Bruno Mendes

# Optimal Deployment of Mobile Gateways in LoRaWAN Environments

UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

2022

Bruno Mendes

# Optimal Deployment of Mobile Gateways in LoRaWAN Environments

**Masters in Computer Engineering**

**Work done under Prof. Dr. Noélia Correia and Dr. Dário Passos supervision.**

UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

2022

# Copyright

## Optimal Deployment of Mobile Gateways in LoRaWAN Environments

Declaration of authorship of work: I declare to be the author of this work, which is original and unpublished. Authors and works consulted are properly cited in the text and appear in the list of references included

## Candidate:

_______________________________

(Bruno Mendes)

Work done at the Research Center of Electronics, Optoelectronics and Telecommunications (CEOT)

## Acknowledgements

First I would like to thank my two thesis mentors, Dr. Noélia Correia and Dr. Dário Passos, their support and knowledge during my two years of masters was the main reason I got so far.

I would also like to thank those who helped me during my undergraduate, masters and CEOT, both teachers and students.

And last but not least my friends and family, who although they are far away and not involved in the field, gave me the motivation to continue.

This work was supported by FCT (Foundation for Science and Technology) from Portugal within CEOT (Center for Electronic, Optoelectronic and Telecommunications) and UID/MULTI/00631/2020 project.

# Abstract

The recent growth of the Internet of Things (IoT) has given rise to new applications and technologies. Of these technologies, LoRa is the one that has stood out recently due to its ability to transmit packets over long distances at low energy costs. In addition to this, this technology also uses unlicensed frequency bands, and all these factors make it possible to build low energy cost networks with large coverage areas at low monetary cost. This makes LoRa very appealing for environments where multiple square kilometers need to be covered for monitoring, such as agriculture. This thesis focuses primarily on positioning gateways in a LoRaWAN in order to achieve energy fairness in the network.The target in question is an environmental sensor network that monitors conditions inside tree canopies in an orange orchard in the Algarve, south of Portugal.The peculiar characteristics of these orange trees, with heights up to 3.5 m and very dense foliage, makes it a very challenging environment for radio waves propagation and causes a rapid drop in signal quality. The power consumption of the end-nodes of the network is defined by 7 combinations of spreading factor and bandwidth (0 to 6) where 0 represents the slowest and most reliable transmission at the cost of higher power consumption while 6 represents the opposite. The combination of bandwidth and spreading factor is denominated data rate. Environmental factors can negatively impact the quality of LoRa packets and the necessary power adjustments of the end-node to overcome this, and increase signal reliability, can easily define whether a device is able to transmit for 1 year or 10! The main factors that can affect signal quality are obstruction, distance and meteorology. In the case study, of these 3 factors, obstruction affects transmission quality the most. Most of the literature suggests solutions within the framework of optimizing the datarate optimization algorithm (ADR). ADR aims to minimize energy consumption while ensuring the best possible packet transmission rate and achieves this by changing the data rate based on the quality of the last 20 packets received.However, this optimization is done directly to individual end-nodes and does not solve the problem of energy fairness

over the whole network because, regardless of how optimized this algorithm is, the algorithm cannot transcend the physical constraints imposed by the devices and the technology itself. Distance and obstruction will always be obstacles to signal quality. Since these physical constraints will always be present in a network and the solutions proposed by the literature only improve performance at the level of individual devices, this ends up creating a large lifetime discrepancy between devices depending on their placement. In the case of LHT65s, the discrepancy in device life expectancy is high. For example the difference between using a data rate of 0 or 5 is about 10 years.

The solution proposed in this thesis to overcome this problem is to precompute the optimal position for the gateways in order to guarantee the highest life expectancy for the network. Given a number of available positions for the gateways and having a certain number of gateways less than the number of positions, the goal is to compute the optimal positioning of the gateways in order to maximize the overall network life expectancy by ensuring a fair energy consumption among different end-nodes.

