



HAL
open science

Joint Load-Balancing and Power Control Strategy to Maximize the Data Extraction Rate of LoRaWAN Networks

Mohamed Hammache, Rahim Kacimi, André-Luc Beylot

► To cite this version:

Mohamed Hammache, Rahim Kacimi, André-Luc Beylot. Joint Load-Balancing and Power Control Strategy to Maximize the Data Extraction Rate of LoRaWAN Networks. *Computer Networks*, 2023, 225, pp.103025. <10.1016/j.comnet.2023.109633>. <hal-03983043>

HAL Id: hal-03983043

<https://hal.science/hal-03983043v1>

Submitted on 31 Mar 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Joint Load-Balancing and Power Control Strategy to Maximize the Data Extraction Rate of LoRaWAN Networks[☆]

Mohamed Hammache^{*,a}, Rahim Kacimi^a, André-Luc. Beylot^a

^a*IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France*

Abstract

LPWAN enabled networks have a dizzying growth and continue to meet an essential need in the Internet of Things market due to their ability to provide low-cost wireless access to geographically spread-out devices. Consequently, an efficient allocation of wireless resources in order to support numerous devices becomes a major concern. In this paper, we propose an SF assignment approach in LoRa networks, paying attention on the traffic load both per Spreading Factor and over the channels. Indeed, our strategy consists in finding a better distribution of the end-devices on the SF by orchestrating an effective load balancing. Moreover, the performance of our solution is evaluated under diverse network configurations, taking into account the capture effect and the non-orthogonality of SFs. Furthermore, we extended the solution by a transmission power control strategy for overall system energy-efficiency. In addition, we validated some assumptions by full-scale experiments like for the 3GPP path loss model, which is used for the first time in LoRa simulations. Our results suggest that Load Shifting leads to better performance in terms of Data Extraction Rate (DER) while guaranteeing good scalability on the network size and density.

Key words: LoRa networks, Spreading Factor allocation, Load Balancing, Transmission Power Control, Urban Path Loss Model, test-bed.

1. Introduction

LPWANs (Low-Power Wide Area Networks) are characterized by power-efficient transmissions, low-rate and wide-range connectivity. Backhauled to the Internet, they are supposed to take us into a fascinating new decade of the next generation of the Internet of Things (IoT). In fact, nowadays, it is estimated that more than 20% of IoT connection are handled by LPWANs which represents billions of connected devices.

Therefore, in order to address the requirements for large-scale deployments, the channel usage was regulated by applying a duty cycle of 1% and the bit rates were minimized. As an example, a LoRa gateway is supposed to handle traffic from thousands of end-devices and even using a low duty-cycle, transmission issues such as collisions are still observed, especially due to the generated traffic load and the number of end-devices using the same spreading factor.

LoRaWAN networks[1] have newly gained a lot of interest in the wireless communications community because of its great potential in terms of the high density of end-device around one single gateway, energy efficiency, long-range capabilities and flexible deployments. All of these capabilities have made it more appropriate for a variety of IoT applications, such as habitat monitoring, smart cities, smart lighting control and smart farming and agriculture.

A transmission between the end-devices and the gateways in LoRaWAN networks is configurable by a set of parameters. These parameters become crucial and have a significant impact on the communication quality in terms of network capacity to handle end-devices, collision rate, data extraction rate, etc. Indeed, the selection of a certain combination of parameters instead of others can enhance or degrade one or more communication quality criteria mentioned above. Among all these parameters, the spreading factor is one of the most important parameters that has a direct impact on the overall throughput. There is no doubt that the choice of the right Spreading Factor (SF) to send a frame strongly contributes to minimize co-SF and inter-SF interference and allows the coexistence of several end-devices in the same region without significantly deteriorating the network quality.

In this paper, we explore a different strategy for spreading factor allocation under different load scenarios for LoRaWAN networks. We argue for the use of a load control and shifting among the different SFs. Therefore, we propose a novel strategy called L3SFA, which enhances the global Data Extraction Rate (DER). The L3SFA strategy takes into account the global system load in terms of both generated traffic and the number of end-devices to allocate the SFs to end-devices. Its main objective is to balance the load on different SFs classes while providing a better DER.

The validation approaches of our solution are twofold: in the first approach, a uniform generation of end-devices was considered. The originality of this evaluation, in addition to the proposed SF allocation strategy, is the utilization of the 3GPP path loss model to describe link behaviour. For the best of our

[☆]This research is supported by “neOCampus”, an operation for research and innovation of the University of Toulouse.

*Corresponding author: Mohamed Hammache.

Email address: Mohamed.Hammache@irit.fr (Mohamed Hammache)

knowledge, this is the first time where such a model is used for LoRaWAN networks. To address some of the simulation limitations, we also considered a second approach. Indeed, emulated end-devices are generated based on frames retrieved from a LoRaWAN dataset gathered during real deployment to feed our simulations. Finally, to demonstrate the feasibility of our solution, we have designed and implemented a flexible and scalable end-to-end LoRaWAN testbed for real experiments.

Furthermore, we extended L3SFA by a transmission power control strategy and came up with L3SFA-TPC which decreases the overall energy consumption and keeps the same performance improvements in terms of DER.

The remainder of the paper is structured as follows: Section 2 critically surveys the related work. In section 3, we describe the LoRaWAN features and basics. We then detail in Section 4 our system model for SF allocation. We, then, proposed a load shifting strategy for SF allocation heuristics in section 5. Section 6 presents an experimental platform in order to validate the link behaviour and provides a thorough performance analysis of the L3SFA scheme. Apart from simulations, an experimental deployment in order to provide valuable insights and to show the performance of a real LoRaWAN system is given. We introduce L3SFA-TPC, an extension to L3SFA with a transmission power control algorithm, in Section 7. In Section 8 we further discuss some key practical issues and reveal their potential weaknesses. Section 9 concludes this paper and gives future research perspectives.

2. Related Work

Research has been particularly productive in recent years on the best combination of transmission parameters in LoRaWAN networks. Undoubtedly, a good setting of the coding rate, the spreading factor, and the transmission power level would lead to a high overall DER and an efficient utilization of the channels. In [2], [3], and [4] authors investigate the densification of LoRaWAN networks by focusing on how to increase the network capacity in single-gateway-centric deployments. Other studies investigate the optimization of a specific performance criteria by new spreading factor allocation schemes. Several approaches were envisioned to come up with the following solutions:

- Algorithms like EXP-SF and EXP-AT in [5] in order to increase the overall data rate, EXPLoRa-TS proposed in [6] perform traffic load equalization among the spreading factors while considering that each end-device, transmits a variable amount of data according to its application profile.
- Optimization problem formulation in [7], [8] where authors modelled LoRaWAN resource allocation as a maximization or minimizing problem solved using meta-heuristics such as Genetic Algorithm and learning methods such as Reinforcement Learning.
- Applied machine learning methods in [6], [7], [9]. Several data science techniques were considered. For instance, k-mean supervised learning has been applied in [6] and

[9] to find suitable clusters for SF allocation. Other papers are based on reinforcement learning, where a gateway plays the role of an agent to control the environment (end-devices) by sending actions (configurations). Convolutional neural networks were also used to solve optimization problems in [7].

These articles either focus on the performance evaluation of LoRaWAN networks such as [2], [3] without providing new mechanisms or solution to enhance their performance. In fact, most of these papers address the issues of network densification, communication coverage, and energy efficiency in order to determine the lifetime of batteries, but they do not investigate on fundamental LoRa issues. Indeed, they:

- neglect in the mathematical model or simulation some LoRa physical phenomena which have a direct impact on LoRaWAN network performance such as capture effect (Intra-SF Interferences) and imperfect SF orthogonality (Inter-SF Interferences) [5], [6], [8].
- mainly use Log-Distance path loss model [7], [10], [11] with a path loss at a distance $d_0 = 40$ meters equals to $127.41dB$. This is a penalizing assumption and can provide only deployment around a gateway with short distances, which contradicts the main purpose of LPWAN technologies.
- weakly take into account the impact of other important factors, particularly high traffic generation rate and large networks sizes (number of devices in high-load LoRaWAN networks [12]).

