


Figure 3-6: Capture effect for SF12BW250 in terms of packet reception rate (PRR) [6]

- The weaker transmission is suppressed by the stronger transmission in both stronger-first and stronger-last scenarios.

If we look at the PRR of the stronger transmission around time offset of 20 symbols, we can observe a PRR peak. The author did not explain why this peak occurs. Also, the distinction of the corrupted packets, whether they come from the stronger transmission or the weaker transmission, is not quite clear. But intuitively, we can assume that those corrupted packets come from the stronger transmission as the weaker transmission is being suppressed by the stronger transmission. In addition, it will be interesting to observe the capture effect when more transmitters are involved.

3-4-3 Interference Model

According to [47], there are two approaches that can be used to design a capture-enabled model for frame collision: packet-level (link-level) modeling in which one or two signal-to-interference-plus-noise-ratio (SINR) computations are performed per packet, and physical-level modeling, which involves a comparison of signal strength from every possible transmitter at all possible receivers for every bit. Packet-level modeling has efficient computational time but suffers in terms of accuracy, whilst physical-level modeling delivers a higher fidelity by sacrificing the computational time. Thus, physical-level modeling may not be suitable for simulating large-scale networks. Two examples of packet-level modeling that are commonly used for simulating interference: additive interference model (AIM), and capture threshold model (CTM).

Additive Interference Model

The additive interference model (AIM) is based on the additive property of interference. This property can affect the performance in some wireless physical layer technologies, depending on their decoding mechanisms. Iyer et al. [48] studied this model, particularly related to network capacity, and also did some comparison with other models. Some experiments using CC2420 radio performed by Maheshwari et al. [49] indicate that the additive assumption is still valid. The additive interference in LoRa transceivers has not been investigated yet and is still an open research question.

The AIM relies on SINR, which compares the received power of desired signal with the cumulative interference powers and thermal noise. Suppose we have a set of concurrent transmissions $\mathbf{S} = \{S_1, S_2, S_3, ..., S_n\}$. The SINR of the desired signal S_i required for successful decoding is given by:

$$SINR(S_i) = \frac{P_r(S_i)}{P_N + \sum_{S_j \in \mathbf{S} \setminus \{S_i\}} P_r(S_j)} \geq \beta_i \tag{3-15}$$

where $P_r(S_i)$ denotes the received power of S_i, P_N denotes the thermal noise power, and β_i denotes the SINR threshold for S_i. The β_i value depends on the modulation and coding scheme used in S_i.

Capture Threshold Model

The capture threshold model (CTM) considers interferences as individual isolated signals. CTM compares the received power of desired signal S_i against each of the interfering signals. A frame S_i can be decoded if the received power is above the receiver sensitivity S_r, and for every interfering signal in $\mathbf{S}$, the received power does not exceed the capture threshold value α_i, which, for LoRa modulation, is equal to the inverse of CCR listed in Table 3-3. These two conditions can be written as:

$$P_r(S_i) \geq S_r(S_i) \qquad \text{and} \tag{3-16}$$

$$\frac{P_r(S_i)}{P_r(S_j)} \geq \alpha_i \qquad \forall S_j \in \mathbf{S} \setminus \{S_i\} \tag{3-17}$$

Several network simulators such as LoRaSIM use CTM for checking frame collisions. Aside from received signal strength comparison, LoRaSIM also checks the arrival time of interfering signals. However, LoRaSIM does not take into account collisions between two frames with different SFs.

Chapter 4

Small-Scale Frame Collision Measurements

In this chapter, we present the measurement and evaluation of LoRaWAN frame collision in a small-scale setup. Additional measurements regarding the influence of spreading factor will be discussed to get a better insight into parameters that affect the LoRaWAN performance.

4-1 LoRaWAN Network Infrastructure

Before experimenting with LoRaWAN devices, one should be aware of the limitations as well as terms and conditions of LoRaWAN networks available in the area of interest. This is done to avoid interference with other users. We decided to build a private LoRaWAN network for experimenting with LoRaWAN frame collisions. We used public LoRaWAN networks, i.e., the TTN and KPN networks, to perform less-intensive measurements.

4-1-1 Private LoRaWAN Networks

A low-cost, single-channel gateway is used to build the private LoRAWAN network. The gateway uses an ESP8266 platform that is connected to a LoRa module (RFM95) using SPI and is injected with firmware from [50]. The gateway forwards uplink messages to, at most, two network servers simultaneously through UDP. Node-RED is used to operate the network server application and is connected to a NoSQL database to store the results. The network architecture is illustrated in Figure 4-1.

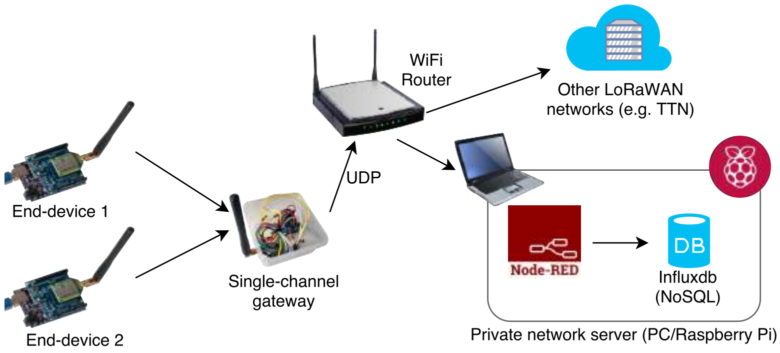


Figure 4-1: Network architecture of the deployed private LoRaWAN network

4-1-2 Public LoRaWAN Networks

TTN

We conducted measurements to test LoRa connectivity through a TTN gateway located at the TU Delft EEMCS building. Note that one needs to create an application and register the end-devices on the TTN platform. This platform provides end-device and gateway configurations as well as network key management. For example, end-devices can be configured to be activated either by using ABP or OTAA. We used TTN for measuring collision in multiple gateways setup while still abiding with the TTN Fair Access Policy [51].

KPN Developer Portal

KPN provides free LoRaWAN network access for developers to try and test connectivity of their end-devices. We used this network to characterize the effect of spreading factor. However, there are some limitations when using this developer portal [52]:

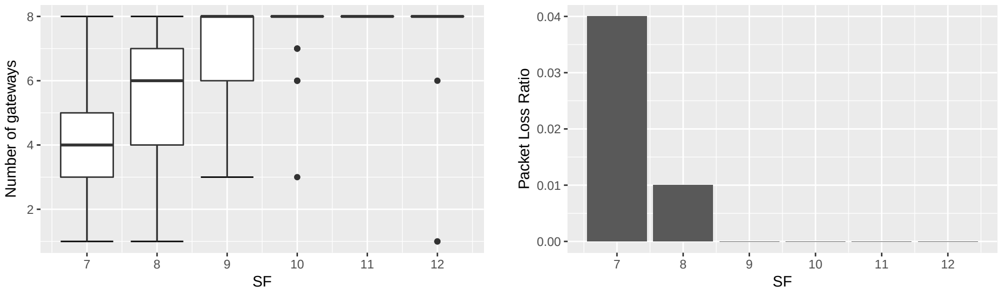
- At most 3 gateway status updates can be received by the network server. Only the gateway that has the strongest link reveals its location.
- The number of connections is limited to 6 uplinks/hour and 2 downlinks/hour.
- The portal only displays at most 10 recent messages, but it allows connection to an application server via HTTPS.

Application Server

The application server is developed using CodeIgniter to collect data from the KPN and TTN networks. The network architecture is similar to the typical LoRaWAN network architecture described in Section 2-2-1. As the KPN network needs HTTPS, the messages are first forwarded to another server that has a valid SSL certificate, and then repacked before sending them to the application server. Node-RED is also used for this purpose [53]. Likewise, messages from TTN networks are forwarded to this intermediate server using the MQTT protocol. From the intermediate server, both KPN and TTN messages are delivered to the application server using HTTPS.

4-2 Spreading Factor Evaluation

We deployed an end-device built using Arduino and Dragino LoRa shield and placed it indoor in a fixed location. The device is connected to the KPN network due to its coverage. The device sent a 9-bytes packet every 10 minutes with a total of 100 packets per SF using $TP = 14$ dBm, $BW = 125$ kHz, and $CF = 868.1$ MHz. The results are shown in Figure 4-2.



(a) SF vs number of gateways **(b)** SF vs Packet Loss Ratio

Figure 4-2: Effects of different SFs

It can be seen that packets with SF7 and SF8 suffer from packet loss, which is apparently due to the coverage outage. Using higher SF increases the probability of a packet being delivered successfully. However, not all packets were received by the maximum number of gateways, i.e., 8 gateways, for SF12. As we used a public network, there might be collisions in some of the gateways so that the packet could become corrupt. Using SF12 increases the chance of collision since the time on air becomes larger. We recommend not to use a high SF like SF12 in such circumstances where multiple gateways exist and are located close to each other. SF9 should be sufficient to obtain good reliability and to prevent frame collisions.

4-3 Frame Collision Characterization

4-3-1 Single Gateway

We performed a collision experiment with 1 ms precision using two RN2483 LoRaWAN modules as transmitters and one single-channel gateway connected to the private network via serial link and WLAN, respectively. Figure 4-3 shows the measurement setup.

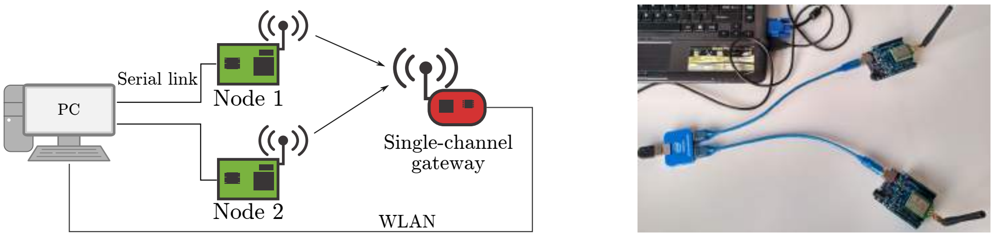


Figure 4-3: Intensive frame collision experiment set-up

By using application threading in Python, both end-devices can send frames simultaneously with 1 ms precision delay. The flowchart of the Python script for the experiment is illustrated in Figure 4-4.

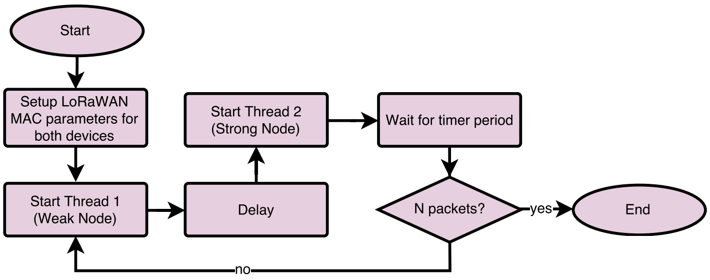

Start
Setup LoRaWAN MAC parameters for both devices
Start Thread 1 (Weak Node)
Delay
Start Thread 2 (Strong Node)
Wait for timer period
N packets?
yes
End
no

Figure 4-4: Python script flowchart

The network server used for this experiment is described in Section 4-1-1. The network server receives data from gateways via UDP, e.g., frames sent by node and gateway status updates. The gateway status updates are filtered out, keeping only the received frames. Afterwards, each frame is decoded to obtain the device address. If the address is not valid, i.e., does not belong to either strong node or weak node, the frame is considered as an unrecognized frame. If the address is valid, the Message Integrity Check (MIC) of the frame will be verified according to the device address. A frame with invalid MIC will be regarded as a corrupt frame. It should be noted that such frame is still forwarded by a gateway to the network server when its payload CRC is valid, meaning that access to gateway log files is required to obtain CRC status. As this thesis focuses on packet delivery from end-devices to application servers or users, we consider only the MIC status.

According to [6], most frames sent by the stronger node can be recovered whenever they are received earlier than the frames sent by the weaker node. Therefore, we conducted the experiment only for the opposite case in which the frames sent by stronger node arrived later than the frames sent by the weaker node. Two scenarios were considered for this experiment:

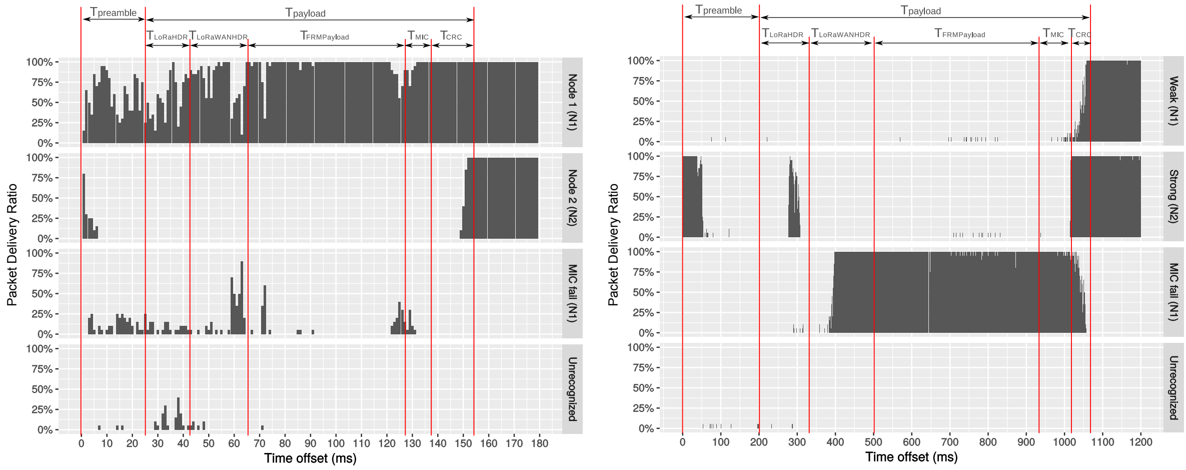
1. Equal received power P_{RX} (RSSI)
2. Different transmission powers P_{TX}, SF combinations, and gateway placements

For both scenarios, the payload size is fixed to 26 bytes, and the packet is transmitted with a regular period tailored to the duty cycle, which varies for different SFs. Each node sends 20 packets for each time offset. The transmissions use $CF = 869.7$ MHz, which is relatively free compared to the default LoRaWAN frequencies, e.g., 868.1 MHz.

In the first scenario, both nodes used $P_{TX} = 2$ dBm. As both nodes use pigtail antennas, the position of the antennas needs to be adjusted until the RSSI values at the receiver are sufficiently equal. The measurement was taken using only one data rate, i.e., SF8BW125. The time offset is calculated with respect to the first node (N1), meaning that the second node (N2) transmission is delayed.

In the second scenario, the stronger node used $P_{TX} = 8$ dBm, and the weaker node used $P_{TX} = 2$ dBm. The time offset is calculated with respect to the weaker node (N1), similar to the previous scenario. The measurement was performed for all SFs using 125 kHz bandwidth and for two different gateway placements, i.e., line-of-sight (LOS) and non-line-of-sight (NLOS). In the LOS scenario, the gateway was placed indoor in the same room with the two end-devices and was separated approximately 5 m away. In the NLOS scenario, the gateway was placed in another room separated 30 m away. Note that the gateway placement in the first scenario is similar as in the NLOS scenario. Ideally, this should also be done in LOS scenario, but due to the time constraints, we leave this as future work.

For the sake of clarity, we only display two results as shown in Figure 4-5. The complete results and the method for determining timing information can be seen in Appendix A-1.



(a) Equal received power (SF8)

(b) Different transmission powers (LOS-SF11)

Figure 4-5: Packet delivery ratio (PDR) for the first and second scenario

In Figure 4-5a, most frames (N1) can be correctly decoded if they arrive earlier than the other frames, whereas most of the delayed frames (N2) are not received by the gateway, except when the delayed frames arrive no later than 4 symbols. According to [18], the receiver needs at least 4 preamble symbols to acquire lock on the LoRa signal irrespective of the programmed preamble length. In this case, the receiver has already locked to the first frame (N1), which further becomes corrupt due to the arrival of the subsequent frames (N2). The PDR for N1, however, is opportunistic in which the gateway indefinitely switches its state, either reading signal from N1 or N2. Therefore, it becomes difficult to conclude the result. The RSSI and SNR values are almost equal, approximately -70 dBm and 12.5 dB respectively, when both frames do not collide. When both frames collide, the RSSI values increase, but the SNR values decrease to approximately 0 dB. Note that some frames are received by the gateway but the device addresses could not be identified. These unrecognized frames may be originated from other end-devices or from N1 and N2 in which the frame headers become corrupt.

For the second scenario, we observe similar patterns for all SF combinations and gateway placements. However, the PDR is slightly higher in LOS scenario, which seems to be reasonable due to multipath fading and shadowing in NLOS scenario. The average RSSI difference is more than 20 dB, which is far above the 6 dB requirement [46]. Figure 4-6 illustrates several conditions that contribute to packet loss and concludes our experiment. Those conditions can be described as follows:

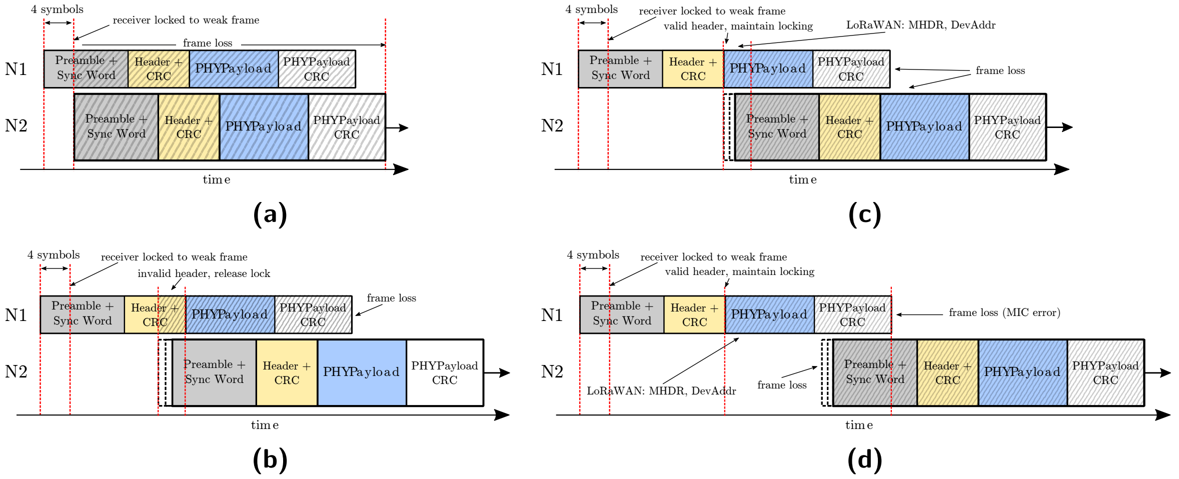


Figure 4-6: Conditions at which packet loss may occur

(a) Both frames are considered lost when the stronger frame arrives later than the receiver locking time, i.e. 4 symbols. The receiver supposedly listens to the weaker frame signal but it is suppressed by the stronger frame signal.

(b) The stronger frame may survive collision when its arrival overlaps with the CRC header of the weaker frame, which may become corrupt. This will cause the receiver to release the locking from the weaker frame and start listening to a new frame, i.e., the stronger frame.

(c) Both frames are considered lost when the stronger frame arrives after the receiver finishes receiving the weak frame header, and the stronger frame overlaps with the LoRaWAN header of the weaker frame. This happens because the receiver keeps locking on the weak frame, and the information inside LoRaWAN header such as device address may get destroyed.

(d) Both frames are considered lost when the stronger frame arrives after the receiver finishes receiving the LoRaWAN header of the weaker frame and slightly before the payload CRC of the weaker frame during which the stronger frame preamble gets destroyed. The weaker frame is successfully received by the gateway but the PHYPayload may get destroyed, resulting in incorrect MIC. Thanks to error correction techniques employed in the LoRa transceiver, the weaker frame can still be decoded whenever the stronger frame slightly overlaps with the payload CRC of the weaker frame. Ultimately, the stronger frame can be decoded whenever at least 4 symbols of its preamble do not collide.

To validate this behaviour, we will examine dataset of the large-scale frame collision experiment in the next chapter. As of this writing, Haxhibeqiri et al. [54] performed similar experiment using one gateway and deduced several collision criteria with respect to the gateway standpoint (CRC). They applied these criteria as a basis for developing simulations. However, they did not segregate the whole frame duration based on the frame structure.

4-3-2 Multiple Gateways

Same Network

We set up an experiment using two end-devices inside a building 64.9 m (12th floor) above sea level and positioned close to the window. Both devices were registered to TTN using ABP without ADR enabled and used the same data rate (SF9BW125), coding rate (4/5), frequency (868.1 MHz), payload size (17 bytes), but with different transmission powers (14 dBm and 8 dBm).

We set the weaker node (N1) as the reference time to calculate time offset with 1 symbol-time precision. Both devices sent 10 frames per time offset and were received by six gateways whose locations are illustrated in Figure 4-7. As the measurements were performed at a relatively high altitude and surrounded by open area, the coverage extended up to 21.5 km.

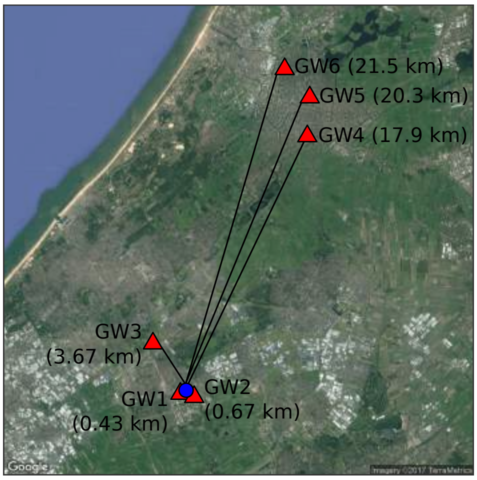


Figure 4-7: Location of gateways that receive frames sent by both devices

Figure 4-8 shows PDR from the perspective of the network server (All GWs) and per individual gateway. The whole frame duration can be partitioned into (1) preamble, (2) LoRa header, (3) LoRaWAN header, (4) frame payload, (5) MIC, and (6) payload CRC.

The result shows that the PDR is improved for the delayed frames sent by the stronger node. As the frames sent by the weaker node could not reach the more distant gateways (GW4, GW5, and GW6), the gateways would not acquire lock to the weaker frames so that the stronger frames could be decoded properly. The closer gateways (GW1, GW2, and GW3) also contribute to the higher PDR level as they increase the probability of receiving correct frames. For the weaker node, there is an improvement of PDR for the collision when the stronger frame arrives at the reception of frame payload from the weaker node, yet it is still opportunistic.

The signal strength indicators, represented in the form of RSSI, SNR, and ESP, can be seen in Appendix A-2. As the RSSI values for both devices seen by the distant gateways are somewhat equal and the SNR values are negative, we use ESP to calculate received power differences. The ESP difference is observed to be proportional to the gateway distance. We observe most of the ESP differences are larger than 6 dB, which is why the PDR patterns are relatively similar to the previous experiment.

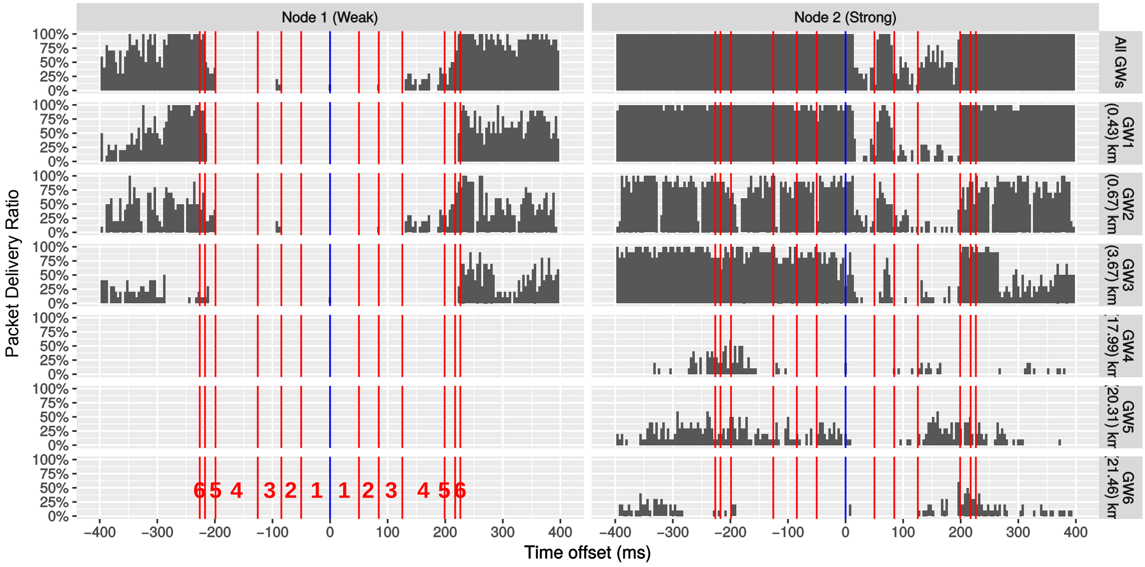


Figure 4-8: Packet delivery ratio (PDR) for the collision experiment with the same network. The number in red indicates the frame timing partition.

Different Networks

We set up a similar experiment but with devices connected to different networks. One end-device was registered to TTN, while the other one was registered to the KPN network. Both devices had ADR disabled. The KPN device acts as an interferer. However, as the KPN development trial period was expired, we could not observe packet received at the KPN network server. Later in this thesis, we found that some of the KPN device transmissions, for uplink and downlink, were recorded in the gateway log files.

Both nodes used the same data rate (SF12BW125), coding rate (4/5), frequency (868.1 MHz), and payload size (17 bytes), but with different transmission powers (14 dBm and 8 dBm). As only one node could be observed (TTN node), we developed two scenarios: **(1) weak interferer**, and **(2) strong interferer**.

The time offset was set using time step of one symbol time with respect to the TTN node. Both end-devices sent 7 frames per time offset. Those frames were received by the same gateways as in the previous experiment.

Based on the KPN gateway log files, both devices started their first transmission using SF9BW125 with ADR and *ADRACKReq* fields enabled, indicating that the network server should respond to it by sending a downlink message that typically contains new configurations, for example, related to frequency channels, data rates, and reception slots. It turned out that the KPN device received two new frequency channels, i.e., 867.7 and 867.9 MHz, whilst the TTN device did not receive new frequency channels and remained to use the 868.1 MHz frequency channel. Ultimately, the KPN device used those three frequencies equally, i.e., 33.3% for each channel, meaning that the probability of collision reduced to 33.3% as both devices might overlap only in the 868.1 MHz frequency given that both devices used SF12.