The first step in this process was to collect information about signal quality from a real case LoRaWAN deployment. This allowed to better understand the constraints and problems associated with its implementation. This was done using 25 LTH65 devices, 1 RAK 7244 gateway and Chirpstack as the framework to manage the network. Regarding the study of the algorithm before applying it to the practical case, a simulator was used to collect data. The simulator chosen for the development of the application was OMNet++, which besides being easier to use is also better documented than the other options considered. This simulator also offers a graphical interface with great detail that allows you to easily observe the behavior of the network. Using the Flora module it was simulated a LoRaWAN network with the structure suggested by the LoRa Alliance® with 25 devices using Oulu's path loss model. The information obtained from this simulation was used as input and test for the algorithm that was compiled by CPLEX. In each simulation about 10,000 packets were sent per device and each experiment was repeated 30 times.

The results show that the optimization model has the ability to identify the best placement for the gateway given a predefined locations and network geometry. This is due to the fact that the algorithm identifies the lowest value in the highest energy consumption per packet, and minimizing this value creates a balance of consumption among the devices and consequently extends the life expectancy of the network. It can then be concluded that this methodology is indeed efficient for deployments where changing network devices cannot be done frequently. Although it is not easy to relocate gateways in already implemented networks, but in new environments where monitoring and optimization are requirements, and these new environments are built considering the network structure, we can use this methodology since it has proven to be able to improve network life expectancy.

**Keywords:** IoT, LoRaWAN, LPWAN, Chirpstack, OMNet++

# Resumo

O recente crescimento da Internet das Coisas (IoT) deu origem a novas aplicações e tecnologias. Destas tecnologias, a LoRa é a que se tem destacado recentemente devido à sua capacidade de transmitir pacotes a longas distâncias a baixos custos energéticos. Além disso, esta tecnologia também utiliza bandas de frequência não licenciadas, e todos estes factores tornam possível a construção de redes de baixo custo energético com grandes áreas de cobertura a baixo custo monetário. Isto torna LoRa muito apelativo para ambientes onde vários quilómetros quadrados precisam de ser cobertos para monitorização, tais como a agricultura. Esta tese centra-se principalmente no posicionamento de *gateways* numa rede LoRaWAN, a fim de alcançar a *energy fairness* na rede. O alvo em questão é uma rede de sensores ambientais que monitoriza as condições dentro de copas de árvores num pomar de laranjeiras no Algarve, a sul de Portugal. As características peculiares destas laranjeiras, com alturas até 3,5 m e folhagem muito densa, torna-o um ambiente muito desafiante para a propagação de ondas de rádio e causa uma queda rápida na qualidade do sinal. O consumo de energia dos nós finais da rede é definido por 7 combinações de factor de propagação e largura de banda (0 a 6) onde 0 representa a transmissão mais lenta e fiável ao custo de um maior consumo de energia, enquanto 6 representa o oposto. A combinação de largura de banda e factor de dispersão é denominada *data rate*. Os factores ambientais podem ter um impacto negativo na qualidade dos pacotes LoRa e os ajustamentos de potência necessários do nó final para ultrapassar isto, e aumentar a fiabilidade do sinal, podem facilmente definir se um dispositivo é capaz de transmitir durante 1 ano ou 10! Os principais factores que podem afectar a qualidade do sinal são a obstrução, a distância e a meteorologia. No estudo de caso, destes 3 factores, a obstrução é o que mais afecta a qualidade de transmissão. A maior parte da literatura sugere soluções no âmbito da optimização do algoritmo de optimização de datarate (ADR). O ADR visa minimizar o consumo de energia, assegurando a melhor taxa de transmissão de pacotes possível e consegue-o alterando a taxa de dados com

base na qualidade dos últimos 20 pacotes recebidos. No entanto, esta optimização é feita directamente aos nós finais individuais e não resolve o problema da *energy fairness* em toda a rede porque, independentemente de quão optimizado este algoritmo esteja, o algoritmo não pode transcender as restrições físicas impostas pelos dispositivos e pela própria tecnologia. A distância e a obstrução serão sempre obstáculos à qualidade do sinal. Uma vez que estas restrições físicas estarão sempre presentes numa rede e as soluções propostas pela literatura apenas melhoram o desempenho ao nível dos dispositivos individuais, isto acaba por criar uma grande discrepância ao longo da vida útil entre os dispositivos, dependendo da sua colocação. No caso dos LHT65s, a discrepância na esperança de vida útil dos dispositivos é elevada. Por exemplo, a diferença entre a utilização de uma taxa de dados de 0 ou 5 é de cerca de 10 anos.