After taking this step back to review the proposed solution, we decided to revisit the Spreading Factor allocation issue in order to investigate a new strategy for high loaded LoRaWAN systems particularly. Indeed, in this work, we take into account high traffic generation rates and large network sizes. In addition, we turned attention to the well-known 3GPP model in order to highlight how LoRa links behave in LoRa networks. So far, to our knowledge, this is unseen in such networks, but we validated this choice of path loss model by a test-bed.

3. Basics of LoRa and LoRaWAN

This section gives a brief overview on LoRa physical layer and LoRaWAN Medium Access layer.

3.1. LoRa: Physical Layer

LoRa (Long Range) is a proprietary Chirp Spread Spectrum (CSS) modulation developed by Semtech in 2004 and incorporated as the physical layer of the LoRaWAN technology. This modulation technique is behind the main characteristics offered by the LoRaWAN networks. Indeed, these features can be given as follows: long-range distances and low-power transmissions, high robustness and Doppler resistance. The transmission quality using the LoRa modulation is determined by the optimal combination of a set of transmission parameters detailed in the next section.

Table 1: LoRaWAN transmission parameters

Parameters	Values
Spreading Factor	7, 8, 9, 10, 11, 12
Transmission Power	2, 5, 8, 11, 14 [dBm]
Bandwidth	125 kHz, 250 kHz, 500 kHz
Coding Rate	4/5, 4/6, 4/7, 4/8
Carrier Frequency	886.1, 886.3, 886.5, ...

3.2. Transmission Parameters

A radio transmission in LoRaWAN can be characterized by a combination of a set of configuration parameters. We can mainly define five parameters [4][13] which have a direct impact on energy consumption, transmission range and robustness. See table 1.

3.2.1. Spreading Factor (SF)

LoRa is a Chirp spreading spectrum modulation. The Spreading Factor is the parameter which controls how these chirps are spread. A lower spreading factor decreases the Signal to Noise Ratio (SNR), and thus sensitivity and range, but also decreases the airtime of the frame. The spreading factor values are between 7 and 12 and the choice is made according to a trade-off between the energy consumption and the robustness.

3.2.2. Transmission Power (TP)

LoRa allows sending frames with different transmission powers in the range [2dBm-14dBm] for the European region. The use of a high TP enhances the probability of frame reception. However, it increases the power consumption of the transmitter and the interference level.

3.2.3. Bandwidth (BW)

it represents the width of a continuous band of frequencies. Higher bandwidth results in a shorter airtime but a lower sensitivity. LoRa is configurable with three possible values of the bandwidth [125 kHz, 250 kHz, 500 kHz] for both Uplink and Downlink.

3.2.4. Coding Rate (CR)

CR is the forward error correction code added to a frame before transmission to offer protection against bursts of interference. LoRa uses four coding rates (4/5, 4/6, 4/7, 4/8). Undoubtedly, using higher CR provides more immunity but increases the time-on-air and consequently the energy consumption.

3.2.5. Carrier Frequency (CF)

it is the center frequency and LoRa uses a multi-channel mechanism for uplink and downlink transmissions. Data rate and communication range are determined by the SF. In fact, SF indicates the received signal strength thresholds. Thus, higher SF values allow receiving frames with lower received signal power, however data rate is reduced. Let us consider the frequency bandwidth BW . Then the chirp length T_c in seconds is given by:

$$T_c = \frac{1}{BW} \quad (1)$$

Thus, the symbol length in seconds is given by:

$$T_s = T_c \times 2^{SF} \quad (2)$$

3.3. LoRaWAN MAC Layer

LoRaWAN is an open-source medium access control protocol [14] standardized by the LoRa Alliance that runs on top of the LoRa physical layer. It provides a set of mechanisms to handle the communication between end-devices and gateways. A LoRaWAN network is based on a star-of-stars topology, as illustrated in Figure. 1). Indeed, end-devices can only send frames to one or more gateways. At reception, the gateways that are connected to a centralized Network Server (NS), forward the received frames to this NS. The NS is in charge of managing the end-devices and gateways, sending downlink frames and MAC commands to the end-devices, if necessary. Finally, the communication ends on the right application server. LoRaWAN layer offers a set of features and mechanisms in order to provide secure and robust transmissions. It provides an Adaptive Data Rate (ADR) mechanism to remotely and dynamically control the end-devices transmission parameters in order to improve the data extraction rate.

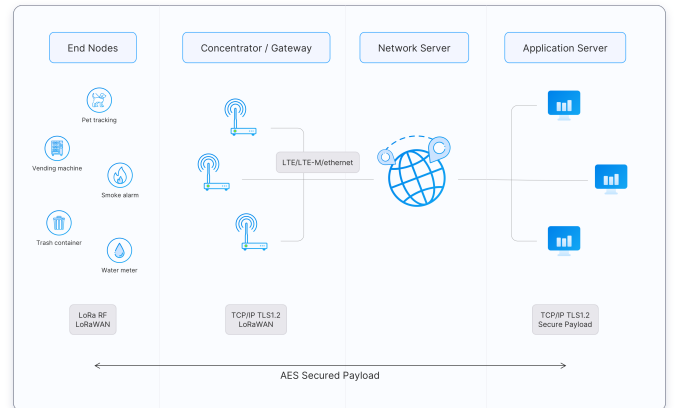


Figure 1: LoRaWAN network architecture [15]

4. SF Allocation System Model

We consider a LoRaWAN network composed of N end-devices around one single gateway. Each end-device is initialized with $SF = i$ according to the Received Signal Strength Indicator (RSSI) and the Signal-to-noise Ratio (SNR) thresholds provided in Table 3. To allocate a Spreading Factor in the range of SF7 to SF12, for a given end-device, we define an accurate load balancing scheme.

4.1. SF shifting

Initially, end-devices are allocated $SF = j$ following the thresholds defined in Table 3 with $j \in \{7, 8, 9, 10, 11, 12\}$. Let $M_{j,i}$ be the number of end-devices which are able to use $SF = j$ but shifted to $SF = i$ and $M_{j,j}$ the number of end-devices that keep their $SF = j$ unchanged.

Table 2: System model variables

Variable	Description
N	Number of end-devices
N_i	Number of end-devices using $SF = i$
T_i	Time On Air using $SF = i$
p	Period of frame generation (per end-device)
$M_{j,i}$	End-devices shifted from $SF = j$ to $SF = i$
$M_{j,j}$	End-devices with unchanged $SF = j$

For ease of presentation, we denote the total number of end-devices that have been assigned a SF i , after the shifting procedure, as M_i . The shifting process is depicted in figure 3 and M_i can be computed as:

$$M_i = \sum_{j=7}^i M_{j,i} \quad (3)$$

4.2. SF Load

Let ρ_i be the load for each $SF = i$ after the SF shifting phase.

$$\rho_i = \frac{M_i \times T_i}{p} \quad (4)$$

ρ_i is computed using equation (4), where T_i is the time-on-air using $SF = i$ and p is the frame generation period per end-device.

Consequently, the Data Extraction Rate DER_i achieved for $SF = i$ with a load ρ_i can be calculated as follows:

$$DER_i = e^{-2\rho_i} \quad (5)$$

4.3. Optimal SF Allocation

We define the SF allocation problem as an optimization problem. More precisely, We consider the maximization problem described in equations (6a-6d).

$$\text{maximize} \quad \frac{1}{N} \sum_{i=7}^{12} DER_i \times M_i \quad (6a)$$

subject to

$$M_{j,i} \in \mathbb{N}, \quad (6b)$$

$$\sum_{j=i}^{12} M_{j,i} = N_i, \quad (6c)$$

$$M_{j,i} = 0 \quad \text{If } j > i \quad (6d)$$

The objective function given by the equation (6a) intends to maximize the global DER as a function of the DER_i of each $SF = i$ and the number of end-devices using this SF once the load shifting phase is achieved.

Equation (6b) makes a strong constraint to the optimization problem. Indeed, the number of end-devices shifting their SF must be a positive integer. The shifting process is possible only

if equation (6d) is satisfied. Indeed, shifting SF is one way direction (from lower to higher SF), the other way is not possible. The shifting process is controlled using equation (6c). This constraint guarantees that the total number of end-devices shifted from $SF = i$ to higher ones must be equal to the number of end-devices using $SF = i$ before the shifting process. This insures that $\sum_{i=7}^{12} M_i = N$.

The optimization problem presents a hard complexity. Indeed, on the one hand, we are facing an Integer Non-Linear Programming (INLP)[16] and on the other hand, the network size network is highly scalable. Undoubtedly, an exhaustive search is not feasible due to the huge number of combinations, and no polynomial-time algorithm is known to be applicable in a general case, especially when the problem under optimization does not present a linear or convex form.