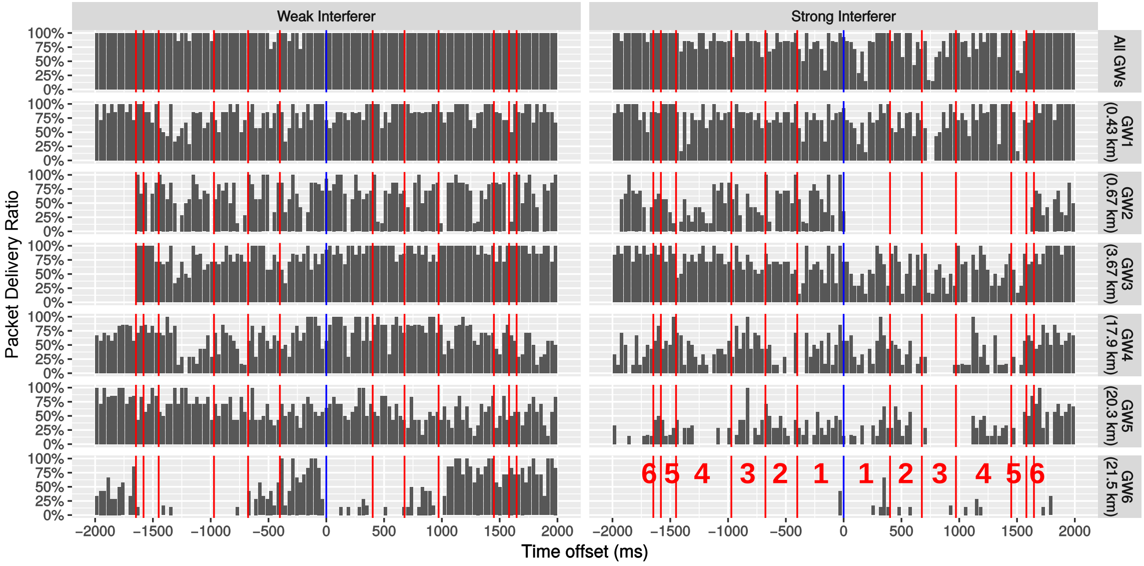


Figure 4-9: Data extraction rate for the weak and strong interferers using different networks. The number in red indicates the frame timing partition.

Figure 4-9 depicts the PDR for both scenarios. In the weak interferer scenario, most frames were delivered to the application layer even when the interferer arrived first, which is indicated by negative time offset. The interferer could not reach the more distant gateways and because of the differences in allocating the frequency channels, less collision can be expected. Frames that are received by those gateways contribute to a high PDR value. Notice that no frames were received by GW2 and GW3 for time offset less than -1600. We believe those gateways might be inactive during the experiment as the measurements for positive and negative time offset happened in two different time periods.

In the strong interferer scenario, the desired frames hardly reach the farthest gateway (GW6) due to a lower transmit power. The frames might also get destroyed by the strong interferer. It should be noted that no frames were received by GW2 for positive time offset, which suggests that either the gateway was inactive or the frames became corrupt. As the other gateways could still receive the frames, we believe that the gateway was inactive at that moment.

The received signal indicators are also depicted in Appendix A-2. The patterns are similar to the previous experiment where the average power difference can exceed 6 dB, but it is rather difficult to analyze signal quality during collisions as there might be other nodes transmitting at the same time.

4-4 Conclusion

From this chapter, we can deduce several factors that should be considered in deploying LoRaWAN networks. First, it is recommended for end-devices not to use the lowest data rate, especially when multiple gateways exist, in order not to interfere on other transmissions. Second, thanks to capture effect, a collision of two frames that use the same data rate and frequency, does not always cause total packet loss, depending on the arrival time and transmission power. Also, deploying more gateways can increase the probability of frame reception at the network level or the end-to-end packet delivery ratio, especially when the network server assigns end-devices to use different frequency channels, which should be varied for different network providers. However, in the case when the weaker and the stronger end-devices are placed at the same location, the capture effect does not significantly improve the weaker end-device performance.

It should be highlighted that these findings need to be validated in large-scale networks, either by performing simulations or by conducting real-world experiments. The next chapter will discuss the evaluation of frame collision through a large-scale experiment.

Chapter 5

Large-Scale Frame Collision Measurements

In this chapter, we present the analysis and evaluation of frame collisions in a large-scale experiment setup. The experiment was performed by Thomas Telkamp (TTN) during the Electronics & Applications (E&A) event at Jaabeurs, Utrecht from 30 May until 1 Jun 2017.

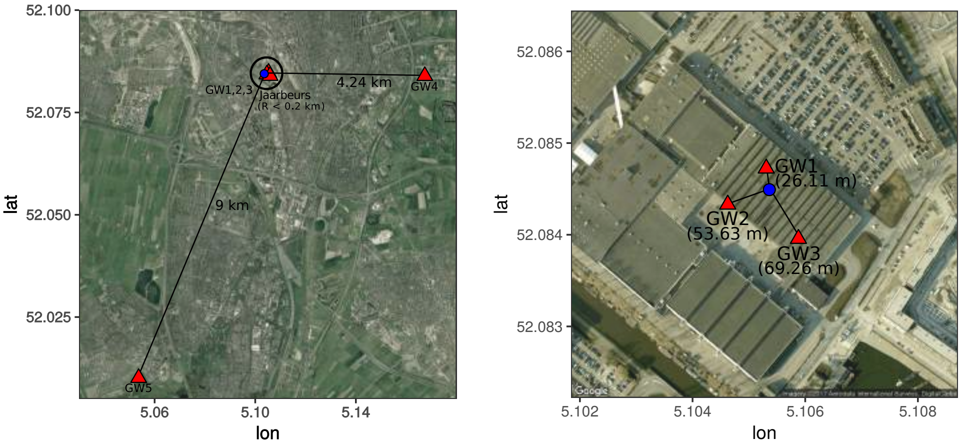
5-1 Experiment Setup

5-1-1 End-devices

During the E&A event, attendees were handed a LoRaWAN device, i.e., KISS LoRa. When this device is powered up, it immediately transmits packets on a fixed device address. All devices were initially programmed to have the same device address (anonymous devices). The anonymous devices contain 2 bytes ID in the encrypted payload, which represents the last 2 bytes of the hardware EUI. Each device then switched to a personal address after personalizing it at the designated booths. The devices sent a fixed 23-bytes packet approximately every 20 seconds in which each transmission used a random spreading factor with probability 1/2 for SF7, 1/4 for SF8, 1/8 for SF9, 1/16 for SF10, 1/32 for SF11, and 1/64 for SF12. The frequency used for transmission was distributed uniformly on eight frequency channels. The devices did not use ADR for transmissions. Traffic from other devices was also observed during the measurements.

5-1-2 Gateways

Up to 8 gateways were found around Jaaberus area, but only five gateways were equipped with GPS. One can synchronize the gateways with no GPS using common reception. However, for packets received only by such gateways, there is no reference time to which the packet can synchronize. Therefore, we only take into account those five gateways with GPS. Three gateways were placed inside Jaabeurs, and the rest were located outside Jaabeurs. Figure 5-1 depicts the gateway locations observed during the event.



(a) Around Jaarbeurs

(b) Inside Jaarbeurs

Figure 5-1: Gateway locations observed during the E&A event

5-1-3 Network Server

The TTN back-end was used for this experiment, and the data were obtained from the TTN network operations centre (NOC). The dataset consists of received uplink, downlink, and gateway status updates. We later processed and merged the uplink and downlink data into a single dataset. The gateway status updates were sent by gateways every 30 seconds. Each gateway started transmitting the status updates on its discretion. The gateway status contains the total number of received uplink frames, forwarded frames with correct CRC, transmitted downlink frames, unsuccessful downlink transmission, and GPS data within the given interval.

5-2 Data Analysis

5-2-1 Time Synchronization

The E&A event dataset contains several fields that represent packet reception time, such as gateway time and network server time. The gateway time field corresponds to the arrival time of packet at a particular gateway, whilst the network server time corresponds to the arrival time of packet at the NOC. Although GPS time field is more suitable to use for the analysis, we found a large number of time-offset with high deviations amongst gateways within a common reception. Thus, it is difficult to synchronize those timestamps for long-run analysis. The network server timestamps have maximum deviations around 1.5 seconds, whilst the gateway timestamps can reach up to 50 minutes. Based on this result, we decided to use the network server time for long-run analysis, and the gateway timestamp for instantaneous or short-interval analysis.

5-2-2 Traffic Characteristics

Traffic Volume

We use gateway status updates data to obtain traffic volume during the event. Figure 5-2 shows maximum uplink and downlink frames from all gateways. The traffic is heavily congested from 10:00 to 16:00 for all days. We will focus on this time-frame for the analysis.

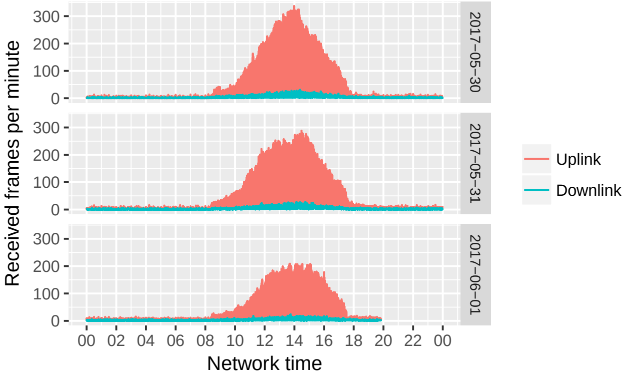


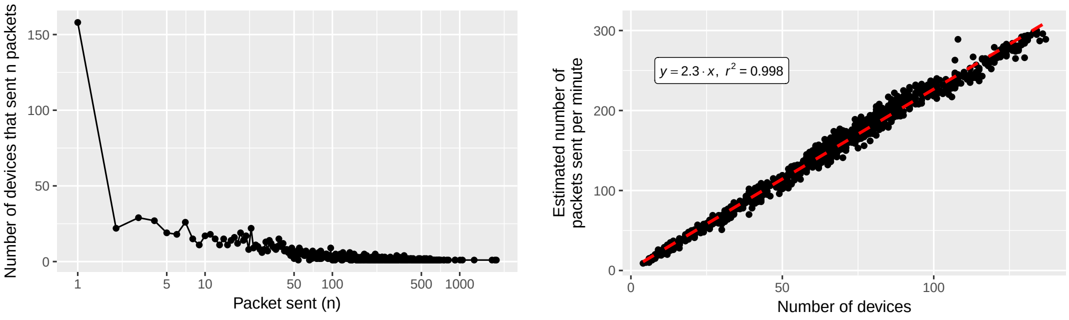
Figure 5-2: Maximum traffic volume captured during the E&A event

Number of End-devices

The dataset is filtered to obtain data from KISS LoRa devices only. The distinction between KISS LoRa and other devices is based on the hardware EUI obtained from the application log files. The anonymous devices initially have the same device address and the same hardware EUI registered at the TTN application server. Such devices can be distinguished by looking into the payload ID inside their encrypted payloads, which have already been decrypted in the application log files. Some of the anonymous devices were switched to personal accounts so that the device addresses and hardware EUIs will be changed. The personalized devices mark the actual hardware EUIs, which later could be used to replace the hardware EUI of the anonymous devices. We found 1426 devices (hardware EUIs) in which 489 anonymous devices were not personalized, 845 anonymous devices were switched to personal devices, and 92 devices were personalized outside the event, e.g., at home.

Not all end-devices sent the same number of packets at that time-frame. Figure 5-3a shows the histogram of end-devices that sent n number of packets. It shows that up to 158 devices only sent one packet and only one device sent 1940 packets in the specified time-frame. Assuming each device can send exactly one packet per 20 seconds, there will be 3240 packets sent per device. This indicates that those devices did not activate all the time and might be restarted several times.

Partitions of a 1-minute interval are created and the number of distinct end-devices and total packets sent per partition are counted to observe the relationship between the number of end-devices and the total packets sent. The result is depicted in Figure 5-3b. The data is fit with a linear model with a gradient of 2.3, meaning that, on average, an end-device sends a packet approximately every 26 seconds, which is close to our initial setup, i.e., 20 seconds.

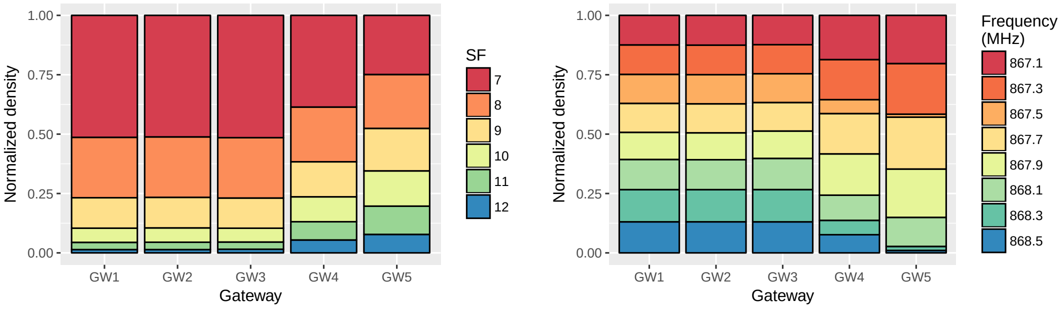


(a) (b)

Figure 5-3: Traffic characteristic: (a) histogram of packet sent vs number of devices, (b) number of devices vs estimated number of packets sent per minute

5-2-3 LoRa Parameters Distribution

The distribution of LoRa parameters is obtained from the successfully received frames at each gateway. Figure 5-4 depicts the distribution of spreading factors and frequency channels. The result shows that the distribution matches the initial setup, particularly for the gateways placed inside Jaarbeurs (GW1, GW2, and GW3). For the gateways placed outside Jaarbeurs (GW4 and GW5), some packets might not be received, especially for the packets with low SFs as they were unable to reach the more distant gateways. Therefore, the probability of getting collision at GW4 and GW5 were significantly influenced by the packets transmitted using high SFs.



(a) (b)

Figure 5-4: Traffic characteristic: (a) histogram of packet sent vs number of devices, (b) number of devices vs estimated packets sent per minute

5-3 Performance Evaluation

5-3-1 Frame Error Rate

The number of lost packets due to CRC error can be extracted from the gateway status updates, as shown in Figure 5-5. The error rate is proportional to the number of received frames per minute and is around 5% for the gateways placed inside Jaarbeurs, and up to 32%

for the gateways placed outside. This also indicates that most frames hardly reach the more distant gateways due to the low spreading factor used. Collisions can aggravate the situation, especially for the frames that used high spreading factors and required longer time on air.

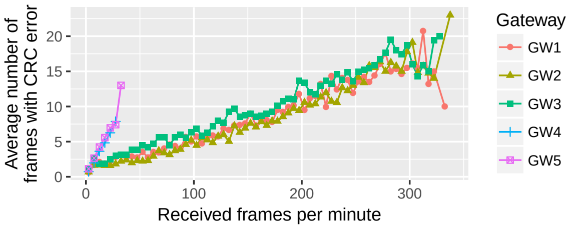


Figure 5-5: Average number of frames with CRC error obtained from gateway status updates

5-3-2 Data Extraction Rate

Experiment

The network performance can be assessed using the data extraction rate (DER) metric. We use DER to determine the advantages of adding more gateways to the network. DER can be easily calculated from simulation since the number of devices and transmitted packets are known within the given measurement period. In our case, the number of devices that sent packets fluctuates. By partitioning the measurement period into sub-intervals, e.g., using a 1-minute interval, the average DER can be estimated. Figure 5-6 shows the proposed method for estimating DER.

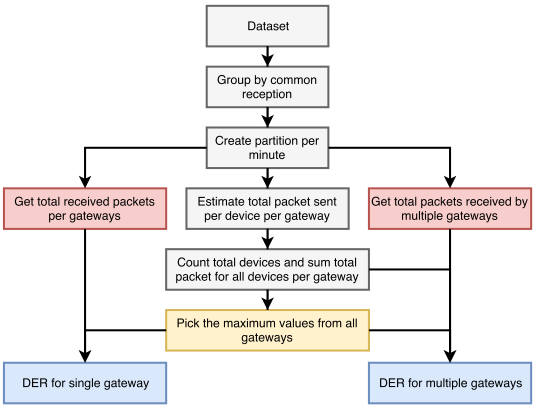
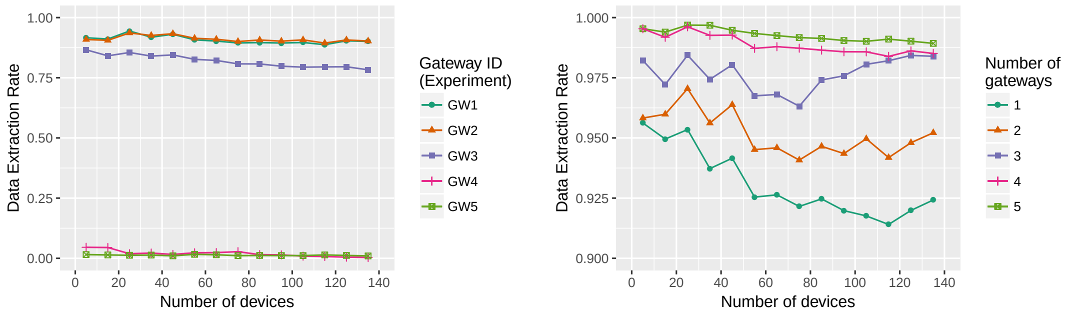

Dataset
Group by common reception
Create partition per minute
Get total received packets per gateways
Estimate total packet sent per device per gateway
Get total packets received by multiple gateways
Count total devices and sum total packet for all devices per gateway
Pick the maximum values from all gateways
DER for single gateway
DER for multiple gateways

Figure 5-6: Method for calculating data extraction rate from empirical data

Assuming a linear relationship between nodes and generated frames (see Figure 5-3b), we can plot DER against the number of active devices per minute. The DER plots are depicted in Figure 5-7.

The results show that for a single gateway, DER mostly depends on the distance between devices and the gateway. Even for the closer gateways, some variations on DER can still be observed. But in general, DER declines gradually as the number of devices increases. Adding more gateways can improve DER around 1% to 5% for approximately 140 active devices per minute.



(a) Single gateway **(b)** Multiple gateways

Figure 5-7: Average DER for single and multiple gateways obtained from the dataset

Simulation

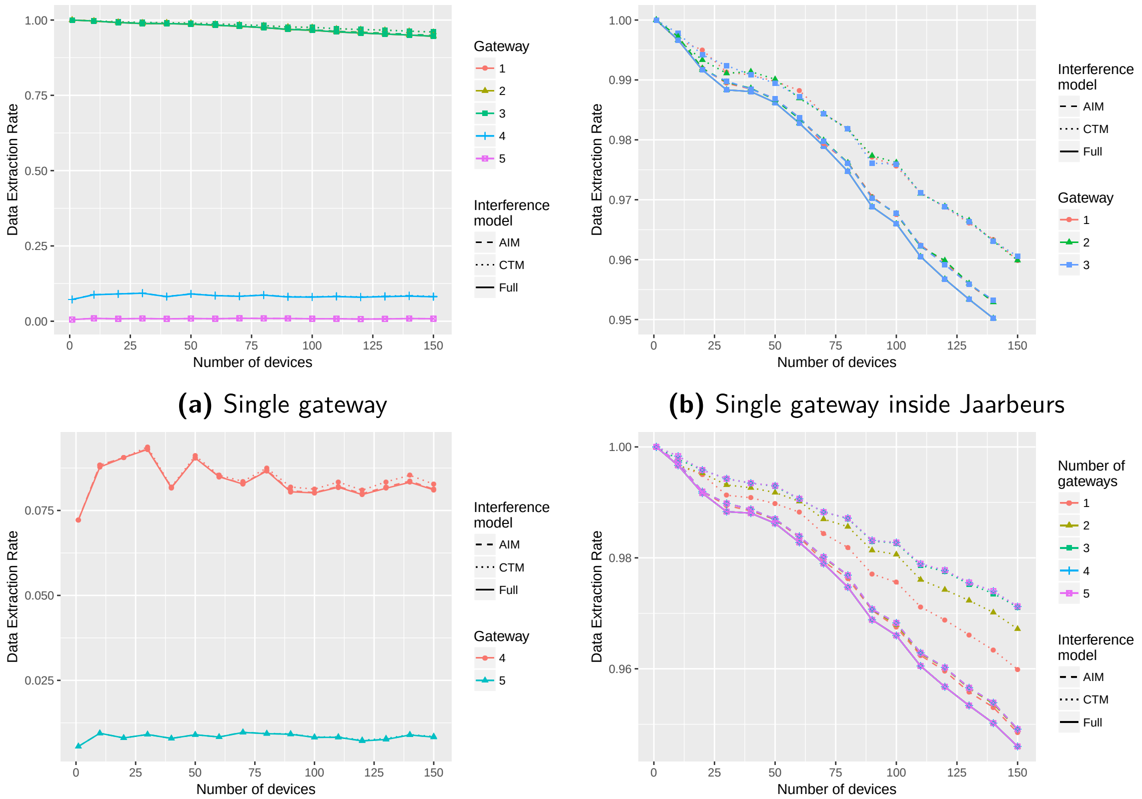
We developed simulations using R to compare and validate against the empirical result (see Listing B.2 in Appendix B). The simulation parameters are made similar to the practical experiment. We employ the simplified path-loss model for simulating transmission propagation, as well as CTM and AIM interference models described in Chapter 3. The path loss exponent (γ) and shadowing (σ) are obtained from the large-scale measurements in Utrecht region, which will be described in the next chapter. As three gateways are placed indoor, the path loss exponent and shadowing variance are different. In this case, we use $\gamma = 3$ and $\sigma = 9$ for the distance below 100 m. We also compare the CTM and AIM interference models with a conservative approach in which frames are marked lost when they occupy the same spreading factor, bandwidth, and frequency, and overlap in time even when the overlapping part is infinitesimally small.

The simulation assumes no interference between spreading factors and no packets survive collision when the stronger frame destroys the LoRaWAN header of the first-arrived weaker frame. In order to do so, one should track the collision time of the previous frames, which becomes more complex when dealing with multiple collisions. As of now, we define that the frame is regarded as a successfully received frame as long as the interfering frames do not overlap with the current frame preamble. Algorithm 5-3.1 describes the simulation procedure.

The measurement duration is set to 1 hour for each number of devices n_i, ranging from 1 to 150 devices. The results are depicted in Figure 5-8. It can be seen that the trends are relatively similar to the experiment. The CTM model provides higher DER values compared to the AIM and the conservative method, especially when the number of devices is sufficiently high. In Figure 5-8b and 5-8c, the gateways located inside Jaarbeurs receives more frames, thereby improving the benefits of capture effect on DER. When multiple gateways are deployed, the conservative method does not show any improvement whatsoever as it, eventually, depends solely on the arrival and end-of-reception time, which can be different for each gateway if the propagation time is added to the simulation. The most notable improvement appears in the CTM model, providing up to 0.5% DER improvement when adding more gateways. Comparing these results with the real experiment, we believe that the optimistic CTM model is more appropriate to use for this case. In particular, the real experiment shows the occurrence of capture effect and its significance on improving the network performance. Our findings are on par with the numerical results of slotted ALOHA protocol with multiple receivers in [55].

Algorithm 5-3.1 E&A Event Simulation Procedure

```
Define a vector $N = \{n_1, n_2, ..., n_i\}$ that represent a sequence of total devices $n_i$.
Define the duration of measurement $T$.
Define a fixed-period of transmission per device $t$.
Generate a vector of packets $P_k$ per device $j$ with total of $(n_i \cdot \frac{T}{t})$ packets.
Assign spreading factor and frequency channel for packet $P_k$ with distribution following the real experiment.
Assign a random transmission start time $t_{start}$ to packet $P_k$.
Clone packet $P_k$ for $G$ number of gateways.
Calculate distance to gateway $l$ and propagation loss for packet $P_k$ per gateway.
Check received power for packet $P_k$ at gateway $l$ and mark it as packet loss if the received power is below the receiver sensitivity.
Check collision for packet $P_k$ with the prior and subsequent packets at gateway $l$ by observing the start time $t_{start}$, time on air $t_{onair}$, and the end time $t_{end}$ of the packets.
Calculate DER for gateway $l$ and the aggregated gateways.
```



(a) Single gateway

(b) Single gateway inside Jaarbeurs

(c) Single gateway outside Jaarbeurs

(d) Multiple gateways

Figure 5-8: Average DER for single and multiple gateways obtained from simulations

Effect of Downlink Transmission

In Figure 5-7a, GW3 yields lower DER compared to GW1 and GW2 even though they were located close to each other. According to the simulation result (see Figure 5-8b), GW3 should output a similar pattern with GW1 and GW2. One reason might be that GW3 was severely blocked by obstacles. To identify other factors causing this problem, we then investigated the number of downlink frames and the downlink utilization. The empirical cumulative distribution function (ECDF) of the number of downlink frames and the downlink utilization are depicted in Figure 5-9.

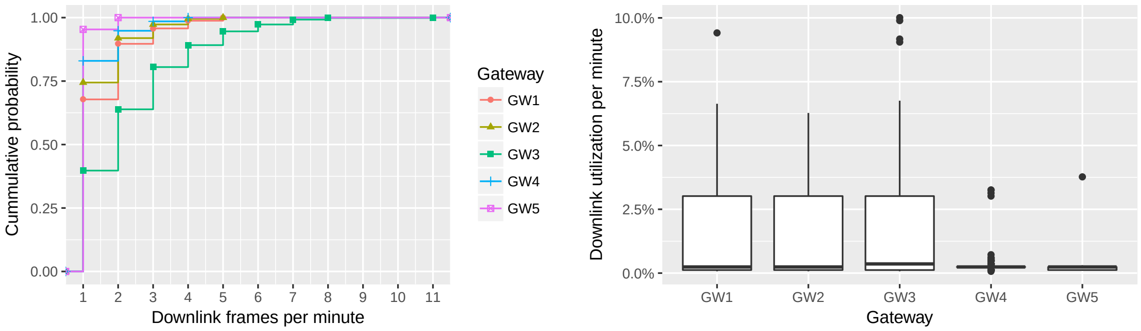


Figure 5-9: Effect of downlink transmission observed per minute for different gateways: (a) cumulative probability of downlink frames, (b) downlink channel utilization

Compared to GW1 and GW2, GW3 transmitted more downlinks, thereby increasing the total time on air. In this condition, the uplink transmissions could not be received as the currently available LoRaWAN gateways operate in a half-duplex mode. Up to 79% of downlink frames sent by GW3 were join-accept messages. Figure 5-10 shows the number of downlink frames transmitted by GW3 for different frequency channels and message types.

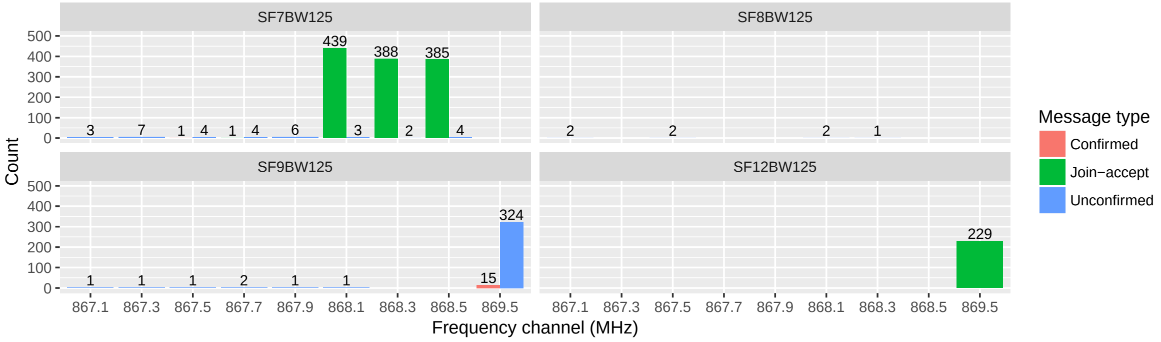


Figure 5-10: Number of downlink messages transmitted by GW3 for different frequency channels, spreading factors, and message types

Note that eight channels were used for uplink and RX1 downlink. The RX2 downlink used the default frequency of 869.525 MHz. Most join-accept messages were transmitted using the default LoRaWAN channels in RX1. However, 229 join-accept messages were transmitted using RX2 and SF12, which significantly increased the downlink utilization. According to the LoRaWAN specification, the default RX2 data rate is SF12BW125, whilst the TTN network uses SF9BW125. When an end-device connects to the network for the first time in a new session, it broadcasts join-request messages and then receives a join-accept message in either RX1 or RX2. In this situation, RX2 uses SF12BW125. Afterwards, it can be modified depending on the setting in the join-accept message. This finding suggests that when many devices send join-request messages, the best gateway that has the strongest signal strength will be easily saturated as it needs to send join-accept messages unless the network server includes the channel utilization aspect to select the other gateways. However, this will cost the reliability of data being sent to the devices as the received signal strength may be decreased. This is why RX2 provides up to 27 dBm transmission power.

