Delft University of Technology
Master of Science Thesis in Embedded Systems

Designing and implementing SFMAC: A MAC protocol for LoRa networks for efficient use of unlicensed bands

Maria Teresa Blanco Abad

Embedded Networked Systems

TU Delft Delft University of Technology

Designing and implementing SFMAC: A MAC protocol for LoRa networks for efficient use of unlicensed bands

Master of Science Thesis in Embedded Systems

Embedded and Networked Systems Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Maria Teresa Blanco Abad
M.T.BlancoAbad@student.tudelft.nl
teresablancoabad95@gmail.com

November 23th

Author
Maria Teresa Blanco Abad (M.T.BlancoAbad@student.tudelft.nl)
(teresablancoabad95@gmail.com)
Title
Designing and implementing SFMAC:
A MAC protocol for LoRa networks for efficient use of unlicensed bands
MSc Presentation Date
November 27th

Graduation Committee

Dr. Ranga Rao Venkatesha Prasad,	TU Delft, supervisor
Ir. Nikolaos Kouvelas,	TU Delft
Dr. Ir. A. J. Van Genderen,	TU Delft
Dr. Ir. Vijay S. Rao,	Cognizant

Abstract

Long Range Wide Area Networks (LoRaWAN) offer easy deployment, robustness against interference, and operational longevity to energy constrained IoT-devices which communicate in a best-effort fashion in extended ranges. However, the simple (ALOHA-like) design of the MAC layer leads to packet collisions in dense LoRaWAN deployments with high traffic loads. To achieve scalability above a few hundreds of devices, time division is not an option, since LoRaWAN is asynchronous regarding communication. Further, feedback mechanisms are discouraged due to duty cycle limitations. In this document, we propose Spreading Factor MAC (SFMAC); a distributed and energy efficient MAC protocol for LoRaWAN, wherein –for the first time to the best of our knowledge– high-SF channels are dedicated strictly to Channel Sensing (CS), while low-SF channels are focused on data-transmission. The Capture Effect (CE) phenomena that is manifested in the PHY layer is extensively evaluated on-field and embodied in the SFMAC operating principle. The dedicated high-SF sensing allows effective revealing of hidden devices' transmissions without affecting the low-SF traffic.

We showcase the impact of SFMAC in scalability by designing a realistic implementation of the mechanism in *ns-3*. We report a x2.08 improvement in channel utilization and x2 goodput compared to LoRaWAN, without substantially increasing complexity.

Preface

I would like to take this unique opportunity to thank VP, Nikos, Vijay for having had great inspiration and support from them, from my friends and from the great engineering community at TU Delft. Even if the MSc thesis could be a lonely path through papers and hours of trying to build something meaningful and new, it has been a very enriching journey.

I will always be thankful for having received from my family unconditional support when embarking myself in every adventure that seemed challenging and worthy.

Maria Teresa Blanco Abad

Delft, The Netherlands
23th November 2020

Contents

Chapter 1

Introduction

Upon the proliferation of small battery-powered devices in numerous applications, new wireless communication protocols emerged as a solution to enable these low-power devices to communicate. One of these communication protocols designed specifically for IoT devices is LoRa [33]. Despite the IP limitations of LoRa, the PHY layer is patented, it has increasingly gained popularity due to the easy deployment of LoRa networks. Any owner of a LoRa gateway and a LoRa device can start broadcasting and receiving LoRa messages without requiring permits or complicated software development thanks to open source projects (i.e ChirpStack [2]). In addition, LoRa public networks (i.e The Things Network [10]) enable users that do not own a gateway to benefit from a public LoRa infrastructure to setup their IoT devices with fewer complications. This communication technology operates in the unlicensed band of the spectrum, therefore it is not required a detailed planning of networks as occurs, for example, with cellular deployments.

All the mentioned advantages battle with the problem of handling an increasing number of users in LoRa networks. Some marketers of LoRa and LoRaWAN [7], claim the ability of LoRa networks to handle thousands of devices. To serve an increasing number of LoRa devices, it exists limitations in the number of packets that can be sent per unit of time in a LoRa network. By enforcing nodes to have limited traffic through duty cycling policies, more users can access the channel without interfering with the rest of the already existing users. Another strategy to cover more devices in a LoRa network is installing more gateways. Increasing the gateway density would unload gateways with heavy traffic by sharing the users between more receivers. There is an underlying limitation to the increase of gateways. Even when increasing the number of receivers, the traffic depends on the transmissions from IoT devices and the traffic is not scheduled to ensure fair access to the medium.

IoT devices in a LoRa network send their packets whenever the device generates the information to be sent (for example the temperature in a industrial refrigeration system, the fuel level in a tank, or the brightness of an automated greenhouse system) in an asynchronous manner. This asynchronous access to the medium is defined as ALOHA [11], and its theoretical capacity limits have been studied in the literature under different amount of packets transmitted in a network. The research community already proposed solutions to the simple ALOHA access to the medium by, for example, enforcing nodes to listen before transmitting. The previous type of solutions are commonly known as Carrier Sensing Multiple Access (CSMA) and have reported an increase of 2.5x in the number of devices that can transmit in a LoRaWAN deployment [25]. Another option is Time Division Multiple Access (TDMA) approaches that are shown to increase by 3 the capacity in LoRaWAN by scheduling the transmission of nodes [19].

Currently, the rules imposed to nodes in a LoRaWAN forbids a transmission if the duty-cycle is exhausted. However we believe that instead of limiting traffic, LoRaWAN deployments could improve significantly the number of nodes served by imposing stricter rules related to sensing the possible ongoing transmissions from the rest of the nodes in the network.

1.1 Main research question

The focus of the investigation is twofold, on the one hand, this work aims to study the PHY layer of LoRa to understand the collisions in a LoRa network and on the other hand, we analyze strategies to increase the number of packets correctly received. This study concludes with the proposal of a MAC (Medium Access Control) protocol to try to reduce the limitations of the current LoRaWAN MAC protocol. Out of the mentioned topics, the questions posed in the present document are the following:

1. How many packets collided and under which conditions can be successfully decoded due to the capture effect in LoRa?
2. Can the current LoRa technology perform carrier sensing in order to implement CSMA protocols in LoRa?
3. How much improvement in channel utilization can be achieved with a CSMA MAC protocol in a LoRa network while keeping the energy consumption of the devices low?

In order to find a response to the previous questions, experiments with LoRa devices are carried out to assess the performance of LoRa devices under collisions. In addition, to tackle the questions involving scalability of LoRa networks, since a considerable amount of devices is required, simulations are carried out in ns-3.

1.2 End goal and contributions

Our contributions are summarized as follows:

1. We design SFMAC, a distributed, energy efficient, CS MAC protocol for LoRaWAN.
   (a) SFMAC utilizes a dedicated control channel and the CAD mechanism to improve capacity in x2.08 compared to LoRaWAN at saturation levels.
   (b) We construct and analyze the optimization problem of choosing Spreading Factor and length of control packets for the dedicated control channel.

(c) SFMAC-devices can coexist with current LoRa-devices without requiring any change in the gateway/infrastructure.

2. We evaluate the behavior of Capture Effect (CE) by conducting on-field experiments. We leverage our observations to adjust SFMAC's parameters of operation.

3. We create the *ns-3* model of SFMAC to simulate scenarios with multiple devices and thousands of packets generated. Our findings in terms of CE are employed in the corresponding modules of *ns-3* to provide realistic simulations.

4. Through our simulations, we offer critical insights on the performance of SFMAC against two state-of-the-art CS-protocols, p-CARMA [25] and LMAC [18], and also against LoRaMAC regarding capacity, PRR, channel utilization, and energy consumption.

5. We evaluate SFMAC by utilizing realistic traffic from a monitoring application that uses cameras to preserve social distancing in public spaces. We prove that SFMAC would achieve the same results more efficiently and without raising any privacy issues, by using a single gateway and wearables connected under LoRaWAN.

1.3 Structure of the document

The rest of the document is organized in different chapters that introduce the reader to the results of the mentioned contributions. Chapters 2 and 3 introduce respectively the state-of-the-art and background of LoRa. In chapter 4 an analysis of collisions in LoRa is presented leading to the confirmation of existence of the capture effect in LoRa. Having studied the PHY layer of LoRa, chapter 5 proposes two MAC protocols for current LoRa networks, followed by chapter 6 that includes the final evaluation of the designed protocols.

Chapter 2

Related works

The current section presents research papers that helped to pinpoint the limitations of the state-of-the-art and to continue aspects from research in LoRa networks that was left unfinished. In section 2.1, MAC protocols that have been proposed for LoRa networks are presented. Following in section 2.2, the usefulness of relying on different channels to pack more transmission is shown. In sections 2.3 and 2.4 the PHY layer of LoRa is investigated to present works that leverage the decoding procedure of the LoRa chirps to obtain higher Packet Reception Ratios (PRR) from LoRa nodes.

2.1 Carrier sensing approaches

Out of the different MAC protocols proposed to substitute the ALOHA-like channel in LoRaWAN, we focus on the carrier sensing, (CS) approaches. A commonly used strategy to avoid collisions in wireless protocols is the adoption of CS techniques. The primer in CSMA was presented by Kleinrock and Tobagi [23, 36], wherein they introduced the variants of CSMA, proved their throughput, and tackled the hidden terminal issue. Hidden terminals appear in a network when the sensing ranges of nodes are limited and nodes that are transmitting to the same gateway do not hear each other's packets when performing carrier sensing. If the sensing ranges of nodes being hidden would be higher, the network operator could consider the problem of hidden terminals solved.

The first investigations of the applicability of CSMA protocols to LoRaWAN were made by To and Duda, and Kouvelas *et al.* [34, 26]. Both works simulated LoRa networks operating under CSMA variants to reduce packet collisions. To and Duda applied binary exponential back-off whenever the medium was sensed as occupied by Clear Channel Assessment (CCA) [34], while Kouvelas *et al.* introduced the application of p-CSMA in LoRaWAN [26]. Pham was the first to test the usage of CAD in CSMA for the LoRaMAC, by mapping the DIFS and the back-off windows of IEEE 802.11 and IEEE 802.15.4 to consecutive CADs [31]. To proceed, we focus on p-CARMA and LMAC, which both evaluate CAD on-field and utilize it reliably to perform channel sensing in LoRa networks [25, 18]. Both these MAC protocols use the local view of each LoRa-device on traffic either to adapt the persistent p-value of transmitting [25] or to perform frequency hopping [18]. Further, both employ gateway-downlinks, broadcasting information to devices regarding the global view of the gateway in terms of traffic. Apart from simulations, in LMAC results are also evaluated using a testbed of 50 LoRa-devices.

2.2 Control channel strategies

Authors in [36] solved the problem of hidden terminals by relying on the transmission of a busy-tone signal on a specific channel defined as busy-tone channel. The dedication of a channel for announcing an ongoing transmission is called as BTMA (Busy Tone Multiple Access) and it is used in our proposed MAC protocol. However in a LoRaWAN network is classified under LPWANs (Low Power Wide Area Networks), therefore the mechanism of BTMA is adapted for devices with power restrictions.

LoRa produces orthogonal signals when modulating an upchirp (increasing frequency) or a downchirp (derceasing frequency). As a result, under the same configured frequency in the RF front end of the device, we can count on practically 2 channels under the same frequency. Therefore, current LoRaWAN deployments uses downlink transmissions generated by the gateway to transmit MAC related commands. End devices demodulate or modulate bits in down chirps to join the network, acknowledge their transmissions, and update their configuration parameters, i.e., Adaptive Data Rate (ADR) messages. As established in the latest LoRaWAN specification, v.1.1 [8], such control information comes from the gateway. The ability of using the downchirps as *control channel* is the most related to our proposed MAC protocol. Therefore, without violating the standard, we enable devices to share information on channel availability with (considerable parts of) the network. As far as we know, this work is the first to dedicate low data-rate channels as control-channels, and define control/data pairs of channels in LoRaWAN.

2.3 Decode collisions

Studying the LoRa PHY layer in order to extract features and solve collisions is an interesting path to solve the challenge of scaling LoRaWANs. Xia *et al.* [40] resolve LoRa collisions by means of analyzing the continuous frequency of each signal in a collision in time domain. Authors in [37], present an algorithm to separate collisions through different peak ratios manifested in FFTs of received collided symbols. Finally, Tong *et al.* [38] leverage peak scaling factors measured in frequency domain to cluster transmissions from packets involved in collisions.

The mentioned solutions inspired this work to consider as receivers USRPs and hardware other than a LoRa gateway (i.e boards with SX1301 chip). Even though these works show promising results in collision resolution, our approach consists on, acknowledging that collisions arrive to a LoRa gateway, studying the survival probabilities of a collision arriving at a LoRa gateway.

2.4 Exploit Capture Effect

The thresholds of power difference and/or delay offset which probabilistically guarantee successful receptions in IEEE 802.11a under the CE were obtained by Lee *et al.* [27]. Whitehouse *et al.* used the capture effect, (CE) to design a flooding mechanism to reduce latency while maintaining goodput in dense networks [39]. The authors, on the one hand, considered trade-offs such as performing aggressively compared to inducing latency in the network and on the other hand show that radios can correctly receive one of several simultaneous transmissions.

Our results replicate this phenomenon in LoRa, therefore we include the capture effect in our protocol and simulations. Since CE is evident in LoRaWAN, as shown by Bor *et al.* [15], several works studied the effect, considering scenarios of both power and delay difference using devices connected to one or multiple gateways [15, 32, 21, 16, 13]. LoRa-frames were partitioned to study which symbols are more critical for the survival of the frame [15, 32, 21]. Further, closed-form expressions were derived regarding successful reception under CE taking into account the radio environment (i.e., fading, interference) [16, 13].

2.5 Limitations of state-of-the-art

CSMA protocols applied to wireless technologies are well-established mechanisms to avoid collisions. However, these techniques are applied to other wireless protocols (ZigBee or 802.11 standards) that have different requirements and PHY layers than LoRa. In the case of ZigBee, the CSMA/CA mechanism includes a collision detection mechanism since the protocol often allows acknowledged traffic, therefore nodes in a ZigBee network can receive feedback upon the outcome of their transmissions after CS. In LoRa, the use of acknowledged traffic is barely used, one of the reasons is that LoRaWAN gateways must adhere to duty cycle restrictions, resulting in the impossibility of acknowledging transmissions to all the nodes. Therefore innovations presented in [28], [24] count with elements that cannot be applied to LoRa deployments such as acknowledgments.

Regarding existing works of CSMA applied to LoRa, authors in [18] use the CAD mechanism to perform CSMA. However, our work differs from [18] in that our protocol does not rely on beacons transmitted from the gateway. In addition, [18] does not consider the use of a control channel to announce the transmissions of nodes in the network. Other proposed MAC mechanism for LoRa networks that rely on carrier sense is [25]. Our proposed MAC protocol compared to p-CARMA, again does not require in its functioning principle information from the gateway and furthermore, our proposed MAC mechanism tries to uncover hidden terminals by performing CAD in higher spreading factors. By using a higher spreading factor to detect a short control packet coming from a node ready to transmit, the nodes could successfully detect this transmission from higher distances compared to lower spreading factors. Compared to the means in which CAD is used in [18], [25], that last claim is leveraged in our proposed MAC protocol to increase the reach of carrier sensing.

Chapter 3

Preliminaries

LoRa is the wireless technology deeply studied in the present document. In order to ease the understanding of future optimizations and results presented, this section explains the main parameters of the LoRa PHY layer, the method that the LoRa gateway employs to decode the uplink traffic and MAC regulations currently enforced LoRaWAN networks.

3.1 LoRa PHY layer

The modulation of LoRa signals are located under frequency modulation techniques. The technology that uses LoRa is called chirp spread spectrum (CSS). CSS is similar to direct-sequence spread-spectrum (DSSS) used in GPS systems since it maintains acceptable performance under noise and interference. However, the patented LoRa wireless transmission scheme enables long range communication with low latency and low power consumption as opposed to DSSS.