5. Load Shifting and SF Allocation Heuristics

In this section, we present a Load Shifting Strategy for Spreading Factor Allocation (L3SFA). The proposed Scheme is designed by default to optimally allocate SF to end-devices to achieve better DER. This is achieved by first allocating the smallest possible SF to the end-devices, while complying with the SNR and RSSI thresholds defined for each SF needed to receive and decode the frame at the gateway. This results in a minimum time on air for the frames. As an example, terminals that generate frames with an $RSSI > -126.5$ and an $SNR > -7.5$ are assigned a $SF = 7$ while for $-127.25 < RSSI < -126.5$ and $-10.5 < SNR < -7.5$ a $SF = 8$ is assigned and so on for the other SF.

It is clear that such an allocation is advantageous, but can cause significant degradation of the data extraction rate due to the increase in the number of collisions if no control is performed on the load generated by each class of end-devices per SF. Indeed, several end-devices may be assigned the lowest SFs and few terminals may use the highest SFs. It is therefore necessary to consider a mechanism for adjusting the SF allocation to ensure a load balancing in the traffic generated by the various end-devices using the same SF. To address all of these considerations, we proposed L3SFA. This SF allocation strategy follows three main phases (See Algorithm 1) described and detailed in the following sections : 1) End-Device Initialization, 2) Spreading Factor Assignment, 3) Spreading Factor Shifting.

Algorithm 1 L3SFA

Input : N : End-devices

Output : End-devices with allocated SF, SF Repartition.

- 1: End-devices Initialization
 - 2: Spreading Factor Assignment
 - 3: Spreading Factor Shifting
-

5.1. End-Device Initialization

L3SFA starts with an initialization phase. Note that two actions are taken: 1) Since the SF allocation mechanism is incremental, all end-devices are initialized with $SF = 7$. 2) End-devices are sorted according to their distance from the gateway

Table 3: SNR and Sensitivity thresholds for 125 kHz Bandwidth in [dB]

SF	SNR	Sensitivity
7	-7.5	-126.5
8	-10	-127.25
9	-12.5	-131.25
10	-15	-132.75
11	-17.5	-133.25
12	-20	-134.5

or their RSSI values. Obviously, this intuitive action allows assigning lower SF on priority basis to the closer end-devices to the gateway or to the ones with the highest RSSI value. This is detailed by the lines [1,2] in Algorithm 2.

5.2. Spreading Factor Assignment

After the initialization, L3SFA strategy begins the second phase of spreading factor pre-allocation. SF_LoRa Algorithm (lines 3 to 12) goes through the list of sorted end-devices produced during the first phase and for each end-device a suitable SF is assigned a function of the SNR and the RSSI values received on the gateway. These values are compared to RSSI and SNR thresholds given in table 3. The SF allocation condition defined in line 7 is necessary and sufficient to guarantee a successful transmission. Undoubtedly, the RSSI higher than the sensitivity insures the frame reception by the gateway and combined to SNR higher than the SNR threshold, the frames will be certainly decoded.

5.3. Spreading Factor Shifting

The SF allocation performed in phase two (SF Assignment) is not final. This allocation is controlled by the last phase of the L3SFA mechanism.

Before assigning definitely a SF to an end-device, SF_LoRa performs a load control on SF distribution (line 13). This control is performed with SF_Shifting heuristic.

Algorithm 3 and Fig. 2 give a description of the SF adjustment mechanism. SF_shifting needs to have a global knowledge on the SF allocation system to perform the adjustments. Indeed, it takes as input a preassigned SF to each end-device, the SF_Distribution (a vector of number of end-devices using $SF = i$ for $i \in \{7, 8, 9, 10, 11, 12\}$) and the loads (a vector of number of end-devices threshold for each $SF = i$) computed using equation 4.

As shown in figure 2, SF_shifting performs a simple checking to decide which SF to assign definitely. Indeed, after that an end-device have been assigned a SF, SF_shifting checks :

1. If the class of the current $SF = j$ (allocated in SF allocation phase) is not overloaded then SF_shifting keeps using $SF = j$.
2. If the class of the current $SF = j$ is overloaded and all classes of $SF = i, i > j$, then SF_shifting keeps using $SF = j$.

3. If the class of the current $SF = j$ (allocated in SF allocation phase) is overloaded and one of the upper classes $SF = i, i > j$ is not, then SF_shifting upgrades $SF = i$ to the first SF class not overloaded.

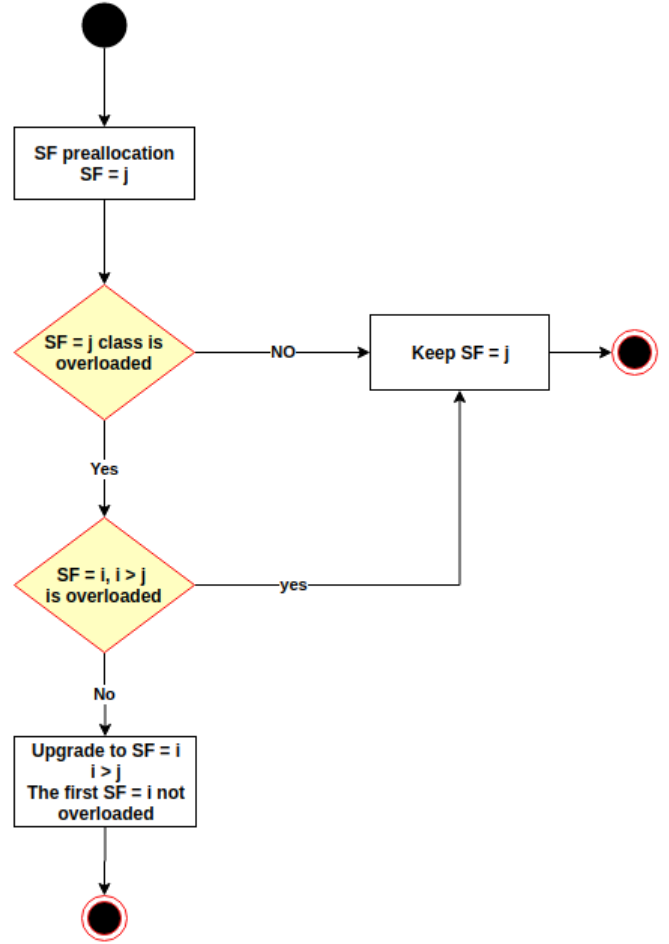


Figure 2: SF Shifting Mechanism Activity Diagram

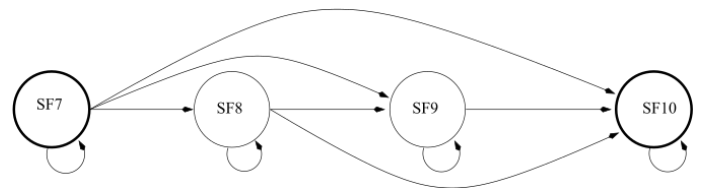


Figure 3: Load Shifting Strategy for Spreading Factor Allocation Finite State Machine (Algo 3) (only SF7 to SF10 are considered)

Without loss of generality, we illustrate in figure 3 how the SF adjustment scheme behaves by giving an example.

Let consider a simple system using only four SF values (from 7 to 10) where N end-devices are sending frames to a single gateway. We suppose that SF Allocation phase allocates $SF = 7$ to an end-device k . SF_Shifting will check if this allocation is appropriate, or it will be better to upgrade it. The finite state machine in Fig. 3 describes such a situation, an end-device with $SF = 7$ is facing two possible situations:

- conserves $SF = 7$ if the class of end-devices this SF is not overloaded or yes it is but all the other classes are also overloaded.
- Upgrades to $SF = j$ with $j > 7$ if the class of end-devices using $SF = 7$ is overloaded and the classes of end-devices using $SF = j$ is not overloaded ($j \in \{8, 9, 10\}$ the first SF not overloaded)

Using such a strategy guarantees that each end-device will be assigned the suitable SF, which is qualified as good for two main reasons: 1) SF allocation in phase two insures that a frame will be received and can be decoded if no-collision happens. 2) SF adjustment in phase three controls the SF load, which reduces significantly the number of collisions. As expected, shifting some end-devices when possible to higher SF reduces the capture effect phenomena. Otherwise, having lower SF classes overloaded is less penalizing in terms of DER than overloaded classes with higher SF. This is due to time-on-air of the frames, which increases by increasing SF and consequently increases the collision probability.