5-3-3 Signal Quality

RSSI and SNR of the received frames are depicted in Figure 5-11. Only frames transmitted from the KISS LoRa devices are considered. The results show that most frames received by the gateways inside Jaarbeurs have strong RSSI values, ranging from -85 to -75 dBm, and positive SNR values, i.e., up to 15 dB. It indicates that collision might frequently occur in such strong signals that have a tendency to have power differences less than 6 dB. In the next subsection, we will discuss this issue.

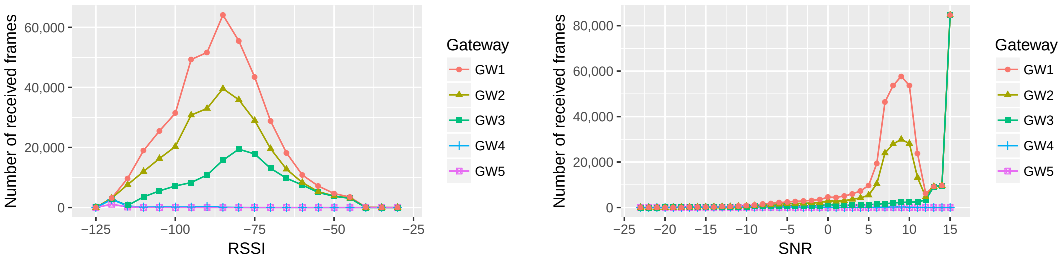


Figure 5-11: Signal quality of the received frames from KISS LoRa devices

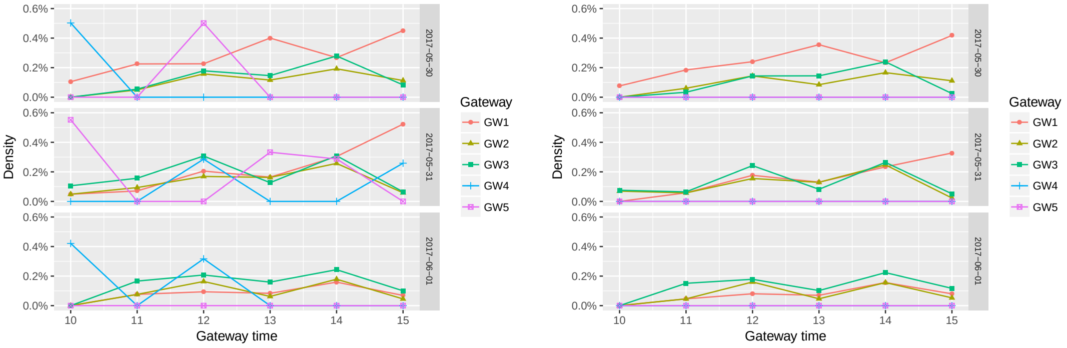
5-3-4 Collision Rate and Capture Effect in Detail

To obtain a complete view on the LoRaWAN frame collisions, we evaluate collision rate and capture effect in detail. The evaluation requires around one symbol precision, i.e., at least 1.024 ms for the typical and fastest data rate (SF7BW125). Therefore, we use the accurate, GPS-enabled, gateway time field and eliminate the outliers.

To check whether the gateway time field in TTN marks the packet arrival time or the end of packet reception, two devices that transmit the same payload but with different SFs (SF9 and SF12) exactly at the same time were used. If the time difference of packet reception at the TTN network server is similar to the difference of time on air, the gateway time field represents the end of packet reception. Based on the result, it turns out that the gateway time field marks the end of packet (t_{end}). After obtaining t_{end}, the start time or packet arrival time (t_{start}) can be estimated by subtracting t_{end} with the corresponding time on air (t_{onair}). These time fields are then used for detecting collision (see Algorithm B.1 in Appendix B). The results from collision detection procedure are collected and grouped per one-hour interval.

Collision Rate

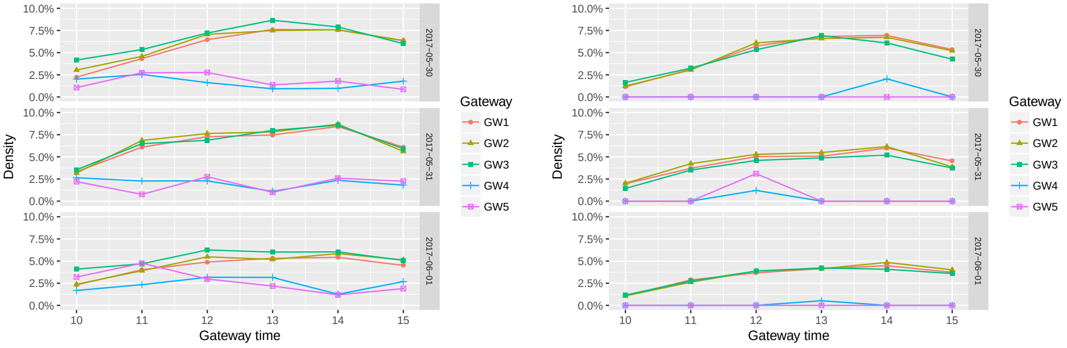
It should be noted that not only packets from KISS LoRa devices were received by the gateways but also from other devices. Figure 5-12 shows a comparison of packet survival rate between all devices and KISS LoRa devices. For all devices, we can expect more collisions due to the higher number of packets being received. It turned out that most packets received by GW4 and GW5 were transmitted from other devices, which were possibly located near these gateways. For KISS LoRa devices, we could not find any packets at GW4 and GW5 that survived collisions. This problem might be caused by either coverage issue or collision in which the colliding packets get destroyed. Nevertheless, up to 0.4% of the received packets experienced and survived collisions.



(a) All devices **(b)** KISS LoRa devices

Figure 5-12: Number of packets survived during collision

The collisions with different data rates can also be obtained as shown in Figure 5-13. As all packets used 125 kHz bandwidth, the collisions depend solely on the SF. The plots have relatively similar patterns and proportional to the plots in Figure 5-12. Also, some packets were received by GW4 and GW5, meaning that packets with different SFs did not significantly interfere each other, even for the distant gateways.



(a) All devices **(b)** KISS LoRa devices

Figure 5-13: Number of colliding packets that have the same frequency but different SFs

Capture Effect in Detail

To observe the capture effect, traffic from all devices is considered. The objective is to get as much traffic as possible so that the capture effect property obtained in the previous chapter can be validated. The first step is to investigate the overlapping symbols of the colliding frames. From the previous chapter, a packet can survive collision if at least 4 symbols of its preamble do not get destroyed. Figure 5-14 shows the histogram of overlapping symbols with respect to the start-of-packet and end-of-packet. The histogram shows that in both directions, most of the packets survive the collisions whenever their preambles slightly overlap with other packets.

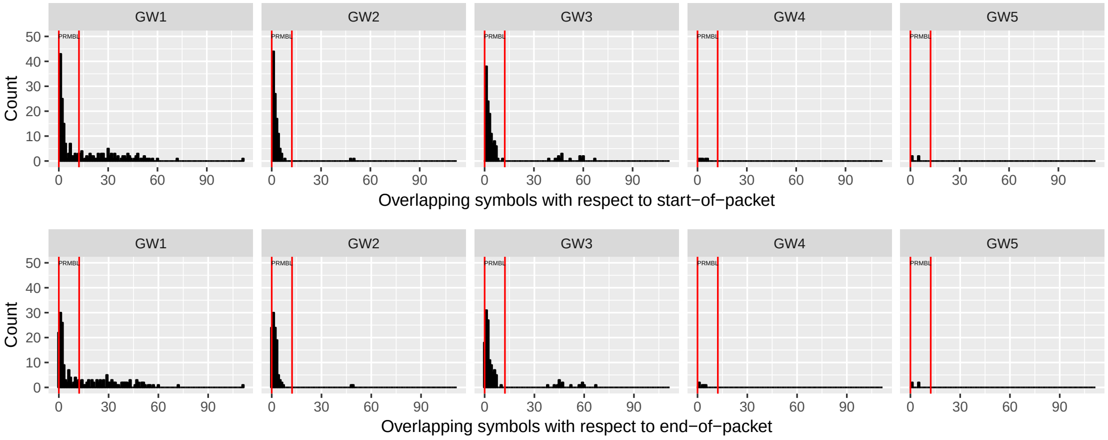


Figure 5-14: Histogram of overlapping symbols

Several packets survived collision even when they overlap after the preamble detection, especially for the packets that were received by GW1. Based on this, we then investigate power difference of overlapping packets in terms of RSSI and ESP to identify whether such packets have at least 6 dB received power difference. The results are depicted in Figure 5-15. It should be noted that only packets that overlap above 25% of the packet durations are considered, and no packets received at GW4 and GW5 fulfill this condition. We also plot the ESP histogram to present the SNR significance on the received power differences. It can be seen that the ESP differences are shifted to the right so that more packets exceed the 6 dB difference. However, even when the power differences exceed 6 dB, only one packet can be received when the packets completely collide. This is somewhat opportunistic and difficult to analyze.

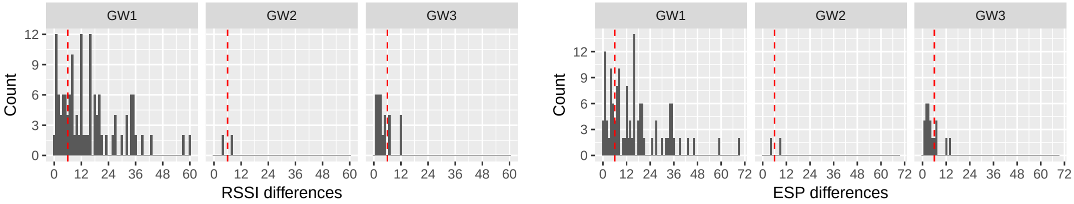
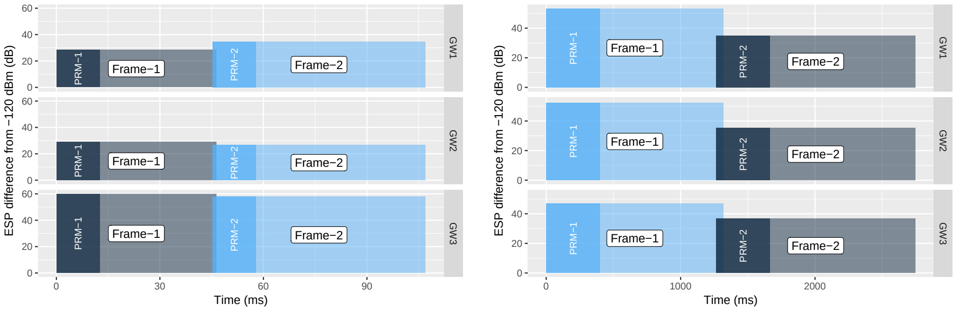


Figure 5-15: Histogram of power differences of packets that have over 25% overlapping time

Figure 5-16 illustrates collision of two packets obtained from the dataset in which both of them can be received successfully. It shows that both packets, regardless of the received power difference, can be decoded whenever the collision does not destroy most of the preamble symbols.



(a) SF7

(b) SF12

Figure 5-16: Collision of two packets in which both packets can be decoded successfully

5-4 Conclusion

From this chapter, several conclusions can be made:

- The LoRaWAN gateways (GW1, GW2, and GW3) can handle up to 300 frames per minute with only 5% of bad CRC packets, assuming that all packets can reach the gateways.
- Packets transmitted at the cell-edge (GW4 and GW5) are prone to CRC error due to collision or low signal strength.
- Adding more gateways can provide 1% to 5% DER improvements for roughly 140 devices that transmit packets approximately every 20 seconds. To some extent, adding more gateways may not improve DER even further, mainly when devices are located in the same area, and some of the devices send packets using lower transmission powers.
- A simulator has been developed to compare DER against the experiment result. The simulation yields similar results with the experiment, and the optimistic CTM model used in the simulation, in this case, provides a more comparable result.
- The number of downlink transmissions significantly affects the DER of a specific gateway (PDR) as LoRaWAN gateways operate in a half-duplex mode, especially when many end-devices join the network for the first time. Uplink transmissions will not be heard if the gateway is still busy transmitting downlink messages.
- Most packets were received with positive SNR and RSSI, which indicates that the devices are positioned close to the gateways (GW1, GW2, and GW3). It leads to the intuition that collisions at the closest gateways could happen mostly to the packets that have strong signal levels.
- The capture effect characteristic observed in this large-scale experiment is similar to the small-scale experiment in which the colliding packets can still be decoded as long as most of the preamble symbols are not destroyed irrespective of the received power difference. Also, packets with different SFs do not significantly interfere each other.

Chapter 6

Real-World LoRaWAN Evaluation: TTN Dataset

In this chapter, we analyze the TTN LoRaWAN network, specifically related to traffic characteristic, signal quality, LoRa parameter utilization, spatial distribution, and coverage aspects. This chapter does not include inter-network interference analysis considering that only packets sent by TTN devices are available in the dataset.

6-1 Data Collection

Large-scale measurement data were collected from TTN server using TTN API over one month period, i.e., June 2016. We obtained the data from [9] and repeated some of the results. All measurements were sent by Class A end-devices activated using ABP with Semtech default keys (NwkSkey = AppSKey = 2B7E151628AED2A6ABF7158809CF4F3C). Table 6-1 describes basic information that can be extracted from the dataset.

Table 6-1: Basic information extracted from TTN dataset

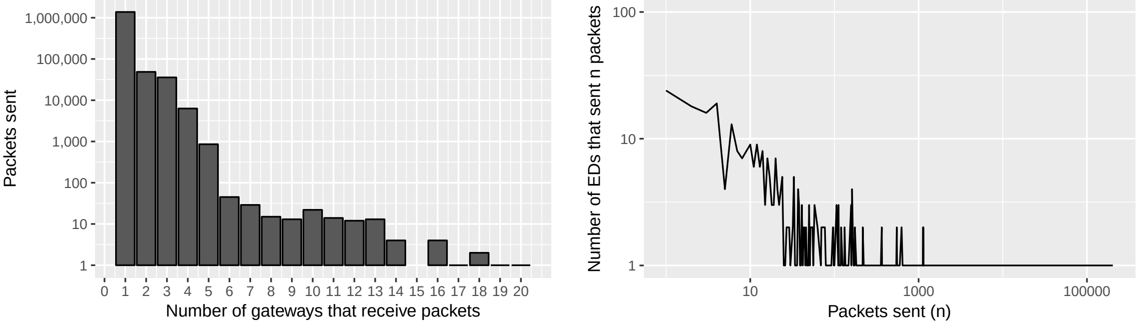
Distinct end-device IDs	488
Gateways: receive frames (active)	328
Gateways: send gateway status (all)	544
Gateways: send gateway status (active)	298
Gateways: have valid GPS location (active, not null or 0)	229
Number of frames received by gateways	1,651,467
Number of frames received by gateways (no duplicate)	1,616,500
Number of unique frames generated by end-devices	1,471,719
Number of gateway status updates sent by gateways (all)	21,816,829
Number of gateway status updates sent by gateways (active)	14,123,189
Number of gateway status updates sent by gateways (active, valid GPS)	12,271,366

In Table 6-1, 544 gateways sent gateway status updates, but only 328 gateways actively received the frames. Note that some frames were received by gateways that either did not send any gateway status or did not have correct locations. Ultimately, only 229 gateways can be used for the analysis.

6-2 Data Analysis

6-2-1 Traffic Characteristics

The packet transmission behaviour incorporates several aspects, namely the number of gateways that receive frames, total messages being sent in a certain period, and periodicity of transmission. For example, in Figure 6-1a most frames were received by one gateway (93.7%) and 2-5 gateways (3.3% - 0.58%), but only one frame could reach 20 gateways.



(a) **(b)**

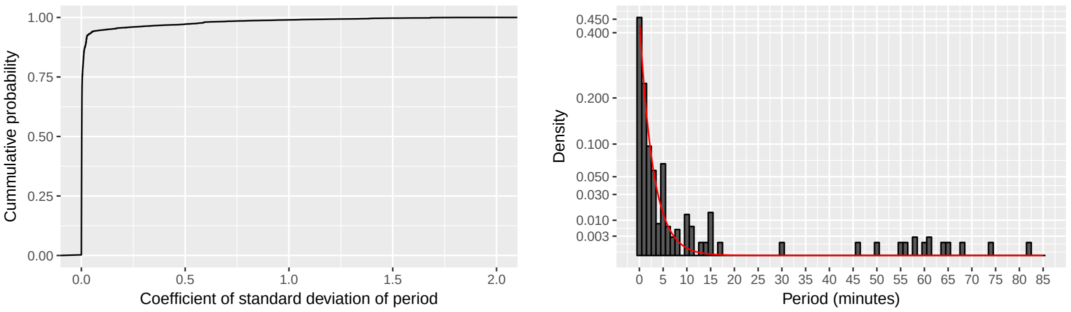
Figure 6-1: Histogram of (a) the number of frames that are received by single or multiple gateways, and (b) the number of end-devices that transmit n unique frames

Some end-devices transmitted only a few messages, less than hundred, and some other transmitted thousands of messages within one month. The number of transmissions can be deduced as an indication whether users doing some testing on LoRa technology or end-devices have been deployed for real use cases. The histogram of the number of end-devices that transmitted n unique frames is illustrated in Figure 6-1b. The distribution is skewed to the left side, which strongly indicates that many users were doing some testing on LoRa.

Transmission Periodicity

To obtain the types of traffic from the dataset, we extracted the period for each end-device by calculating the time difference between two consecutive unique frames. The sequence of unique frames was based on the FCnt in ascending order. As there might be some packet loss in the dataset, the difference of FCnt could be larger than one. One can estimate each individual period by dividing the total period with the FCnt difference. However, as we did not know yet whether the end-device sent unique frames in periodic or irregular manner, we did not take into account such values. The procedure to extract the periodicity of unique frames sent is described in Algorithm B-1.1 (see Appendix B).

The periods were grouped by end-device ID and the number of restarts as some end-devices might have restarted several times within one month. By taking the average (μ) and standard deviation (σ) of those grouped periods, the coefficient of variation ($c_v = \sigma/\mu$), also known as relative standard deviation (RSD), could be obtained and subsequently could be used to estimate the periodicity and irregularity of the traffic.



(a) Cumulative probability of c_v

(b) Transmission period histogram

Figure 6-2: Periodicity of packet transmission per session per end-device

The value of c_v close to zero indicates that the traffic tends to have a stronger periodic behaviour, whereas c_v larger than zero indicates that the traffic tends to have a higher irregularity level. Figure 6-2a illustrates the empirical cumulative probability (ECDF) of c_v. Around 93.7% of the average periods have $c_v = 0.0$, 98.58% have $c_v < 1$, and only 1.49% have $c_v >= 1$. The results indicate that most traffic in the currently deployed LoRaWAN networks, particularly in the dataset being analyzed, is periodic. Roughly 46% of transmission sessions have periods less than 1 minute, 23% have periods of 1 to 2 minutes, and the rest are spread up to 82 minutes. The histogram of different transmission periods is depicted in Figure 6-2b, and it can be fitted by an exponential distribution $n = a \cdot \exp(-b \cdot t)$ where $a = 0.46$ and $b = 0.69$. Note that only 10 packets were sent with periods less than 1 second, which violates the LoRaWAN duty-cycle.

Packet Retransmission, FCnt Overflow, and End-Device Restart

At least 5302 retransmission packets (0.36%) were obtained from the dataset of which 14.5% retransmit every 5 seconds, 9.92% every 8 seconds, and the rest were distributed evenly from 1 second up to 20 hours. Only one end-device (ID: 02031902) encountered FCnt overflow, three times at most. Finally, up to 33.6% of end-devices had never been restarted, whilst 10.04% had restarted once, and 7.17% twice.

Total Packet Loss

The result for packet loss is rather difficult to interpret since the calculation of packet loss depends solely on the difference of FCnt, that is, the end-devices may start sending packets and then move beyond the coverage of gateways for quite a long time, resulting in the large gap of FCnt. In other words, the distinction between packet loss due to collision and due to coverage outage could not be determined. Nevertheless, we roughly estimated that around 40.4% of packets were received successfully and 60.6% were lost. In order to obtain more reliable results, the spatial distribution of gateways and end-devices needs to be known.

6-2-2 Signal Quality

Signal quality analysis was performed using RSSI and SNR values as illustrated in Figure 6-3. The RSSI and SNR are obtained from frames because different gateways might receive the same unique frames but with different RSSI and SNR levels. The dataset does not contain any information regarding transmission power and receiver characteristics, such as receiver placement, antenna type, and gain. We assume that end-devices use transmission power $P_{TX} = 14$ dBm and are equipped with a pigtail antenna with typical gain $G_{TX} = 2$ dBi, as most LoRaWAN devices are deployed with these default settings. We also assume that the receivers have antenna gain $G_{RX} = 2$ dBi.

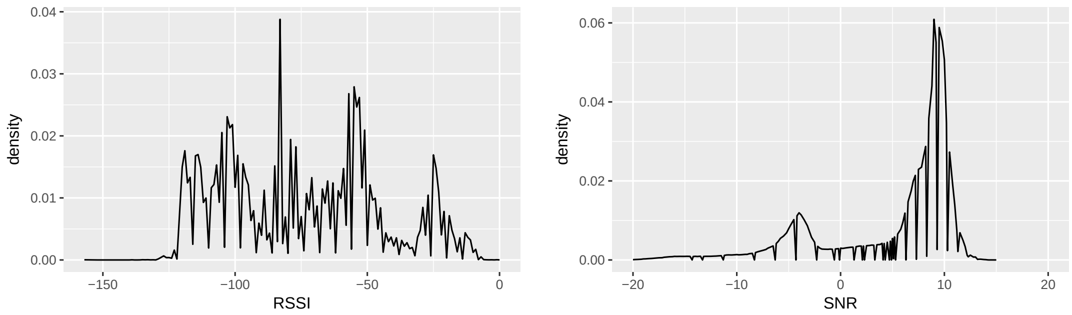


Figure 6-3: Signal quality indicator

In Figure 6-3, most end-devices have positive SNR, around 10, and RSSI above -100. The RSSI threshold or the receiver sensitivity is limited to around -130 dBm. The higher RSSI and SNR values indicate that the end-device was positioned close to the gateway. It is nevertheless not always the case, for example, if there are many interferers, which may be originated from other LoRa end-devices or coexisting radios, the RSSI may increase but the SNR value will drop drastically even though the end-device is located near the gateways. The spatial distribution of gateways and end-devices, as well as the time the packets are being sent, are also required to address such problems.

6-2-3 LoRa Parameter Utilization

Carrier Frequency Usage

There are three major frequency bands obtained from the dataset, namely 433 MHz, 863-870 MHz, 902-928 MHz. The histogram of those frequency bands is depicted in Figure 6-4.

Most end-devices were operating in 863-870 MHz frequency band and were located in Europe. Up to 60% of packets were received by the gateways located in the Netherlands. For the frequency band of 902-928 MHz, 28% of packets originated from the United States. However, for the frequency band of 433 MHz, the location of gateways and end-devices could not be obtained from the dataset. This thesis will focus on end-devices operating in the European frequency band of 863-870 MHz.

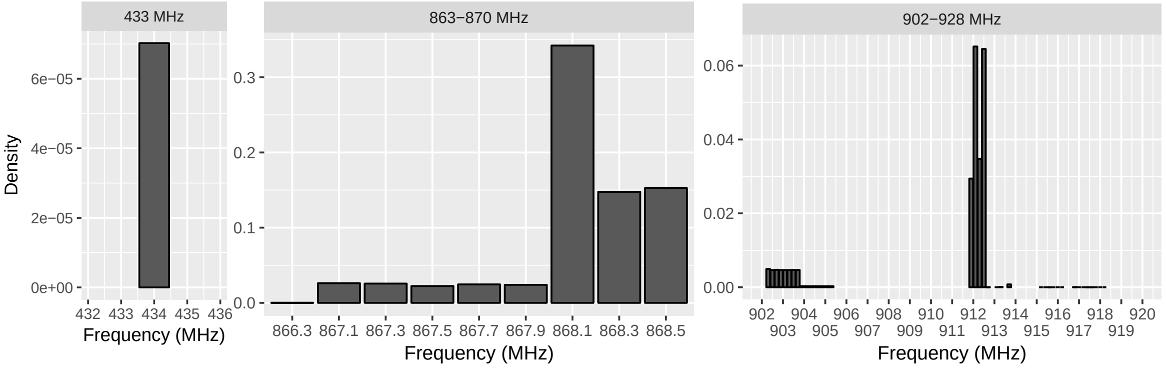


Figure 6-4: Frequency usage in three different bands

Data Rate

The histogram of data rates used by end-devices to transmit packets is illustrated in Figure 6-5. It can be seen that most end-devices used SF7 with 125 kHz BW, which is likely to be the default setting of typical LoRaWAN devices. We observed only 10 packets sent using DR6 (SF7BW250), which is typically not used in the commercial LoRaWAN deployment as we will discuss in the next chapter.

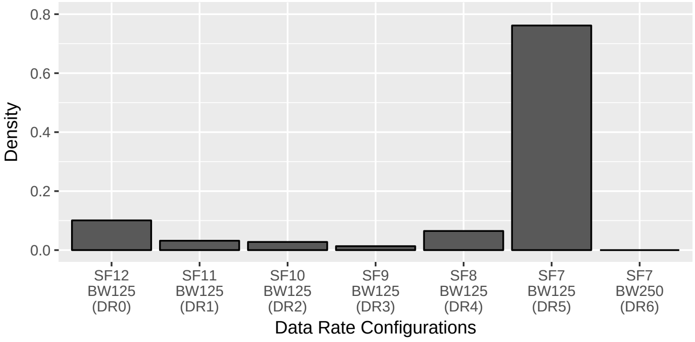


Figure 6-5: Histogram of data rates used by end-devices to transmit packets

6-2-4 Payload Analysis

The dataset contains decrypted data representing `FRMPayload`, and encrypted raw data denoting `PHYPayload`. Both data are encoded in Base64 format. The raw data can be decrypted using Semtech default keys mentioned earlier in Section 6-1. The raw data includes LoRaWAN headers comprising several parameters as shown in Figure 2-5 in Section 2-2-5.

Payload size

The size of payload influences the duration of packet transmission, which is also affected by data rate, particularly by SF. Transmission with higher SF and bigger payload requires longer time on air. It is therefore essential to obtain typical FRMPayload size. Figure 6-6 depicts the histogram of payload size for different data rates.

In accordance with Table 2-4 in Section 2-2-6, none of the packets exceeds the maximum payload size for each data rate. Most packets, that is, around 54% of packets have payload size less than 10 bytes, 38% have 10 to 50 bytes, and only 8% have payload size greater than 50 bytes. However, the number of packets that have payload size close to the maximum value of the corresponding data rate is significantly high, especially for the lower data rates (higher SFs). This condition prolongs the packet time on air and increases the probability of collision.