3.1.1 Parameters associated to modulation

Since LoRa performs frequency modulation, several parameters allow the waveform to change its frequency over time. The shape of the waveform defined by In-Phase and Quadrature phase (I/Q) allows the receiver to decode the bits modulated by the transceiver. In order to decode the data correctly, the following parameters need to be set.

- **Carrier frequency, f_c.** Is the frequency of the carrier wave. In Europe the carrier frequencies allowed are 863 to 870 MHz range of the unlicensed spectrum [29]. LoRa gateways perform frequency-hopping to tune to the carrier frequency of the transmitting nodes.

- **Increasing or Decreasing frequency**. When the frequency of the waveform increases from a negative frequency value (concerning the carrier frequency, fc) to a positive value, the so-called upchirp is generated. On the contrary, a downchirp is a chirp that changes frequency from highest to lowest value.

- **Bandwidth,** BW. LoRa transmitters can make a wider use of the spectrum in order to reduce the duration of the LoRa transmissions in time. LoRa nodes choose from a wide range of bandwidths but the most commonly used are: 125, 250 or 500 kHz.

- **Spreading Factor,** SF. The rate at which the frequency reaches from its lowest value to its highest is defined by the spreading factor. The spreading factor is defined by a value from 7-12.

- **Symbol duration,** T_s. Depending on the spreading factor, the LoRa symbol has a different duration in time. The abscissa of Fig. 3.1 shows the different lengths of symbols ranging from SF7 (0.001024 s that is the more channel efficient configuration) to SF12 (with a T_s of 0.0327 s).

- **Data rate,** DR. Represents the inverse of the time on air of a LoRa symbol. DR values range from 5 to 0 which respectively are denoted high data rate and low data rate.

- **Coding rate,** CR. Redundancy is included in LoRa packets in order to recover bits in case of bursts of interference. The lower the value of CR, the longer time on air but the more protection against interference is offered. Usually, this value is 4/5 (out of 4 bits, one redundancy bit is added).

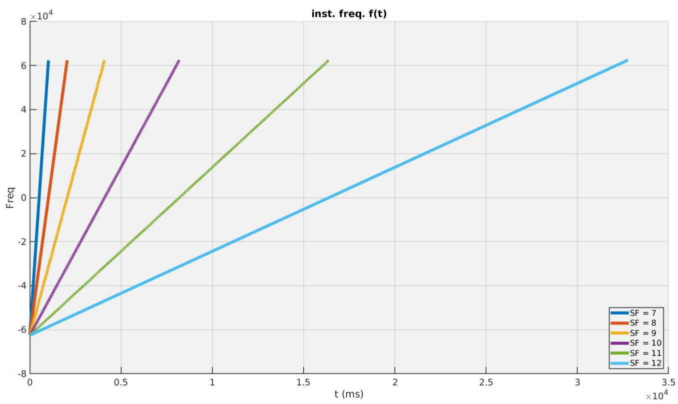


Figure 3.1: **Instantaneous frequency depending on the spreading factor, (**SF**)**

3.1.2 Waveform

Chirps are the main building blocks of transmissions in LoRa, manifesting change of frequency over time t, $f(t)$, described in eq. (3.1). The chirps encode 6-12 bits per symbol depending on the SF used. The decoder obtains the bits of a symbol from the initial frequency (i.e., $f(t = 0)$). The initial frequency manifests a peak of amplitude when performing the Fast Fourier Transformation (FFT) of the signal multiplied by a downchirp of the same SF. The value of the FFT-bin is an integer representing the encoded bits carried by a LoRa-symbol.

In eq. (3.2), a is the value of the symbol (an integer of length SF bits), therefore a is the FFT-bin with maximum amplitude of the frequency spectrum. In a LoRa-chirp the initial frequency is equal to the final frequency. The indicator I ensures that the chirp rolls around itself to preserve the frequency continuity as shown in Fig. 3.2. In addition, $\phi(t)$, B, T_s represent in eq. 3.2 the instantaneous phase, the bandwidth and the length of the symbol respectively. The resulting LoRa-signal is described by eq. 3.3.

$$f(t) = \frac{\partial \phi(t)}{\partial t}, t \in [0, T_s) \tag{3.1}$$

$$\phi(t) = \frac{a}{2^{SF}} - \frac{1}{2} + \frac{BWt}{2^{SF}2} - I\left(t - \frac{2^{SF} - a}{B}\right), t \in [0, T_s) \tag{3.2}$$

$$s(t) = \mathrm{e}^{jBW2\pi\phi(t)} \tag{3.3}$$

The waveform of a LoRa symbol is presented in Fig. 3.2. which reveals several unique traits that are summarized below. These characteristics could explain why a gateway can decode overlapping signals in time.

- LoRa signals are relatively long in time (e.g., the Time on Air (ToA) of a Wi-Fi symbol is 4 us), which may benefit the receiver from having a pattern to match against during 1 ms compared to 4 us of Wi-Fi. In addition, CSS modulation is robust against interference since the transmissions can be decoded under noise floor.

- The instantaneous frequency, and especially when the frequency is null, e.g., around t = 200 ms in Fig. 3.2 can be exploited by the decoder to ensure a high cross correlation value around the $f(t) = 0$ of the signal. Therefore, if this $f(t) = 0$ is kept intact even in a superimposed signal, the decoder can recover the original LoRa transmission.

3.2 Decoding of LoRa signals

Gateways are the receivers responsible of detecting LoRa waveforms and obtaining the bits encoded in the signals detected. As mentioned earlier, gateways perform frequency hopping to check possible transmissions in all the allowed frequency bands for LoRa.

Once a preamble is detected in a specific frequency, the gateway locks in that frequency to listen for the whole length of the transmission. The preamble is a sequence of symbols encoding the bit 0 (usually 8 to 12 upchirps) followed by 2 downchirps that have the value modulated of 0. The preamble is a common pattern to all LoRa transmissions that signal to the gateway the start of a LoRa packet.

The RF front end of the gateway collects the I/Q samples through the antenna and after the digital decoder has obtained the bits of the payload, these are transferred over SPI to the main processing unit of the gateway. Since LoRa is a patented technology, access to these raw I/Q samples through the LoRa gateway (SX1301) was not possible. Therefore to analyze the decoding process, a software defined radio (USRP) was used as a gateway to analyze the raw I/Q samples. The collected samples from the USRP were processed through

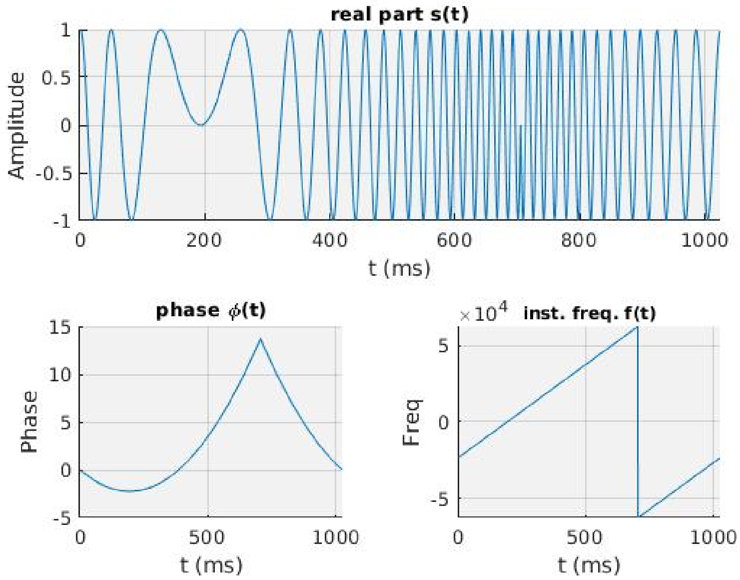


Figure 3.2: **Waveform, phase and instantaneous frequency of LoRa chirp generated in MATLAB. SF = 7, BW = 125 kHz, T_s = 1024 ms, a = 40**

GNURadio and Python. In order to mimic the process that a gateway follows to detect a LoRa transmission, we generated the I/Q samples of a LoRa symbol of a preamble in Fig. 3.2 via software and we performed cross-correlation to understand how the gateway locks into an incoming preamble.

As observed in Fig. 3.3, the process of correlating signal f and g resulted in a peak in the correlation $f * g$ due to the detection of a LoRa symbol hidden under noise. Therefore the I/Q samples collected by a gateway would be processed with this method to lock onto a symbol. This would be considered the first step of the decoding process. Further steps consist on multiplying every symbol (in this case every g by a downchirp and perform decoding in the frequency domain through FFT operations. Analysis in the frequency domain returns more accurate results since, as can be seen in Fig. 3.3, even though f and g are LoRa symbols, g waveform is seriously distorted by noise.

Note: the process described for decoding symbols in a gateway is also performed by LoRa end devices when performing Channel Activity Detection (CAD).

3.3 Analysis of LoRa traffic

Previously to designing a MAC strategy for the nodes of the network, and following works such as [22], the traffic in current LoRaWAN networks is analyzed in this section. Authors in [22] analyze the dynamics of the 802.11 traffic to predict where possibly *white spaces* could allow ZigBee nodes to transmit without creating interference. Similarly, LoRa traffic is analyzed in this section to extract features by treating the data collected as a time series. There are existing works

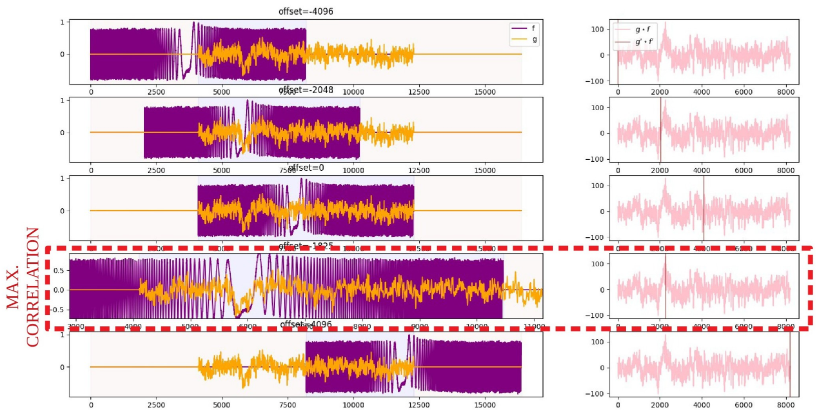
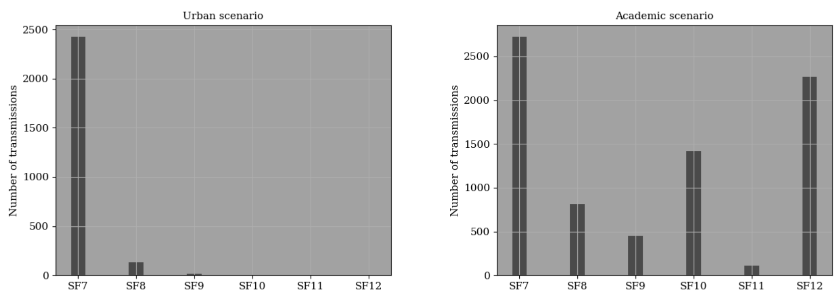


Figure 3.3: **correlation of LoRa symbol with preamble**



(a) **Urban scenario (Zaragoza, Spain)**

(b) **University campus scenario (TU Delft, The Netherlands)**

Figure 3.4: **LoRa transmissions collected in different scenarios**

in the literature that extensively collected data from LoRa deployments, such as [14]. However our data collection is directed towards performing statistical analysis to know how crowded are LoRa networks in different scenarios.

3.3.1 Data collection

The objective of the research was to obtain a dataset from real LoRa deployments. The dataset can be used to extract the most popular configurations of the nodes and detect for example crowded channels in the unlicensed spectrum.

The result of 4 days of logging LoRa transmissions in the city center of Zaragoza, Spain is presented in Fig. 3.4a. The results differ significantly from Fig. 3.4b since the technology is still not widely established in this city. The justification of that statement can be found in Appendix A.

A trend is spreading factor usage and can be extracted from Figs. 3.4a, 3.4b and Fig. 3.5. It is noticed that regardless of the scenario considered, the most used spreading factor is 7. The main reason for its popularity is the duration of the transmission in SF7. Even though the transmission range is lower by

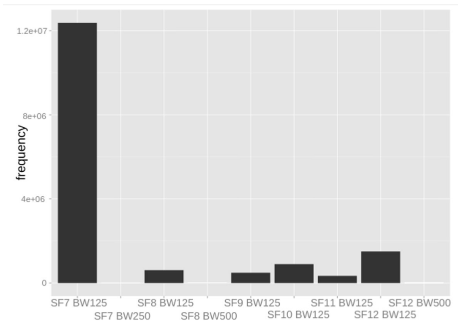


Figure 3.5: **Histogram of the used spreading factors [14]**

configuring a LoRa device in SF7 than in SF12, the time on air is reduced and the battery life gets extended by using SF7. This is an important observation and it is applied in the MAC mechanism proposed in Section 5.2.

3.3.2 LoRa traffic data as time series analysis

By treating the dataset as a time series, one could benefit from the extensive existing literature in models to predict events, or even to extract constant features from the time series. One possible outcome from the analysis of the LoRa traffic is the period of transmission of nodes in case their transmissions are periodic. If a gateway is capable of extracting features from a sufficiently-large recording period, it could provide information to the nodes about when to transmit and not to transmit.

Another possible outcome from analyzing the traffic as a time series analysis is to implement a slotted MAC protocol as proposed in [35]. The traffic is analyzed and depending on the spreading factor, slotted schedules are generated for nodes to avoid collisions. Since there seems to be advantages of utilizing a centralized protocol where the gateway is responsible for avoiding collisions, we present the results of our analysis of traffic.

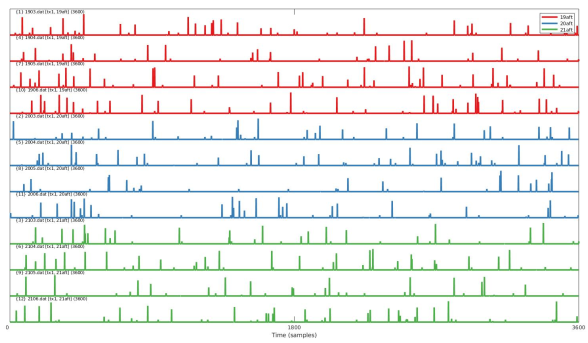


Figure 3.6: **Traffic per hour in the city of Zaragoza analyzed at the same hours for 4 days. Color classification distinguish transmissions collected at the same hour in the evening**

By using the framework presented in [17], we can apply classification methods to extract common features in the time series to see if there is correlation between the traffic of one day at a specific hour and the next day at the same hour. If the data analysis is performed properly and our LoRa traffic has a specific pattern, the Highly comparative time-series analysis (HCTSA) [17] tool will highlight if we can consider either periodic, semi-periodic or completely random traffic.

In order to determine if there is correlation between traffic seen during several days in the same hours of the transmissions presented in Fig. 3.6, a Principal Component Analysis (PCA) is carried out. A PCA analysis is a 2-dimensional feature-based representation of a time-series dataset. PCA highlights what are the most representative features that could be used to distinguish or on the contrary relate time series, therefore classifying different time series. If the result of the PCA is of 100% accuracy, it means that we found 2 dimensions (2 main features) that completely classify the involved time series. In Fig. 3.7 we test the accuracy of the linear classification model (SVM_linear), and what we obtain is a low classification rate. Therefore we conclude that there is no time relation between the time series considered for classification.