Algorithm 1 is composed of the two algorithms 2 and 3 and has a linear execution time, its complexity is given as follows: $O(\alpha N)$, where N is the number of end-devices and α a constant.

6. L3SFA Performance Analysis

In this section, we discuss the performance of the spreading factor allocation strategy through in-depth simulations using implementation of the algorithms detailed in section 4.

In addition to designing the L3SFA strategy, we also developed a simulation tool called LoRa-L3SFA to demonstrate the efficiency of the proposed strategy. LoRa-L3SFA simulator is based on the both: the Simpy, a process-based discrete-event simulation framework based on standard Python, and a path loss model based on the 3GPP Urban Macro model. Moreover, the simulator is also based on LoRaSim [4] and LoRaFREE [10]. In fact, to emphasize the feasibility of SF allocation strategy on LoRaWAN networks, we enhanced LoRaSim with new feature given as follows:

- It considers the imperfect orthogonality of spreading factors and the fading impact for frame collisions check.
- It integrates a 3GPP propagation model for random generation end-devices. To the best of our knowledge, there is no other work that uses this model in LoRa performance studies.
- It also considers emulated end-devices, as inputs, realistic RSSI and SNR values from an experimental recorded dataset are used to generate end-devices.

6.1. Path Loss Model

We consider that a transmission in LoRaWAN networks is successfully received at the gateway, as well as the received

signal strength P_{rx} is greater than the receiver sensitivity. P_{rx} is given by the equation 7.

$$P_{rx} = P_{tx} - PL + GL \quad (7)$$

where P_{tx} is the transmission power, PL is the path loss and GL combines all general gains and losses.

1

Algorithm 2 SF_LoRa

Input : N : end-devices

Output : N with SF allocated, SF_Distribution

```

1: Sort end-devices by distance  $d$  to the gateway;
2: Initialize all end-devices with SF =7 and SF_Distribution to zeros
3: for  $n$  in  $N$  do
4:   SF = n.sf
5:   Get SNR and RSSI thresholds for the corresponding SF from the table 3
6:   for  $s$  in range (SF,12) do
7:     if  $n.snr < SNR_{sf}$  or  $n.rssi < sensibility_{sf}$  then
8:       if  $n.sf < 12$  then
9:          $n.sf = n.sf + 1$ 
10:      end if
11:     Get SNR and RSSI thresholds for the corresponding SF from the table 3
12:   end if
13:    $n.sf = SF\_Shifting(n.sf, SF\_Distribution)$ 
14: end for
15: SF_Distribution[n.sf] = SF_Distribution[n.sf] + 1
16: end for

```

Algorithm 3 SF_Shifting

Input : SF, SF_Distribution, load

Output : SF

```

1: for  $s$  in range (SF,12) do
2:   if SF_Distribution[s] < load[s] then
3:     return SF
4:   else
5:     if SF_Distribution[s]  $\geq$  load[SF] and SF_Distribution[s+1] < load[s+1] then
6:       if  $s < 12$  then
7:         return s+1
8:       end if
9:     end if
10:   end if
11: end for

```

We use in this work the 3GPP Urban Macro path loss model [17] which is commonly used to model deployments in urban macro-cells areas. We choose this model as it matches the environments where LoRa networks are expected to be deployed. Using this path loss model depends on the communication distance d and can be described as:

$$\begin{aligned}
PL &= 44.9 - 6.55 \times \log_{10}(GH) \times \log_{10}\left(\frac{d}{1000}\right) \\
&+ 45.5 \\
&+ 35.46 \times 1.1 \times NH \times \log_{10}(f_c) \\
&- 13.82 \times \log_{10}(GH) \\
&+ 0.7 \times NH \\
&+ \kappa
\end{aligned} \tag{8}$$

where GH is the gateway antenna height in meters, NH the end-device antenna height in meters, f_c the carrier frequency in MHz , d is the distance between the gateway and the end-device in meters, and κ is a constant factor ($\kappa = 0$ dB for suburban macro and $\kappa = 3$ dB for urban macro).

6.2. Experimental Platform For Link Behaviour

In order to better understand how the RSSI behaves according to the distance between the end modules and the LoRa Gateway, we conducted the following experimental study. The objective is also to validate and support the chosen path loss model. On the hardware side, we used for the experiments a MultiTech Conduit Gateway and a RN2483 microchip end-device scratched on a Raspberry Pi3. In addition, we equipped this platform with a Xbee GPS module (See Fig. 4). The LoRa end-device is attached to a bicycle and transmits the GPS coordinates every 20 seconds under the following parameters: $SF = 12$, $BW = 125kHz$, $Ptx = 14dBm$ and $CR = 4/5$.

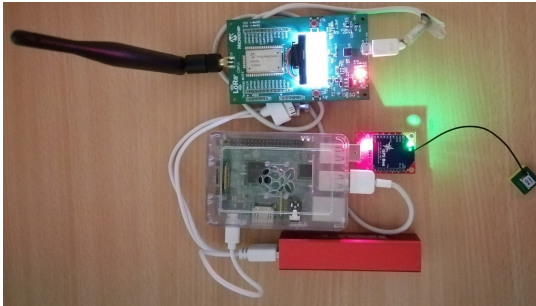


Figure 4: Experimentation Setup

Fig. 5 shows the experimentation circuit traced using the GPS coordinates sent by the RN2483 microchip end-device. The aim of this experiment is to compare the propagation models with measurements from a real environment and using a real hardware. The results of this comparative study are represented in Fig. 6.

As all the path loss models depend on the distance d between the end-device and the base station, we plotted the evolution of the RSSI as a function of the distance for log-distance model, 3GPP models and RSSI values from experimentation (distances are computed using GPS coordinates). We can see that all the curves have the same appearance. In fact, the value of the RSSI decreases while moving away from the gateway.

$$PL = PL(d_0) + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + \chi \tag{9}$$



Figure 5: GPS Coordinates Transmitted With RN2483 Equipped With Xbee GPS Module

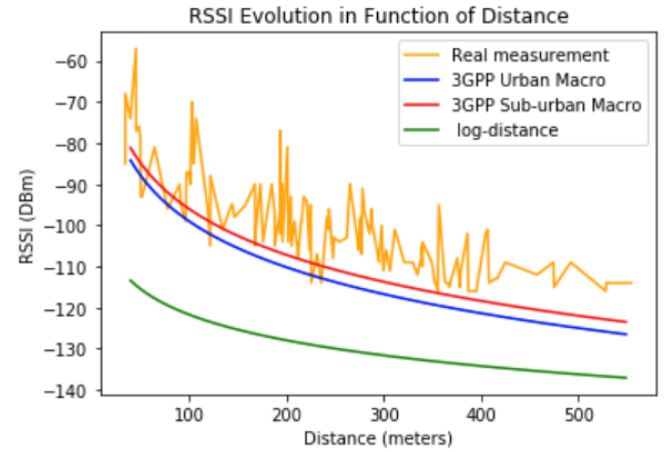


Figure 6: Received Signal Strength Indicator: Mathematical Models vs Real Measurements

Log-distance model (equation 9) is very penalizing and gives overestimated RSSI values compared to the empirical results obtained through the experiments. This is due to an overestimation of the path loss value at a reference distance d_0 , where $PL(d_0)$ at distance $d_0 = 40$ meters is supposed to be $127.41dB$. γ is the Path Loss Exponent with a value of 2.08 and χ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ . This variable is used only when there is a shadowing effect. If there is no shadowing effect, then this variable is zero (we suppose that $\chi = 0$).

3GPP models as shown in Fig. 6 are suitable for LoRaWAN networks, we notice that the 3GPP models are not too penalizing in terms of path loss and reflects results obtained by experimentation.

As a conclusion, we decide to use 3GPP Urban macro in our simulations because it gives RSSI values which correspond to real environments where LoRaWAN networks are the most adopted.