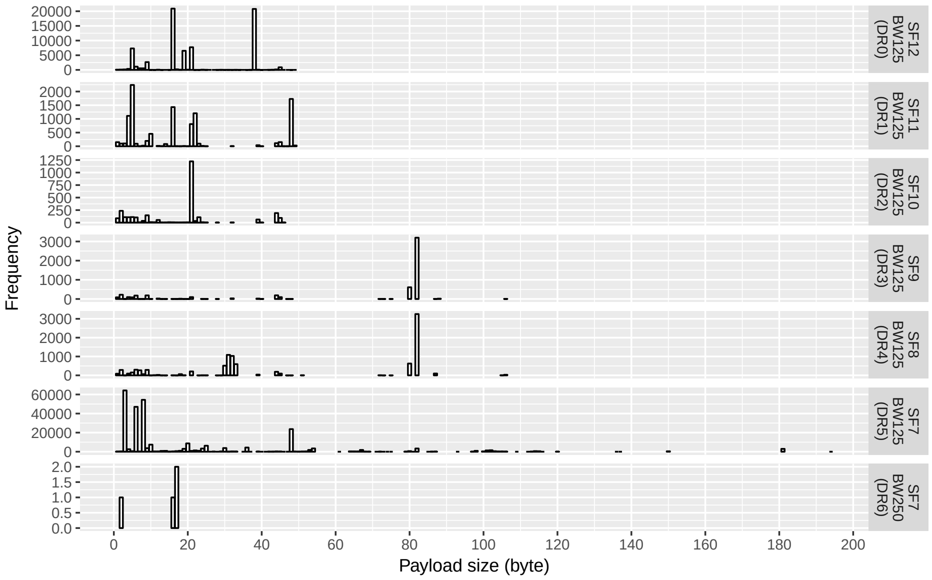


Figure 6-6: Histogram of payload size for different data rates

Frame Port (FPort)

In some cases, an end-device can serve multiple applications that require different application payload formats. FPort can be used as an identity to distinguish those formats. It helps the network server, particularly the application server, to decode messages correctly.

From the dataset, it is found that most end-devices are programmed to send packets with default FPort = 1. Some packets have FPort equally distributed from 10 to 255. Those packets originated from only one end-device, which might indicate user testing different kinds of FPort. Note that FPort 225-255 are reserved for further use [20].

No packets were sent with FPort = 0 (FRMPayload contains only MAC commands). MAC commands should be placed in FOpts field if FOptsLen is not 0. For this case, the FPort value should also be set different from 0. It can be concluded that all packets in the dataset did not contain any MAC commands inside the FRMPayload but rather placed them inside the FOpts fields. It also suggests that MAC commands should be piggybacked with application payload to reduce the number of uplinks.

Message Type

The type of message can be useful to determine the number of uplink and downlink messages that are sent within the specified period. The type of message can be revealed through the MType field. It is also interesting to take into account the frame control status (FCtrl) so that we can see whether the end-devices request for ADR or have MAC commands piggybacked in FOpts. The distribution of message types with their FCtrl options is listed in Table 6-2.

Table 6-2: Message type and frame control status (FCtrl) of packets sent

Message Type	ADR	ADRAckReq	ACK	FOpts	Frequency	Density
Unconfirmed Message	No	No	No	No	1158848	78.7411%
	Yes	No	No	No	245836	16.7040%
				Yes	15928	1.0823%
			Yes	No	4	0.0003%
				Yes	17	0.0012%
		Yes	No	No	12178	0.8275%
Confirmed Message	No	No	No	No	17278	1.1740%
	Yes	No	No	No	5519	0.3750%
				Yes	15077	1.0244%
			Yes	No	3	0.0002%
				Yes	347	0.0236%
		Yes	No	No	684	0.0465%
Total					1471719	100.00%

In Table 6-2, up to **96.53%** of packets were sent without requiring confirmations. This high number of unconfirmed packets strongly indicates that the downlink messages happened infrequently. However, the network server might arbitrarily schedule a downlink message, and the end-devices will receive it in one of their two receive windows. Since we do not have any dataset regarding downlink, we could not determine the total number of downlink frames, but we can still estimate it by looking into the number of confirmed messages as well as the status of ADRAckReq, ACK, and FOpts.

According to the LoRaWAN specifications, ADRAckReq bit will be available if end-devices use ADR. Whenever an end-device sends packets with ADR enabled, it will increment its ADR counter (ADR_ACK_CNT). If the ADR_ACK_CNT exceeds the ADR counter limit (ADR_ACK_LIMIT) without getting any downlink message from the network server, the end-device will send an ADRAckReq message, and the network server should respond by sending a downlink message. The ACK bit is enabled and sent to the network server whenever the network server requires a downlink message to be confirmed, meaning that the number of ACKs can be used to represent the number of downlink messages. A packet that contains MAC commands in FOpts fields might be followed or preceded by a downlink message, depending on the command identifier (CID) of the MAC commands. LinkCheckReq (0x02), for example, demands the network server to send a downlink message containing LinkCheckAns (0x02). If the packet is sent in the form of a confirmed message, then the LinkCheckAns command will be piggybacked with ACK. Another example is LinkADRReq (0x03), which is triggered by the network server. It requires the end-device to send an uplink message containing LinkADRAns (0x03), meaning that this command is preceded by a downlink message. Based on these facts, we could estimate the number of downlink messages from the dataset to be approximately **4.55%**, assuming no downlink retransmissions.

Adaptive Data Rate (ADR)

The characteristic of ADR messages can be analyzed so that one can have a clear recommendation whether implementing ADR increases network performance or not. The network performance may be degraded due to the high number of downlink messages carrying ADR parameters (data rate, channel, and transmission power) that need to be sent by gateways. It is important therefore to observe the average number of such messages.

ADR status can be obtained from `FOpts`. In Table 6-2, 31369 packets have `FOpts`, can be indicated by a non-zero value in `FOptsLen`. `FOpts` varies from 0 up to 15 bytes, and in the dataset, the maximum length is 8 bytes. `FOpts` field contains MAC commands of which the same command identifier (CID) can exist. For example, `LinkADRAns` (0x03) can be found multiple times in an uplink message. We focused on observing only the packets that contain `LinkADRAns` command as it indicates that the end-devices sending such packets have acknowledged the ADR parameters from the network server.

The total number of packets that contain `LinkADRAns` is 31113 transmitted from 24 end-devices. However, those packets do not include GPS coordinates, which can be used to analyze the impact of enabling ADR with different end-device density on the performance, particularly on the packet loss. Therefore, only temporal characteristics could be obtained. One method to get the temporal characteristics of ADR is by analyzing the time differences between two consecutive `LinkADRAns` packets sent by an end-device. The results, however, were difficult to interpret as they fluctuate significantly over time. Hence, an alternative method was used, that is, by counting the number of `LinkADRAns` packets per day per end-device. Figure 6-7 shows the histogram of the number of end-devices that sent `LinkADRAns` packets per day.

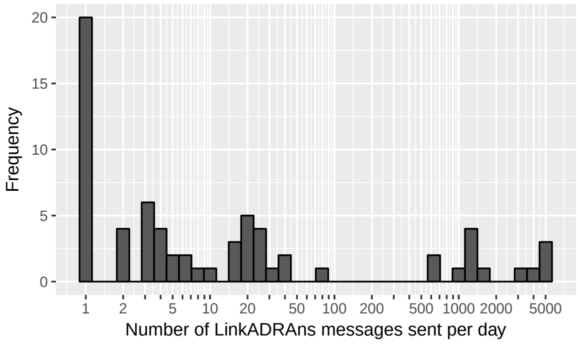


Figure 6-7: Histogram of the number of `LinkADRAns` messages sent per day

The results show that the end-devices typically sent `LinkADRAns` once a day or from 2 up to 5000 `LinkADRAns`. Such a high number of `LinkADRAns` is proportional to the number of packets sent, meaning that either ADR parameters could be received and acknowledged every time a packet was sent, or the `LinkADRAns` messages were repeatedly sent to ensure the network server receives the messages. Thus, the number of ADR transactions per day depends on the packet transmission periods and in addition, on the way the ADR mechanism is implemented in the network server. In the next chapter, we will evaluate ADR from downlink dataset.

6-2-5 Spatial Distribution

Gateways

The location of gateways can be revealed from the gateway status updates in the form of GPS latitude and longitude coordinates. We filtered out the outliers and checked the time consistency of gateway locations whether, at some point, the gateways are moved to the other places. We started by grouping the gateway status per day. For each day, the average and standard deviation of lat-long values were calculated. A threshold value of 0.01 was set to indicate whether the gateway can be categorized as a moved gateway or a fixed gateway. If a gateway has a standard deviation value larger than the threshold in one of the given days, then we consider the gateway as a moved gateway. Only gateways that have fixed locations are eventually considered for further analysis. From 229 gateways, 50 gateways had been moved and 179 gateways were steady.

Most gateways were located in Europe, especially in the Netherlands, up to 111 gateways. The detailed map of the distribution of gateways in the Netherlands is depicted in Figure 6-8. The map is categorized into major urban, urban, and rural landscape types based on population density taken from [56]. Note that the dots represent the gateway positions, and the shaded circle areas represent the typical coverage of gateways up to 15 km in relatively open space area when CF 868 MHz, SF12, and TP 14 dBm are used for the transmission [38].

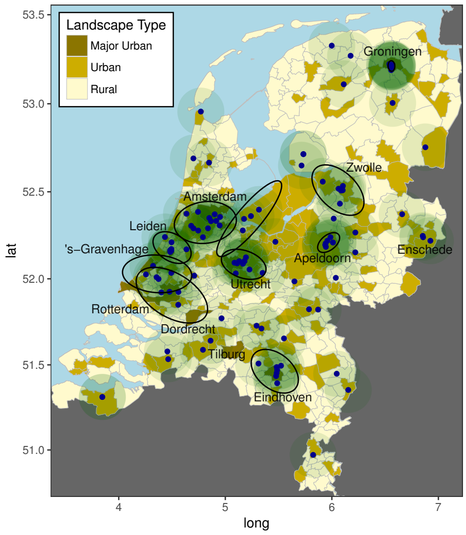
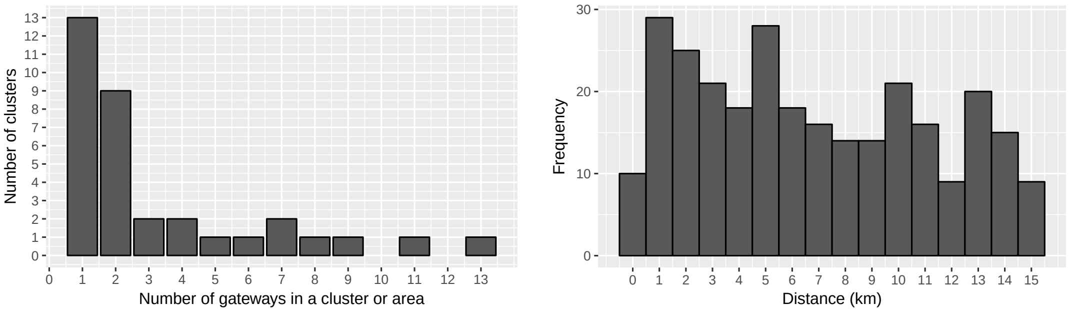


Figure 6-8: Spatial distribution of TTN gateways from the dataset in the Netherlands

In Figure 6-8, most gateways are densely distributed in major urban areas like Amsterdam, Utrecht, and Eindhoven. The black ellipse shapes represent the cluster of gateways classified using hierarchical agglomerative clustering method with a linkage criteria selected based on the average distance that is calculated from the distance matrix. The distance matrix is a square matrix that contains spatial-distance values for each pair of gateways.

The clustering information can be used to determine the number of gateways that overlap in a particular region. Figure 6-9a shows the histogram of the number of gateways found in a cluster. Most clusters contain only one gateway, meaning that the gateway either has a slight overlapping area or even does not overlap completely with other gateways. The maximum number of gateways that can be found in one cluster belongs to the region of Amsterdam, where 13 gateways are densely spread. In Groningen, the number of gateways is not as many as in Amsterdam, but the distance among the gateways is smaller, resulting in a denser distribution. Nevertheless, the result shows that the average number of gateways for a particular area within a radius of 15 km is 3.



(a) **(b)**

Figure 6-9: Histogram of (a) the number of gateways in one cluster or area, (b) distance between two gateways

In addition to the number of gateways, the distance distribution between gateways is required for developing a more realistic simulation. Figure 6-9b shows the histogram of distance between two gateways taken from the distance matrix. The histogram is limited to 15 km, which corresponds to the typical coverage radius of LoRaWAN gateway. The results indicate that the typical distance of gateways ranges from 1 to 2 km and from 5 to 6 km, which seems to be appropriate for urban areas.

End-devices

The location of end-devices can be obtained through GPS coordinates that are contained in the packet payloads. Several end-devices sent such packets with different kinds of format, encoded either in binary or string. For example, LoRaMote, a LoRaWAN demo platform made by Semtech, encodes GPS coordinates in a binary format as specified in [57]. As no information about other binary formats was available, we only focused on extracting data from LoRaMotes. Figure 6-10 shows the process of extracting GPS coordinates from the dataset using MySQL scripting. Note that the altitude values could also be extracted, but most of them were unreliable and were filled with 0.

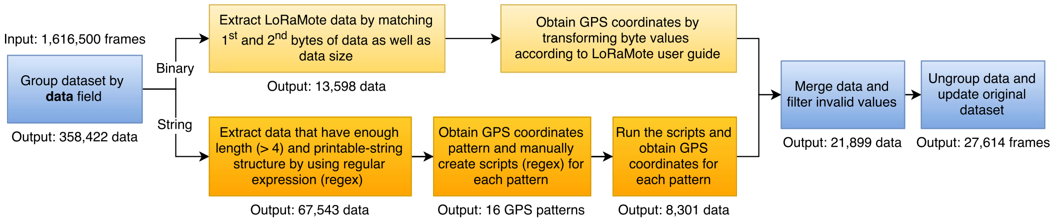
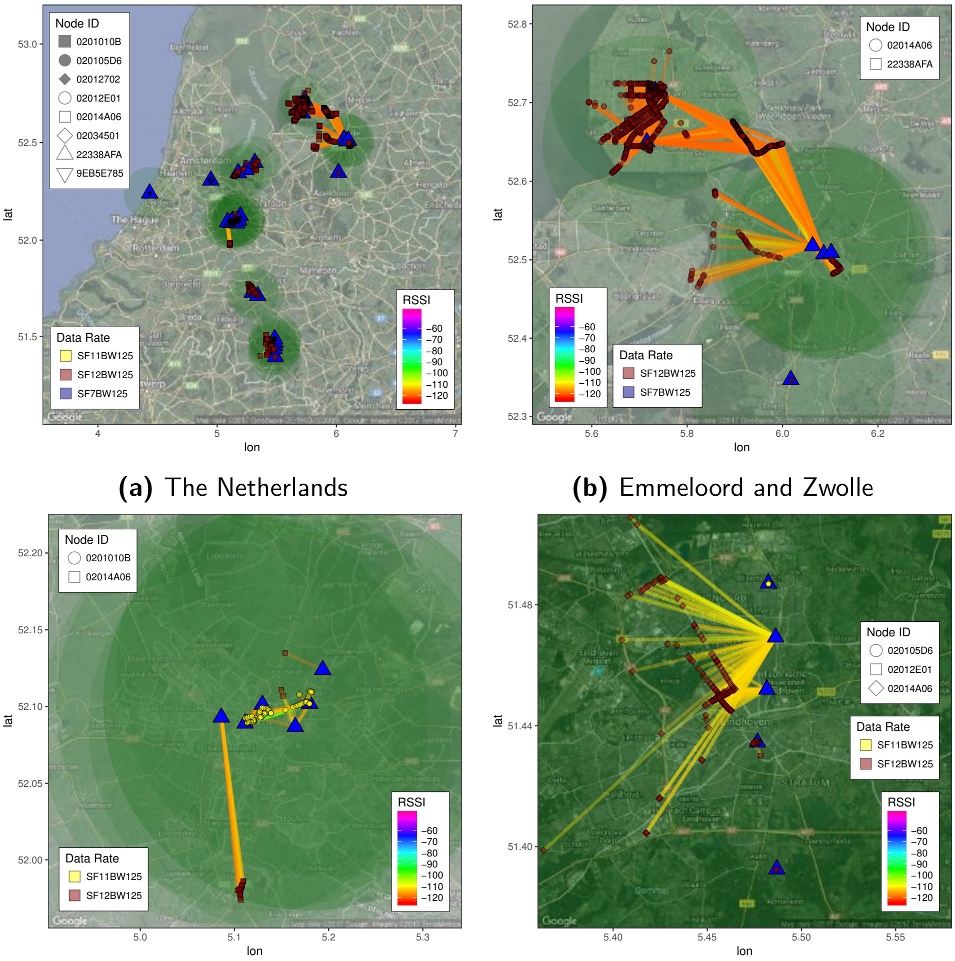

Input: 1,616,500 frames
Group dataset by **data** field
Output: 358,422 data
Binary
String
Extract LoRaMote data by matching 1st and 2nd bytes of data as well as data size
Output: 13,598 data
Obtain GPS coordinates by transforming byte values according to LoRaMote user guide
Extract data that have enough length (> 4) and printable-string structure by using regular expression (regex)
Output: 67,543 data
Obtain GPS coordinates pattern and manually create scripts (regex) for each pattern
Output: 16 GPS patterns
Run the scripts and obtain GPS coordinates for each pattern
Output: 8,301 data
Merge data and filter invalid values
Output: 21,899 data
Ungroup data and update original dataset
Output: 27,614 frames

Figure 6-10: Procedures taken for extracting GPS coordinates from the dataset

Ultimately, only 8 end-devices were located in the Netherlands in which up to 6937 frames (6460 unique frames) were captured in total by 25 gateways. Those end-devices were relatively mobile as can be seen in Figure 6-11.



(a) The Netherlands

(b) Emmeloord and Zwolle

(c) Utrecht

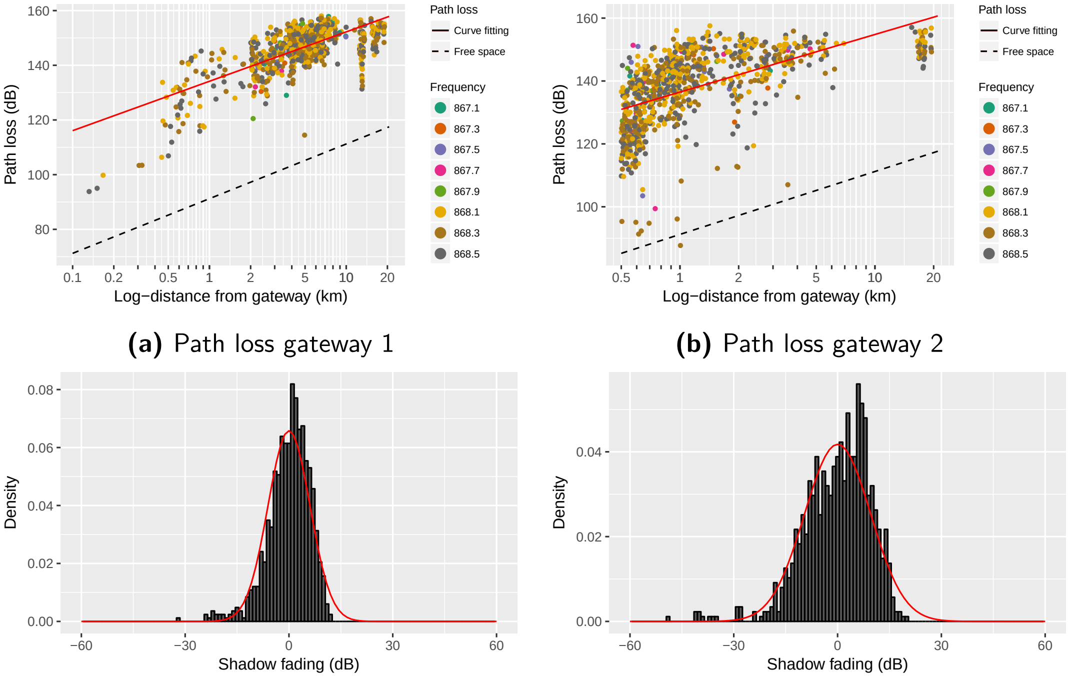
(d) Eindhoven

Figure 6-11: Spatial distribution of TTN end-devices with GPS observed in the dataset

In Figure 6-11, the points with different shapes represent the location of end-devices at the time of transmissions, whilst the blue triangles represent gateways that have fixed locations. The end-devices and the gateways are connected with rainbow-colored lines representing signal strength or RSSI of the transmissions. Most end-devices transmitted packets using SF11 and SF12 with 125 kHz bandwidth. Such packets were received by one or multiple gateways with RSSI values ranging from -100 to -120 dBm. Some packets could reach a gateway beyond the typical range of 15 km, whilst some others were transmitted while being very close to a gateway. One factor that could influence this coverage range is the type of landscape, which correlates to the path loss exponent. The extraction of path loss exponent will be discussed in the next section.

6-2-6 Coverage

The empirical path loss can be calculated from RSSI and SNR using Equation 3-5 in Section 3-2-3. We assume gateways and end-devices have 2 dBi antenna gain. As the measurement data were concentrated in the Emmeloord area (5631 frames), we initially extracted path loss and shadowing for this area, which can be classified as a flat and open rural area. Only frames with SF12BW125 were considered as approximately 99% of the frames used this data rate.



(a) Path loss gateway 1

(b) Path loss gateway 2

(c) Log-normal shadowing gateway 1

(d) Log-normal shadowing gateway 2

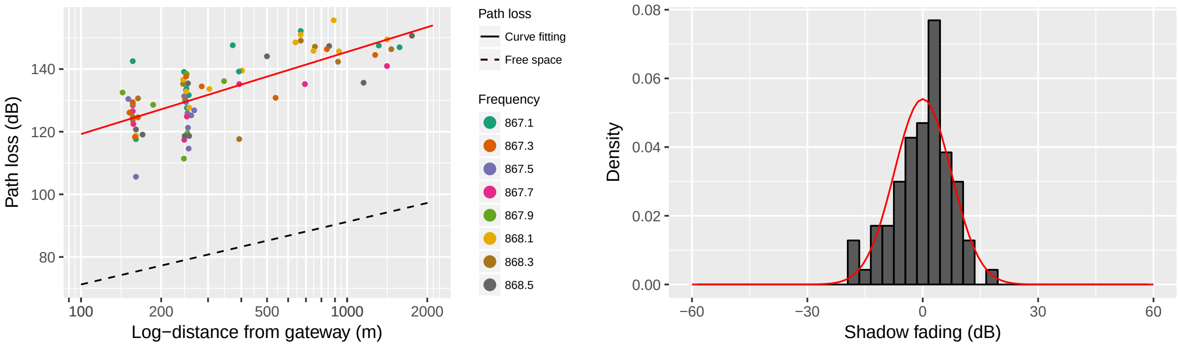
Figure 6-12: Path loss and log-normal shadowing distribution in Emmeloord (SF12 and BW125)

Two gateways, GW1 (ID: 1DEE0CAA2B8B08C9) and GW2 (ID: FFFEB827EBF7A0CA), were located in Emmeloord. The distance between these two gateways is approximately 7.4 km. The reference distance d_0 is set to 100 m, which is appropriate for outdoors. Some measurements with distance shorter than 500 m from GW2 have very strong signal powers, which appear to be the outliers. We performed linear curve-fitting based on Equation 3-6 to get the path loss exponents and plotted the result as expected path loss. The difference between the measured path loss and the expected path loss is used to obtain the standard deviation (σ_{SF}) of log-normal shadowing distribution. The path loss and log-normal shadowing distribution for the Emmeloord region are depicted in Figure 6-12.

It is worth noting that the aforementioned path loss model is intended for all channels, not for a particular channel as only few frames were received. Thus, the mean frequencies of all the channels, i.e. 868.2129 MHz for GW1 and 868.0636 MHz for GW2, are used for calculating free-space path loss [38].

Path loss and shadowing from other regions could also be obtained, specifically for the Utrecht region. Since only few measurements could be found in this region, three gateways (ID:

008000000000A888, 1DEE02A9759F35A1, B827EBFFFE4614EB) that are located on the same landscape type, i.e. urban, were selected for the analysis. Most frames received by these gateways are sent from one end-device using SF11 and BW125 (78 frames). The results are shown in Figure 6-13



(a) Path loss

(b) Log-normal shadowing

Figure 6-13: Path loss and log-normal shadowing distribution in Utrecht (SF11 and BW125)

After obtaining the path loss exponent, the coverage radius of the gateways can be estimated by using Equation 3-7. The results can be seen in Table 6-3.

Table 6-3: Coverage, path loss exponents and shadowing for Emmeloord and Utrecht regions

			Emmeloord		**Utrecht**	**Free Space**
			GW1	**GW2**	**All GWs**	
Average frequency ($\bar{f_c}$) **[MHz]**			868.2129 MHz	868.0636 MHz	867.8026 MHz	868 MHz
Path loss exponent (γ)			1.806	1.83113	2.6234	2
Mean path loss at d_0 ($\bar{L_{pl}}$) **[dB]**			116.0952	118.17138	119.2509	71.22
Shadowing (σ_{SF})			6.064119	9.551812	7.387089	-
Coverage radius (R) **[km]**	**SX1301**	**DR0**	19.6	14.1	2.9	2060.6
		DR1	13.4	9.6	2.2	1458.8
		DR2	9.7	7	1.8	1094
		DR3	7.1	5.1	1.4	820
		DR4	5.1	3.8	1.1	615.2
		DR5	3.7	2.7	0.9	461.3
	SX1276	**DR0**	12.6	9	2.1	1377.2
		DR1	8.6	6.2	1.6	975
		DR2	7.5	5.5	1.5	869
		DR3	5.1	3.8	1.1	615.2
		DR4	3.5	2.6	0.9	435.5
		DR5	2.4	1.8	0.7	308.3
	SX1272	**DR0**	14	10.3	2.3	1545.3
		DR1	11.1	8	1.9	1227.4
		DR2	8.6	6.2	1.6	975
		DR3	5.8	4.3	1.2	690.2
		DR4	4	2.9	1	488.7
		DR5	2.7	2	0.7	345.9

For the Emmeloord region, the results show that both gateways have similar path loss exponents slightly below free-space path loss, which is reasonable as the landscape is relatively open and flat. For the Utrecht region, the path loss exponent fits with typical path loss exponents for suburban and urban area. Adding more gateways in the urban area is recommended to reduce packet loss caused by coverage outage and collisions.

6-3 Conclusion

The TTN LoRaWAN network has evolved substantially since the data was collected in June 2016 and is still being developed to provide more LoRa coverage empowered by the users as a community. At the time the dataset was collected, most packets were received only by one gateway, and only one packet was received by 20 gateways. It implies that the TTN gateways are sparsely distributed, particularly in the rural areas. This situation is supported by the fact that up to 60.6% of packets were estimated lost apparently due to coverage outage as only 488 end-devices were observed. The end-devices sent packets predominantly in a periodic fashion in which up to 93.7% of unique packets sent with regular periods, and thus the Poisson processes, which is typically used to model the packet arrival, may give less accurate results for such periodic transmissions [29].