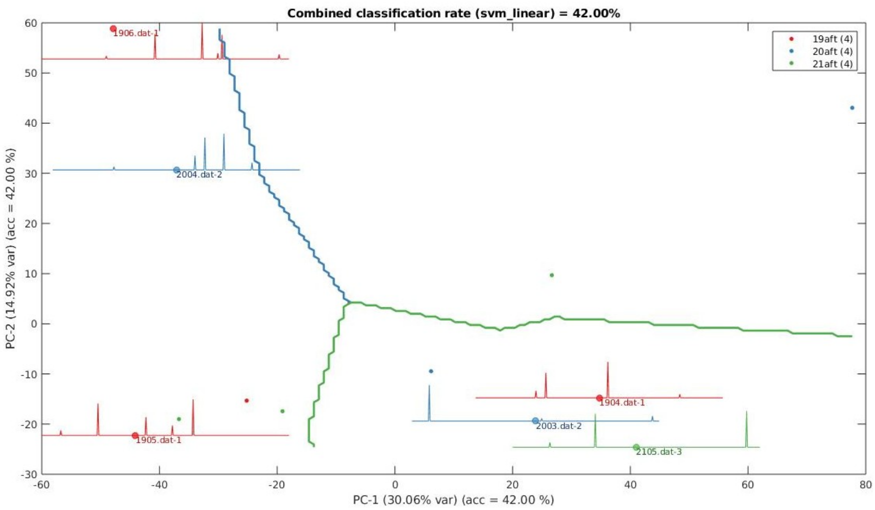


Figure 3.7: **PCA analysis. A 42% classification accuracy is obtained**

Since the feature space obtained with hctsa does not satisfy the requirements of finding a structure in LoRa traffic in the scenario considered, it shows that the traffic is random. Therefore, the idea of having a centralized entity (gateway) responsible for monitoring the traffic and informing the nodes of that underlying structure of the traffic is abandoned. However, it is useful to keep the tools used to classify the LoRa traffic in a specific deployment (i.e a private LoRa network, where the operator is aware of the transmissions from the nodes).

Chapter 4

Characterization of the capture effect in LoRa

This chapter presents an analysis of the LoRa PHY layer to disentangle the behavior of LoRa waveforms when collisions occur. An increase in the number of devices in a LoRaWAN may result in a higher number of collisions, therefore we study the survival probability of nodes colliding. To obtain the probability of successful reception even under overlap in time with other transmissions, results of the on-field experiments are presented in this chapter.

4.1 Capture effect analysis

This section describes the setup of the experiments and the results summarized in rules for a more realistic representation of the physical collision mechanism in our simulations in ns-3. The capture effect is a physical phenomenon that consists on having transmissions of a device being received successfully even if they overlap in time with other transmissions.

4.1.1 Experiments with LoRa nodes

This section deals with CE using SX1261 LoRa-radios [9]. Extensive on-field experiments have been performed. To avoid the near-far effect and antenna perturbations, a considerate distance of around 40 m was kept between the LoRa-nodes and the gateway (IMST Lite Gateway [5]). LoRa radios are configured with SF 7 and the time on air of the transmissions is 52 ms. The main objective of the experiments is to reflect in a probabilistic manner the proper reception of packets when two or three nodes are involved in a collision. Two scenarios are considered based on the capture effect; differences in power and/or overlapping differences in the reception time.

Capture effect due to delay offset. This experiment involves two/three nodes: (i) *early* node which transmits first and (ii) *delayed* node(s) which are delayed by different offsets with respect to early node. The results of this experiment are summarized in Fig. 4.1, wherein each node transmits 30 packets per delay configuration and the percentage of correctly received packets is the Packet Reception Ratio (PRR).

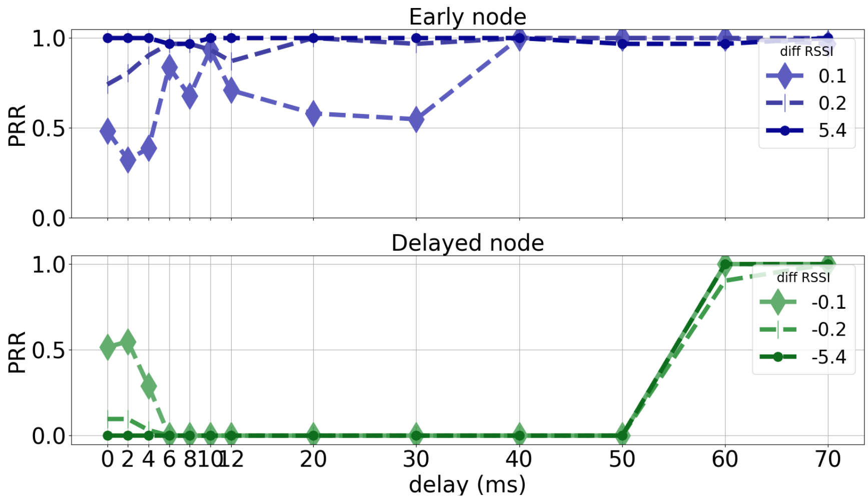


Figure 4.1: **Capture effect due to delay. Experiment with 2 nodes.**

Since RSSI reports a significant variance under the same conditions of the environment, the experiments are repeated and the RSSI values of the nodes are averaged per experiment run. As can be extracted from Fig. 4.1 that involves two devices, during the preamble duration ($t \in [0, 12]$ ms) and if the averaged difference in RSSI is almost 0 (i.e 0.1, 0.2), the early node benefits from at least 40% PRR while the delayed node manages to achieve merely 10%. The rest of the packets from both sides are either corrupted or lost.

Fig. 4.2 involves three LoRa-nodes of the same power. Two of them, nodes 1 and 2, are delayed with delay-values as indicated in the figure. It can be extracted that under the same reception power, early node gets affected in the preamble region (i.e., 0 to 10 ms), however after the middle of its packet (i.e., 20 ms delay), early node behaves the same as if it was interfered by one device. Further, regardless of the performance of the early node, the two delayed nodes neutralize each other, reconfirming our observation from Fig. 4.5.

Capture effect due to power difference. To characterize the difference in power needed for correct packet reception, two nodes (node A and B) are aligned in time and only their power difference is considered for analysis as observed in Fig. 4.3. In Fig. 4.3, the critical region where node A starts to reduce its PRR coincides with the region where node A and node B have practically the same RSSI. In addition, the PRR of node B stays stable with around $PRR = 1$ once the difference in RSSI is higher than 2 dBm.

Rules extracted from capture effect To compile the probabilities of correct packet reception and the needed power difference to achieve a specific PRR, the results from Fig. 4.1, Fig. 4.2 and Fig. 4.3 are accounted in the corresponding CE-module of the *ns-3*. In order to implement the model, the rules proposed in Table 4.1 and Table 4.2 dictate how to resolve the collision of two nodes arriving at the gateway. Regarding cases of more than two nodes: **(i)** if the early node passes the preamble the rules are the same as in two node cases, **(ii)** if the early node collides at the preamble and it is less powerful than 2 dBm from its opponents all packets are considered lost.

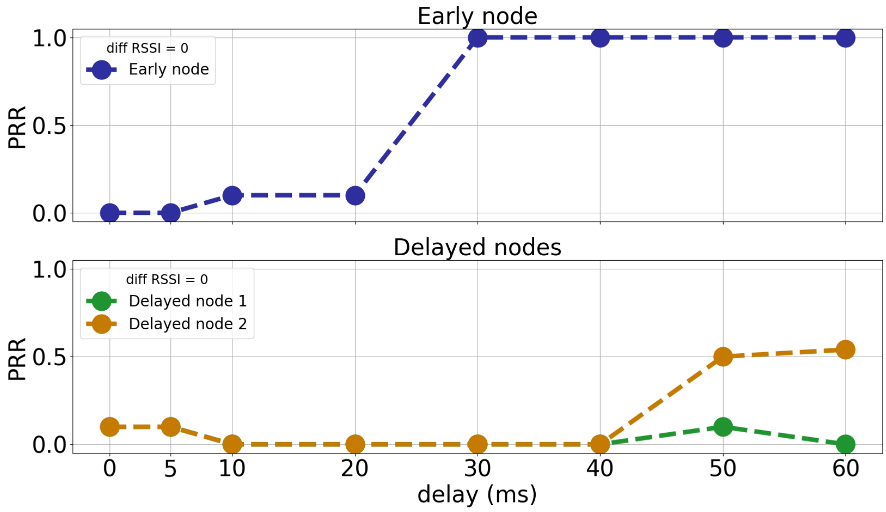


Figure 4.2: **Capture effect due to delay. Experiment with 3 nodes.**

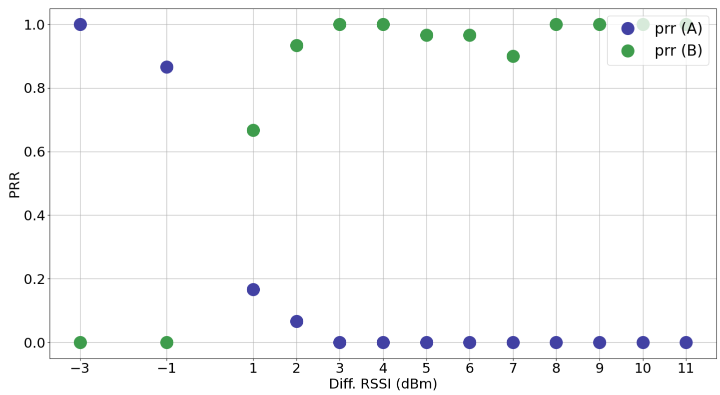


Figure 4.3: **Capture effect due to power difference.**

4.1.2 Analytical expression for capture effect

It can be noticed that describing the physical phenomena of the capture effect using a table with rules may result in more steps to implement. However previous publications [30] offer the following closed expression calculates the signal to interference noise ratio (SINR) to determine whether a transmission will result in a collision or not.

$$SINR_{i,j} = \frac{P_{rc,0}}{\sigma_w^2 + \sum_{l \in I_j} P_{rc,l}} \tag{4.1}$$

eq. 4.1 contains the power of the node to analyze if will survive the collision: $P_{rc,0}$ and the power from the rest of the interferers $\sum_{l \in I_j} P_{rc,l}$ overlapping in time with the node. The value of SINR calculated through eq. 4.1 is compared against a value of in dBm obtained from a SINR threshold matrix included in [20] to determine if a collision happened or not. As observed in Fig. 4.3, this

Table 4.1: **Early node conditions in RSSI for ensuring PRR = 1**

Collision region	**Time ranges (ms)**	**RSSI diff.**
First half of preamble	[0, 6]	>0.2
Second half of preamble	(6, 12]	>2
Payload	(12, 55]	>2
Last 1/10 of packet	(~55, 61]	>0

Table 4.2: **Delayed node conditions in RSSI for ensuring PRR = 1**

Collision region	**Time ranges (ms)**	**RSSI diff.**
First half of preamble	[0, 6]	>3
Second half of preamble	(6, 12]	>6
Payload	(12, 55]	>6
Last 1/10 of packet	(~55, 61]	>6

SINR threshold (between transmissions of the same SF is 6 dBm) indeed ensures the packet survival under collisions, but it is a rather conservative value since even until 2 dBm one of the colliding nodes obtained PRR = 1. Considering a difference in RSSI lower than 2 dBm requires a survival probability lower than 1 but different from 0 to replicate the results obtained in the experiments.

4.1.3 Superimposition of LoRa-signals

Analyzing collisions with a LoRa gateway results on simply obtaining an outcome of (un)successful reception of a packet. Therefore, a diagnostic tool, USRP B200mini [1], was used to study how signals get distorted in a collision.

Through GNURadio [3], In-phase and Quadrature (I/Q) samples from LoRa transmitters were collected. We emulate LoRa-radios through software by utilizing Software Defined Radio (SDR) as a transmitter. By sending packets of 20 B at SF7 with a ToA of 53 ms, we build a superimposed signal representing simultaneous transmissions of signals defined by eq. (3.3). It is assumed that at each instant, a collision results in a sum of I/Q of each of the individual transmitters involved. The mathematical expression of the superimposed signal, s_{sup}, is enclosed in eq. (4.2), with its real and imaginary part, Re and Im, stated.

$$s_{sup}(t) = \sum_{i=1}^{N} s_i(t) = \sum_{i=1}^{N} Re(s_i(t)) + jIm(s_i(t)) \tag{4.2}$$

Depending on how transmissions overlap, the sum of the I/Q can be constructive or destructive. In Fig. 4.4, we plot the captured I/Q samples of Tx_1 and Tx_2 from USRP and Tx_{sup} is the result of applying eq. (4.2). Fig. 4.4 shows constructive behavior since the overlap in time is considerably small and the shape of the superimposed signal does not differ from the shape of the individual transmissions.

Since the region of $f(t) = 0$ in Tx_1 coincides with the high frequency part of Tx_2, the shape of the superimposed signal in $t_{overlap}$ can capture the shape of Tx_1 and Tx_2. Further, we increase the number of emulated concurrent LoRa-

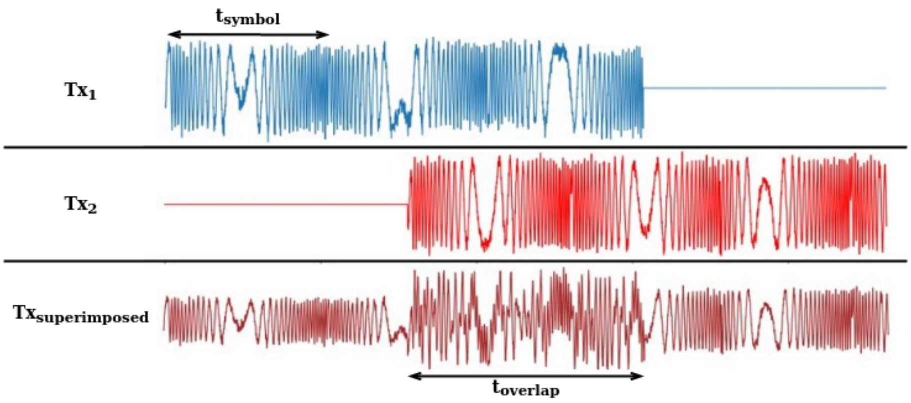


Figure 4.4: **Real part of the waveform during a collision of two LoRa-transmissions. Tx_1 (blue, upper) arrives first, Tx_2 (red, middle) arrives 1.5 symbols later. Notice the increase in amplitude of Tx_{sup} signal (dark red, bottom) as stated in eq. (4.2), sum of Tx_1 and Tx_2**

transmissions, still under the concept of the superimposed signal. Instead of 2 transmitters, in this case, 3-50 transmitters send simultaneously.

Fig. 4.5 includes the results of the transmissions seen by the gateway. Node A in Fig. 4.5 represents the so-called susceptible-to-interference node, (in Fig. 4.4 is Tx_1). Node B represents the sum of all the interferers depending on the experiment, with a specific delay. Each interferer node is delayed in each run of transmissions concerning the first node by 2 ms in 2 ms in the preamble (for example 2 to 4 ms) and by 10 ms in 10 ms in the payload (i.e 20 to 30 ms). On top of that delay, we add a unique delay on each interferer with regards to the first node by following ToA/q, where $ToA = 50\,\text{ms}$ and $q \in [0, N]$, with N interferers.

In Fig. 4.5 there is an increase in the number of transmissions of node A when the number of interferers grows. Even though it may seem contradicting, the susceptible-to-interference node obtains advantage from more collisions happening after it, since interferers neutralize each other.

The reason for claiming less interference under a higher number of transmitters can be found in Fig. 4.6a and Fig. 4.6b. In those figures, the SNR and RSSI values of the packets from the susceptible-to-interference node are presented. Even if the SNR gets degraded with the increment of transmissions, the RSSI maintains the same trend regardless of the number of concurrent transmissions and therefore the gateway still properly decodes the payload from the early arriving node (the so-called susceptible-to-interference node). From 50 ms of delay onwards the susceptible-to-interference node is not overlapping in time with 3, 4, 10, 20, 50 nodes anymore since the experiment is designed to not have interference after a delay equal to the ToA of the node. As a result, the maintained RSSI values of the susceptible-to-interference node with or without overlap in time, justify the correct reception of packets from Node A regardless of the increased interference.