6.3. Simulation Approaches

We simulate and compare results from L3SFA Load Shifting for SF Allocation approach and SF Allocation without load control (the allocation derived only by the LoRaWAN specifications). In all scenarios, we consider an application that generates frames of 20 Bytes according to a Poisson process. We also assume that generation period p is the same for all end-devices. Before the frame transmission phase, the L3SFA scheme starts and follows three different steps that we describe in section 4. Indeed, let remind that L3SFA initializes all end-devices with $SF = 7$ and sorts them according to the distance d or their $RSSI$, allocates the SFs while controlling the load generated on the different SF classes. Hence, these operations produce a set of end-devices with SF assignment. We simulate frames transmission by N end-devices using SFs allocated previously for 2 hours. We examine the scalability and the performance of our approach in terms of Data Extraction Rate (DER) metric. Indeed, in an effective LoRa deployment all transmitted messages should be received at the gateway level, the higher the DER the effective and efficient is the deployment. To achieve higher accuracy of simulation results, two kinds of approaches were considered:

6.3.1. Approach 1 - Uniform end-device generation

In this approach, the end-devices are randomly generated and positioned in a 2-dimensional space, forming a cell of $radius = 600$ meters. Several simulation runs are achieved for each scenario with different seed values in order to ensure the accuracy of the results.

Table 4: Simulation Configuration

parameters	Values
End-devices	100-10000 Randomly Scattered
BW	125 [kHz]
CR	4/5
TP (P_{tx})	14 [dBm]
SF	7, ..., 12
Channels	3 up-links [868.1, 868.3, 868.5]
Concurrent Receptions	8
SNR Thresholds	Table 3
Sensitivity	Table 3
Inter-SF and Intra-SF Interferences	Matrix in equation 10
p	100, 300, 600 (<i>second</i>)
Payload	20 [bytes]
Path Loss Model[17]	3GPP Urban Macro
End-device Height (NH)	1 [meter]
Gateway Height (GH)	15 [meter]

We run 2400 simulations for N end-devices, $N \in [500, 10000]$ with a step of 500, under different configurations. All simulation settings, propagation model variables, transmission and L3SFA parameters are summarized in table 4

To better understand how L3SFA behaves, we represent the spreading factor distribution for 5000 end-devices randomly generated and deployed in a cell of 600 meters radius in Fig. 7. We can see in Fig. 7a that initially, all end-devices are assigned a $SF = 7$. Thereafter, a SF reallocation according to SNR and

RSSI thresholds is performed, results are given in SF8, SF9 and non-use of SF10, SF11, SF12. Finally, Fig. 7c represents the SF adjustment considering a load of 0.5, we can see clearly that some end-devices shifted to higher SFs to lighten the load of the system.

For this approach, we plotted in Fig. 8d, to 8f the spreading factor distribution after SF allocation with and without load control for a LoRa network of 5000 end-devices. Similarly, we highlighted the impact of the Sf redistribution on providing better global DER in Fig. 8a to 8c. It is also worth noting that, the traffic load generated decreases by using respectively frame generation rate $\frac{1}{100}$, $\frac{1}{300}$, and $\frac{1}{600}$.

We observe from Fig. 8a to 8c that L3SFA leads to better results in terms of DER. Indeed, shifting some end-devices to higher SF lightens the system load and reduces collisions due to the capture effect, hence it increases LoRa network capacity.

For instance, for a frame generation ratio of $\frac{1}{600s}$ and to achieve a DER higher than 80%, L3SFA supports 8500 end-devices for just 6000 end-devices using L3SFA without SF adjustment phase (there is no SF classes load control).

Final SF Distribution is reported in Fig. 8g, 8h, 8i for $\rho = 0.5$, $\rho = 0.3$, $\rho = 0.2$ respectively for 5000 end-devices in a cell of 600 meters radius. Compared to Fig. 7b which represents SF distribution for 5000 end-devices before SF adjustment process, we note that some end-devices shifted to higher spreading factors which results in the formation of rings inside the cell.

6.3.2. Approach 2 - Dataset based end-device generation

In approach 2, we used a real dataset of LoRaWAN messages obtained in the city center of Antwerp [18]. Simulated end-devices positions are picked from actual positions of end-devices in the dataset. This dataset holds 130 430 messages which are collected from November 17th, 2017 until July 19th, 2019. This data-set is a wealth of information which contains end-device payloads combined to meta-data describing the channel state (RSSI, SNR) and the transmission parameters (SF, BW, CR, etc). Thus, we used these RSSI and SNR values as simulation inputs to emulate end-devices with real radio parameter, like in a real deployment. All other simulation parameters are the same as the first approach (See Table 4).

Similarly, as in the random generating study, we evaluate the simulation done with real emulated end-devices to demonstrate the effectiveness of our solution and improve the accuracy of our evaluation. The obtained results are shown in Fig. 9. Fig. 10a clearly shows that shifting some end-devices to higher SFs (See Fig. 9b) in order to lighten SF classes suffering from high load leads to better global DER. Hence, using L3SFA increases the network capacity. Indeed, we report from Fig. 9 that to achieve a DER higher than 85%, the network can support 6000 end-devices versus only 4500 without SF adjustment.

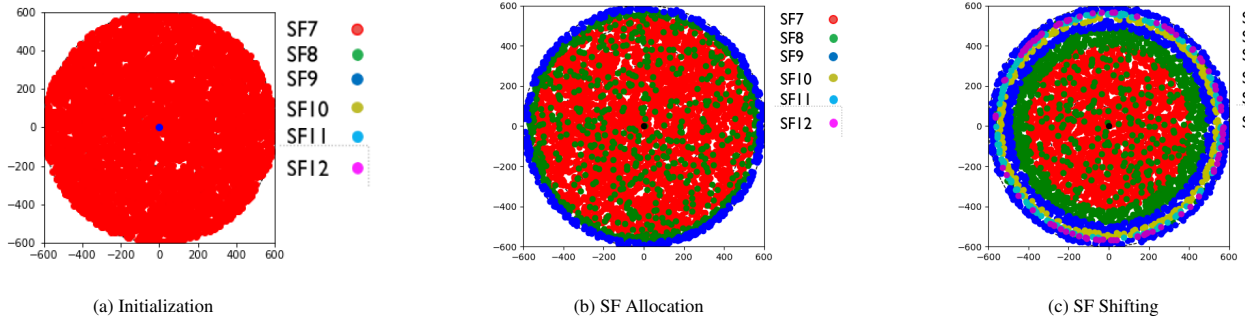


Figure 7: SF Distribution for 5000 end-devices in a cell of 600 meters radius for Different L3SFA Steps

7. L3SFA With Transmission Power Control

In LoRaWAN networks, concurrent transmissions using different spreading factors can be handled at the gateway-level. However, the different strengths of the received signals of these concurrent transmissions may increase drastically the error rate. This is due to the fact that the spreading factors are not completely orthogonal, which causes mutual interference. The matrix SIR [19] represents the Signal-to-Interference Ratio thresholds among spreading factors. when receiving two overlapping signals using different or similar spreading factors on the same channel does not mean that both are loss. In fact, if the difference between their signal strengths is less than the corresponding interference threshold, the two signals or at least the strongest one can be successfully received. Therefore, a signal isolation mechanism is required to successfully receive all concurrent transmissions. In wireless transmission, this isolation can be carried out either by transmission power control or channel assignment.

$$SIR = \begin{bmatrix} 6 & -8 & -9 & -9 & -9 & -9 \\ -11 & 6 & -11 & -12 & -13 & -13 \\ -15 & -13 & 6 & -13 & -14 & -15 \\ -19 & -18 & -17 & 6 & -17 & -18 \\ -22 & -22 & -21 & -20 & 6 & -20 \\ -25 & -25 & -25 & -24 & -23 & 6 \end{bmatrix} \quad (10)$$

7.1. Inter-SF Interference

Since the spreading factors are not perfectly orthogonal, two concurrent frames, i and j , sent with different spreading factors, sf and sf' respectively, are both correctly received only if the received signal strengths are higher than the thresholds defined in the SIR matrix. Otherwise, if equation 11 is verified, only the frame i is correctly received.

$$RSS_{sf}^i - RSS_{sf'}^j \leq SIR_{sf, sf'} \quad (11)$$

7.2. Intra-SF Interference

When two frames are sent in the same channel using the same spreading factor $sf = sf' = k$, it does not necessarily mean that both of them are lost due to the capture effect. Indeed, if the absolute difference of these two received signals strengths is less

than 6 dB (Equation 12), both of frames are lost. Otherwise, the frame with the strongest signal could be decoded.