Frequency utilization in the three LoRaWAN default channels is still considerably high for confirmed and unconfirmed uplink messages. This situation is understandable due to the nature of the early-stage development and the use of ABP in which the users need to set the frequency channels of the devices manually. The network server can add new channels or modify the existing channels later, and it is recommended to use different frequency channels other than the default channels for each packet transmission to reduce collision. In addition, the probability of collision can be decreased when the time on air of packets become shorter by reducing SF and payload size. The SF settings can be changed by the network server through ADR, which, from this dataset, is typically sent once a day.

Most of the TTN gateways are densely distributed in urban areas and are mainly separated 1-2 km apart. However, the coverage radius in the urban area is smaller compared to the rural area due to obstacles, e.g., buildings. For example, in Utrecht, the uplink transmission can be received by the gateway with SX1301 chipset positioned 2.9 km away when SF12 is used, while it can achieve 19.6 km in the rural area like Emmeloord. As the end-devices typically use SX1272 and SX1276 chipsets, downlink transmission will have a lower coverage radius, and thus up to 27 dBm transmission power is supported in RX2.

Chapter 7

Real-World LoRaWAN Evaluation: KPN Dataset

This chapter presents data processing, analysis, and evaluation of the KPN LoRaWAN dataset. This dataset is newer than the TTN dataset evaluated in the previous chapter, and it represents commercial usage of a LoRaWAN network.

7-1 Dataset Overview

The KPN dataset comprises 33.2 GB gateway log files collected from 18 to 25 June 2017. It was handed to us by Bart Hendriks and Ronald Rust from KPN. The log files were generated from 725 gateways deployed across the Netherlands, which comprises a large subset of KPN's gateways. Two different log formats were found. One for gateways with prefix ID "08" and "29", and the other one for gateways with prefix ID "FF" indicating the upgraded version. We developed a Python script to handle various log formats as well as to extract and merge uplink and downlink data.

7-2 Data Processing and Analysis

7-2-1 Gateway Statistics

The statistics of KPN gateways can be seen in Table 7-1. We have received a list of gateway position in the form of lat-long coordinates, but not all gateways found in the log files appear in the list. We end up with 616 gateways for the evaluation.

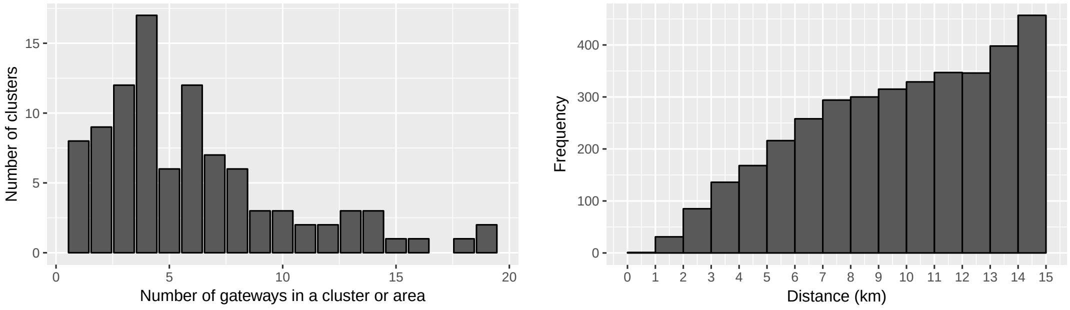
Spatial Distribution

The KPN gateways locations can be grouped into clusters using the same method as in the TTN analysis. Figure 7-1a shows the histogram of the number of gateways found in a

Table 7-1: KPN Gateway statistics

Description	Number of gateways
Receive uplink	717
Transmit downlink	640
Utilize GPS timestamp	692
Utilize GPS timestamp and have lat-lon coordinate	616

cluster. Most clusters contain 4 gateways, and the average number of gateways for a particular area within a radius of 15 km is 6, which is denser compared to TTN. Figure 7-1b shows the histogram of distance between two gateways trimmed to 15 km. It shows that the distribution of the distance between two gateways tends to be uniform for distances large than 7 km, but it has a lower probability for the shorter distance. This finding suggests that the KPN gateway placement is more uniform than in TTN, where users may deploy their gateways close to each other.



(a)

(b)

Figure 7-1: Histogram of (a) the number of gateways in one cluster, (b) distance between two gateways

Most of the gateways are located in urban areas like Amsterdam, Eindhoven, The Hague, and Rotterdam. The high density of gateways is expected to give an improvement in the network capacity, especially in the urban area where the demand for IoT applications using LoRaWAN and other coexisting technologies is relatively high. Such dense gateway placements affect the packet reception. As can be seen in Figure 7-2, only 20.46% of packets were received by one gateway, while 78.43% of packets were received by 2 up to 30 gateways. This result shows that the KPN network is more mature, providing better coverage and redundancy than observed in the older TTN dataset.

7-2-2 End-device Statistics

The number of active end-devices can be revealed from device address that is embedded in the frame payload. However, frames with CRC error do not contain any frame payload. Table 7-2 shows end-device statistics based on device address.

A device address is not unique and may be used by multiple end-devices that can be identified from their unique hardware serial (EUI). The gateways have no information regarding hardware EUI upon reception of uplink transmission. Hardware EUIs are stored in the network

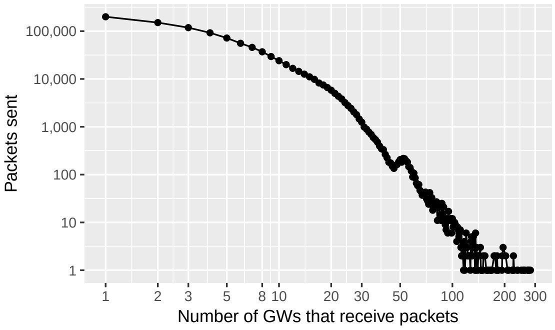


Figure 7-2: Histogram of the number of packets that are received by multiple gateways

Table 7-2: End-device statistics based on device address

Description	Number of unique device addresses
Transmit uplink and/or receive downlink	334,117
Transmit uplink and receive downlink	6,556
Transmit uplink only	311,819

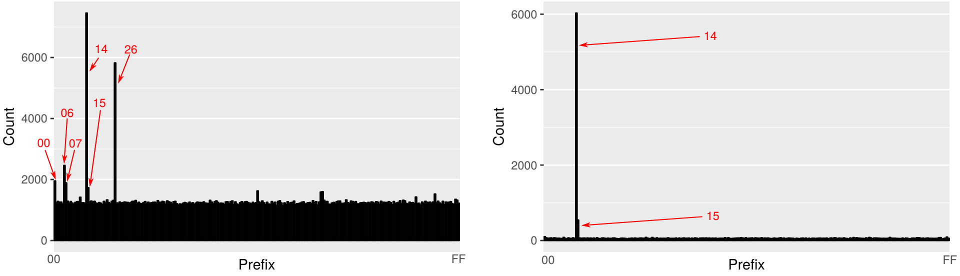
server. When the network server sends a downlink packet to a gateway, the target end-device and its hardware EUI will be informed to the gateway. By examining the downlink dataset, hardware EUI with its corresponding device address can be mapped, but only for end-devices that are registered to KPN network.

Packets that are intended for different network operators are also received by the KPN gateways in the vicinity. By using device address prefix, we can distinguish the network to which the end-device is transmitting the packet. Table 7-3 shows prefix assignment for several network operators [58], and Figure 7-3 depicts the histogram of the number of device addresses for each prefix found in the dataset.

Table 7-3: Device address prefix assignment

Network	NetID	Prefix
Experimental nodes (World)	0x00	0x00
		0x01
	0x01	0x02
		0x03
Proximus (Belgium)	0x03	0x06
		0x07
KPN (Netherlands)	0x0a	0x14
		0x15
TTN (World)	0x13	0x26
		0x27

For uplink transmission, most end-devices are registered to KPN and TTN networks. Some end-devices have 0x06 and 0x07 prefixes, which are seemingly registered to the Proximus network (Belgium), while other devices are marked as experimental nodes. For downlink transmission, end-devices from other networks are not included in the dataset. The rest prefixes are assigned typically for OTAA. It should be noted that even though the received packet type is join-request, the packet has an arbitrary device address in the gateway log files, which should not be the case because a join-request frame does not include device address in



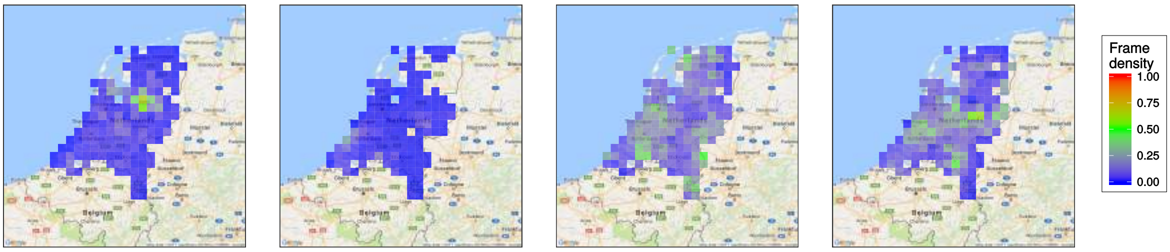
(a) Uplink

(b) Downlink

Figure 7-3: Histogram of device address assignment

its payload.

Intuitively, packets sent by Proximus devices should be received most by the gateways close to the Belgium-Netherlands border. Figure 7-4 shows the distribution of captured frames sent by end-devices registered to different networks. The color gradient represents the ratio between the number of received frames of a particular network to the aggregated frames from all networks. We can see that frames sent by Proximus devices can be received by most KPN gateways, and thus it is most likely that such frames are not transmitted from those devices unless they move around the Netherlands. As our goal is to observe the influence of other networks to the performance, we will focus on the noticeable KPN and TTN devices and leave the rest as other networks.



(a) Experimental nodes **(b)** Proximus **(c)** KPN **(d)** TTN

Figure 7-4: Distribution of frames sent by end-devices that are registered to different networks

7-2-3 Traffic Characteristics

The number of frames received and transmitted by gateways can indicate the congestion level in the surrounding area where the gateways are located. Table 7-4 describes the number of frames classified based on packet direction and message type.

Roughly half of the total uplink frames are received with CRC error status, which would not be forwarded to the network server. Such high number of bad CRC frames might be originated from end-devices that are located close to specific gateways, but far from the other gateways. Also, collisions might happen at those gateways. However, as long as one frame is received correctly, it will not significantly impact on DER. We obtained over 4.5 million

Table 7-4: Frame statistics based on packet direction and message type

By directions				
Uplink	Frames	CRC OK	GPS timestamp at gateway	21,939,429
			No GPS timestamp at gateway	274,435
			Total	**22,213,864**
		CRC Error		**15,403,806**
		Total		**37,617,670**
	Packets			**4,579,705**
Downlink	Transmitted			377,947
	Error			26
	Other			64
	Total			**378,037**

By message type	
Join-Request	1,166,028
Join-Accept	15,614
Unconfirmed Uplink	17,439,086
Unconfirmed Downlink	362,409
Confirmed Uplink	2,086,528
Confirmed Downlink	0
RFU	307,007
Proprietary	281,172
Total Uplink	**21,279,821**
Total Downlink	**378,023**

unique uplink packets from the received frames, which is more than 10 times larger than the number of downlink frames. Those downlink frames are already unique of which each packet is generated by only one best-selected gateway. However, some downlink frames cannot be transmitted due to the busy gateway state or late response from the network server.

Frame statistics can be obtained based on message type from the uplink dataset that contains CRC OK status. The result shows that the unconfirmed uplink frames outnumbered the confirmed uplink frames, which also correlates to the low number of downlink frames. Some frames have RFU and proprietary types and appear to have a non-LoRaWAN compliant format. Such frames might be originated from network testing or system development.

Payload Size

Most unique packets contain payload with less than 60 bytes as seen in Figure 7-5. It indicates that the current LoRaWAN network is used by applications that require only a few bytes in each transmission, such as for environmental monitoring.

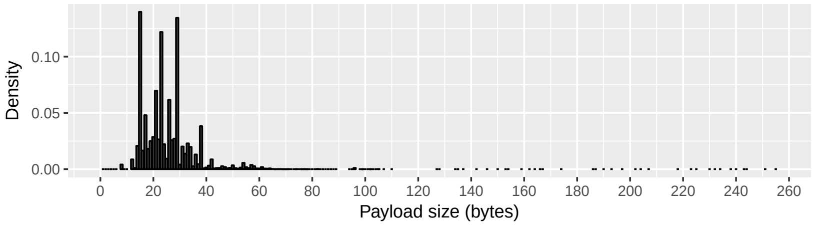


Figure 7-5: Histogram of PHY payload size

ADR Traffic From KPN End-Devices

In this chapter, ADR traffic is obtained from the downlink dataset by counting the `LinkADRReq` (0x03) command in the unconfirmed downlink frames targeted for up to 6344 unique end-devices. Other MAC commands such as `RXParamSetupReq` (0x05), `DevStatusReq` (0x06), and `NewChannelReq` (0x07) could be observed as well. The `RXParamSetupReq` command is used to change the RX2 data rate and frequency. The `DevStatusReq` command is used to obtain the status of end-devices in terms of battery level and SNR margin. We figured out that `DevStatusReq` command is regularly sent once a day for each end-device. The `NewChannelReq` command can be used to either create a new frequency channel or modify the existing one. Figure 7-6 shows histogram of the number of end-devices to the number of downlink messages sent per day for different MAC commands.

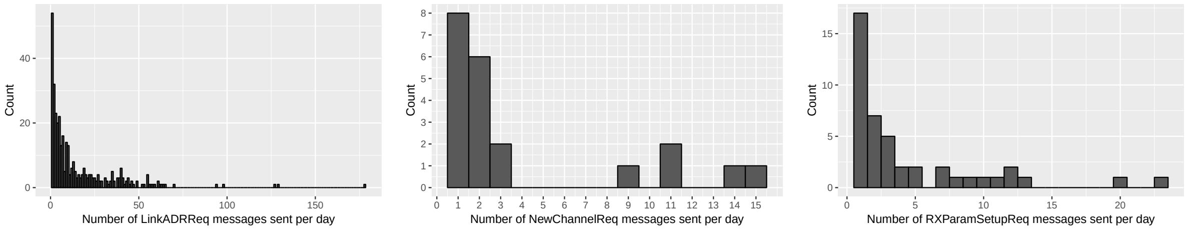


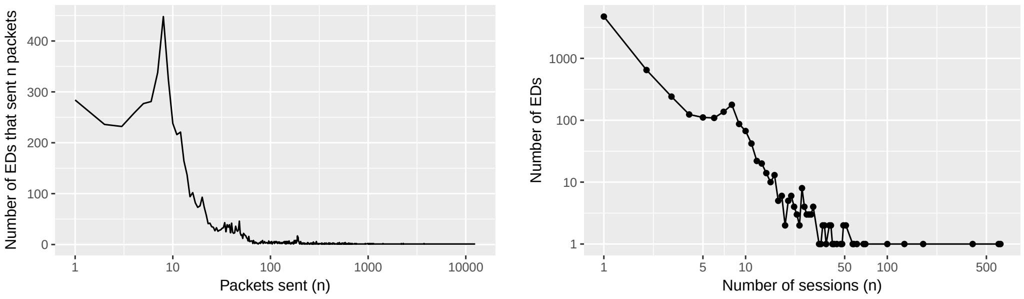
Figure 7-6: Histogram of downlink messages with MAC commands that are transmitted per day

The results are similar to the previous chapter in which most end-devices tend to receive ADR message once a day. In the `NewChannelReq` and `RXParamSetupReq` plots, the count number and the messages sent per day are relatively low. It indicates that only a few end-devices receive such messages, which may not be transmitted in the subsequent days. Ultimately, we found that the new frequency channel list will be assigned by the network server when end-devices initiate uplink transmission for the first time in a new session. The network server then utilizes ADR commands to enable or disable certain frequency channels. For example, when one of the eight channels is congested, the network server can instruct end-devices not to use that channel. By doing this, the number of collisions is expected to decrease.

KPN End-Devices Packet Transmission Periodicity

We extract the uplink periodicity of KPN end-devices and compare it with the TTN network. The first step is to filter out the dataset, keeping only packets that are transmitted from KPN end-devices (device address prefix 0x14 and 0x15), and then select only one packet from any receiving gateways to obtain unique confirmed and unconfirmed uplink packets. Up to 6646 KPN devices were found in the filtered dataset with a total of 977,551 unique packets. In Figure 7-7a, however, most of the KPN devices sent only a few packets over one week period. This might indicate either users testing the network or infrequent message transmissions. Later, we will investigate the periodicity of packet transmission per end-device in a particular session where FCnt is reset back to 0. The histogram of the number of end-devices experiencing n network sessions is depicted in Figure 7-7b. Most end-devices renewed their sessions or restarted less than 10 times over the specified period.

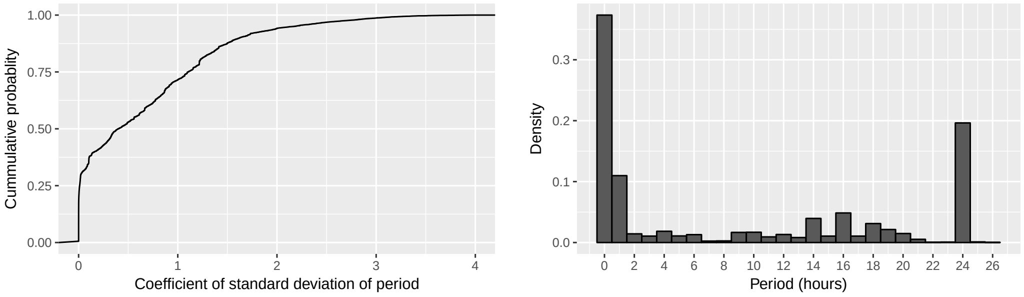
Similar to the TTN analysis, the unique packets are grouped by device address and network session to determine the transmission interval (T) of two consecutive packets while ignoring packet loss and retransmission. The coefficient of variation (c_v) is also used to estimate the transmission periodicity. The result is depicted in Figure 7-8a, and it shows that the KPN traffic periodicity has more variations compared to TTN and tends to have a higher irregularity. It can be seen that 31.43% of sessions have $c_v = 0$, 70.27% $c_v < 1$, and 29.72 % $c_v \geq 1$. Figure 7-8b shows the histogram of the KPN end-devices transmission period with $c_v < 1$. In most sessions, packets were transmitted with an interval less than one hour. Most packets were also sent once a day, which is different from TTN traffic. As the TTN dataset was captured in the early-stage of development, many users favoured testing their end-devices



(a) **(b)**

Figure 7-7: Histogram of (a) the number of end-devices that transmit n unique frames, and (b) the number of end-devices experiencing n sessions

using default settings with fixed transmission periods, while in the KPN network, we believe that most end-devices were used for real use-cases.



(a) Cumulative probability of c_v **(b)** Transmission period histogram

Figure 7-8: Periodicity of packet transmission per session per end-device for KPN network

7-2-4 LoRa Configuration Statistics

The selection of parameters used for transmitting a packet can contribute to frame collisions, particularly when the packets are sent with the same bandwidth, spreading factor, and carrier frequency. Figure 7-9 shows the histogram of spreading factor and frequency utilization of the unique packets for different networks found in the dataset. It should be noted that all packets in the dataset are transmitted using 125 kHz bandwidth.

The spreading factor histogram signifies that the usage of SF12 remains high, similar to SF7. The high number of SF12 utilization can increase the frame collision probability due to the longer air time. The situation might become worse as most packets are sent using the LoRaWAN default frequency channels. It can be seen that KPN devices utilize most of the available channels, while the TTN devices are mainly concentrated in the default channels, especially in 868.1 MHz. By forcing end-devices to switch to the other less-occupied channels, the probability of frame collision can be reduced but at the expense of more downlink overhead. In the next section, we will see that the devices registered to KPN network have a lower aggregated packet loss ratio compared to the other networks.

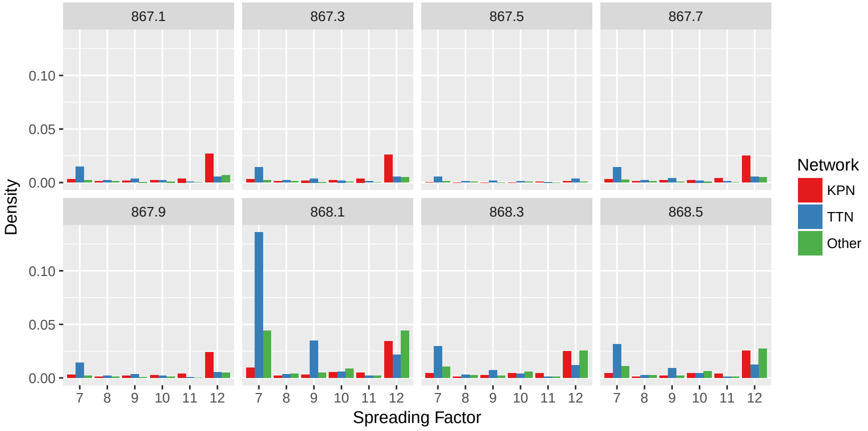


Figure 7-9: Histogram of the usage of LoRa configuration parameters from all gateways

Around 99% of packets use coding rate 4/5 and only 0.1% use coding rate 4/8. The higher coding rate is typically used to transmit more important messages, such as confirmed uplink and join-request messages. We believe that most end-devices will still send packets using coding rate 4/5, and thus the contribution of coding rate to the frame collision will be less.

7-3 Performance Evaluation

7-3-1 Aggregated Packet Loss

In this section, the network performance can be assessed by observing the average packet loss ratio (PLR) of a group of end-devices registered to certain networks over a period. The analysis is intended for the unconfirmed and confirmed uplink packets as such packets have FCnt values incremented after transmission. This procedure is similar as in the previous chapter in which the dataset from entire gateways is grouped by device address and network session to estimate the total number of packet loss and packets sent. However, we found frequent timestamp glitches for both gateway log time and GPS time fields, thereby making the FCnt sequences unordered, especially when dealing with multiple network sessions as the same FCnt can occur multiple times. Therefore, instead of using the whole one-week time-frame, the packet loss measurement is partitioned into sub-intervals per minute, per hour, and per day. The use of different intervals is to check whether the result in one particular interval holds in the other intervals and to accommodate the packets that are transmitted with larger periods. Figure 7-10 depicts the cumulative probability of average packet loss aggregated from all end-devices within the same network.

The KPN with OTAA devices curve includes not only KPN devices with prefix 0x14 and 0x15 but also devices that have addresses listed in the downlink dataset. The dataset contains 132 OTAA devices contributing to the 2.6% of the total packets, i.e., 3,734,866 packets where 29.7% from 6701 KPN devices, 45% from 4629 TTN devices, and 22.7% from 6945 other devices. The results show that the OTAA devices do not significantly influence the packet loss ratio for KPN devices, whereas for the other devices, it does reduce the probability of packet loss ratio as these OTAA devices most likely utilize all the eight allocated channels.

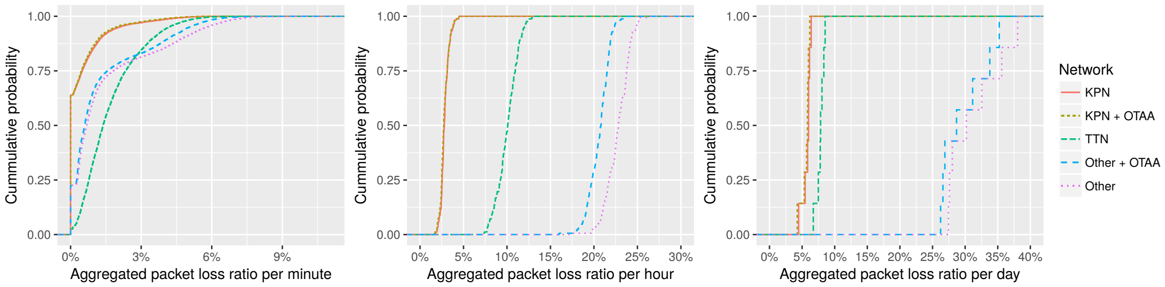


Figure 7-10: Cumulative probability of average packet loss with different time intervals

We can clearly see that the KPN network outperform TTN and the other networks in terms of average packet loss ratio. The KPN network yields a promising result where up to 63.89% probability of getting 0% packet loss per minute can be achieved, with only 2% packet loss per minute at 95th percentile, whereas TTN and the other networks provide 4% and 5.7% packet loss per minute at 95th percentile, respectively. The results also hold for the larger intervals. The fact that most TTN devices use default 868.1 MHz frequency becomes a strong reason why the packet loss ratio increases for the TTN network. In the next sub-section, we will validate this by evaluating channel utilization.

7-3-2 Channel Utilization and Collision Rate

In order to observe the influence of packets sent by other networks, one can measure the utilization of communication channels over a time period. The intuition is that when other networks use the same channel, the channel becomes quickly saturated or even over-utilized, leading to collision and packet loss. Several gateways in a particular area are selected based on gateway density, traffic volume, end-device count, spreading factor distributions, and time consistency to reduce the complexity of the analysis. The selection goes to 27 gateways located around The Hague and Rotterdam as seen in Figure 7-11.

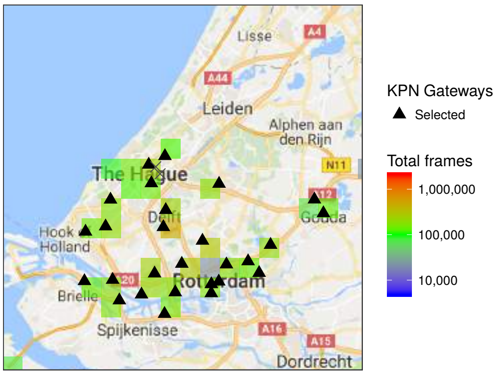
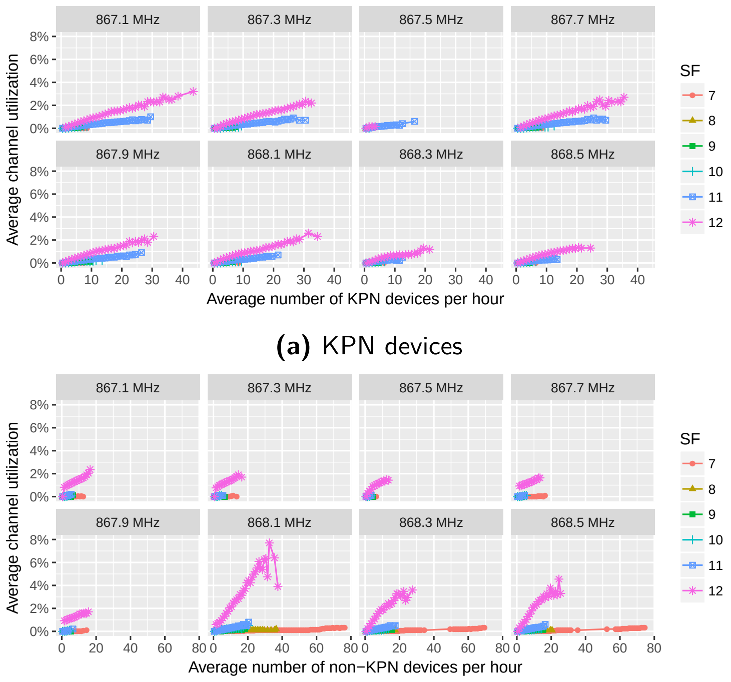


Figure 7-11: Selected gateways around The Hague and Rotterdam area

The time on air of packets received at these selected gateways is calculated and grouped based on SF and CF. Note that apart from CF, SF can also be considered as a channel, assuming the orthogonality of SF. By sorting the arrival timestamp, the packets are then grouped into partitions of one-hour length, and the time on air of those packets are summed up per SF and CF. Finally, the average channel utilization per hour from all selected gateways over a week period can be measured. The results can be seen in Figure 7-12. It is obvious that higher SFs give longer time on air and increase the channel utilization. The main point is that the utilization is relatively uniform in all CFs for KPN devices, whilst for devices from other networks, the utilization is quickly escalated in the three LoRaWAN default CFs, even for the small number of devices. This corresponds to the number of packets being sent by these devices. The result is also on a par with the LoRa configuration distributions from all gateways as previously depicted in Figure 7-9.