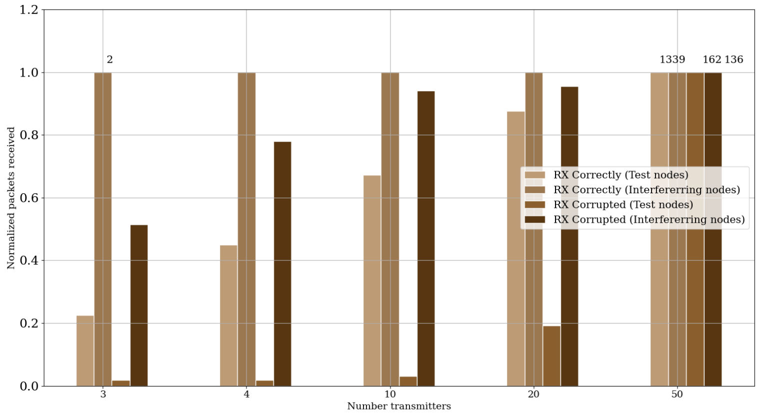
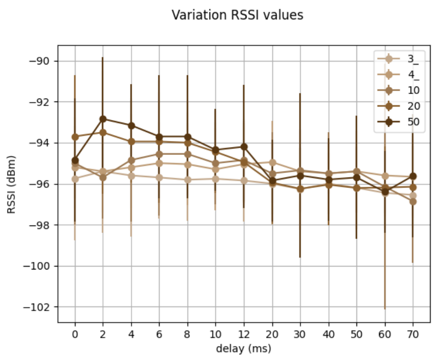
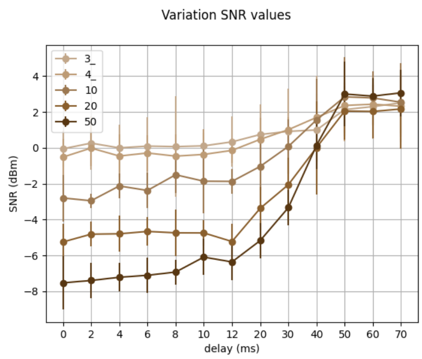


Figure 4.5: **Normalized number of packets decoded based on several transmissions. Values on top show the absolute value of the maximum values per series of values.**



(a) **RSSI values per experiment**



(b) **SNR values per experiment**

Chapter 5

Design of MAC protocol

5.1 CADMAC

CADMAC is an end-node MAC protocol that relies on the LoRa radios being able to detect the transmissions of non-hidden devices through CAD.

Since the CAD mechanism is used to provide nodes with CS capabilities, CADMAC results in a MAC protocol that could improve current LoRaWAN deployments. However, the collisions due to hidden devices remain unsolved since the nodes have limited reach of CAD sensing.

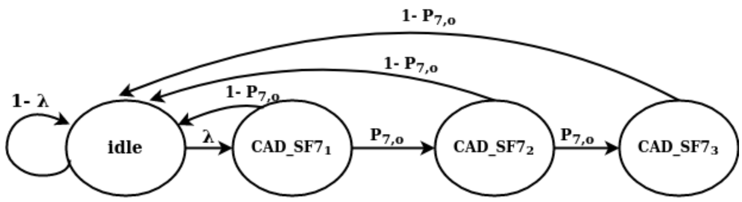


Figure 5.1: **Markov chain representing CADMAC. Note that SF7 has been chosen as SF example. However any SF can be used under the same CADMAC schema.**

The specific MAC actions that CADMAC enforces in the nodes are summarized in these steps: **(1)** Upon generation of a packet, the node performs CAD in SF7. The node transmits in SF7 and returns to sleep in case the channel is idle. If the channel is occupied, the node backs-off for W_1. **(2)** After W_1, the node probes again the channel in SF7, leading to transmission in SF7 and sleeping or backing-off for W_2 if channel is found occupied. **(3)** The last CAD attempt forces the node to transmit its frame with probability 1 either directly in case of free channel or after a W_1 back-off in case of busy channel. Table 5.1 summarizes the back-off values and the justification leading to them.

The choice of the duration of the back-offs depends on the length of the data-packets in our network, i.e, ToA of our frames is 70 ms since they are transmitted in SF7. However, if the node has already sensed the channel in the first place, performed the backoff [70, 90] ms and it still senses the channel occupied, it will perform a longer random back-off duration between [200, 400] ms. This back-off enables the node to choose randomly from x10 values and within the range 3

Table 5.1: **Backoffs enforced in CADMAC**

Symbol	Justification	Value random Bw.$[t_1, t_2]$
W_1	Linked to ToA of data packets	[70, 90] ms
W_2	Spreads back-off to reduce collision probabilities	[200,400] ms

nodes in best case scenario could choose spread enough random values (i.e 202, 274, 346) with a slack of approximately +50 ms . Consequently it should allow the different nodes concurrently accessing the channel to be more spread and reduce collision probabilities.

Since the CAD mechanism is used to provide nodes with CS capabilities, CADMAC results on a MAC protocol that could improve current LoRaWAN deployments.

5.2 SFMAC

SFMAC extends the utilization of CAD for CS to a separate control channel, of low data-rate, dedicated strictly to the transmission of control packets. A schema of the idea is presented in Fig. 5.2. The control channel allocated to a higher spreading factor is used to announce the transmission of a packet in a lower spreading factor, since lower spreading factors are the preferred configuration in LoRa devices for energy and speed in transmission reasons.

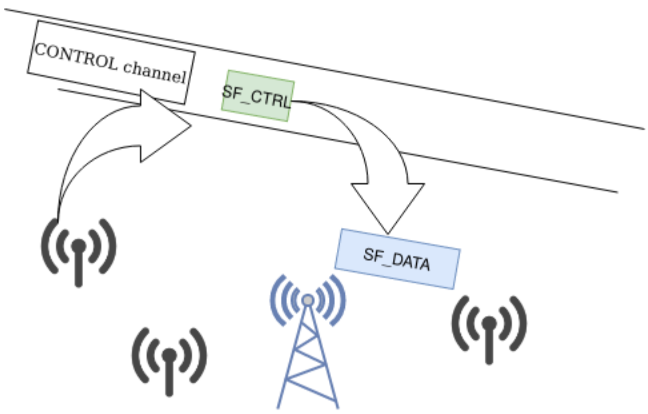


Figure 5.2: **SFMAC description**

In this way, a LoRa device can be informed regarding the state of the medium even by devices at relatively higher ranges than its neighbors, i.e., possible hidden terminals. Further, it allows transmissions of control-packets without affecting regular traffic, i.e., traffic in SF7. Additionally, in case of collision in the control-channel, having limited the traffic of an SF solely to control packets still allows the devices in the vicinity to sense the power from the collided control packets without the need to decode the payload. The control packet is a short sequence of downchirps with encoded value 0, i.e., only preamble symbols.

5.2.1 Design of control channel

SFMAC design can be formulated as an optimization problem to find possible improvements when operating under different configuration parameters. We take three approaches to find out the optimal design choices. The first approach consists on the formulation of an optimization problem to obtain the minimum energy consumption. The second approach focuses on setting an objective regarding the reduction in PRR needed to compensate for the extra energy by the use of control-packets and CAD in each SF. Thirdly, a heuristic approach is used to analyze in the time domain the change in the number of collisions depending on the parameters used.

Optimization problem

To define the parameters of the control channel, we address as an optimization problem the revealing of the maximum number of hidden terminals while simultaneously keeping the lowest possible energy consumption and collisions. The optimization was carried out using Gurobi 9.0.3 [4]. Most solvers offer optimization of linear objective functions. However, our objective function uses Gurobi solver since it minimizes non-convex quadratic objective functions. Our objective function is presented in eq. (5.1).

$$\mathbf{min}(\alpha[i,j](E[i,j]L[i,j] - D[i,j])) \quad \mathbf{s.t} \ \ (5.2)(5.3)(5.4) \tag{5.1}$$

eq. (5.1) minimizes $L[i,j]$ length in time of control packet and maximizes $D[i,j]$ the range for reliable CAD. E[i,j] is a ratio representing the energy cost of switching between SFs (i.e., $E[SF7]/E[SF12]$). In eq. (5.1), index i represents SF7 (i.e., starting point of choice of control channel), and index j is any other SF as possible outcomes for minimization. $\alpha[i,j]$ is a score to qualify the improvement from SFi to SFj. $\alpha[i,j]$ restricts the minimization problem since the sum of all the scores must equal 1, as stated in eq. (5.2). For solving the minimization, the model must be feasible and we must define the constraints in eq. (5.3) and eq. (5.4).

$$\sum \alpha[i,j] == 1 \tag{5.2}$$

$$L > [2,2,2,4,4,4] \tag{5.3}$$

$$D < [\frac{D_{SF7}}{D_{SF7}}, \frac{D_{SF8}}{D_{SF7}}, (...), \frac{D_{SF12}}{D_{SF7}}] \tag{5.4}$$

The minimum values of variables $L[i,j]$ are the number of symbols needed to perform CAD, derived from the datasheet of Semtech on CAD [6]. D_{SF7}-D_{SF12} are obtained from CAD experiments and they represent the maximum distance at which CAD is successful. Therefore by using the ratio (e.g., D_{SF12}/D_{SF7}), we obtain a measure of the improvement in reaching more nodes when choosing higher SFs. The optimization solver returns the majority of our variables equal to the bounds from eq. (5.2)-(5.4), except α[SF7, SF8] = 1 which would result in choosing SF8 as control channel according to our definition of score α. However, the time domain has not been yet considered in our optimization problem.

Throughput increase based on SF

The extra energy consumption when transmitting control packets in high SFs must be justified by the corresponding improvement in PRR. At each collision the energy of a whole transmission is lost. Therefore, if there is a substantial decrease in the number of collisions in scenarios utilizing a dedicated channel compared to scenarios using solely CAD, the extra cost in energy for the control packets is compensated.

Excluding the energy for data transmission, the remaining energy that is spent in a MAC protocol using a dedicated channel is denoted by E_1. Therefore, E_1 comprises of the total energy spent in CAD at the control-channel and at the data-channel ($E_{1,cad}$) and the energy spent in transmitting the control-packets ($E_{1,ct}$), as shown in eq. (5.5).

The energy of control-packets ($E_{1,ct}$) equals the product of the current needed per symbol (I_{ct}), the ToA of the dedicated SF, ToA_{ct}, and the symbol-length of the packet L_{ct}, seen in eq. (5.6). E_2 is the energy spent for CAD in a MAC protocol which does not use a dedicated channel, as seen in eq. (5.7). For both the above protocols, the energy spent in data transmissions taking place at SF7 is E_{Tx}, as seen in eq. (5.8); where ToA_7 is the Time on Air of a LoRa-symbol in SF7, L_{Tx} is the number of symbols of payload, and I_{Tx} is the value of current needed per symbol.

$$E_1 = E_{1,cad} + E_{1,ct} \tag{5.5}$$

$$E_{1,ct} = I_{ct}\ L_{ct}\ \ ToA_{ct} \tag{5.6}$$

$$E_2 = E_{2,cad} \tag{5.7}$$

$$E_{Tx} = I_{Tx}\ L_{Tx}\ \ ToA_7 \tag{5.8}$$

In eq. (5.9) we equal the energy consumed by a MAC using control- and data-channel to a MAC using only data-channel, excluding this part of energy spent for successful packets in each case, $E_{Tx}PRR_1$ and $E_{Tx}PRR_2$. We add E_1 which represents the energy spent in CAD and sending the control packet to the energy of transmission E_{Tx} and we subtract the energy of successful packets: $E_{Tx}PRR_1$. On the scenario 2 the calculations are the same as scenario 1.

$$E_1 + E_{Tx} - E_{Tx}PRR_1 = E_2 + E_{Tx} - E_{Tx}PRR_2 \tag{5.9}$$

Reducing the expression above, we obtain eq. (5.10), which is a relationship between the energy expense and the PRR improvement that should be accomplished to match the equality of eq. (5.9).

$$diff_{PRR} = PRR_1 - PRR_2 = (E_1 - E_2)/E_{Tx} \tag{5.10}$$

From Fig. 5.3, it can be inferred that higher SFs such as SF11 and SF12 enforce strict improvements in PRR. Situations wherein the $diff_{PRR}$ is higher than 1 are non-obtainable. Fig. 5.3 sets an upper limit on the collisions to be avoided once energy is spent in the control messages. From Fig. 5.3 we observe that compensations in PRR are not achievable when utilizing SF10-SF12 for dedicated channels. Therefore the next SF available, SF9, under the minimum difference in PRR is 0.13 when 1 symbol in SF9 is used as a control packet.

However, to increase the chances of detecting the control packet under CAD, 2 symbols are required [6]. Consequently, an objective of reducing the PRR on a 20% is chosen to compensate the energy consumed in transmissions of control packet of duration 2 symbols in SF9.

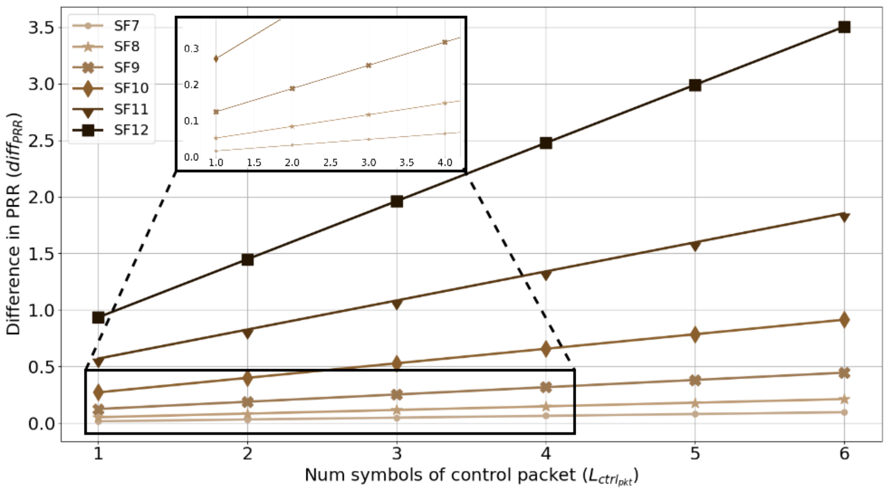


Figure 5.3: **PRR improvement required per SF and length of control packet**

Determination of optimal length and Spreading Factor of control packets

The previous analysis relied on global statistics. However, the optimal length of the control packet and its SF also depends on interarrival time of packets and topology. These dependencies of time and space have been modeled as shown in Fig. 5.4. The matrix of distances D_{SF} summarizes if a device, i, could sense through CAD the transmission of another device, j, according to our on-field CAD experiments. The matrix of state of the channel, H, represents the transmissions per node in millisecond resolution. Therefore, H, will contain 1 in (i, j) if the node i is transmitting at that moment in time. Since a transmission length of 50 ms is considered (ToA of SF7), elements (i, j) to $(i, j + 50)$ will contain 1 whenever node i transmits its LoRa packet starting at time j. In addition, H includes the control packet transmissions. To differentiate from the transmission of the payload, we label the presence of a control packet in the medium with *pkt*. The number of elements filled with *pkt* depends on the length of the control packet.