$$|RSS_{sf}^i - RSS_{sf'}^j| \leq 6 \quad (12)$$

Algorithm 4 L3SFA-TPC

Input : N : end-devices

Output : N with SF allocated and Tx adjusted, SF_Distribution

- 1: End-devices Initialization
 - 2: Spreading Factor Assignment
 - 3: Spreading Factor Shifting
 - 4: Transmission Power Control (Algorithm 5)
-

In this section, we present a transmission power control strategy called, *TPC_LoRa* which extends the *L3SFA* algorithm to L3SFA-TPC given in algorithm 4. More precisely, *TPC_LoRa* algorithm looks at the difference in power strengths between all frame pairs from all the end-devices sent on the same channel. As defined in LoRaWAN specifications, transmission power ranges from 2 to 14 dBm in the European ISM Band regulation with a step of 3 as follows: $TP \in \{2, 5, 8, 11, 14\}$. The proposed strategy calculates the maximum possible steps to decrease the transmission power based on the difference between the transmission power and the thresholds defined in the SIR matrix, as given by the lines [5 – 14] of the algorithm 5. Moreover, the number of steps is used to decrease the power transmission of the lost frame end-device. This algorithm presents a sub-quadratic execution time and its complexity is given as follows: $O(N^2)$

To validate the TPC_LoRa approach combined with L3SFA, we simulated 5000 end-devices under the same simulation parameters given in table 4 for $p = 100$ seconds, and we evaluated the energy consumption and the DER.

Equation 13 gives the cumulative energy consumption E over all sent frames from N end-devices where T_n is the time-on-air for the end-device n using the assigned SF , Tx_n is the transmission consumption in mA, $Nframe_n$ is the number of frames sent from end-device n and finally V is the voltage.

$$E = \sum_{n=1}^N T_n \times Tx_n \times Nframe_n \times V \quad (13)$$

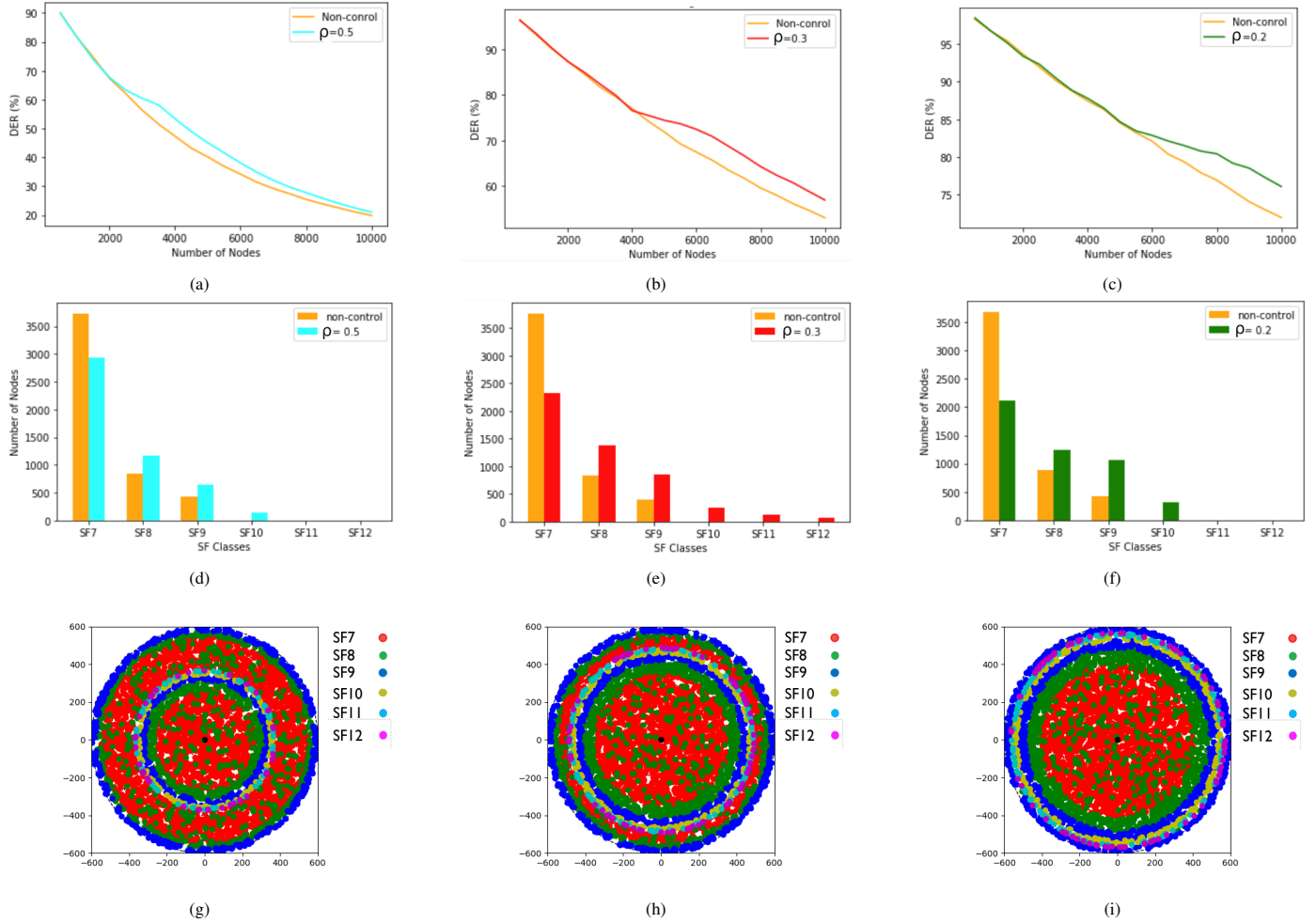


Figure 8: L3SFA Random Generation Performance Study. [a-c] DER as a function of the number of end-devices for a frame generation rate, respectively $\frac{1}{100s}$, $\frac{1}{300s}$, $\frac{1}{600s}$. [d-f] SF Distribution for 5000 end-devices. [g-i] Final SF Distribution for 5000 end-devices in a cell of 600 meters radius.

We plotted in Fig. 10b the cumulative energy consumption E in *joule* over all sent frames from 5000 end-devices for both *L3SFA* and *L3SFA-TPC* under $\rho = 0.2$. It is worth noting that our results demonstrate that adopting *TPC_LoRa* decreases energy consumption by 27% compared to *L3SFA*. Moreover, *L3SFA-TPC* presents better performances in terms of DER for $\rho = 0.2$ and keeps the same performances as *L3SFA* while improving energy consumption.

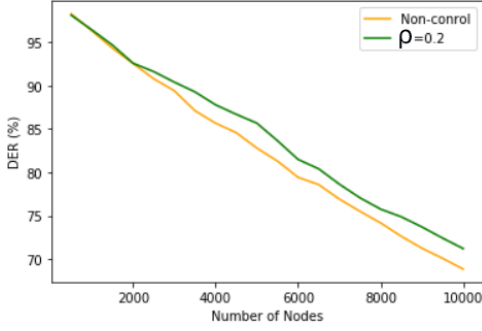
8. Testbed Setup

8.1. Design and Configuration

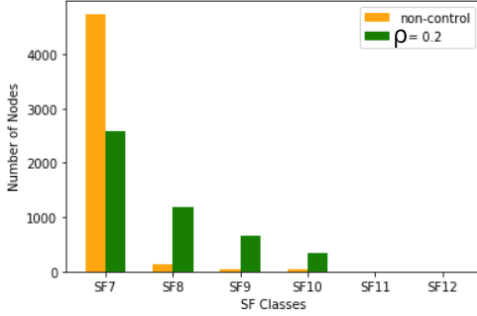
In this section, we cover some concrete concerns related to experiment reproduction and scalability. In order to avoid the tedious work of re-deploying a LoRaWAN network for each experiment, we present and design a testbed for LoRaWAN experiments. Furthermore, this testbed will be used to validate the proposed SF allocation approach by considering a set-up of a real experimentation platform consisting of a Multitech Conduit Gateway, a microchip-based RN2483 and a lopy 4 end-devices.