(a) KPN devices

(b) Devices from other networks

Figure 7-12: Average channel utilization per hour

One can expect packet loss and DER measurement after obtaining the channel utilization distributions. However, these metrics are difficult to be measured as the mobility and spatial distributions of the devices are not known. Also, the devices and the network server might alter the LoRa transmission parameters, making it difficult to classify packet loss based on SF and CF. Instead, we use packet survival rate representing the number of packets that survived collisions, which is mainly attributed to the capture effect. The result is depicted in Figure 7-13 for all devices.

As we expected, the collision rate increases significantly for SF12 in 868.1 MHz frequency. Up to 80 devices sent packets using SF7 within one hour, but the collision rate is still low. Based on these findings, it is recommended not to use SF12 under a multiple gateways environment and to avoid using the default LoRaWAN channels for transmitting uplink packets.

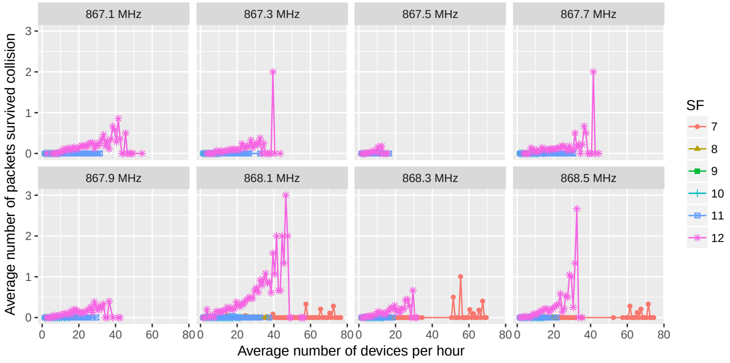


Figure 7-13: Average number of packets survived collisions

7-4 Conclusion

From this chapter, we can see that the commercial KPN LoRaWAN network provides better coverage and reliability compared to the older data from the crowdsourced TTN network, providing up to 63.89% probability of getting 0% packet loss per minute. Also, the KPN network server effectively used MAC command to instruct end-devices to enable certain frequency channels and switch the data rates. This process is typically performed when the devices connect to the network for the first time in a new session. In terms of channel occupancy, the current use of LoRaWAN is still considerably low, i.e., only 8% utilization. As most devices from other networks use the default LoRaWAN frequency, the effect on the overall channel occupancy will be less. However, further investigation needs to be done in case of bursty traffic as it can cause packet loss and alter the calculation of channel utilization.

Chapter 8

Conclusion and Future Work

8-1 Conclusion

In this thesis, an extensive evaluation of LoRaWAN performance in small-scale and large-scale networks has been performed, in particular, on the characteristics of LoRaWAN frame collisions. Also, two major LoRaWAN networks in the Netherlands, namely The Things Network (TTN) and KPN, which respectively represent the crowdsourced and commercial usage of LoRaWAN networks, have been evaluated.

Starting with small-scale experiments, we tried to answer the research question regarding the aspects that degrade the performance. Spreading factor is one of the leading parameters that influence the time on air. Our experiments show that packets with SF7 and SF8 suffer from packet loss, which is apparently due to the coverage outage, while SF12 does improve the coverage but increases the collision probability at several gateways, leading to packet loss. However, from the application layer standpoint, there will be no packet loss as long as one gateway receives the packet. Based on the coverage estimation in Chapter 6, it is recommended to use SF7 when devices are located close enough to a gateway, approximately below 700 meters in an urban area. SF9 should be sufficient for typical usage. Using higher coding rate also increases the time on air. However, from Chapter 7, most end-devices stick with the default coding rate 4/5 so that the influence of coding rate may not be significant.

From the large-scale collision experiments, the number of downlink transmissions affects the performance because the currently available gateways operate in a half-duplex mode, limiting them from listening to the uplink while transmitting downlink messages. In Chapter 7, traffic from other networks can increase the occupancy of certain LoRaWAN channels, depending on the frequency and spreading factor used. In the KPN dataset, end-devices from other networks predominantly use the three default LoRaWAN frequency channels and SF12, while KPN end-devices distribute the traffic uniformly on the eight allocated channels.

The main contribution of this thesis is the evaluation of LoRaWAN frame collision under multiple gateways environment through practical measurements, which are then compared with simulation results. This answers the main research question as well as the sub-questions regarding the significance of the capture effect and the addition of gateways to the performance. By deploying multiple gateways, the Data Extraction Rate (DER), which is related to packet loss, is expected to be improved. In contrast to [5] in which DER is used in simulations, we propose a method to estimate DER from the large-scale measurement dataset. A simulator has been developed in R using the same setup as in the experiment to validate the results. The simulator uses different collision models: conservative, Capture Threshold Model (CTM), and Additive Interference Model (AIM). The results show that the optimistic CTM model provides a more comparable result to the experiment. Combining the simulation and experiment results, adding more gateways does improve DER due to the capture effect, specifically by 1% to 5% for 140 devices sending packets approximately every 20 seconds. To some extent, adding more gateways may not improve DER even further, especially when end-devices are closely located in the same area, and some of the devices send packets using lower transmission powers.

In Chapter 6 and Chapter 7, we have explored several aspects that can be extracted and evaluated from the real-world TTN and KPN LoRaWAN networks. Traffic characteristics, coverage estimation, gateways and end-devices spatial distributions, and LoRa parameters utilization have been evaluated to obtain the differences between crowdsourced and commercial LoRaWAN networks. Most of the KPN and TTN gateways are densely distributed in the urban area in which the coverage is smaller compared to the rural area due to obstacles, e.g., buildings. From the TTN dataset, we have estimated that the LoRaWAN coverage can reach up to 2.9 km in Utrecht when SF12 is used, while it can achieve 19.6 km in rural areas like Emmeloord. The KPN network provides more diverse channel utilizations compared to TTN. As most end-devices that registered to TTN and other networks use default LoRaWAN channels, the overall performance of the KPN network is not significantly affected by the inter-network interference. Based on the channel utilization, the current use of TTN and KPN LoRaWAN networks can be considered low, i.e., only 8% per channel. Although our evaluation does not explicitly address the collision aspect due to the obscurity and many unknown variables in the real-world networks, we have provided an overview of the currently deployed LoRaWAN networks and possible topics that can be studied further from the dataset.

8-2 Contributions

To the best of our knowledge, this thesis delivers several novel contributions:

- Intensive collision measurements with respect to the application layer. We have evaluated Packet Delivery Ratio (PDR) and signal quality (RSSI and SNR) during collisions under line-of-sight (LOS) and non-line-of-sight (NLOS) environments. The whole packet duration has been partitioned based on the LoRaWAN frame structure to investigate under which circumstances collisions lead to frame loss. We have classified the corrupted frames based on the device IDs and investigated the Message Integrity Check (MIC).
- Frame collision measurements in a multiple gateways environment. We have investigated the influence of adding more gateways to the Data Extraction Rate (DER) through practical measurements. The results have been validated and compared with simulations developed using R. Also, we have extended the small-scale collision measurements to investigate the effect of using different network providers.

- Evaluation of downlink traffic. We have observed through practical measurements that the number of downlink messages, particularly related to join-accept messages, significantly affect the performance of a specific gateway, i.e., the best gateway selected for transmitting downlink.
- Characterization of traffic in real-world, large-scale TTN and KPN LoRaWAN networks.
- Evaluation of inter-network interference from the KPN large-scale measurement dataset.

8-3 Future Work

The small-scale and large-scale frame collision experiments are still limited to the setup in which end-devices are closely located each other. It will be interesting to investigate the spatial distributions of end-devices to the performance, starting from developing a simple model and then followed by simulation and practical measurement in a controlled environment. Also, the experiments involving various power levels in LOS and NLOS scenario can be improved. In the physical layer, the effect of interference from coexisting technologies, the presence of out-of-band interference, and the uplink-downlink collision should be studied and validated through practical measurements. Also, the impact of different sync words for different LoRaWAN networks to the packet loss ratio can be investigated. Theoretically, the receiver will unlock the signal reception whenever it detects an unintended sync word, thereby improving the reception of other packets, especially for the packet that has stronger signal strength.

The real-world LoRaWAN dataset can be explored even further, for example, estimating the mobility patterns and comparing its characteristics with the cellular mobile networks. A solid and generic framework needs to be developed for this purpose. In this thesis, we have obtained and collected several parameters that can be used to develop a more realistic LoRaWAN simulator. Due to the time constraints, we did not develop analytical models and simulations for real-world LoRaWAN networks. Also, only the frame collision aspect was analyzed. It opens up the possibility to study the scalability and capacity aspects through analytical modeling, simulation, and experimentation.

Our study is limited and tied to the ALOHA-like LoRaWAN protocol. The feasibility of a collision-aware protocol, such as Carrier Sense Multiple Access (CSMA) or Listen Before Talk in combination with Adaptive Frequency Agility (LBT-AFA), can be evaluated by utilizing the Carrier Activity Detection (CAD) feature in LoRa devices. One of the possible evaluations is regarding the trade-off between energy consumption and reliability.

As of this writing, LoRaWAN 1.1 specification was released at the end of October 2017. The specification includes support for LoRaWAN Class B, security improvement, new MAC command, and roaming capabilities. It will be interesting to evaluate this specification, particularly for the roaming feature, as it involves coordination between network operators and improves manageability of end-devices belonging to other networks.

Bibliography

[1] LoRa Alliance. *A technical overview of LoRa and LoRaWAN*. Tech. rep. November. 2015, pp. 1–20.

[2] The Things Network. *TTN Wiki: The Things Network Backend*. 2016. URL: https://www.thethingsnetwork.org/wiki/Backend/Home (visited on 04/06/2017).

[3] Orne Brocaar. *LoRa Server Documentation*. 2017. URL: https://docs.loraserver.io/loraserver/ (visited on 04/06/2017).

[4] Congduc Pham. "Building Low-Cost Gateways and Devices for Open LoRa IoT Test-Beds". In: *Testbeds and Research Infrastructures for the*. Vol. 2006. 2017, pp. 70–80. ISBN: 1424401062 (ISBN); 9781424401062 (ISBN). DOI: 10.1007/978-3-319-49580-4_7.

[5] Thiemo Voigt et al. *Mitigating Inter-network Interference in LoRa Networks*. Nov. 2016. arXiv: arXiv:1611.00688v1.

[6] Martin Bor et al. "Do LoRa Low-Power Wide-Area Networks Scale?" In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2016, pp. 59–67. ISBN: 978-1-4503-4502-6. DOI: 10.1145/2988287.2989163.

[7] K Whitehouse et al. "Exploiting the Capture Effect for Collision Detection and Recovery". In: *The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II.* 2005, pp. 45–52. DOI: 10.1109/EMNETS.2005.1469098.

[8] K. Leentvaar and J. Flint, K Leentvaar, and J Flint. "The Capture Effect in FM Receivers". In: *IEEE Transactions on Communications* 24.5 (1976), pp. 653–658. ISSN: 0090-6778. DOI: 10.1109/TCOM.1976.1093327.

[9] Norbert Blenn and Fernando Kuipers. *LoRaWAN in the Wild: Measurements from The Things Network*. 2017. arXiv: 1706.03086. URL: http://arxiv.org/abs/1706.03086.

[10] Semtech. *LoRa Modulation Basics*. Tech. rep. May. 2015, pp. 1–26.

[11] Semtech. *SX1301 datasheet*. Tech. rep. May. 2017, pp. 1–40.

[12] Semtech. *LoRa Modem Low Energy Consumption Design*. Tech. rep. July. 2013, pp. 1–11.

[13] Semtech. *LoRa Modem Design Guide*. Tech. rep. July. 2013, pp. 1–9.

[14] Matt Knight. *Reversing LoRa*. 2016. URL: https://github.com/matt-knight/research (visited on 02/15/2017).

[15] RevSpace. *Decoding LoRa*. 2016. URL: https://revspace.nl/DecodingLora (visited on 05/05/2017).

[16] Link Labs. *What is LoRa?* 2015. URL: https://www.link-labs.com/blog/what-is-lora (visited on 05/05/2017).

[17] Aloÿs Augustin et al. "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things". In: *Sensors* 16.9 (2016), p. 1466. ISSN: 1424-8220. DOI: 10.3390/s16091466.

[18] Semtech. *SX1276 datasheet*. Tech. rep. August. 2016.

[19] Semtech. *SX1272 datasheet*. Tech. rep. March. 2015.

[20] N. Sornin et al. *LoRaWAN Specification v1.0.2*. Tech. rep. 2015.

[21] Semtech. *Reading channel RSSI during a CAD*. Tech. rep. 2014, pp. 1–12.

[22] Martin Bor, John Vidler, and Utz Roedig. "LoRa for the Internet of Things". In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks* (2016), pp. 361–366.

[23] Orange. *LoRa Device Developer Guide*. Tech. rep. 2016, p. 42. URL: https://partner.orange.com/wp-content/uploads/2016/04/LoRa-Device-Developer-Guide-Orange.pdf.

[24] LoRa Alliance. *LoRaWAN Regional Parameters*. Tech. rep. 2016.

[25] Semtech. *ETSI Compliance of the SX1272/3 LoRa Modem*. Tech. rep. July. 2013, pp. 1–25.

[26] The Things Network. *LoRaWAN Wiki*. 2017. URL: https://www.thethingsnetwork.org/wiki/LoRaWAN/Home (visited on 05/07/2017).

[27] Semtech. *LoRa FAQ*. 2016. URL: http://www.semtech.com/wireless-rf/lora/LoRa-FAQs.pdf (visited on 04/07/2017).

[28] The Things Network. *Adaptive Data Rate*. 2017. URL: https://www.thethingsnetwork.org/wiki/LoRaWAN/ADR (visited on 05/09/2017).

[29] Zhuocheng Li, Steeve Zozor, and Jean-marc Brossier. "2D Time-frequency interference modelling using stochastic geometry for performance evaluation in Low-Power Wide-Area Networks". In: *IEEE International Conference on Communications (ICC)*. May. 2017. arXiv: 1606.04791v2.

[30] Orestis Georgiou and Usman Raza. "Low Power Wide Area Network Analysis: Can LoRa Scale?" In: *IEEE Wireless Communications Letters* (2017), pp. 1–4. ISSN: 2162-2337. DOI: 10.1109/LWC.2016.2647247. arXiv: arXiv:1610.04793v1.

[31] Brecht Reynders, Wannes Meert, and Sofie Pollin. "Range and coexistence analysis of long range unlicensed communication". In: *2016 23rd International Conference on Telecommunications, ICT 2016* c (2016). DOI: 10.1109/ICT.2016.7500415.

[32] B Vejlgaard et al. “Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks”. In: *Ieee Wireless* (2017).

[33] D Bankov, E Khorov, and A Lyakhov. “On the Limits of LoRaWAN Channel Access”. In: *2016 International Conference on Engineering and Telecommunication (EnT)*. Nov. 2016, pp. 10–14. DOI: 10.1109/EnT.2016.011.

[34] Juha Petäjäjärvi et al. “Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage”. In: *International Journal of Distributed Sensor Networks* 13.3 (2017), p. 155014771769941. ISSN: 1550-1477. DOI: 10.1177/1550147717699412.

[35] Konstantin Mikhaylov, Juha Petäjäjärvi, and Janne Janhunen. *On LoRaWAN Scalability: Empirical Evaluation of Susceptibility to Inter-Network Interference*. 2017.

[36] Ferran Adelantado et al. *Understanding the limits of LoRaWAN*. July 2016. arXiv: arXiv:1607.08011v1.

[37] Juha Petäjäjärvi et al. “Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring”. In: *International Symposium on Medical Information and Communication Technology, ISMICT*. Vol. 2016-June. 2016. ISBN: 9781509028498. DOI: 10.1109/ISMICT.2016.7498898.

[38] Juha Petäjäjärvi et al. “On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology”. In: *2015 14th International Conference on ITS Telecommunications, ITST 2015*. 2016, pp. 55–59. ISBN: 9781467393829. DOI: 10.1109/ITST.2015.7377400.

[39] Konstantin Mikhaylov, Juha Petäjäjärvi, and Tuomo Hänninen. “Analysis of the Capacity and Scalability of the LoRa Wide Area Network Technology”. In: *European Wireless 2016;* (2016). URL: http://ieeexplore.ieee.org/abstract/document/7499263/.

[40] Oana Iova et al. “LoRa from the City to the Mountains : Exploration of Hardware and Environmental Factors”. In: *International Conference on Embedded Wireless Systems and Networks (EWSN) 2017*. 2017, pp. 317–322. ISBN: 9780994988614.

[41] Alexandru-Ioan Pop et al. *Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?* arXiv: 1704.04174. URL: https://arxiv.org/pdf/1704.04174.pdf.

[42] Peter R. Egli. *Low Power Wide Area Network*. 2015. URL: http://www.indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf.

[43] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[44] Navid Nikaein et al. “Simple Traffic Modeling Framework for Machine Type Communication”. In: (2013), pp. 783–787.

[45] Semtech. *Recommended SX1276 Settings for EU868 LoRaWAN Network Operation*. Tech. rep. January. 2015, pp. 1–16.

[46] C Goursad and J M Gorce. “Dedicated networks for IoT: PHY/MAC state of the art and challenges”. In: *EAI Endorsed Transactions on the Internet of Things* 1.1 (2015), pp. 1–12.

[47] Behnam Dezfouli et al. “CAMA: Efficient Modeling of the Capture Effect for Low-Power Wireless Networks”. In: *ACM Trans. Sen. Netw.* 11.1 (Aug. 2014), 20:1–20:43. ISSN: 1550-4859. DOI: 10.1145/2629352. URL: http://doi.acm.org/10.1145/2629352.

[48] A. Iyer, C. Rosenberg, and A. Karnik. “What is the right model for wireless channel interference?” In: *IEEE Transactions on Wireless Communications* 8.5 (2009), pp. 2662–2671. ISSN: 1536-1276. DOI: 10.1109/TWC.2009.080720.

[49] Ritesh Maheshwari, Shweta Jain, and Samir R. Das. “On estimating joint interference for concurrent packet transmissions in low power wireless networks”. In: *Proceedings of the third ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WiNTECH ’08* (2008), p. 89. DOI: 10.1145/1410077.1410094. URL: http://portal.acm.org/citation.cfm?doid=1410077.1410094.

[50] Maarten Westenberg. *ESP8266 Single Channel LoRaWAN Gateway v4.0.2*. 2017. URL: https://github.com/things4u/ESP-1ch-Gateway-v4.0 (visited on 04/30/2017).

[51] The Things Network. *TTN Fair Access Policy*. 2017. URL: https://www.thethingsnetwork.org/forum/t/limitations-data-rate-packet-size-30-seconds-uplink-and-10-messages-downlink-per-day-fair-access-policy/1300.

[52] KPN. *KPN LoRa Developer Portal*. 2016. URL: https://zakelijkforum.kpn.com/lora-forum-16/kpn-lora-developer-portal-8429 (visited on 06/22/2017).

[53] IoT Acedemy. *Node-RED KPN LoRa*. 2017. URL: https://github.com/iotacademy/NodeRed_KPN_LoRa (visited on 06/22/2017).

[54] Jetmir Haxhibeqiri et al. “LoRa Scalability: A Simulation Model Based on Interference Measurements”. In: *Sensors* 17.6 (2017), p. 1193. ISSN: 1424-8220. DOI: 10.3390/s17061193. URL: http://www.mdpi.com/1424-8220/17/6/1193.

[55] Andrea Munari, Federico Clazzer, and Gianluigi Liva. “Multi-receiver Aloha systems - a survey and new results”. In: *2015 IEEE International Conference on Communication Workshop (ICCW)* Jun. (2015), pp. 2108–2114. DOI: 10.1109/ICCW.2015.7247493. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7247493.

[56] Statistics Netherlands. *NETHERLANDS : Provinces and Major Urban Areas*. 2016. URL: https://www.citypopulation.de/Netherlands-UA.html (visited on 05/07/2017).

[57] Semtech. *LoRaMote User Guide*. Tech. rep. July. 2014, pp. 1–23. URL: http://www.semtech.com/images/datasheet/User_Guide_LoRaMote_STD.pdf.

[58] The Things Network. *Address Space in LoRaWAN*. URL: https://www.thethingsnetwork.org/wiki/LoRaWAN/Address-Space (visited on 09/29/2017).

Appendix A

Frame Collision Experiment Results

A-1 Single Gateways

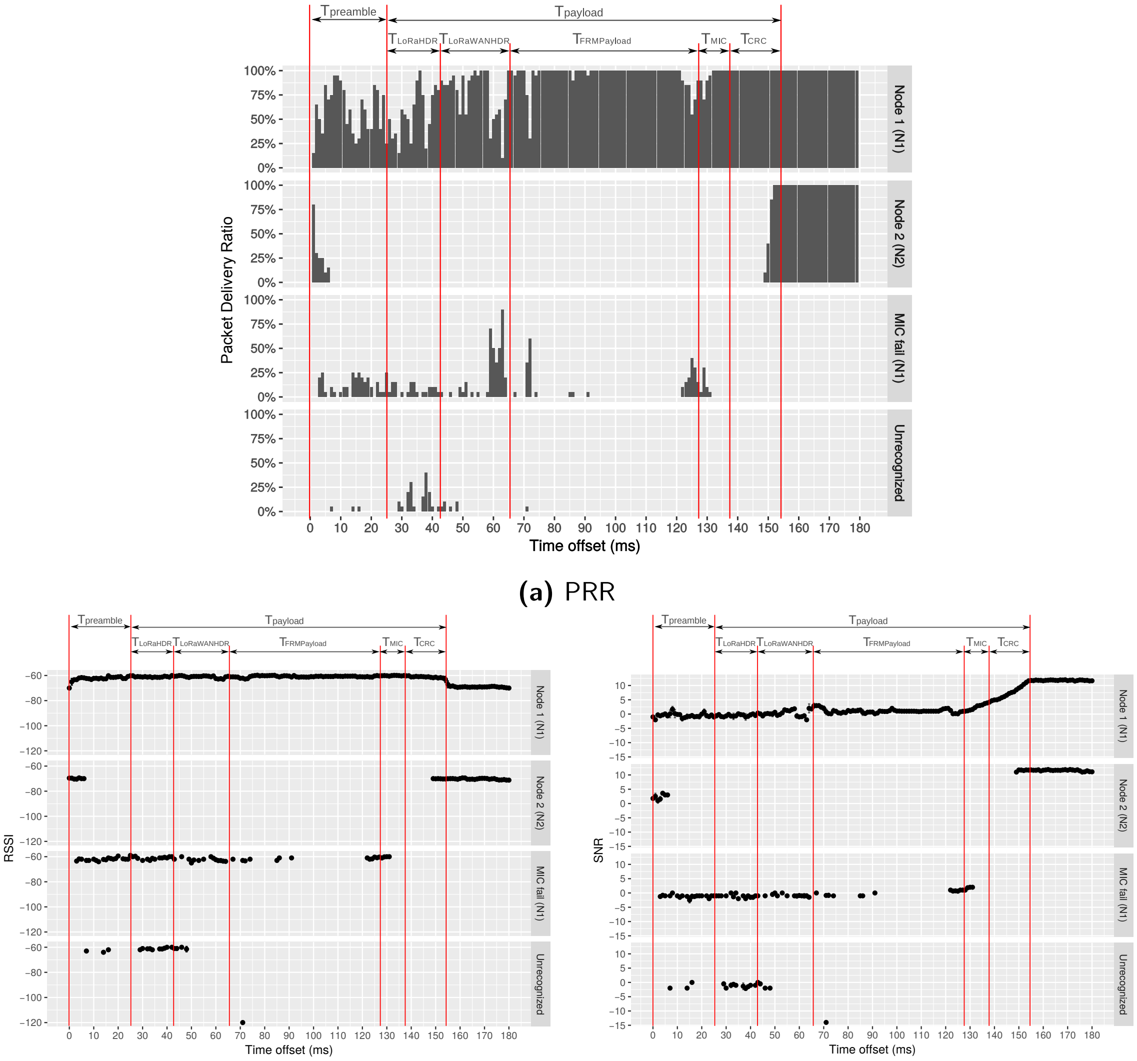
A-1-1 LoRaWAN Frame Timing Decomposition

In this section, we will describe the approach taken to decompose and obtain the timing information from a LoRaWAN frame. The first step is to calculate the symbol time (T_S), preamble time (T_{preamble}), payload time (T_{payload}), and packet time (T_{packet}) using a set of equations described in Chapter 3, Section 3-3-3. It should be noted that the payload consists of LoRa header, LoRa header CRC, PHYPayload, and PHYPayload CRC. The PHYPayload incorporates LoRaWAN header and footer (MIC) as well as the encrypted application payload (FRMPayload). A typical LoRaWAN header has 13 bytes, including 4 bytes MIC. As the number of bytes are known, the number of symbols required for transmitting LoRaWAN header and MIC can be estimated using Equation 3-10. According to the LoRa reverse engineering in [15], the length of LoRa header is approximately 8.4 symbols or 20 bits encoded with $CR = 4/8$. We assume that the header length is fixed for all configurations as such information could not be found in the LoRa documentations. The header time (T_{LoRaHDR}) can be calculated by multiplying the header length with the corresponding T_S. Table A-1 shows the estimated timing information for all scenarios.