In order to characterize LoRaWAN under different length and SF of control-packets, we randomize and repeat the following process: (i) out of 1000 nodes forming a ring around the gateway, 200 nodes are selected, ensuring heterogeneity of location of nodes, (ii) each node generates packets with an inter-arrival time defined by an exponential random variable. The scheduled transmissions of the nodes follow then a Poisson process preserving the randomness in time. To find the optimal length of the control packet, we test from 2 up to 10 symbols. To reflect the effect of using a different SF, the matrix D_{SF} adds a 1 in locations $D_{SF}[i, j]$ and $D_{SF}[j, i]$ when nodes i, j can sense each other when increasing the SF. For every outcome of the setup, the elements of each column of matrix H are summed. As highlighted in Fig. 5.4 a collision (*col.*) occurs if the sum of a column in H is greater than 1. In Fig. 5.5, the average number of collisions out of ten runs of the process described above is presented for each different

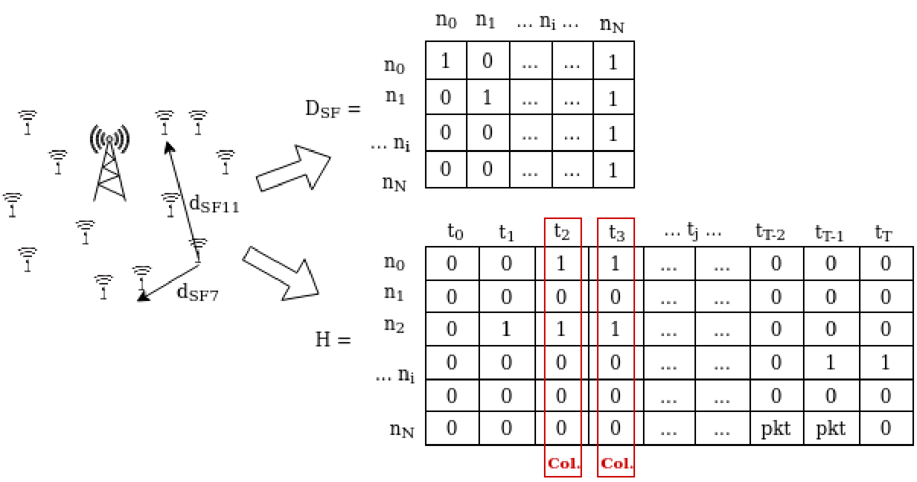


Figure 5.4: **Representation of method to find the best parameters for the control channel. Values in matrices are an illustration of information that could be contained.**

configuration. As observed in Fig. 5.5, the benefit of reaching further nodes when using higher SFs gets limited by the number of nodes that listen concurrently to the control packet. In order to balance the mentioned trade-offs, our final choice is SF9 and the length of 2 symbols, i.e., 8.192 ms of control packet. SF9 is preferred because as seen in Fig. 3.4b, the use of SF9 represents 5.8% of all transmissions. In the case of implementing SFMAC in current LoRaWAN deployments, the impact of limiting SF9 to control would be reduced compared to other SF.

5.2.2 SFMAC model

A schema of the SFMAC mechanism is presented in Fig. 5.6. After a generation of a packet, nodes perform carrier sensing on the so-called control channel: CAD_CTRL ($Default$) to detect if other nodes are announcing their imminent transmissions in lower spreading factors (SF7, SF8) labeled as SF_DATA. In case of channel found free with probability (P_{free}), nodes transmit in the spreading factor dedicated to control, (higher SF like SF9, SF10, SF11, SF12) labeled as SF_CTRL the control packet (TX_CTRL) and the transmission of the payload (TX_DATA) in SF_DATA. If the channel is found occupied when the outcome of CAD is $(1-P_{free})$, the node sleeps for the duration of transmission of the payload (TX_DATA). Upon wake up, nodes contend for the channel by listening for a random time duration in the state: CAD_CTRL ($Contend$). In case of unsuccessful CAD up to a specific number of attempts, nodes drop their generated packet.

The improved performance of SFMAC lies in the ability of our protocol to establish an order under a random arrival of transmissions. As can be observed in Fig. 5.7a, under current LoRaWAN deployments, nodes will transmit as soon as they generate a packet resulting in several collisions. For example around t = 200.2 node 168 collides with node 212 since it did not perform channel

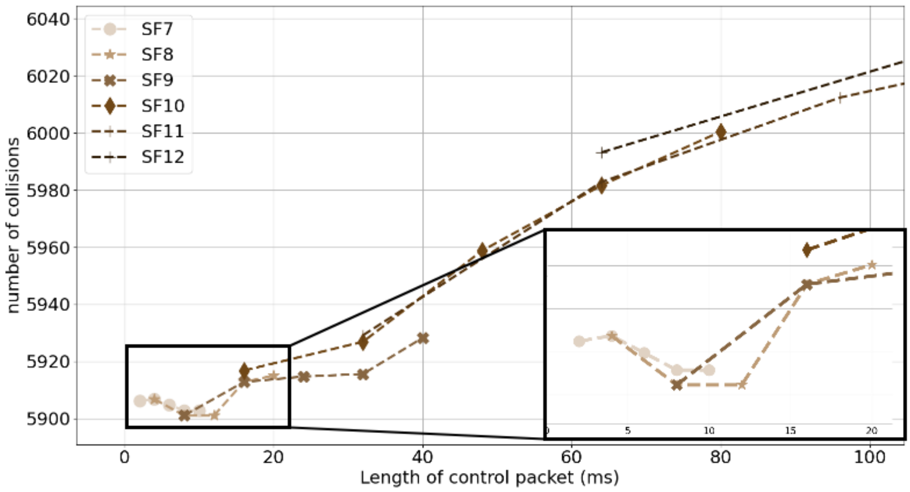


Figure 5.5: **Number of collisions depending on the length of the control packet and spreading factor. Average of 10 runs of the randomized process described above.**

assessment prior to transmitting. Note that the packet from node 212 is still correctly received due to the capture effect. However our protocol, as seen in Fig. 5.7b, can order the transmissions and spread them by allowing a structured contention period and in a worst-case scenario a dropping of the packet. In the case of the node with ID 168, since it is enforced to listen in SF_CTRL before transmitting, it manages to detect the ongoing transmission from node 212 and avoid the collision. In Fig. 5.7b, the dashed blue lines represents the time the node is either listening or sleeping until it determines based on algorithm 1 to transmit.

The MAC actions that would be imposed in the nodes using SFMAC are summarized in algorithm 1. The simple algorithm is built on top of the observations of section 3.3 regarding the uneven use of spreading factors observed in the LoRa traffic measured. By using the CTRL channel we make an efficient

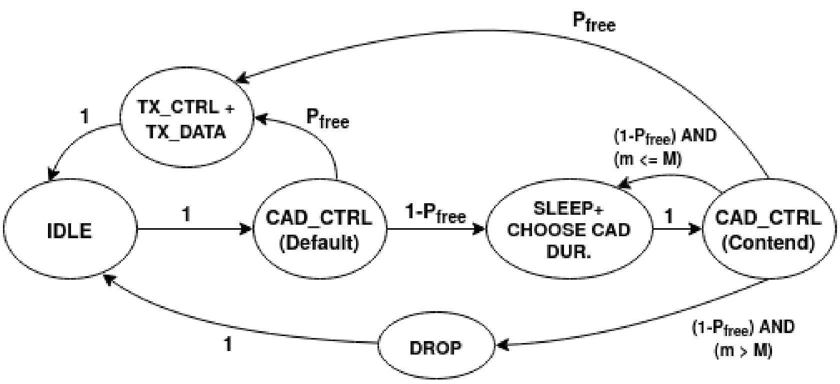
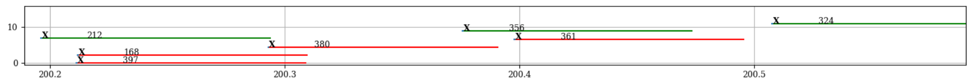
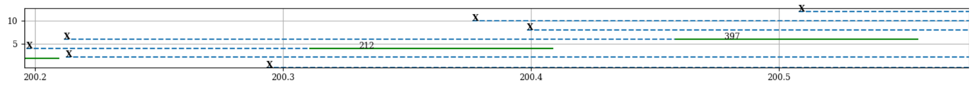


Figure 5.6: **SFMAC schema**



(a) **LoRaWAN**



(b) **SFMAC**

Figure 5.7: **Timeline under different MAC protocols. The crosses describe the generation of a packet. Red bars represent collided packets. Green represent successful packets.**

use of the spectrum, and as a preliminary evaluation of SFMAC in Fig. 5.7b, the collision reduction results evident.

Algorithm 1: SFMAC
(CTRL: control-channel, DATA: data-channel)

---

1 **Wait** for packet arrival
2 Perform **CAD_CTRL**
3 **if** *CTRL channel found free* **then**
4   **Send** packet in CTRL channel
5   **Send** packet in DATA channel
6   **Sleep** until next packet generation
7 **else**
8   **Sleep** for s
9   **if** $m <= MAX\_attempt$ **then**
10   **Choose** random listening time
11   **Go To** line 2
12   **else**
13   **Drop** DATA packet
14   **Sleep** until next packet generation
15   **end**
16 **end**

5.2.3 SFMAC Parameters

The increase of the PRR of LoRa nodes under SFMAC depends on the choice of the most appropriate values of the parameters presented in Table 5.2. When traversing the flow graph presented in Fig. 5.6, the node is restricted to parameters associated to CAD, backoff and transmission duration. In order to ease the optimization of the value of parameters, some parameters specified with values are fixed according to the most common LoRaWAN applications. The multiplication parameter, N is a constant relating the duration of transmission in SF_DATA and D which is the duration of transmission in SF_CTRL. In addition, the maximum number of attempts that the nodes can be contending for the channel is labeled as M.

Table 5.2: **Parameters**

Parameter name	**Symbol**	**Value**
Duration transmission payload	D	97.5 ms
Multiplication parameter	N	12
Duration initial listening time	X	$X = (N + 1)\ D$
Duration transmission control packet	d	$d = D/N$
Duration sleep time	s	$s = (N - 2)\ d$
Number of attempts allowed	M	-
Current attempt value	m	-
Min. contention window size	$C_W^{\min}$	-
Max. contention window size	$C_W^{\max}$	-
Contention window update policy	-	-
Contention window reset policy	-	-

After detection of a control packet and the corresponding sleeping time, the contention period starts with the choice of a random listening duration. The choice of the random value depends on the size of the contention window. The size and change in size per attempt of the contention window, $C_W^{\min}$, $C_W^{\max}$ are explained in Fig. 5.8.

The contention window reset policy refers to the strategy the node follows to reset the parameter, m. The node can reset the current attempt number m either per generation of packet (with reset), per drop (after drop) or per idleness of channel for a reasonable amount of time (after idle).

5.2.4 Optimization of parameters

Duration of transmission control packet

In order to maintain low overhead and energy in the implementation of the MAC protocol, efficiency in the duration of the control packet is found in this subsection. From subsection 5.2.1, the minimum values of collisions occurred under usage of 2 - 4 symbols of control packet for SF9. Therefore our protocol is implemented and tested under both 2 or 4 symbols of control packet. We omit the analysis with 3 symbols since our objective is to find a trend and we assume that 2 and 4 symbols would already show the trend of best-performing length of control packet.

In Fig. 5.9, a better channel utilization is achieved under shorter control packets. By using 2 symbols of the control packet ($d = 8$ ms) we benefit from the correct detection of the control packet due to the sufficient initial listening time, X, and a less delayed transition to transmitting the payload in SF7.

Contention window size, update policy and number of attempts

After the length of the control packet is fixed, the contention period parameters need to be evaluated. Following the description of Fig. 5.8, we fixed the length of the control packet to 8 ms, reset the policy to update the number of attempts after dropping and number of attempts to 4. As observed in Fig. 5.10, the

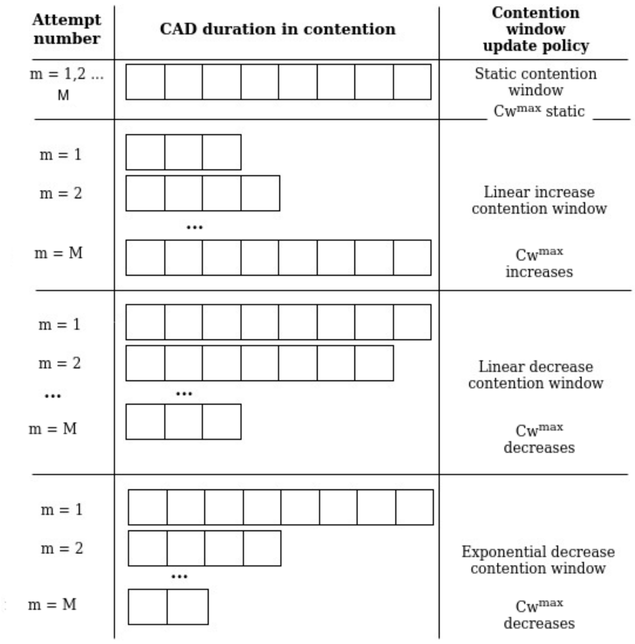


Figure 5.8: **Contention window update policies**

choice between exponential update or linear update is evaluated. The static contention window is not analyzed since in Fig. 5.9 did not report better results than linear decrease. The reason of the better performance of linear decrease lies in the values from where the nodes can choose from after attempting claiming the channel (after m=1 or first attempt). Since we consider 4 attempts, with exponential update we set for the $C_W^{\max} = 16$, so that we reduce in 4 steps to $C_W^{\min} = 2$.

The values for exponential increase coincide with an exponential decrease but the value of $C_W^{\min}$ is assigned to $C_W^{\max}$ and vice versa. In the case of linear decrease, we set $C_W^{\max} = 10$ and $C_W^{\min} = 4$ to obtain the same result at m = 2 (in both exponential decrease and linear decrease the most common value of attempt is m = 2, and both share the same window size of 8 at that m = 2). Since the linear decrease starts with a smaller $C_W^{\max}$, the normalized standard deviation is the lowest for linear decrease (fewer values to choose from under m = 1 compared to exponential decrease). Regarding the change in window size, the slowest the window reduces its size, the less number of packets collide since nodes can still choose from a considerable pool of values to not coincide with the choices from neighbors. Consequently, our study continues with a linear decrease update policy. The final metric considered to choose the update policy is the ratio of total energy consumed and the packets successful, as can be observed in Fig. 5.10 with linear decrease we achieve a 4% lower energy compared with the exponential decrease update policy.

In Fig. 5.11 it is presented the impact of changing the maximum number of attempts allowed having fixed the update policy (linear update policy). As reflected in Fig. 5.11, the increase in channel utilization after 5 attempts start to become stable, however the time that the nodes keep trying to access the me-

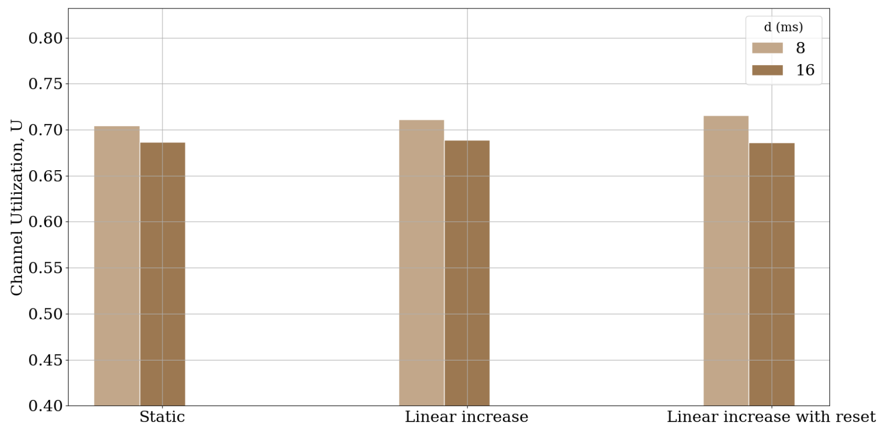


Figure 5.9: **Channel utilization based on length of control packet**

dium and therefore the delay in transmission increases significantly in medium and big CW sizes. With more attempts, the number of dropped packets diminishes, especially with the smallest window size since the nodes chooses amongst a very low duration of listening time, reducing the probability of increasing the value of m increases to the maximum allowed value. However more collisions happen since either nodes choose the same value or a higher number of nodes are involved in the contention period (more attempts means more time trying to access the channel and more delay for other nodes to join the contention period) and there are more chances to choose the same random number leading to a certainty of collision. There is therefore a tradeoff between the number of values to choose from and the delay associated to higher window sizes. Our final choice is 5 attempts since it reports a high channel utilization value with low delay and energy consumption considering the mentioned tradeoff.