Several LoRaWAN backend solutions exist. For the design and the deployment of this testbed we used the ChirpStack [20] solution. Indeed, ChirpStack is an open source LoRaWAN stack that provides open-source modules as illustrated in Fig. 11 to deploy LoRaWAN networks. The detail of these modules is given as follows:

- **Gateway Bridge:** This module is responsible for converting the frame into a suitable format that can be easily analysed by the network server. In addition, this module manages the communications between the gateway and the network server.
- **Network Server:** Is the ChirpStack core module. It is in charge of receiving frames and provides authentication mechanisms, manages the MAC layer, supports downlink programming and handles communications with the application server.
- **Application Server:** This module manages the end-devices, decrypts the application payloads and enables integration with external services (cloud, end-device-red, databases, ..).



(a) DER as a function of the number of end-devices



(b) SF Distribution for 5000 end-devices

Figure 9: L3SFA Dataset Generation Performance Study

ChirpStack is an open source scalable and extensible LoRaWAN tools due to its modularity design. It should be noted, however, that like other available LoRaWAN projects, ChirpStack does not support sending personalized downlink control messages containing MAC commands in order to provide an over the air end-devices configuration capability. More precisely, ChirpStack is designed to send control commands only when the Adaptive Data Rate algorithm is enabled. In other words, it is not possible to send custom control commands in the payload by setting the "Port" to 0 or by using the "fOpts" and "fOptsLen" header fields. In the end, sending control commands in the downlink appears to be a tedious task that complicates the testing of new approaches that offer gateway-centric end-device control.

Our goal is to take advantage of the ChirpStack modularity by extending it with a new flexible module called "L3SFA Extension" as illustrated in Figure. 11). This new module offers valuable features. As designed, it consists of two main components: 1) *Core Engine*: this is the flexible part of the module, it provides the control intelligence of LoRa end-devices and allows LoRaWAN developers to provide end-device transmission parameters control mechanisms. For instance, L3SFA strategy is implemented in the core engine and other allocation parameters strategies can be added. 2) *Flask Server*: provides access to a set of endpoints to interface with the ChirpStack network server. It handles the MAC commands sequences computed by the core engine using a given strategy and feeds the network server downlink queue to be scheduled later.

We are planning to use the designed platform in our future work to evaluate L3SFA in a real environment deployment and investigate downlink issues.

Algorithm 5 TPC.LoRa

Input : N : end-devices.

Output : N with Tx adjusted.

```

1: for  $i$  in  $N$  do
2:   for  $j$  in  $N$  do
3:     if  $i \neq j$  then
4:       if  $i.freq == j.freq$  then
5:         if  $(i.rssi - j.rssi) < SIR[i.sf - 7][j.sf - 7]$  then
6:           if  $i.sf == j.sf$  then
7:              $diff = SIR[i.sf][j.sf] - (i.rssi - j.rssi)$ 
8:           else
9:              $diff = SIR[i.sf][j.sf] - (i.rssi - j.rssi)$ 
10:          end if
11:           $steps = \lceil diff/3 \rceil$ 
12:        end if
13:        if  $steps > 4$  then
14:           $steps = 4$ 
15:        end if
16:         $newRSSI = 0$ 
17:         $newTx = 0$ 
18:        for  $s$  in range (1, steps) do
19:          if  $j.rssi - s \times 3 > Sensitivity_{j,sf}$  &  $j.tx - s \times 3 \geq 5$  then
20:             $newRSSI = j.rssi - s \times 3$ 
21:             $newTx = j.tx - s \times 3$ 
22:          end if
23:          if  $newRSSI \neq 0$  and  $newTx \neq 0$  then
24:             $j.rssi = newRSSI$ 
25:             $j.tx = newTx$ 
26:          end if
27:        end for
28:      end if
29:    end if
30:  end for
31: end for

```

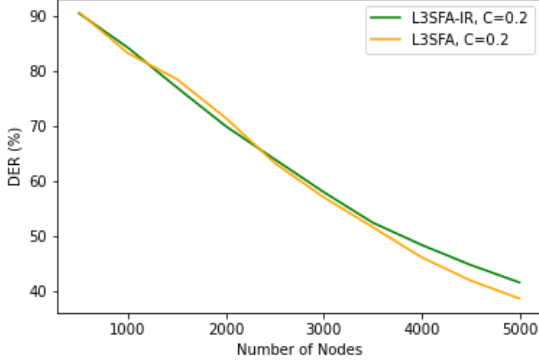
8.2. Motivating use case: Coverage study

To experiment on our LoRaWAN platform, we chose as a first use case to study the coverage with all our gateways. More exactly, we perform an experiment using the testbed described in section 8 when disabling L3SFA extension. In this experiment, we consider a standard LoRaWAN network architecture composed of two gateways receiving data from one mobile end-device. In this study, the end-device sends periodically a frame every 5 seconds containing its GPS position collected with the GPS module to the gateways while moving on the university campus, as shown in Fig. 12.

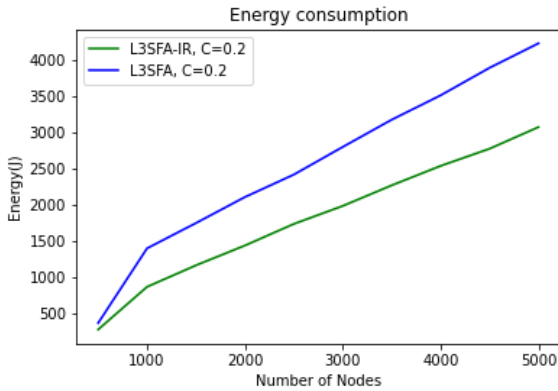
To conduct our study, we used the same end-device presented in Fig. 4 configured with the following parameters: $SF = \{8, 10, 12\}$, $BW = 125kHz$, $Ptx = 14dBm$ and $CR = 4/5$. We recorded a total of 204, 234, and 393 received frames with SF8, SF10, and SF12 respectively. The data extraction rate is distributed as depicted in Table 5 and described as follows:

- Gateway n°1 has received *with SF8*: 117 frames among

ER in Function of Number of Nodes using 3GPP Urban Macro Model (T=1)



(a) DER as a function of the number of end-devices



(b) SF Distribution for 5000 end-devices

Figure 10: L3SFA-TPC Performance Study

204 sent, representing a data extraction rate of 57,35% with the best RSSI and SNR. 26 of these 117 frames are only received on this gateway. *With SF10*: 128 frames among 234 sent, representing a reception rate of 54,7% with the best RSSI and SNR. 10 of these 128 frames are only received on this gateway. *With SF12*: 308 frames among 393 sent, representing a data extraction rate of 78,37% with the best RSSI and SNR. Moreover, 39 of these 308 frames are exclusively received on this gateway.

- Gateway n°2 has received *with SF8*: 87 frames among 204 sent, representing a data extraction rate of 42.65% with the best RSSI and SNR. 15 only of these 87 frames are only

Table 5: Coverage experiment study results

	GW 1		GW 2		GW 1 & 2
	Best	Only	Best	Only	Both
SF8	117 57.35 %	26	87 42.65 %	15	163 79.90 %
SF10	128 54.70 %	10	106 45.30 %	24	200 85.47 %
SF12	308 78.37 %	39	85 21.63 %	14	340 86.51 %

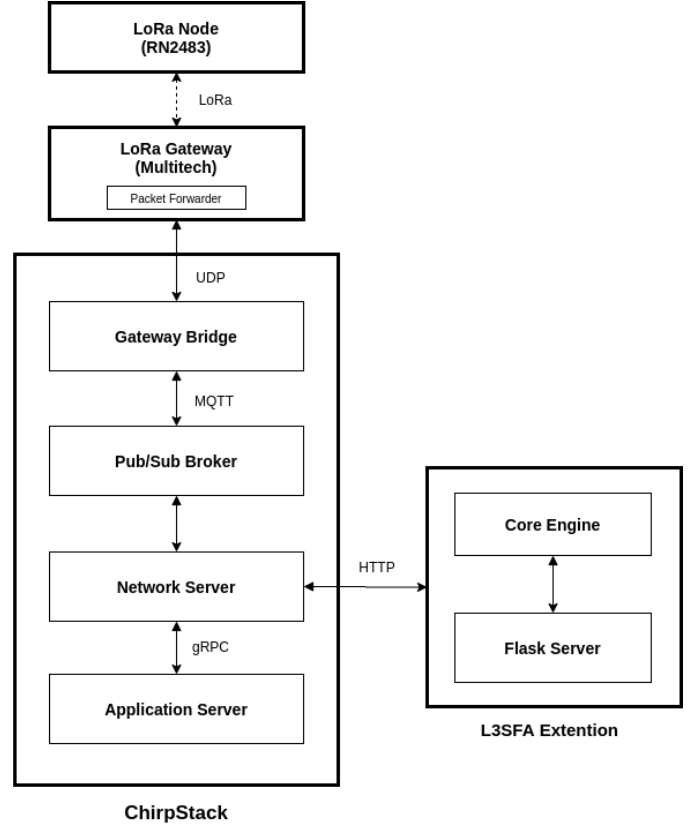


Figure 11: Extended ChirpStack Architecture

received on this gateway. *With SF10*: 106 frames among 234 sent, representing a data extraction rate of 45.3% with the best RSSI and SNR. Besides, 24 frames among these 106 frames are exclusively received on the gateway n°2. *With SF12*: 85 frames among 393 sent, representing a data extraction rate of 21.63% with the best RSSI and SNR. In addition, 14 frames among the 85 are only received on this gateway.