Table A-1: Timing information of LoRaWAN frames represented in milliseconds

Scenario	SF	$\mathbf{T_{sym}}$	$\mathbf{T_{preamble}}$	$\mathbf{T_{payload}}$	$\mathbf{T_{packet}}$	$\mathbf{T_{LoRaHDR}}$	$\mathbf{T_{LoRaWANHDR}}$	$\mathbf{T_{FRMPayload}}$	$\mathbf{T_{MIC}}$	$\mathbf{T_{payloadCRC}}$
Equal P_{RX}	8	2.048	25.088	129.024	154.112	17.2032	24.576	70.8608	10.24	6.144
Different P_{TX}	7	1.024	12.544	69.632	82.176	8.6016	13.312	38.5024	6.144	3.072
	8	2.048	25.088	129.024	154.112	17.2032	24.576	70.8608	10.24	6.144
	9	4.096	50.176	217.088	267.264	34.4064	40.96	108.9536	20.48	12.288
	10	8.192	100.352	393.216	493.568	68.8128	73.728	201.5232	32.768	16.384
	11	16.384	200.704	868.352	1069.056	137.6256	163.84	435.8144	81.92	49.152
	12	32.768	401.408	1572.864	1974.272	275.2512	294.912	806.0928	131.072	65.536

A-1-2 Equal Received Power



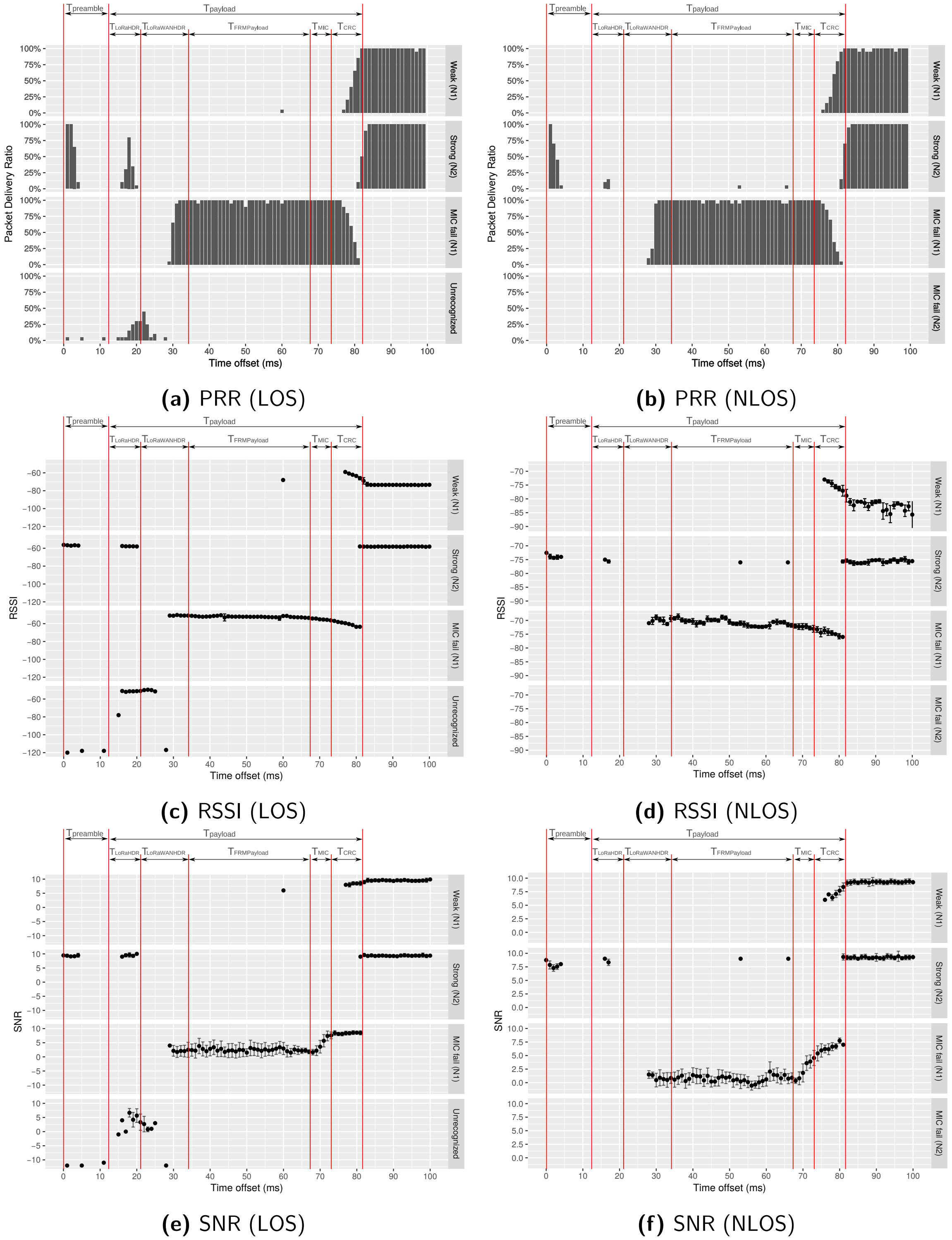
(a) PRR

(b) RSSI

(c) SNR

Figure A-1: PRR, RSSI, and SNR for the first scenario using SF8BW125

A-1-3 Different Transmission Powers and SFs



(a) PRR (LOS)

(b) PRR (NLOS)

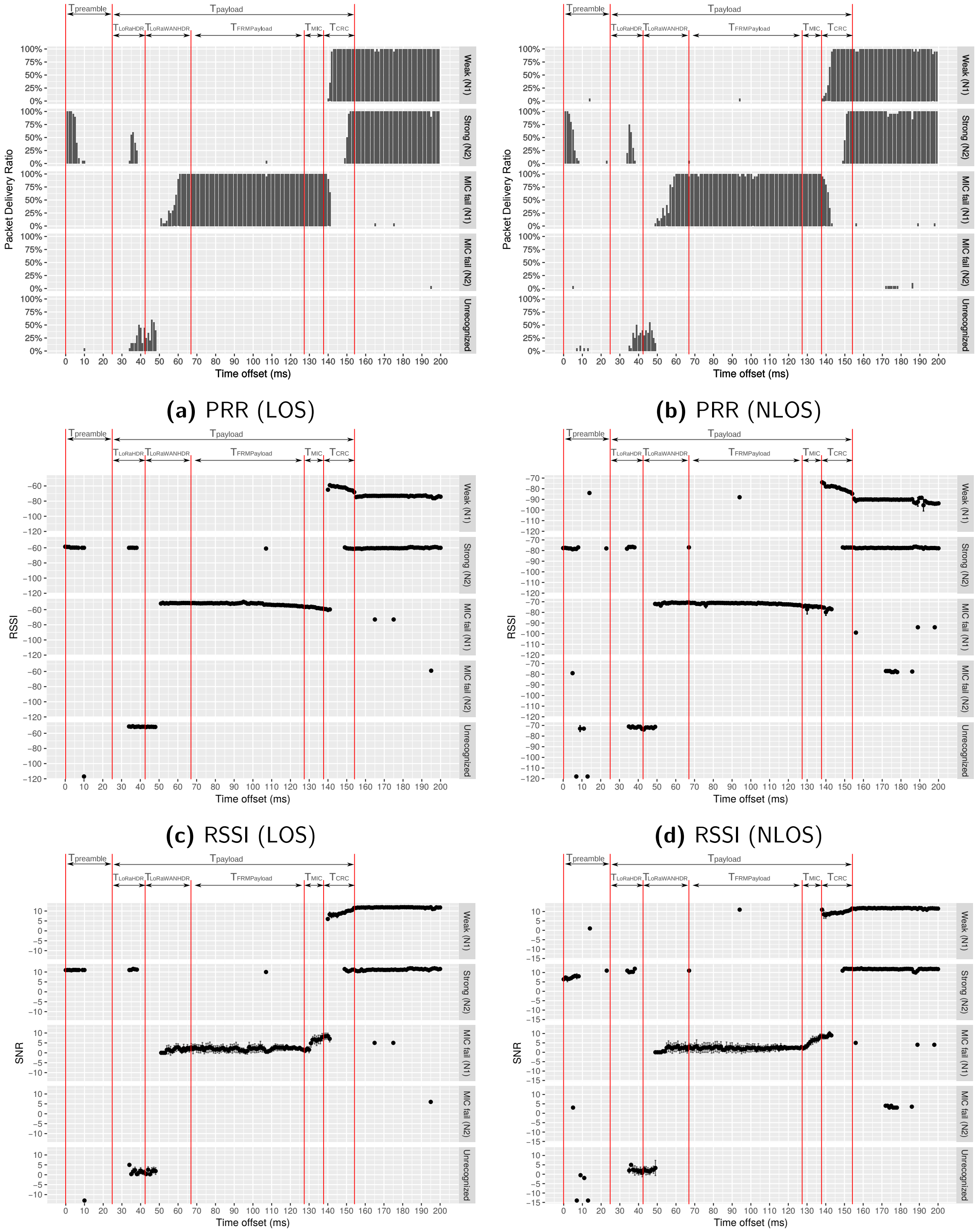
(c) RSSI (LOS)

(d) RSSI (NLOS)

(e) SNR (LOS)

(f) SNR (NLOS)

Figure A-2: PRR, RSSI, and SNR for the second scenario using SF7BW125



(a) PRR (LOS)

(b) PRR (NLOS)

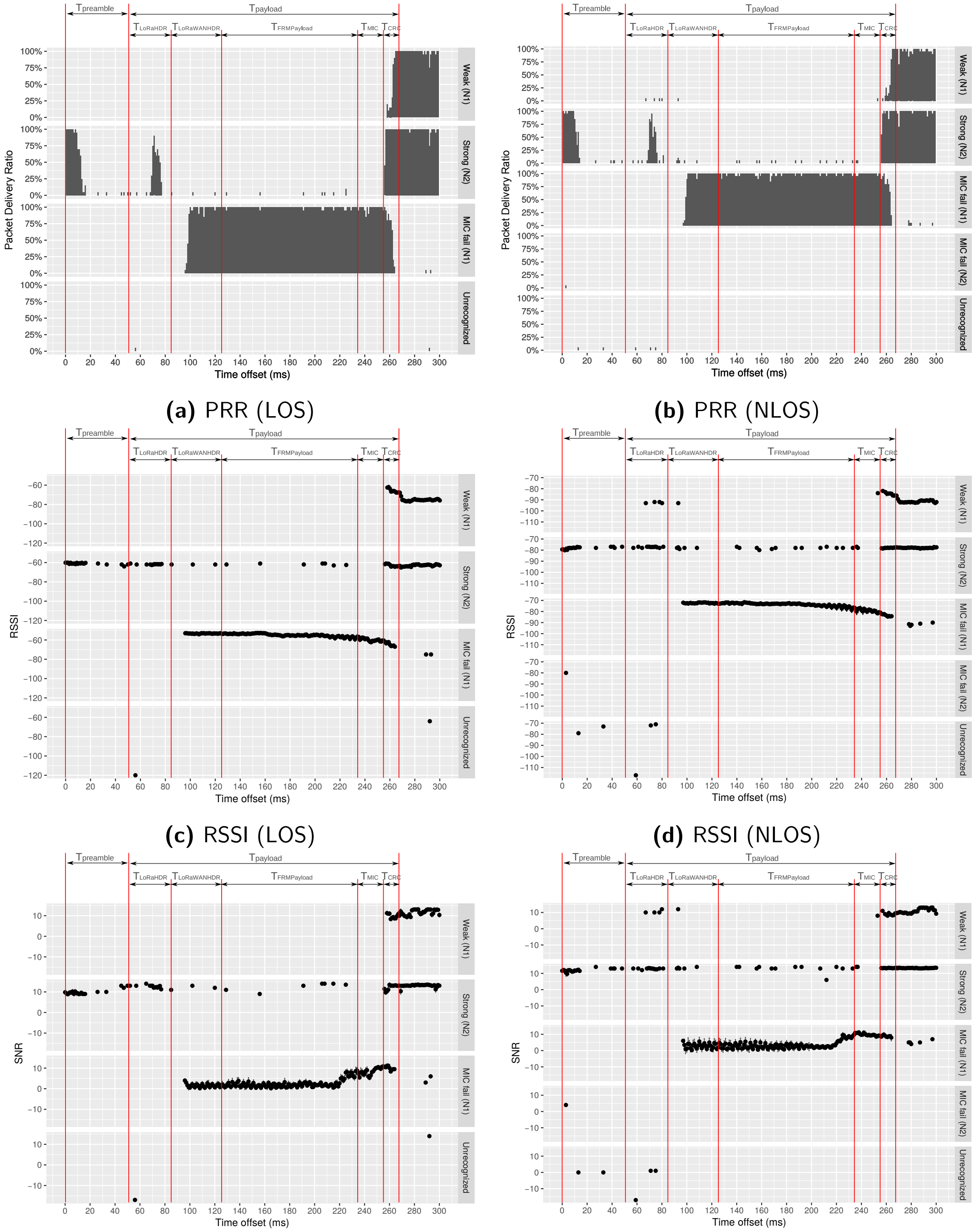
(c) RSSI (LOS)

(d) RSSI (NLOS)

(e) SNR (LOS)

(f) SNR (NLOS)

Figure A-3: PRR, RSSI, and SNR for the second scenario using SF8BW125



(a) PRR (LOS)

(b) PRR (NLOS)

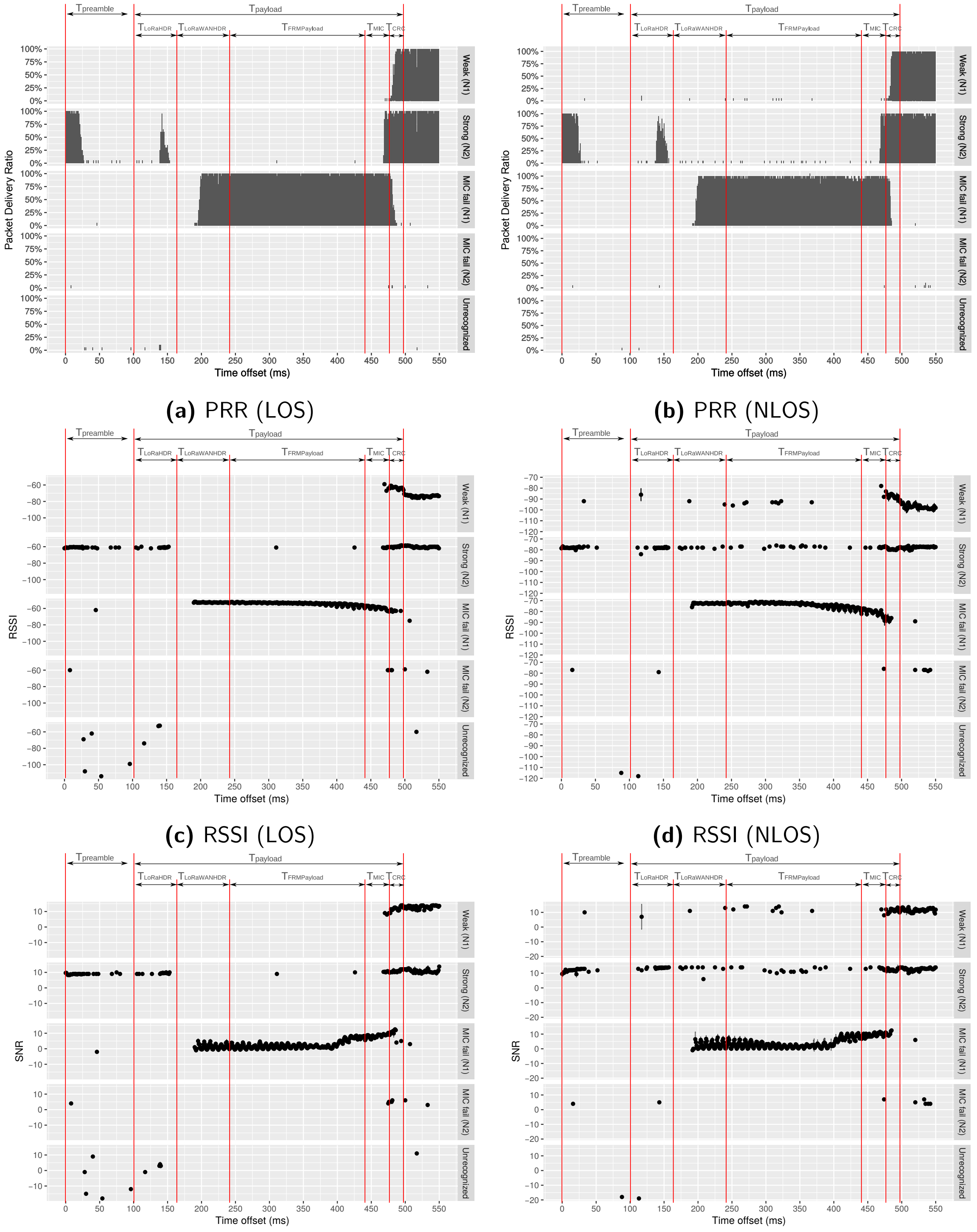
(c) RSSI (LOS)

(d) RSSI (NLOS)

(e) SNR (LOS)

(f) SNR (NLOS)

Figure A-4: PRR, RSSI, and SNR for the second scenario using SF9BW125



(a) PRR (LOS)

(b) PRR (NLOS)

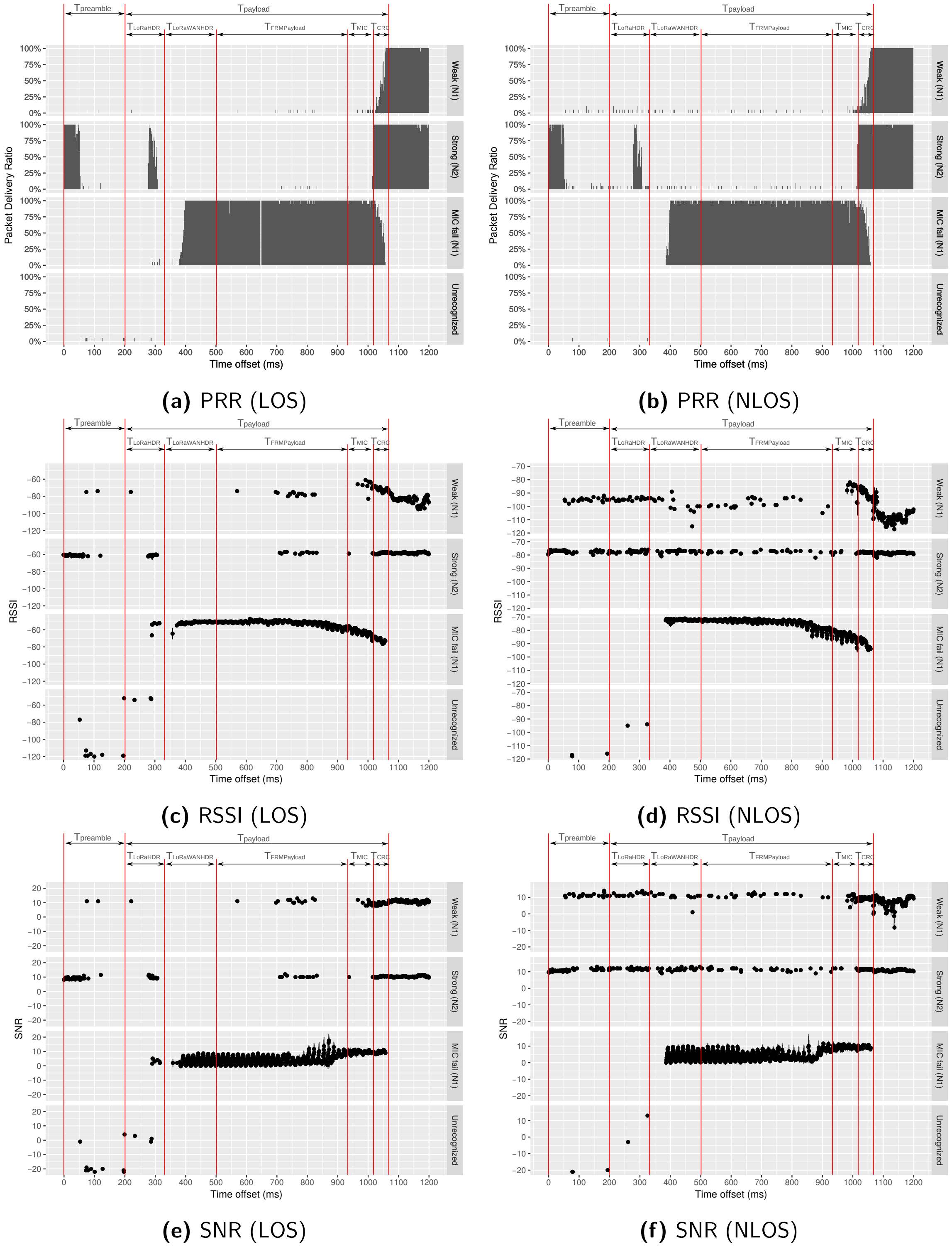
(c) RSSI (LOS)

(d) RSSI (NLOS)

(e) SNR (LOS)

(f) SNR (NLOS)

Figure A-5: PRR, RSSI, and SNR for the second scenario using SF10BW125



(a) PRR (LOS)

(b) PRR (NLOS)

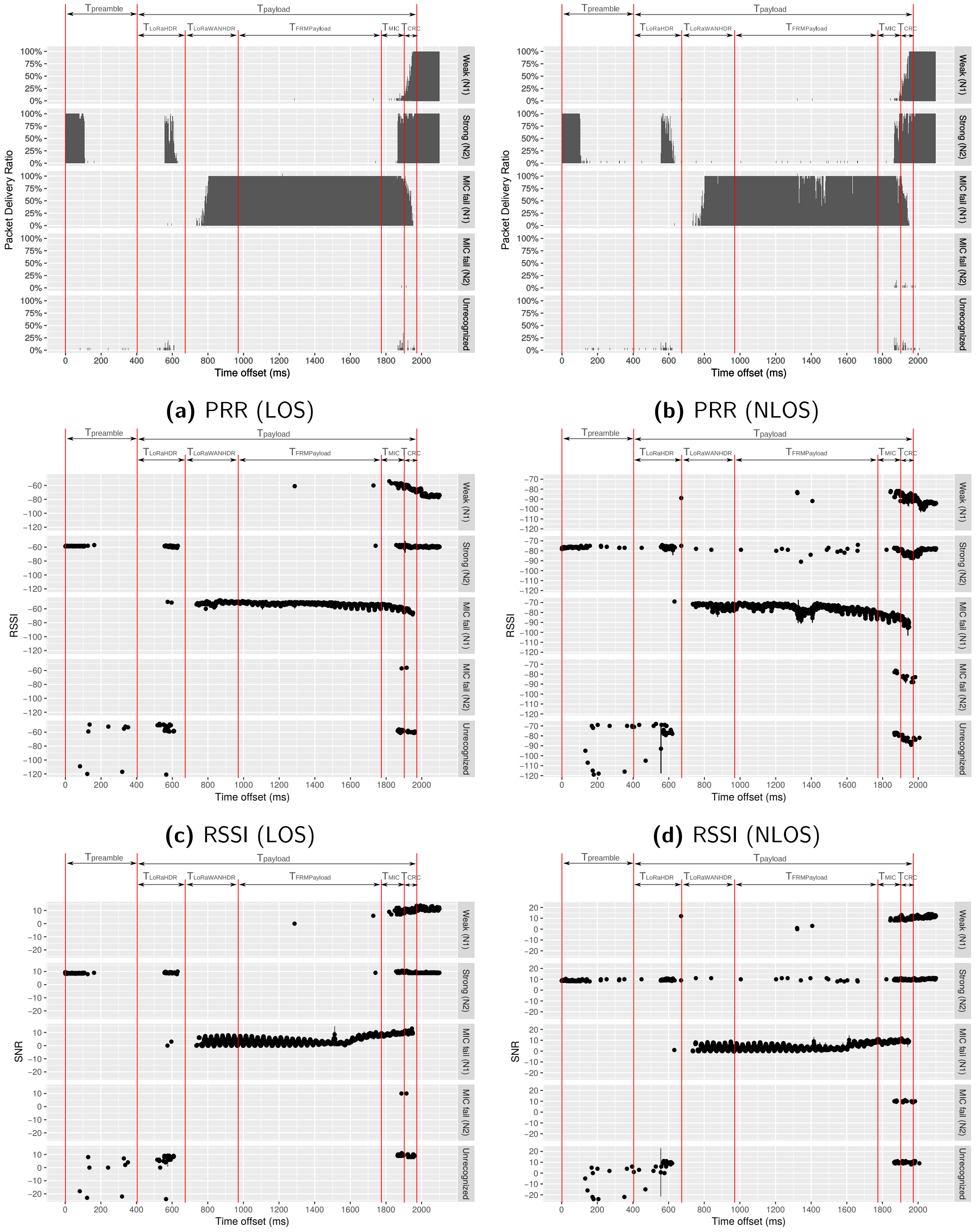
(c) RSSI (LOS)

(d) RSSI (NLOS)

(e) SNR (LOS)

(f) SNR (NLOS)

Figure A-6: PRR, RSSI, and SNR for the second scenario using SF11BW125



(a) PRR (LOS)

(b) PRR (NLOS)

(c) RSSI (LOS)

(d) RSSI (NLOS)

(e) SNR (LOS)

(f) SNR (NLOS)

Figure A-7: PRR, RSSI, and SNR for the second scenario using SF12BW125

A-2 Multiple Gateways

A-2-1 Same Network

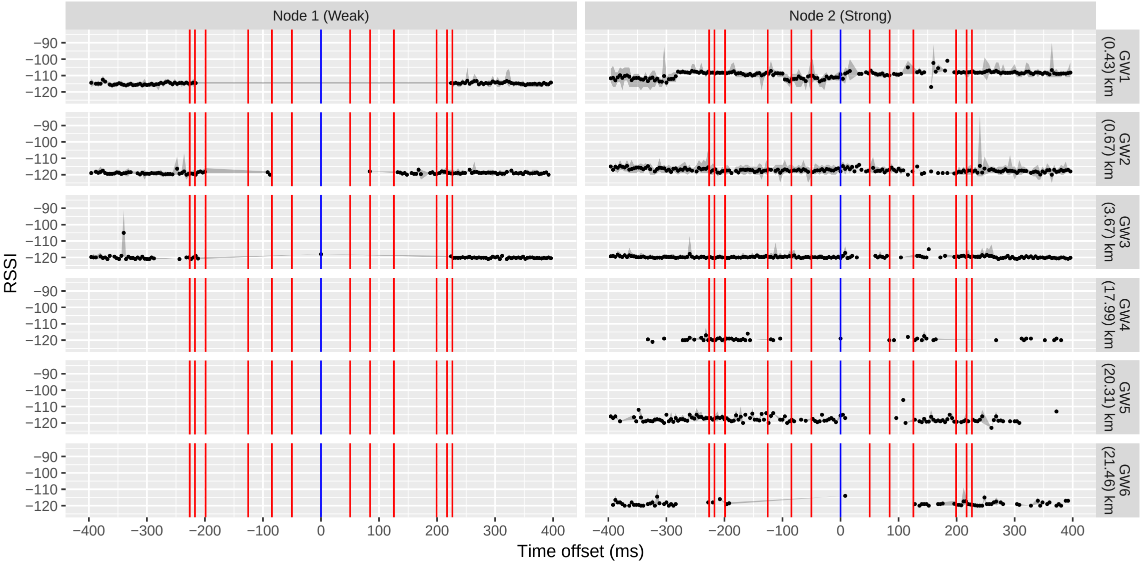


Figure A-8: RSSI received by multiple gateways for the experiment using the same network

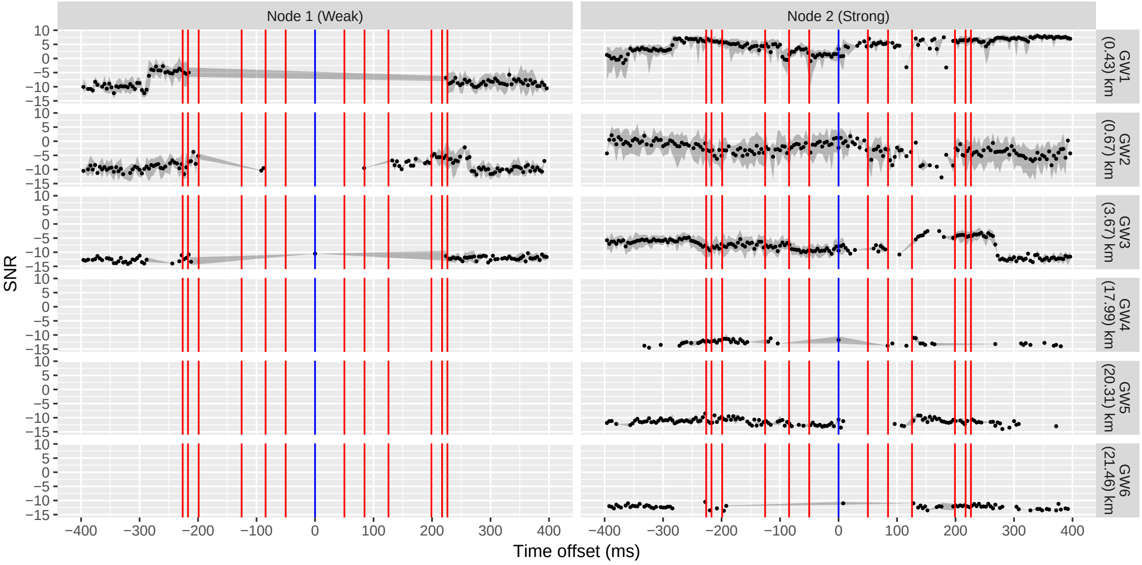


Figure A-9: SNR received by multiple gateways for the experiment using the same network