Analysis of parameters under different traffic loads

Considering a crowded network, SFMAC maintains a high channel utilization due to the dropping and sensing policies. Independently of the offered load, linear decrease with reset policy after generation reports the best result. However, under a higher offered load (G = 1.5) our system reaches saturation since the best value of channel utilization reaches a limit for all the policies. From Fig. 5.12 we choose the medium size of contention window and reconfirm 5 attempts and linear decrease with reset as the most suitable number of attempts and policy respectively since for almost all offered values the highest channel utilization is achieved.

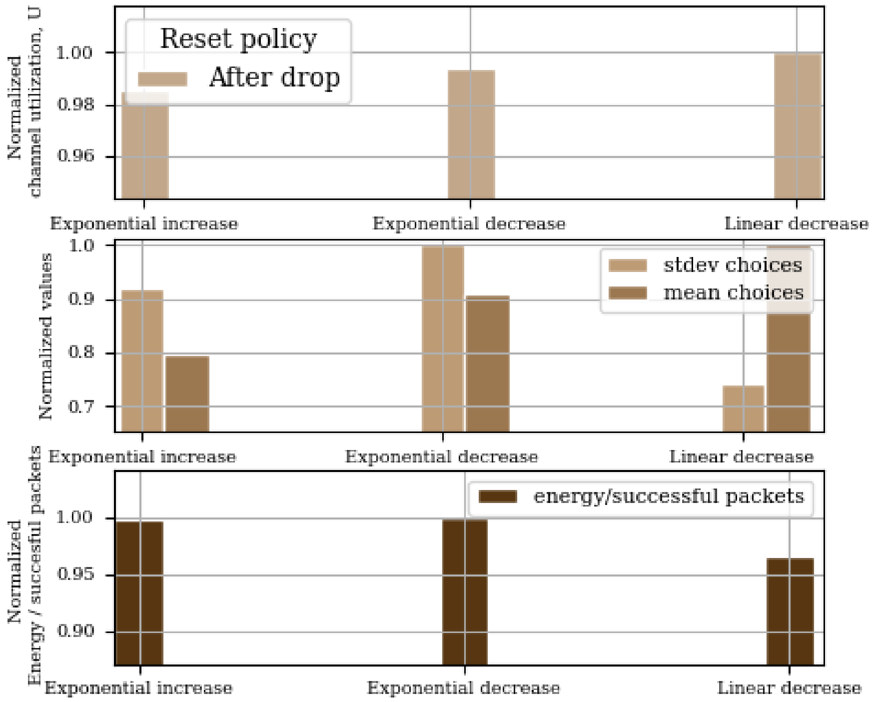


Figure 5.10: **Evaluation of update policies. The standard deviation, mean choices, collided and dropped packets are normalized with respect to the highest value per metric considered. Therefore a normalized value of 1 means that value holds the highest value of that metric.**

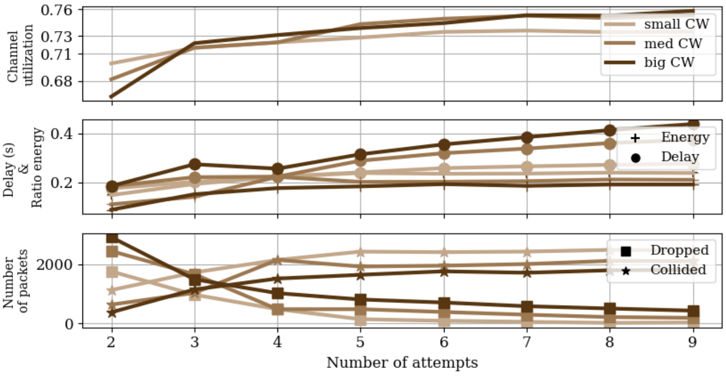


Figure 5.11: **Channel utilization based on a number of attempts and window size. Small, medium and big window size is respectively defined as $C_W^{\mathbf{max}} = 5$, $C_W^{\mathbf{max}} = 10$ and $C_W^{\mathbf{max}} = 15$**

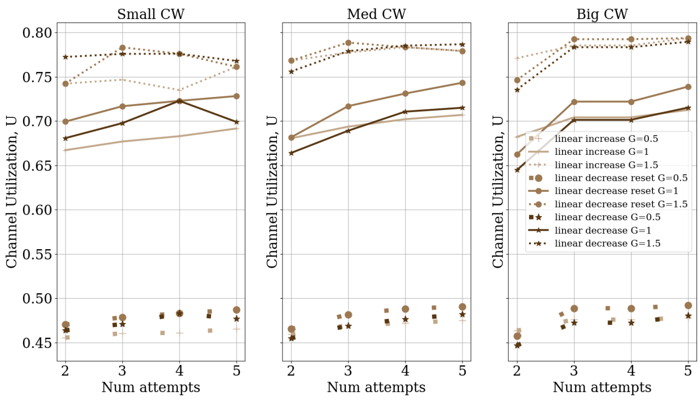


Figure 5.12: **Channel utilization based on different traffic loads.**

Chapter 6

Evaluation of MAC protocols

6.1 Parameters of simulation and evaluation

After having described the mechanism that guides the nodes to access the channel and send their frames, simulations of SFMAC have been performed in *ns-3*. Table 6.1 summarizes the simulation parameters. An exponential random variable with mean λ with a sufficiently low packet inter-arrival time ensures a realistic LoRaWAN deployment where collisions take place. The reduction of collisions achieved under Poisson traffic is transferable to other types of traffic. The end devices are located homogeneously around the gateway in a circle. The radius selected ensures that no hidden terminals are present in the deployment in order to analyze the highest potential of both CADMAC and SFMAC.

Table 6.1: **Parameters of simulation**

Parameter	**value**
Frequency of channel (MHz)	868.1
Number of devices, N	500
Number of gateways	1
Topology	Circle
Radius (m)	500
Packet size (B)	40
Coding rate, CR	4/5
SF of transmissions	7
SF of control channel	9
ToA (s)	0.0975
Offered load	G
Poisson rate (1/s)	λ = G / ToA
Transmission rate per node (1/s)	r = λ / N
Start time (s)	$\in [0, 100]$

6.2 Evaluation metrics

The LoRaWAN network performance is assessed using global and per-node metrics. The definition of the metrics to be analyzed is the following:

- **Channel Utilization:** It represents the normalized ratio of cumulative time of correctly received transmissions over the total observation time. This metric assesses how effectively the devices utilize the channel. The ratio of cumulative time of correctly received transmissions and the total time in a simulation run. Since we are considering a scenario where a network is congested, it is interesting to quantify from that congestion, how long we are sending correctly received packets.
- **Goodput:** It is defined as the number of correctly received units of information (bits) per second in our network. A closed mathematical expression of the goodput, (Gpt) is presented in eq. 6.1. The total number of correctly received packets, P_i is multiplied by number of bits in a packet and divided by the total observation time, denoted as T^{total}.
- **PRR per node:** PRR reflects the ratio between the number of frames successfully delivered to the gateway and the total number of generated packets.
- **Effective energy:** It is the product of Packet Error Rate (PER), defined as $PER = 1 - PRR$ multiplied by the energy consumed on average per packet. The effective energy is a score to weight the energy consumed based on the outcome of packet reception.

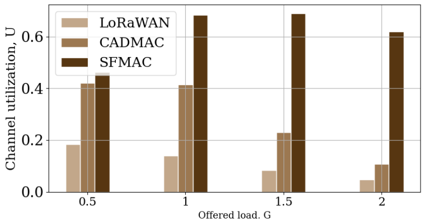
Apart from the aforementioned metrics, we also specifically compute the number of collided and correctly received packets under different MAC protocols and the corresponding energy that is consumed.

$$Gpt = \frac{\sum_{i=1}^{N} P_i \ b_P}{T^{total}} \tag{6.1}$$

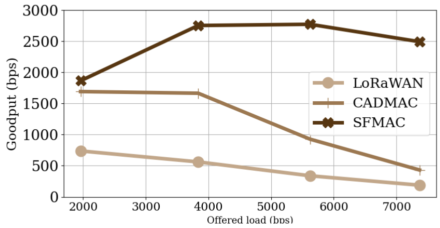
6.3 Performance evaluation without capture effect

Through simulations we evaluate the design and validate the assumptions made for SFMAC. Global measures such as channel utilization reflect the impact of the MAC protocol when scaling the network. In addition we show the distribution of metrics per node to evaluate if fairness is achieved among devices.

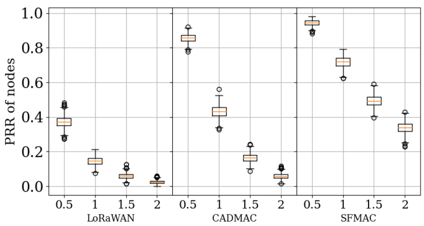
As observed in Fig. 6.1a, SFMAC achieves an x10 increase in channel utilization compared to LoRaWAN at G = 2. The reason for this difference is the dropping policy of SFMAC. The network is already saturated at G = 2, since even though a 62% of channel utilization is achieved, nodes have a PRR in average x0.31 lower PRR compared to G = 0.5 where the network is further from saturation. Under an offered load of G = 0.5, 0.18 of channel utilization is achieved in LoRaWAN, which is the theoretical limit of ALOHA deployments. When the offered load is 1 or higher, the performance of both CADMAC and



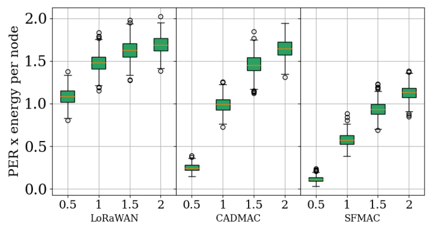
(a) **Channel Utilization**



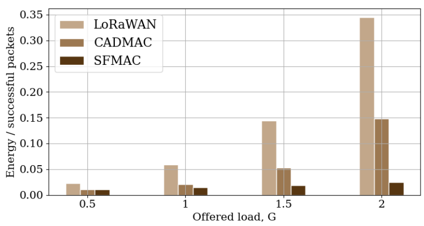
(b) **Goodput. A demand of 3800 bps represents an offered load of G = 1**



(c) **PRR**



(d) **Effective energy (J)**



(e) **Total energy (J) / Successful packets**

Figure 6.1: **Results without considering capture effect**

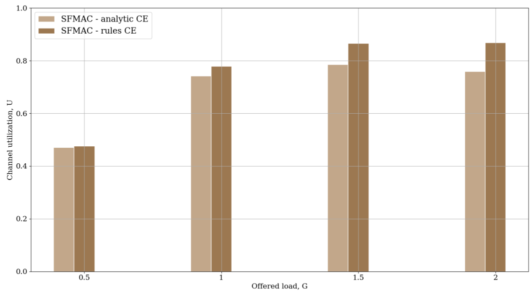


Figure 6.2: **Evaluation of simulated capture effect**

LoRaWAN decreases. The transmission rate is on average of 10 pkts/s under G = 1 and the time on-air of a packet is around 100 ms. Consequently, only under perfect TDMA we could achieve a 100% of channel utilization at G = 1. Since nodes perform packet dropping under SFMAC when a maximum number of attempts of transmitting is reached, the network is slightly unloaded. The linear decreasing update policy tries to avoid some nodes from constantly dropping its packets.

The fairness in SFMAC is reflected in the values of PRR presented in Fig. 6.1c. The biggest difference between the maximum and minimum PRR under SFMAC is 0.21 at G = 1.5. Nodes in SFMAC have x1.12 times higher PRR compared to CADMAC in the worst-case scenario, at G = 0.5.

Results in Fig. 6.1 do not reflect the real performance of LoRa networks. In reality the capture effect presented in section 4.1 allows to have better packet reception. If LoRa nodes would not benefit from the capture effect, LoRaWAN deployments could reach in the best'case scenario a goodput of 740.9 bps.

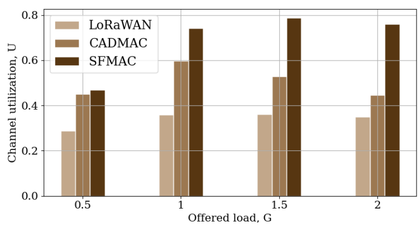
6.4 Performance evaluation with Capture effect

We add our observations of CE to LoRaWAN, CADMAC and SFMAC to obtain a closer to reality representation of collisions in a LoRa network benefiting from the capture effect. In Fig. 6.2, it is shown the effect of implementing the rules described in Table 4.1 and Table 4.2 compared to the analytic study of eq. 4.1. The revealing results of the on-field experiments where collisions resulting under some conditions in successful delivery are reflected in Fig. 6.2. The difference gets more evident with higher traffic loads since it is more likely that two nodes collide. Under the premise of 2 transmitters we apply the rules of capture effect.

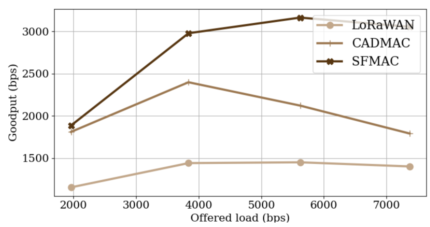
To expand our analysis, we add the following metrics to the ones described in section 6.3:

- **Energy per transmission per node:** This metric shows the energy consumed per transmission per node. It is a measure to evaluate the cost of implementing a more complex protocol than ALOHA.

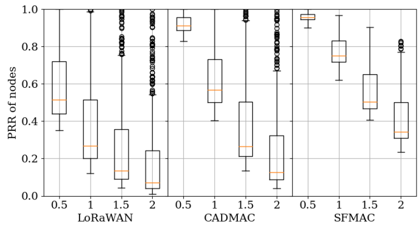
- **Packet Transmittance Ratio (PTR):** Since we impose a dropping policy in SFMAC, we evaluate the ratio of transmitted packets over generated packets.



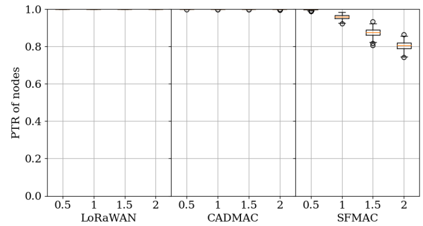
(a) **Channel Utilization**



(b) **Goodput**



(c) **PRR**



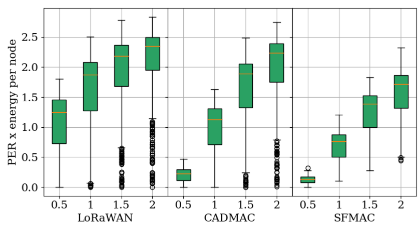
(d) **PTR**

Figure 6.3: **Comparison between MAC protocols including the capture effect**

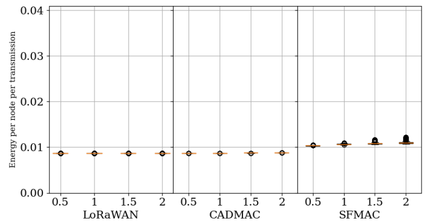
Performance of global metrics. Fig. 6.3c present the results of the simulations in ns-3 carried out for 2 hours per configuration. There is a noticeable boost in performance in LoRaWAN and CADMAC as compared to Fig. 6.1 since the capture effect has been taken into account. SFMAC improves LoRaWAN by x2.08 under a heavy loaded network. A value of G = 2 means the generation of 18 packets per second. In current LoRa deployments that traffic can seem high considering the duty cycle limitations. However it is relevant to still consider the scenario of a heavy loaded network because a) regulations of LoRa in some countries does not enforce duty cycle and b) The effect from the gateway perspective of for example 10 nodes transmitting 1 packet per node is the same as 1 node transmitting 10 packets.