- Both of the gateways n°1 and n°2 received *with SF8*: 163 frames among the 204 sent representing 79.9%. *With SF10*: 200 frames among the 234 sent representing 85.47%. *With SF12*: 340 frames from 393 sent representing 86.51%.

The takeaways from this experiment are as follows: 1) The position of the gateways is very important. Indeed, if we remove Gateway n°1 and keep the network only with Gateway n°2, 39 frames will not be received. Moreover, we will have frames received with a lower RSSI and SNR since most of the selected frames are those from Gateway n°1. 2) Deploying more gateways is undoubtedly more advantageous in terms of network coverage and resilience.

9. Conclusion

In this paper, we addressed the Spreading Factor assignment issue in large-scale LoRaWAN networks with low duty-cycle.

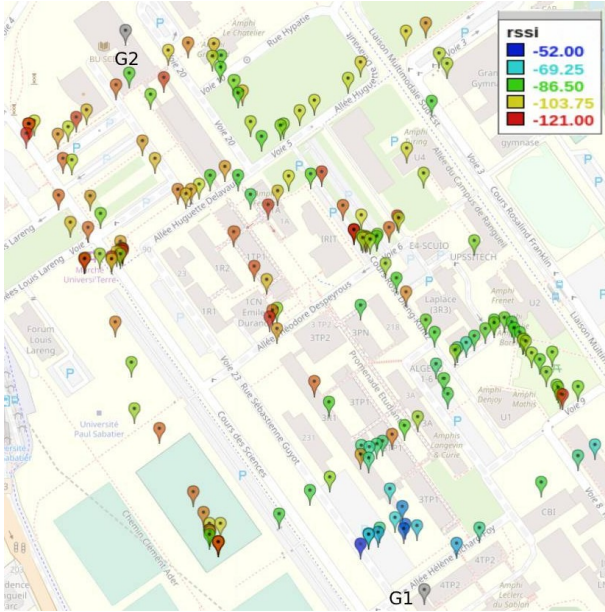


Figure 12: RSSI Mapping of LoRa communications on the campus using the platform given in fig.4 with an RPi, a N2483 LoRa module, and a xbee GPS module.

We demonstrated that traditional spreading factor assignment policies either suffer from low efficiency or simply fail to handle the traffic load of the network. We showed that the problem is similar to a load balancing problem, and consequently, we presented a new approach for SF allocation that aims to find the best distribution of the end-devices among different SF classes by performing an effective load balancing orchestration. Additionally, our solution also motivated an experimental study that relies on the correlation between the RSSI and the distance. We evaluated the performance of the proposed strategy under various network settings. The results suggest that load balancing leads to better performance in terms of Data Extraction Rate while guaranteeing high scalability even with a large network size and a high density. Moreover, the extension of L3SFA with a transmission power control algorithm demonstrates that the L3SFA-TPC approach saves 27% of power consumption while maintaining the same Data Extraction Rate performance.

In the future, we aim to improve the performance of our strategy, as L3SFA is not the last chapter spreading factor allocation in LoRaWAN systems. Besides, we would like to deploy our algorithms on our LoRaWAN network server, and conduct experiments to investigate how it behaves in real infrastructure, especially on the down-link. Moreover, through the takeaways of our case-study on macro-diversity reception, there is certainly a room for a fascinating challenge on the distribution of our algorithms in a multi-gateway enabled SF allocation system.

Acknowledgment

This research study is supported by “neOCampus”, an operation for research and innovation of the University of Toulouse.

References

- [1] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martínez, J. Melia-Segui, T. Watteyne, Understanding the limits of lorawan, *IEEE Communications Magazine* 55 (9) (2017) 34–40.
- [2] D. Bankov, E. Khorov, A. Lyakhov, Mathematical model of lorawan channel access, in: 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017, pp. 1–3.
- [3] O. Georgiou, U. Raza, Low power wide area network analysis: Can lora scale?, *IEEE Wireless Communications Letters* 6 (2) (2017) 162–165.
- [4] M. Bor, U. Roedig, T. Voigt, J. Alonso, Do lora low-power wide-area networks scale?, in: Proc. of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWIM’16, 2016, p. 59–67. doi:10.1145/2988287.2989163.
- [5] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, P. Pisani, Explora: Extending the performance of lora by suitable spreading factor allocations, in: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2017, pp. 1–8.
- [6] F. Cuomo, J. C. C. Gámez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, P. Pisani, Towards traffic-oriented spreading factor allocations in lorawan systems, in: 2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2018, pp. 1–8.
- [7] R. M. Sandoval, A. Garcia-Sanchez, J. Garcia-Haro, Optimizing and updating lora communication parameters: A machine learning approach, *IEEE Transactions on Network and Service Management* 16 (3) (2019) 884–895.
- [8] R. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, Performance optimization of lora nodes for the future smart city/industry, *EURASIP Journal on Wireless Communications and Networking* 2019. doi:10.1186/s13638-019-1522-1.
- [9] M. Asad Ullah, J. Iqbal, A. Hoeller, R. D. Souza, Alves, Hirley, K-means spreading factor allocation for large-scale lora networks, *Sensors (Basel, Switzerland)* 19 (21) (2019) 4723, 31671700[pmid]. doi:10.3390/s19214723. URL <https://pubmed.ncbi.nlm.nih.gov/31671700>
- [10] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, D. Pesch, *free*—fine-grained scheduling for reliable and energy-efficient data collection in lorawan, *IEEE Internet of Things Journal* 7 (1) (2020) 669–683.
- [11] R. M. Sandoval, D. Rodenas-Herraz, A. Garcia-Sanchez, J. Garcia-Haro, Deriving and updating optimal transmission configurations for lora networks, *IEEE Access* 8 (2020) 38586–38595.
- [12] A. Farhad, D.-H. Kim, J.-Y. Pyun, Resource allocation to massive internet of things in lorawans, *Sensors* 20 (2020) 20. doi:10.3390/s20092645.
- [13] J. Liando, A. Gamage, A. Tengourtius, M. Li, Known and unknown facts of lora: Experiences from a large-scale measurement study, *ACM Transactions on Sensor Networks* 15 (2019) 1–35. doi:10.1145/3293534.
- [14] J. Haxhibeqiri, E. De Poorter, I. Moerman, J. Hoebeke, A survey of lorawan for iot: From technology to application, *Sensors* 18 (2018) 3995. doi:10.3390/s18113995.
- [15] T. T. Network, Lorawan architecture (Jun 2020). URL <https://www.thethingsnetwork.org/docs/lorawan/architecture.html>
- [16] R. Hemmecke, M. Köppe, J. Lee, R. Weismantel, Nonlinear integer programming, in: 50 Years of Integer Programming, 2010.
- [17] 3GPP, Etsi tr 125 996 v10.0.0 (2011). URL https://www.etsi.org/deliver/etsi_r/1259001_25999/125996/10.00.00
- [18] M. Aernouts, R. Berkvens, K. Vlaenderen, M. Weyn, Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas, *Data* 3 (2018) 13. doi:10.3390/data3020013.
- [19] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, I. Tinnirello, Impact of lora imperfect orthogonality: Analysis of link-level performance, *IEEE Communications Letters* 22 (4) (2018) 796–799. doi:10.1109/LCOMM.2018.2797057.
- [20] ChirpStack, Chirpstack: open-source lorawan network server stack. URL <https://www.chirpstack.io/>