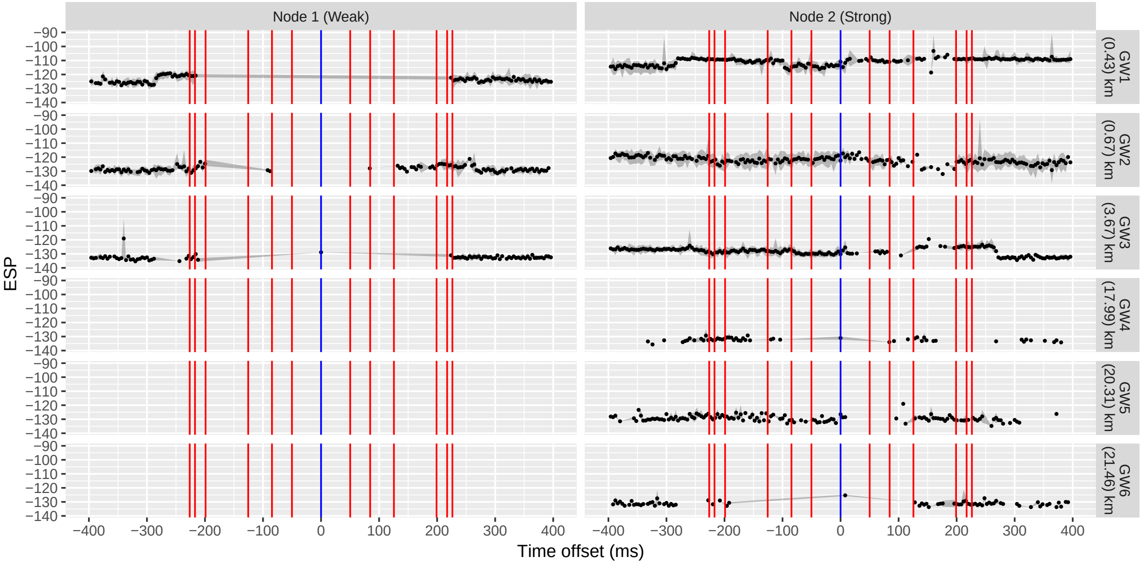


Figure A-10: ESP received by multiple gateways for the experiment using the same network

A-2-2 Different Networks

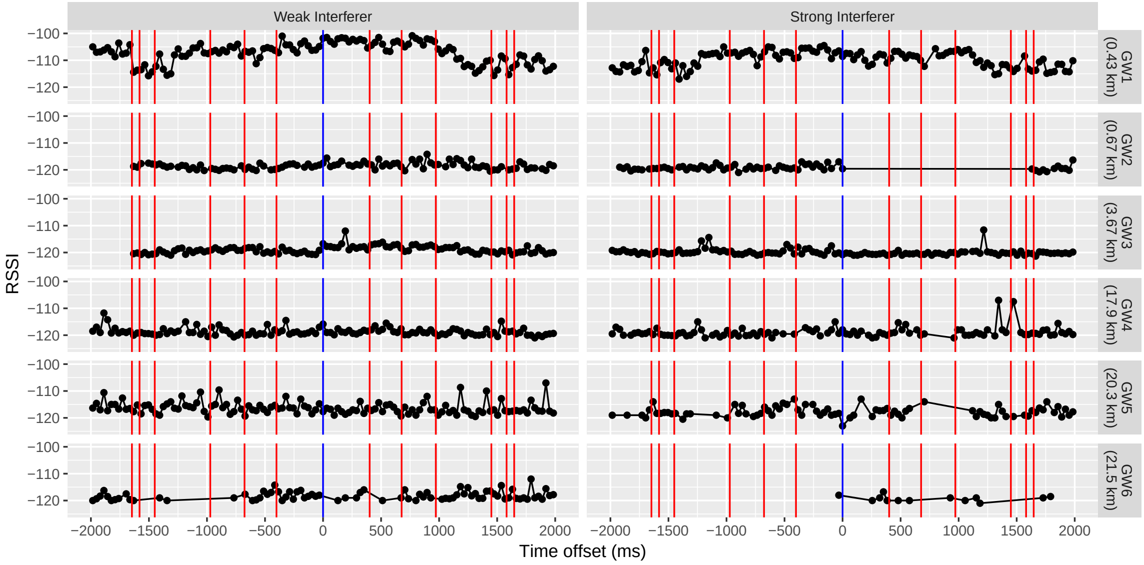


Figure A-11: RSSI received by multiple gateways for the experiment using different networks

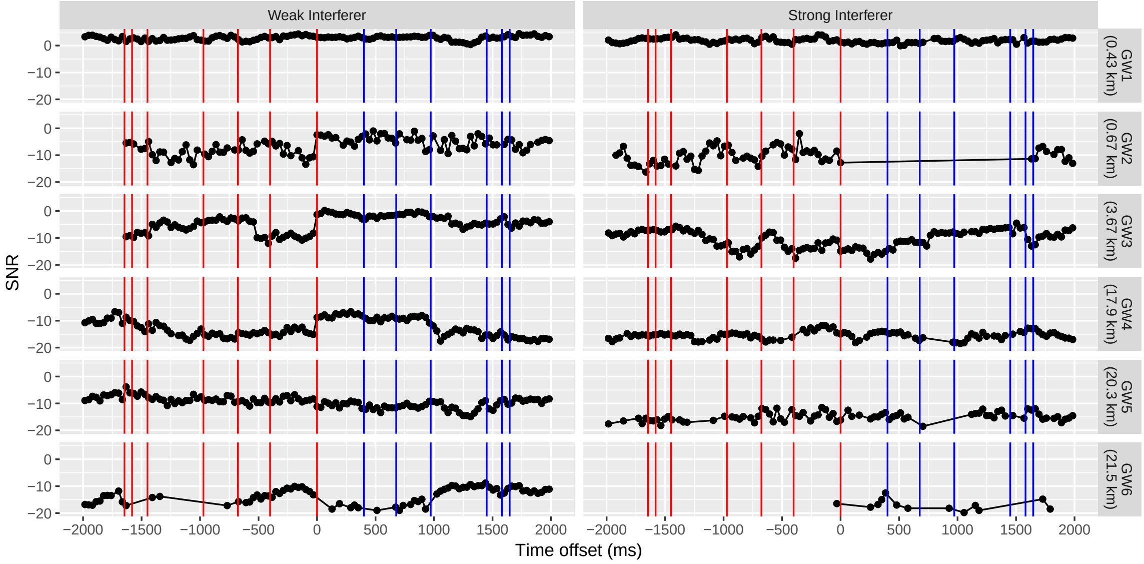


Figure A-12: SNR received by multiple gateways for the experiment using different networks

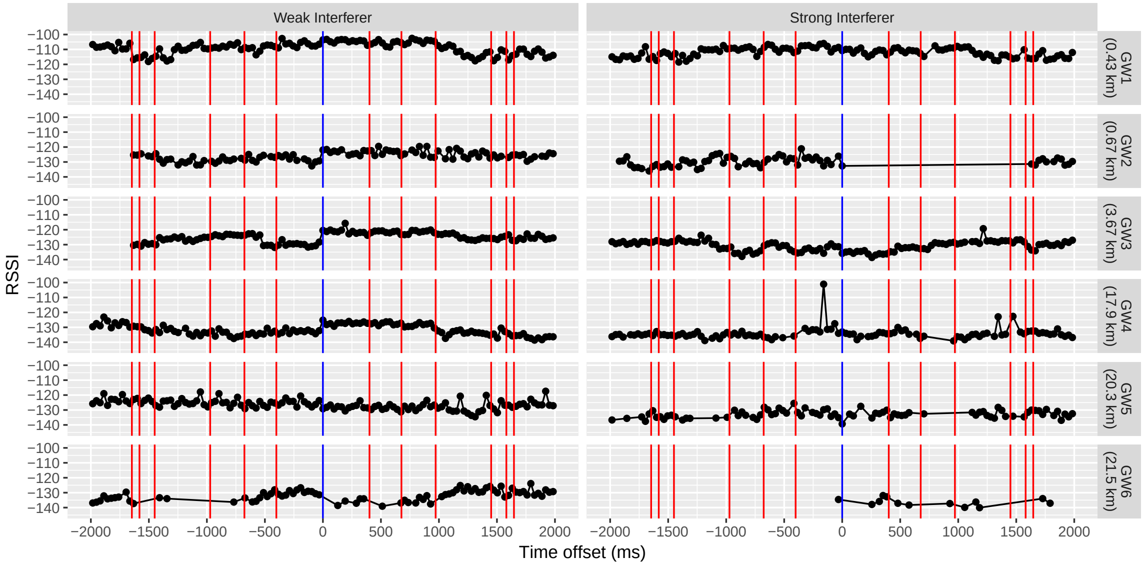


Figure A-13: ESP received by multiple gateways for the experiment using different networks

Appendix B

Algorithms and Source Code

B-1 Traffic Characteristics

Algorithm B-1.1 is used for extracting periodicity of transmission, retransmission, packet loss, counter overflow, and the number of restarted sessions obtained from large-scale measurements dataset.

Algorithm B-1.1 Extract traffic characteristic and packet loss from dataset

```
procedure ExtractTraffic($data\_ordered\_by\_nodeID\_and\_time\_ascending$)
    for $line$ in $data\_ordered\_by\_nodeID\_and\_time\_ascending$ do
        $x_1 \leftarrow$ FCnt previous $line$
        $x_2 \leftarrow$ FCnt current $line$
        $t_1 \leftarrow$ timestamp previous $line$
        $t_2 \leftarrow$ timestamp current $line$
        $nodeID_1 \leftarrow$ node ID previous $line$
        $nodeID_2 \leftarrow$ node ID current $line$
        if $nodeID_2 \neq nodeID_1$ then                 ▷ Start new readings of a different node
            reset results
        else
            if $x_2 > x_1$ then
                if $x_2 \neq x_1 + 1$ then
                    packet loss
                end if
                $period \leftarrow (t_2 - t_1)/(x_2 - x_1)$          ▷ Add to period list
            end if
            if x2 = x1 then
                if $message\_type = confirmed\_message$ then
                    retransmission
                    $period\_retransmission \leftarrow (t_2 - t_1)$ ▷ Add to retransmission period list
                else
                    if $t_2 - t_1 > threshold$ then
                        node restarted
                    end if
                end if
            end if
            if $x_1 = 2^{16} - 1$ or $x_1 = 2^{32} - 1$ then ▷ FCnt length might be 16-bit or 32-bit
                packet FCnt overflow
            else
                node restarted
            end if
        end if
    end for
end procedure
```

B-2 Collision Detection

Listing B.1 is used to detect frame collision obtained from the empirical dataset and simulation results. It checks overlapping transmission time for both forward and backward directions. The algorithm counts the number of overlapping frames that have the same spreading factor, bandwidth, and frequency in time. It also marks the time a new frame starts colliding and the time the previous frame ends colliding with respect to the packet being checked.

Listing B.1: Source code of function for detecting frame collision

```r
checkCollision <- function(data_in) {
  # Rearrange data based on tstart
  d <- data_in %>%
    mutate(id = as.character(paste(gw_id, dev_id, pkt_id, sep = "-"))) %>%
    select(id, sf, cf, tstart, tend, esp) %>%
    arrange(tstart)

  # Obtain maximum time on air from input data
  max_toa <- data_in %>% summarise(max(toa))

  # Prepare for output
  output <- matrix(nrow = nrow(d), ncol = 14)
  class(output) <- "numeric"

  # Set fixed param
  bw <- 125

  # Iterate per packet
  for(i in 1:nrow(d)){
    tsym_ms <- (2^(d$sf[i]))/bw
    cur_id = d$id[i]

    # Reset all counters
    n_col_cf <- 0
    n_col_sf <- 0
    n_col_all <- 0
    n_col_time <- 0
    packet_loss_coverage <- 0
    packet_loss_no_capture <- 0
    packet_loss_capture_ctm <- 0
    packet_loss_capture_aim <- 0
    col_at_fwd_ms <- NA
    col_at_bck_ms <- NA
    col_at_fwd_symb <- NA
    col_at_bck_symb <- NA

    # Flag for exit loop
    col_fwd_flag <- 0
    col_back_flag <- 0

    # Check for collision in forward direction
    if(i < nrow(d)) {
      for(j in (i+1):nrow(d)) {
        # If tstart of the next packet is less than tend of the current packet (collision)
        if(d$tstart[j] < d$tend[i]) {
          n_col_time <- n_col_time + 1

          # Check if the colliding packets have the same SF and CF
          if((d$cf[i] == d$cf[j]) & (d$sf[i] == d$sf[j])) {
            n_col_all <- n_col_all + 1

            # Mark only the first collision
            if(col_fwd_flag == 0) {
              col_at_fwd_ms <- d$tstart[j]

              # Number of symbol when collision happen w.r.t. tstart
              col_at_fwd_symb <- ceiling((col_at_fwd_ms - d$tstart[i])/(tsym_ms*1000))
              col_fwd_flag <- 1
            }

            # CTM
            if(d$esp[j] > esp_ctm_fwd) {
              esp_ctm_fwd <- d$esp[j]
            }

            # AIM
            esp_aim_fwd <- esp_aim_fwd + 10^(d$esp[j]/10)
          }

          # CF collision
          if(d$cf[i] == d$cf[j]) {
            n_col_cf <- n_col_cf + 1
          }

          # SF collision
          if(d$sf[i] == d$sf[j]) {
            n_col_sf <- n_col_sf + 1
          }
        } else {
          break
        }
      }
    }

    # Check for collision in backward direction
    if(i > 1) {
      for(j in (i-1):1) {
        # If tend of the previous packet is larger than tstart of the current packet (collision)
        if(d$tend[j] > d$tstart[i]) {
          n_col_time <- n_col_time + 1

          # Check if the colliding packets have the same SF and CF
          if((d$cf[i] == d$cf[j]) & (d$sf[i] == d$sf[j])) {
            n_col_all <- n_col_all + 1

            # Mark only the first collision
            if(col_back_flag == 0) {
              col_at_bck_ms <- d$tend[j]

              # Number of symbol when collision happen w.r.t. tstart
              col_at_bck_symb <- ceiling((col_at_bck_ms - d$tstart[i])/(tsym_ms*1000))
              col_back_flag <- 1
            }

            # CTM
            if(d$esp[j] > esp_ctm_bck) {
              esp_ctm_bck <- d$esp[j]
            }

            # AIM
            esp_aim_bck <- esp_aim_bck + 10^(d$esp[j]/10)
          }

          # CF collision
          if(d$cf[i] == d$cf[j]) {
            n_col_cf <- n_col_cf + 1
          }

          # SF collision
          if(d$sf[i] == d$sf[j]) {
            n_col_sf <- n_col_sf + 1
          }
        # Break until tstart of other packets < maximum time on air
        } else if(d$tstart[j] < (d$tstart[i] - max_toa)) {
          break
        }
      }
    }

    # Sensitivity (SX1301, 125 kHz)
    if(d$sf[i] == 7) rx_sens <- -126.5
    else if(d$sf[i] == 8) rx_sens <- -129
    else if(d$sf[i] == 9) rx_sens <- -131.5
    else if(d$sf[i] == 10) rx_sens <- -134
    else if(d$sf[i] == 11) rx_sens <- -136.5
    else rx_sens <- -139.5

    # Packet loss due to coverage
    if(d$esp[i] < rx_sens) {
      packet_loss_coverage <- 1
    }

    # Collision (simulation)
    if(n_col_all > 0) {
      packet_loss_no_capture <- 1

      # CTM
      ## Check collisions with previous packets
      if(!is.na(col_at_bck_symb)) {
        ## Interferer is stronger, arrives earlier
        if(d$esp[i] - esp_ctm_bck < -6)
        {
          packet_loss_capture_ctm <- 1
        }
        ## Interferer is weaker, arrives earlier
        else if(d$esp[i] - esp_ctm_fwd > 6)
        {
          # The current preamble frame is overlapped by the previous frames
          if(col_at_bck_symb > 4) {
            # For now, we consider this as packet loss
            packet_loss_capture_ctm <- 1
          }
        }
        ## Interferer is less than 6 dB, arrives earlier
        else
        {
          if(col_at_bck_symb > 4) {
            # For now we consider this as packet loss
            packet_loss_capture_ctm <- 1
          }
        }
      }

```

```r
      # Check collisions with the next packets
      if(!is.na(col_at_fwd_symb)) {
        ## Interferer is stronger, arrives later (delayed)
        if(d$esp[i] - esp_ctm_fwd < -6) {
          # For now, we consider this as packet loss
          packet_loss_capture_ctm <- 1
        }
        ## Interferer is weaker, arrive later (delayed)
        ## No packet loss at all
        else if(d$esp[i] - esp_ctm_fwd > 6) {
        }
        ## Interferer is less than 6 dB, arrive later (delayed)
        else
        {
          # Randomize packet loss (uniform distribution)
          packet_loss_capture_ctm <- round(runif(1, 0, 1))
        }
      }

      # AIM
      if(d$sf[i] == 7) rx_demod <- -7.5
      else if(d$sf[i] == 8) rx_demod <- -10
      else if(d$sf[i] == 9) rx_demod <- -12.5
      else if(d$sf[i] == 10) rx_demod <- -15
      else if(d$sf[i] == 11) rx_demod <- -17.5
      else rx_demod <- -20

      ## Noise floor: -174 + 10 * log10(125000)
      noise_floor <- -123.0309

      # Check collisions with the next packets
      if(esp_aim_fwd != 0) {
        esp_aim_fwd_s <- d$esp[i] - 10*log(esp_aim_fwd + 10^(noise_floor/10), 10)
        # If less than demodulator SNR
        if(esp_aim_fwd_s < rx_demod) {
          packet_loss_capture_aim <- 1
        }
        # Otherwise, check overlapping time
        else
        {
          # Check if we have collision with same
          if(!is.na(col_at_fwd_symb)) {
            if(col_at_fwd_symb > 4) {
              packet_loss_capture_aim <- 1
            }
          }
        }
      }
      # Check with previous packet
      if(esp_aim_bck != 0) {
        esp_aim_bck_s <- d$esp[i] - 10*log(esp_aim_bck + 10^(noise_floor/10), 10)
        # If less than demodulator SNR
        if(esp_aim_bck_s < rx_demod) {
          packet_loss_capture_aim <- 1
        }
        # If not. Check time overlap
        else
        {
          # Check if the colliding packets have the same SF and CF
          if(!is.na(col_at_bck_symb)) {
            if(col_at_bck_symb > 4) {
              packet_loss_capture_aim <- 1
            }
          }
        }
      }
    }

    # Write to matrix
    output[i, 1] <- cur_id
    output[i, 2] <- n_col_cf
    output[i, 3] <- n_col_sf
    output[i, 4] <- n_col_all
    output[i, 5] <- n_col_time
    output[i, 6] <- col_at_fwd_ms
    output[i, 7] <- col_at_fwd_symb
    output[i, 8] <- col_at_bck_ms
    output[i, 9] <- col_at_bck_symb
    output[i, 10] <- packet_loss_no_capture
    output[i, 11] <- packet_loss_capture_ctm
    output[i, 12] <- packet_loss_capture_aim
    output[i, 13] <- packet_loss_coverage
    output[i, 14] <- d$tstart[i]
  }

  # Output to data frame
  data.frame(tstart = output[, 14],
             id = as.character(output[, 1]),
             n_col_cf = as.numeric(output[, 2]),
             n_col_sf = as.numeric(output[, 3]),
             n_col_all = as.numeric(output[, 4]),
             n_col_time = as.numeric(output[, 5]),
             col_at_fwd_ms = as.numeric(output[, 6]),
             col_at_fwd_symb = as.numeric(output[, 7]),
             col_at_bck_ms = as.numeric(output[, 8]),
             col_at_bck_symb = as.numeric(output[, 9]),
             packet_loss_no_capture = as.numeric(output[, 10]),
             packet_loss_capture_ctm = as.numeric(output[, 11]),
             packet_loss_capture_aim = as.numeric(output[, 12]),
             packet_loss_coverage = as.numeric(output[, 13]),
             stringsAsFactors=FALSE)
}
```

B-3 E&A Event Simulation

Listing B.2: Simulation code for the E&A event

```r
# Time on air calculation
toa <- function(SF, BWkhz, CR58, PL){
  if(!is.na(SF) & !is.na(BWkhz) & !is.na(CR58) & !is.na(PL)) {
    BW <- BWkhz
    # Coding rate 4/n, where n from 5 to 8, default 4/5 for LoRaWAN
    CR <- CR58
    # PL, PHY payload size
    npreamble <- 8       # Number of preamble, default 8 for LoRaWAN
    IH <- 1              # Explicit header = 1 for LoRaWAN
    CRC <- 1             # Payload CRC = 1 for LoRaWAN
    DE <- 0
    if((SF == 11 || SF == 12) && (BW == 125)){
      DE <- 1              # Low data rate optimization = 1 for LoRaWAN
    }

    # Calculate Tysmbol
    tsym <- (2^SF)/(BW)

    # Calculate preamble time
    tpreamble <- (npreamble+4.25)*tsym

    # Calculate n symbols of payload
    payloadSymbNb <- 8+max(ceiling((8*PL-4*SF+28+16*CRC-20*(1-IH))/(4*(SF-2*DE)))*CR,0)

    # Calculate Tpayload
    tpayload <- payloadSymbNb * tsym

    # Function to calculate Tpacket (Time on Air)
    tpacket <- tpreamble + tpayload

    tpacket
  } else {
    NA
  }
}

# Generate Packtes
## -- Output data structure
## gw_id | dev_id | pkt_id | sf | freq | tstart | tend | rssi
generatePackets <- function(num_node, period, duration) {
  # Number of gateways
  num_gws <- 5

  # Distance of gateways w.r.t. Jaarbeurs
  dist_gws <- c(26.11, 53.63, 69.26, 4240, 9001)

  # Calculate total number of packets per device
  n_pkt <- round(duration/period)
```

```

# Set a fixed payload, i.e., 23 bytes
py_size <- 23

# Prealocate output matrix
output <- matrix(ncol = 9, nrow = n_pkt * num_node * num_gws)

# Reset all
sf <- NA
cf <- NA
tstart <- NA
tend <- NA
rssi <- NA

# Loop per device
for(i in 1:num_node) {
  # Initial start time for devices (random)
  t_dev_start <- runif(1, min = 0, max = period) * 1000

  # Packets sent per device
  for(j in 1:n_pkt) {
    # Different gateways
    for(k in 1:num_gws) {
      cur_idx = n_pkt*num_gws*(i-1) + num_gws*(j-1) + k
      # GW ID
      output[cur_idx, 1] <- k
      # Node ID
      output[cur_idx, 2] <- i
      # Packet ID
      output[cur_idx, 3] <- j

      # Spreading factor distributions
      if(k == 1) {
        sf_p <- sample(seq(1,64,1),1)
        if(sf_p == 1) {
          sf <- 12
        } else if(sf_p > 1 & sf_p < 4) {
          sf <- 11
        } else if(sf_p > 3 & sf_p < 8) {
          sf <- 10
        } else if(sf_p > 7 & sf_p < 16) {
          sf <- 9
        } else if(sf_p > 15 & sf_p < 32 ) {
          sf <- 8
        } else {
          sf <- 7
        }
      }
      output[cur_idx, 4] <- sf

      # Frequency distributions
      if(k == 1) {
        cf_p <- sample(seq(1,8,1),1)
```

```
          if(cf_p == 1) {
            cf <- 867.1
          } else if(cf_p == 2) {
            cf <- 867.3
          } else if(cf_p == 3) {
            cf <- 867.5
          } else if(cf_p == 4) {
            cf <- 867.7
          } else if(cf_p == 5) {
            cf <- 867.9
          } else if(cf_p == 6) {
            cf <- 868.1
          } else if(cf_p == 7) {
            cf <- 868.3
          } else if(cf_p == 8) {
            cf <- 868.5
          }
        }
        output[cur_idx, 5] <- cf

        # Calculate time on air and tstart
        if(k == 1) {
          t_toa <- toa(SF = sf, BWkhz = 125, CR58 = 5, PL = 23)

          # Calculate tstart (t_dev_start + period + some random num)
          tstart <- t_dev_start + (j * period * 1000) +  runif(1, 0, 5000)
        }
        output[cur_idx, 6] <- tstart

        # Calculate tend
        if(k == 1) {
          tend <- tstart + t_toa
        }
        output[cur_idx, 7] <- tend

        # Calculate RSSI (using simplified path loss model)
        ## -- Outdoor path loss and shadowing parameters from measurements in Utrecht (adjusted)
        d0 <- 100
        lpl_d0 <- 115
        shadow_var <- 7.387089
        pl_exp <- 2.6234

        # Indoor path loss model
        d0_in <- 1
        c <- 3e8
        f <- mean(c(867.1, 867.3, 867.5, 867.7, 867.9, 868.1, 868.3, 868.5)) * 1000000
        lpl_d0_in <- 20*log(4*pi*d0_in*f/c, 10)
        pl_exp_in <- 3
        shadow_var_in <- 9

        # Calculate path loss
        ## Add distance deviation
        dist <- dist_gws[k] + abs(rnorm(1, mean = 0, sd = 30))
        if(dist > 1000) {
          lpl <- lpl_d0 + 10 * pl_exp * log(dist/d0, 10) + rnorm(1, mean
             = 0, sd = shadow_var)
        } else {
          lpl <- lpl_d0_in + 10 * pl_exp_in * log(dist/d0_in, 10) + rnorm
              (1, mean = 0, sd = shadow_var_in)
        }

        # Calculate RSSI
        ## Assumption: GRX = 2 dBi, GTX = 2 dBi, PTX = 14 dBm
        grx <- 2
        gtx <- 2
        ptx <- 14
        rssi <- ptx + grx + gtx - lpl
        output[cur_idx, 8] <- rssi

        # Add time on air to the output
        output[cur_idx, 9] <- t_toa
      }
    }
  }

  # Construct output data frame
  data.frame(
    gw_id = output[,1],
    dev_id = output[,2],
    pkt_id = output[,3],
    sf = output[,4],
    cf = output[,5],
    tstart = output[,6],
    tend = output[,7],
    esp = output[,8],
    toa = output[,9])
}

# Generate packets for 1 to 150 nodes incremented every 10 steps
d_sim_all <- data.frame()
for(i in seq(0, 150, 10)) {
  if(i == 0) i <- 1
  d_sim_all <- bind_rows(d_sim_all,
                         (generatePackets(num_node = i, period = 20,
                            duration = 3600) %>%
                            mutate(n_dev = i))
  )
}

# Save to file
save(d_sim_all, file = "d_sim_all_1_to_150.RData")
```