Therefore if more devices join the public LoRa infrastructure without control, an offered load of 2 could be realistically reached. Under that traffic, SFMAC could serve nodes achieving an average PRR of x2.3 higher compared to LoRaWAN. LoRaWAN keeps a stable channel utilization in G = 1, 1.5, 2 while CADMAC slowly converges to the value of channel utilization of 0.36. The reason for not seeing reduced the value of 0.36 regardless on the increment of G can be explained with the outlier values of PRR in Fig. 6.3c. Since we have some *privileged* nodes located close to the gateway its frames will be received with the highest power. The frames of these *privileged* nodes are successful regardless on the number of transmitters colliding, reaching PRRs close to 1 (this phenomena is observed in Fig. 4.5 where the strongest node always obtains a high PRR regardless on the number of interferers).

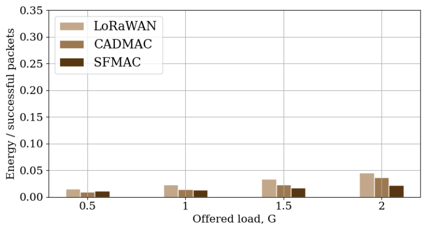
In addition, the values of channel utilization observed in Fig. 6.3 could determine when the nodes in our network should change from CADMAC to SFMAC.



(a) **Effective energy** (J)



(b) **Energy per transmission per node** (J)



(c) **Energy** (J) **/ Successful packets**

Figure 6.4: **Comparison of energy between existing MAC protocols considering the capture effect**

Since under G = 0.5 the differences between CADMAC and SFMAC are not that considerable in channel utilization and the energy spent in CADMAC per node per transmission is x0.8 the energy in SFMAC, nodes could receive feedback from the gateway to modify their protocol under a detected traffic close to G = 0.5. However, for increased traffic, SFMAC outperforms CADMAC by up to x1.78. The stability in the channel utilization of SFMAC is achieved after G = 1.5 since the value of channel utilization remains the same.

Performance of individual metrics. Fig. 6.3 reflects the fairness of the MAC protocol used in terms of number of packets delivered by the worst and best-performing nodes. In LoRaWAN and CADMAC, since there is not a dropping policy, the nodes closer to the gateway manage a PRR of 1 compared to 0.1 reported by other nodes further away. It should be noted that if the topology considered would be a ring, these differences in best and worst PRR would be lower. On the contrary, SFMAC manages to maintain the PRR of nodes at least at 0.22 under G = 2, which is the worst-case scenario.

Energy overhead. SFMAC reports in Fig. 6.4b a 37% increase in energy compared to LoRaWAN per transmission under the worst-case scenario. However, this does not reflect the amount of energy spent for correct receptions or wasted for collided packets. Therefore, in Fig. 6.4a we calculated $(1 - PRR) * E_{per\_Tx}$ which shows the effective energy consumed during the observation time under the different MAC protocols. Considering that the energy overhead per transmission in SFMAC was 37% higher than the energy of LoRaWAN, Fig. 6.4a shows that SFMAC decreases the energy spent in collisions per node by x6.25 under G=0.5. A global assessment of the energy consumed in our network is presented in Fig. 6.4c, it has been scaled to have the same y-axis as Fig. 6.1e. By means of scaling, we can again assess the impact in reduction

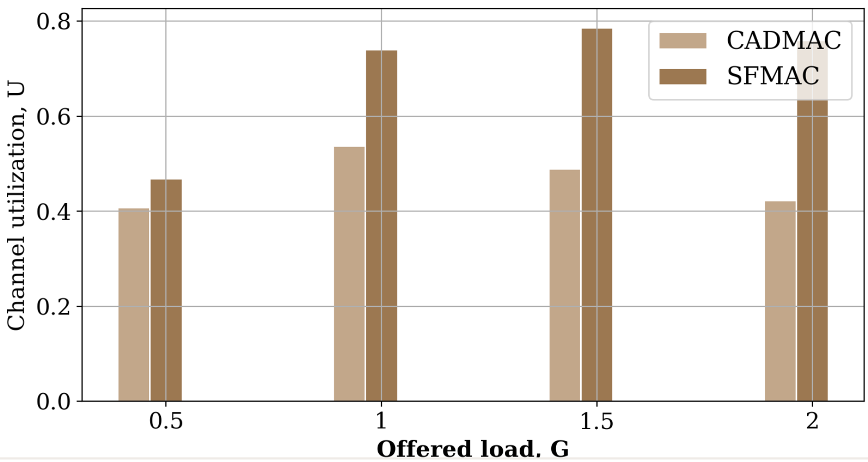


Figure 6.5: **Impact of reducing 33.3% sensing range of CADMAC**

of collisions (and therefore boost in correctly received packets) that LoRa PHY layer offers thanks to the capture effect.

The results of Fig. 6.3 and Fig. 6.4 are obtained considering that CAD in SF of higher data rates (i.e 5) is the same as the performance of CAD under lower data rates (i.e 3). The reason for not imposing a different sensing range in CADMAC (CAD in SF7) and SFMAC (CAD in SF9) is because the radius of our circle is 500 m, which leads to a maximum distance between nodes of 1 km, distance covered by both SF7 and SF9. However, in the literature it is found that the sensing range of CAD under higher spreading factors is higher. In order to show the impact of hidden terminals and compare SFMAC and CADMAC, the sensing range of nodes in CADMAC deployment is reduced by 33.3% in Fig. 6.5(Since the differences in range between SF7 and SF9 leads to that decrease in CAD sensing range). Compared to G = 0.5 in Fig. 6.3a, CADMAC reduces its performance in the best-case scenario by 9.5%.

6.4.1 Effect of imposing failures in CAD in SFMAC performance

Until now the success probability of CAD mechanism in our simulations is 1. However in a real LoRa deployment attenuation, path loss and interference joined to hardware imperfections can lead to false negative outcomes after channel sensing. Therefore in Fig. 6.6 we analyze the impact of inducing CAD errors, leading to upmost a 6.75 % downgrade of performance compared to perfect scenario.

The degradation per traffic load and percentage of failure is presented in Table. 6.2.

6.4.2 Comparison with state-of-the-art

The most relevant publications regarding proposed MAC protocols for LoRa networks are studied in order to perform a fair comparison between SFMAC and

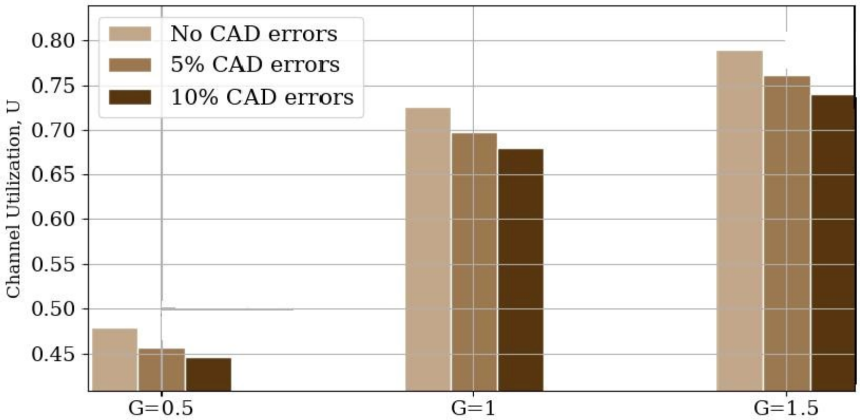


Figure 6.6: **Performance of SFMAC under CAD errors**

Table 6.2: **Percentage of channel utilization degradation compared to no CAD failures**

Offered load	% failure	% degradation
0.5	5	4.29
0.5	10	6.75
1	5	3.88
1	10	6.27
1.5	5	3.57
1.5	10	6.14

the state of the art. LMAC [18] proposes several versions of a MAC protocol for LoRa with CAD as carrier sense mechanism. We will compare to LMAC-2 since it reports the best performance for class A devices. Whereas LMAC use 16 combinations of 8 channels and 2 SFs, we use SF7, SF9 and one frequency. While LMAC uses 16 B transmissions, 50 nodes and could use in extreme cases 450 CAD, SFMAC uses 40 B payload, 500 nodes and up to 5 CAD attempts. p-CARMA [25] differs from SFMAC, apart from the use of a dedicated control channel, in that p-CARMA needs a convergence time to find optimal p values, while SFMAC behaves equally over time. Note in comparisons with p-CARMA there is only one value to compare against, since the simulations were not run with an offered load higher than 0.5.

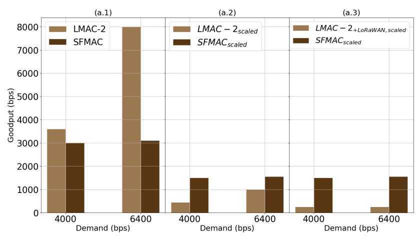
Goodput. Fig. 6.7a presents a goodput comparison between LMAC-2 and SFMAC. Since LMAC uses 16 combinations of SF and transmission channel, we scaled it down to compare to SFMAC. In Fig. 6.7 (a.3) LMAC-2 is implemented under the restrictions of current LoRaWAN specification (guard times, two receive windows...) As observed, the goodput offered by SFMAC at high traffic outperforms LMAC-2. Specifically, SFMAC achieves x6.25 times higher goodput at 4 kbps. The increase in goodput highlights the potential of using a control channel under dense traffic and minimum resources, i.e., one frequency and two SFs (data/control channel).

A reason for improvement with SFMAC, can be the fact that LMAC and p-CARMA perform carrier sensing in the same channel as the transmissions of the payload. However, SFMAC does more reliable channel sensing, because it dedicates a channel for assessing the state of the channel.

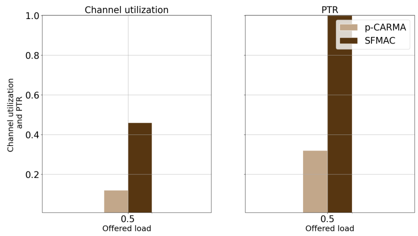
Channel Utilization. The channel utilization shown in Fig. 6.7b show a x3.6 increase in channel utilization compared to p-CARMA. The above confirms the advantage of direct channel assessment given by the control-channel than the indirect, probabilistic adaptation of p-value that takes place in p-CARMA. In addition, the PTR of Fig. 6.7b shown in SFMAC transmits x2.63 more compared to p-CARMA, even if both mechanisms incorporate dropping mechanisms.

PRR. Since the demand of the network differs significantly between p-CARMA and LMAC-2, SFMAC is compared individually with p-CARMA in Fig. 6.7 (c.1) and with LMAC-2 in Fig. 6.7 (c.2). SFMAC leads to a x1.72 increase in PRR compared against p-CARMA thanks to the existence of a control channel, the spread in time of backoffs and dropping policy. Compared to LMAC-2, our PRR results inferior unless we scale to considering the PRR per channel and SF where we report a x3 improvement under a demand of 4000 bps.

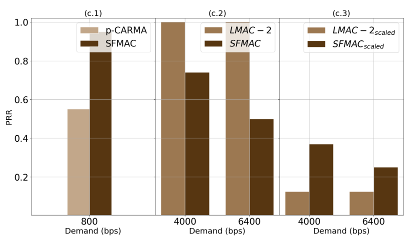
Increase in energy consumption compared to ALOHA. The ratio of energy consumed with LoRaWAN (ALOHA) and the proposed MAC protocol is presented in 6.7d. The energy overhead of transmission of packets in SF_CTRL is needed to achieve the improvement in previous metrics. Fig. 6.7 (d.1) calculates $E_{p-CARMA}/E_{LoRaWAN}$ and the energy overhead compared to p-CARMA is x0.85 times lower with SFMAC. In case of Fig. 6.7 (d.2) we calculate the ratio of energies consumed on packets correctly received $E_{LMAC-2}/E_{LoRaWAN}$, since in LMAC-2 it is measured the energy consumption per frame reception. Since successful packets are used, there is a decrease in energy ratios in SFMAC under higher loads in Fig. 6.7 (d.2) achieving the best result of a decrease of x0.42 times the value of E_{LMAC-2}.



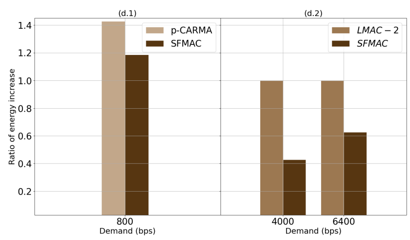
(a) **Goodput**



(b) **Channel Utilization and PTR**



(c) **PRR**



(d) **Energy**

Figure 6.7: **Comparison of SFMAC and state-of-the-art**

6.4.3 Multiple gateway scenario

The authors in [12] performed extensive data collection of LoRa transmissions in different gateways spread in the city of Antwerp. In order to test our algorithm in a real city deployment, and obtaining the location of several gateways spread in the city of Antwerp from [12], the performance of SFMAC is assessed. As expected, in topology 1 (the real locations of gateways in Antwerp) we obtain very similar results as the previous sections since there is not overlap in the circles of r=500 m that is the distribution of nodes in our simulations. In order to get a closer result to what LoRaWAN deployments in smart cities will look like, (where the population of gateways will increase to serve the increase of end devices), topology 2 in Fig. 6.8 halves the distances between gateways in topology 1. As noticed in Fig. 6.8, gateways that are not in contact with other still preserve the same channel utilization. On the other hand, networks that are closer to each other experiment higher traffic than usual, in our network we are modelling an offered load of G = 1. SFMAC still offers a 0.58 channel utilization even under interference from close deployments.

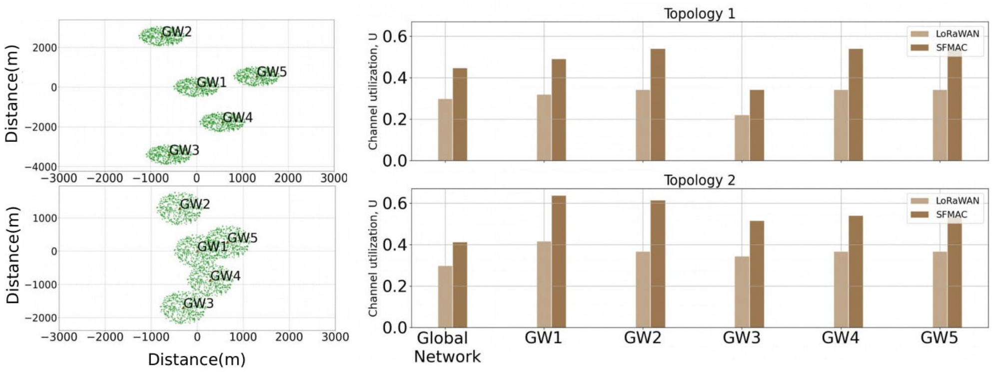


Figure 6.8: **Performance of SFMAC in multiple gateway scenario**

6.4.4 SFMAC evaluation in a practical application

In this real life scenario, we build on top of an existing project [41] that has the objective of monitoring with a camera if social distancing is preserved in public spaces to avoid the spread of COVID-19. Yang *et al.* [41] use several datasets of images to perform image processing and detect violations in social distances. We will take as example of application one of the datasets used which contains a footage from NYC Grand Central Terminal. The footage was introduced by Zhou *et al.* [42]. Using the outcome of these processed images and distance calculations, we can model traffic and distribution of nodes in our own scenario. Our scenario would consider instead of cameras monitoring, wearable devices tracking whether the distance between individuals has been preserved in a public space. These devices report to a gateway whenever it a social distance violation is detected. This section evaluates whether SFMAC can handle all the generated transmissions.

In order to simulate how SFMAC would handle all the transmissions, both violations of distance per time and space have been adapted. We have gathered the number of violations happening per unit time and we have simplified the space to cover with a grid the available space. In order to use the potential of performing social distancing monitoring with LoRa, we increase the distance covered by the footage until 1 km in one dimension and we sparse the data obtained from the image processing of Fig. 6.9.

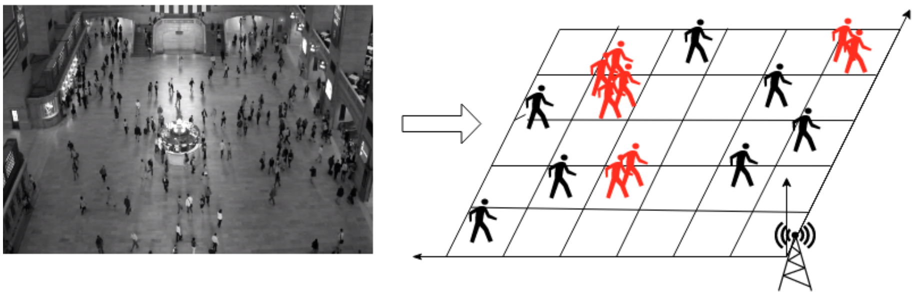


Figure 6.9: **Diagram of social distancing monitoring using LoRaWAN. Red icons represent areas where social distancing is not being preserved. In those areas, pedestrians using LoRa-enabled wearable devices will transmit a frame to the closest gateway. Example of image processed, [42].**

Compared to deploying cameras and dealing with privacy issues because of recording on public spaces, using long-range-transmitting wearable devices could be a more efficient solution to raise awareness to keep social distance. As observed in Fig. 6.10 by using a LoRa network with SFMAC, we could cover a monitoring area of 1 km x 1 km and obtain 80% correct packet deliveries in some areas with a single gateway.

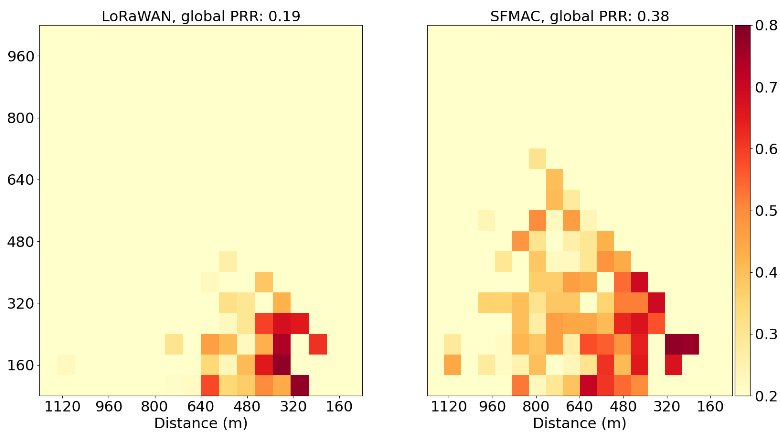


Figure 6.10: **PRR per area. Most of the color map corresponds to PRR = 0 due to nonexistent distance violations in the footage from NYC Grand Central terminal**

Chapter 7

Conclusions

The conclusions of the work presented can be defined by revisiting our initial research questions. The present document has analyzed the LoRa PHY layer and applied the results of the collision experiments to represent more accurately the capture effect. Thanks to the robustness of LoRa PHY layer, also under severe interference, LoRa signals can be decoded. Even if the outcome of the collision is a wrong CRC, the auto-correlation of the incoming signal with a LoRa symbol can report a peak enabling the receiver to at least determine if the medium is idle or busy. By reducing the listening requirements to detection via CAD and not to enforce decoding, we ease the implementation of SFMAC. Thanks to eliminating the need of decoding, we trust that in case of collision between control packets, the nodes performing carrier sensing could still assess the state of the channel as occupied. The Long Range capabilities of LoRa create a spectrum of different received signal strengths at the gateway, and from these power differences, nodes arriving first or with stronger power, benefit from seeing their frames decoded. We evaluated the impact of the capture effect when scaling the network compared to the deployment without capture effect and found a best-case scenario of 6 times the number of correctly received packets in LoRaWAN when considering the capture effect.

After evaluating the performance of SFMAC, we have significantly increased the successfully received packets in a LoRa network under heavy traffic. Our proposal of MAC protocol has followed the progressive systems development methodology where from an initial version of a MAC protocol, CADMAC, we have reached an improved version, SFMAC, building on top of the features of the previous version. Our implementation has been compared against the state of the art. While LMAC benefits from using a wide variety of channel configurations, our protocol manages to improve channel utilization while making sure that the majority of spreading factors are being utilized without requiring nodes to change the frequency of the channel assigned. Since we are dealing with a scarce availability of frequencies due to the allocation of LoRa in unlicensed bands, the better the usage per frequency, the closer the implementation is from solving scalability issues. In order to introduce the least number of changes in current LoRaWAN deployments, our nodes follow the LoRa alliance specification of opening two receive windows after transmission, and when needed, nodes could adhere to duty cycle limitations. In addition, the gateway, as compared to p - CARMA, does not transmit any feedback to the nodes apart from the established MAC messages from LoRaWAN. Therefore gateways can, instead of performing expensive beaconing as with LoRaWAN class-B devices, save transmission time to use it in more useful applications such as firmware updates of LoRa nodes over-the-air.

Finally, by analyzing data from real LoRa networks we have demonstrated the correctness of the design choices reasoned prior to extensively simulating the MAC protocol. The potential of developing new applications for industry, academia, civil sectors with LoRa has been shown in an example with traffic that a social distancing monitoring tool could be generating. If the research community continue joining efforts to consider LoRa as connectivity solution, I am confident that the scalability problem in LoRaWAN would be solved.

Chapter 8

Future Work

There exist possible technical limitations with regards to the implementation of SFMAC in current LoRa nodes. An example is the switching time from one spreading factor to the spreading factor used for the control channel. This switching time could be negligible or could induce latency in the network. Also, it has been considered a minimum listening time for obtaining a successful CAD, and in a real LoRaWAN deployment the value could be lower resulting in faster actions coming from the nodes. In order to evaluate the impact of these practical limitations, future work should be directed towards a real implementation of SFMAC in LoRa radios. In addition, a more flexible schema of SFMAC considering variable packet sizes could help to generalize the algorithm developed.

Regarding the study of the PHY layer of LoRa, a closed mathematical expression including the results of the experiments could be obtained. It is a challenge considering that the input for determining the survival probabilities in case of overlap in time depends on the RSSI values. The standard deviation of RSSI values is normally high since interference and fading modifies the received signal strength. Despite the challenge, it could help to implement the capture effect in LoRa in an easier manner.

As a whole, the project could benefit from more extensive real-life testing. However, having set in the present document the basis and most optimal parameters can serve as starting in future research.

Bibliography

[1] B200 mini. https://kb.ettus.com/B200/B210/B200mini/B205mini. [Online; accessed: 2020-09-16].

[2] Chirpstack open-source lorawan® network server. https://www.chirpstack.io/.

[3] gnuradio. https://github.com/gnuradio/gnuradio. [Online; accessed: 2020-07-19].

[4] Gurobi solver. https://www.gurobi.com/.

[5] Lite gateway. https://wireless-solutions.de/products/lora/development-tools/lite-gateway/. [Online; accessed: 2020-09-14].

[6] LoRa Channel Activity Detection (CAD) with SX126x. https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000Q1EX/pKB1DjG1BOAF0cdgLeDqC1JurOhnVUlLcUH3brdO504. [Online; accessed: 2020-08-14].

[7] Lora enables millions of iot devices in latin america. https://blog.semtech.com/lora-technology-enabling-millions-of-iot-devices-in-latin-america.

[8] LoRaWAN 1.1 Specification. https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf. [Online; accessed: 2020-10-10].

[9] Semtech SX1261 Long Range Low Power LoRa RF Transceiver. https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1261.

[10] The things network. https://www.thethingsnetwork.org/.

[11] Norman Abramson. The aloha system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70 (Fall), page 281–285, New York, NY, USA, 1970. Association for Computing Machinery.

[12] Michiel Aernouts, Raf Berkvens, Koen Vlaenderen, and Maarten Weyn. Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas. *Data*, 3:13, 04 2018.

[13] D. Bankov, E. Khorov, and A. Lyakhov. Mathematical model of LoRaWAN channel access with capture effect. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5, 2017.

[14] Norbert Blenn and Fernando A. Kuipers. Lorawan in the wild: Measurements from the things network. *CoRR*, abs/1706.03086, 2017.

[15] Martin C. Bor, Utz Roedig, Thiemo Voigt, and Juan M. Alonso. 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*, MSWiM '16, pages 59–67. ACM, 2016.

[16] J. M. d. S. Sant'Ana, A. Hoeller, R. D. Souza, H. Alves, and S. Montejo-Sánchez. Lora performance analysis with superposed signal decoding. *IEEE Wireless Communications Letters*, pages 1–1, 2020.

[17] Ben D. Fulcher, Max A. Little, and Nick S. Jones. Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of The Royal Society Interface*, 10(83):20130048, 2013.

[18] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. Lmac: Efficient carrier-sense multiple access for lora. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.

[19] Branden Ghena, Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity. Association for Computing Machinery, 2019.

[20] Claire Goursaud and Jean-Marie Gorce. Dedicated networks for IoT : PHY / MAC state of the art and challenges. *EAI endorsed transactions on Internet of Things*, October 2015.

[21] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke. LoRa Scalability: A Simulation Model Based on Interference Measurements. *Sensors*, 2017:1193, 05 2017.

[22] J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *The 18th IEEE International Conference on Network Protocols*, pages 305–314, 2010.

[23] L. Kleinrock and F. Tobagi. Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, December 1975.

[24] Linghe Kong and Xue Liu. Mzig: Enabling multi-packet reception in zigbee. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, page 552–565, New York, NY, USA, 2015. Association for Computing Machinery.

[25] Nikolaos Kouvelas, Vijay S Rao, R. Venkatesha Prasad, Gauri Tawde, and Koen Langendoen. P-carma: Politely scaling lorawan. USA, 2020. Junction Publishing.

[26] Nikos Kouvelas, Vijay Rao, and R. R. Venkatesha Prasad. Employing p-CSMA on a LoRa Network Simulator. *CoRR*, abs/1805.12263, 2018.

[27] Jeongkeun Lee, Wonho Kim, Sung-Ju Lee, Daehyung Jo, Jiho Ryu, Taekyoung Kwon, and Yanghee Choi. An experimental study on the capture effect in 802.11a networks. In *Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WinTECH '07, page 19–26, New York, NY, USA, 2007. Association for Computing Machinery.

[28] F. Li, J. Luo, G. Shi, and Y. He. Art: Adaptive frequency-temporal co-existing of zigbee and wifi. *IEEE Transactions on Mobile Computing*, 16(3):662–674, 2017.

[29] A. Shahid M. Saelens, J. Hoebeke and E. De Poorter. Impact of eu duty cycle and transmission power limitations for sub-ghz lpwan srds: an overview and future challenges. *EURASIP Journal on Wireless Communications and Networking*, 2019.

[30] Davide Magrin. Network level performances of a lora system. 2016.

[31] Congduc Pham. Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks. pages 1–6, 04 2018.

[32] Andri Rahmadhani and Fernando Kuipers. When LoRaWAN Frames Collide. In *Proceedings of the 12th International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WiNTECH '18, page 89–97. Association for Computing Machinery, 2018.

[33] Olivier Bernard Andre Seller and Nicolas Sornin. Low power long range transmitter, European Patent Office, EP2763321A1, Feb. 2013.

[34] T. To and A. Duda. Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018.

[35] T. To and A. Duda. Timemaps for improving performance of lorawan. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–7, 2020.

[36] F. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.

[37] S. Tong, Z. Xu, and J. Wang. Colora: Enabling multi-packet reception in lora. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 2303–2311, 2020.

[38] Shuai Tong, Jiliang Wang, and Yunhao Liu. Combating packet collisions using non-stationary signal scaling in lpwans. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, MobiSys '20, page 234–246, New York, NY, USA, 2020. Association for Computing Machinery.

[39] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect for collision detection and recovery. In *The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II.*, pages 45–52, 2005.

[40] X. Xia, ACM, Y. Zheng, and T. Gu. Ftrack: Parallel decoding for lora transmissions. *IEEE/ACM Transactions on Networking*, pages 1–14, 2020.

[41] Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill, and Ümit Özgüner. A vision-based social distancing and critical density detection system for covid-19, 2020.

[42] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878, 2012.

Appendix A

Appendix A

The reason for claiming a difference between the transmissions collected in a specific urban scenario in Zaragoza and TU Delft can be explained by the satellite images collected by the Things Network [10] in both scenarios where data was gathered. The assumption made is that there is a direct relationship between the number of public gateways in an area and the number of users in a LoRa network in that same area. In both Fig. A.1 and Fig. A.2. Another explanation is the type of scenario: whereas in the city the most likely application for the LoRa devices is a non intensive monitoring activity, in a University scenario experiments with more intensive traffic may be carried out.

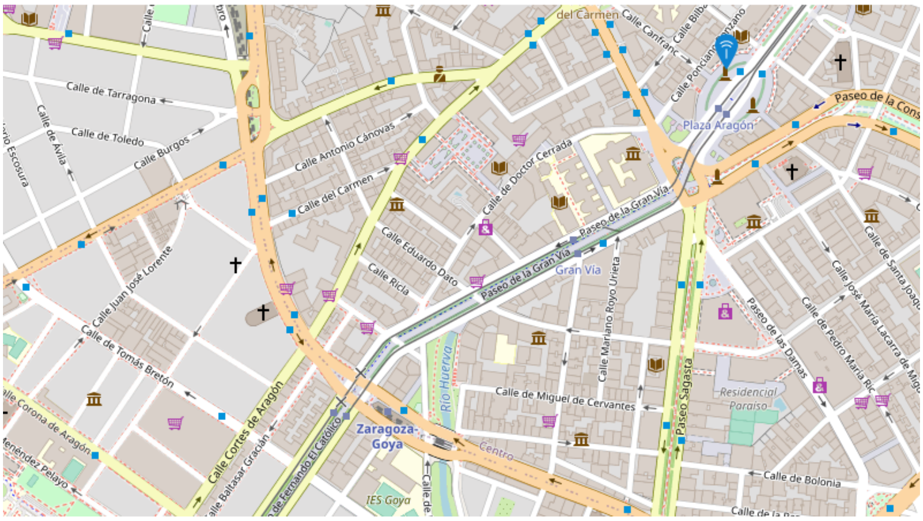


Figure A.1: **Location of The things network public gateways in area of Zaragoza where transmissions were collected, [10]**

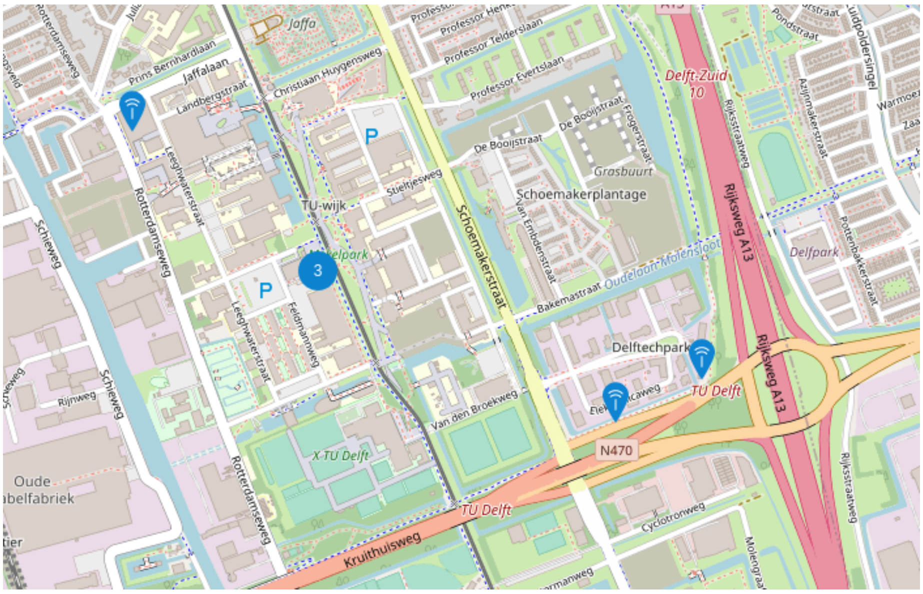


Figure A.2: **Location of The things network public gateways in area of TU Delft where transmissions were collected, [10]**