A solução proposta nesta tese para ultrapassar este problema é a de pré-computação da posição óptima para as *gateways*, a fim de garantir a maior esperança de vida da rede. Dado um número de posições disponíveis para as *gateways* e tendo um certo número de *gateways* inferior ao número de posições, o objectivo é calcular o posicionamento óptimo das *gateways* a fim de maximizar a esperança de vida global da rede, assegurando um consumo justo de energia entre os diferentes nós finais.

O primeiro passo neste processo foi recolher informações sobre a qualidade do sinal a partir de um caso real de implantação do LoRaWAN. Isto permitiu compreender melhor os constrangimentos e problemas associados à sua implementação. Isto foi feito utilizando 25 dispositivos LTH65, 1 RAK 7244 *gateway* e Chirpstack como estrutura para gerir a rede. Quanto ao estudo do algoritmo antes da sua aplicação ao caso prático, foi utilizado um simulador para a recolha de dados. O simulador escolhido para o desenvolvimento da aplicação foi o OMNet++, que além de ser mais fácil de utilizar está também melhor documentado do que as outras opções consideradas. Este simulador também oferece uma interface gráfica com grande detalhe que lhe permite observar facilmente o comportamento da rede. Utilizando o módulo Flora foi simulada uma rede LoRaWAN com a estrutura sug-

erida pela LoRa Alliance® com 25 dispositivos utilizando o modelo de perda de trajecto de Oulu. A informação obtida desta simulação foi utilizada como entrada e teste para o algoritmo que foi compilado pelo CPLEX. Em cada simulação foram enviados cerca de 10.000 pacotes por dispositivo e cada experiência foi repetida 30 vezes.

Os resultados mostram que o modelo de optimização tem a capacidade de identificar a melhor colocação para a *gateway* dada uma localização e geometria de rede predefinidas. Isto deve-se ao facto de o algoritmo identificar o valor mais baixo no maior consumo de energia por pacote, e minimizar este valor cria um equilíbrio de consumo entre os dispositivos e consequentemente prolonga a esperança de vida da rede. Pode-se então concluir que esta metodologia é de facto eficiente para implantações onde a mudança de dispositivos de rede não pode ser feita com frequência. Embora não seja fácil relocalizar *gateways* em redes já implementadas, mas em novos ambientes onde a monitorização e a optimização são requisitos, e estes novos ambientes são construídos considerando a estrutura da rede, podemos utilizar esta metodologia uma vez que provou ser capaz de melhorar a esperança de vida da rede.

**Palavras-chave:** IoT, LoRaWAN, LPWAN, Chirpstack, OMNet++

# Contents

# List of Figures

# List of Tables

# Nomenclature

## Acronyms

| | | |
|---|---|---|
| ABP | : | Activation By Personalization |
| ADR | : | Adaptive Data Rate |
| AWGN | : | Additive White Gaussian Noise |
| BW | : | Bandwidth |
| CayenneLPP | : | Cayenne Low Power Payload |
| CAS | : | Chirpstack Application Server |
| CEOT | : | Center of Electronics, Optoelectronics and Telecommunications |
| CGB | : | Chirpstack Gateway Bridge |
| CNS | : | Chirpstack Network Server |
| CR | : | Code Rate |
| CSS | : | Chirp Spread Spectrum |
| DER | : | Data Extraction Rate |
| FSK | : | Frequency Shifting Keying |
| gRPC | : | google Remote Procedure Call |
| IoT | : | Internet of Things |
| LGR | : | LoRa Gateway Relocating |
| LPWAN | : | Low Power Wide Area Network |
| LoRa | : | Long Range |
| LoRaWAN | : | Long Range Wide Area Network |
| LoS | : | Line of Sight |
| MIC | : | Message Integrity Check |
| NEC | : | Network Energy Consumption |
| NLOS | : | Non Line of Sight |
| NS | : | Network Server |
| OTAA | : | Over The Air Activation |
| PDR | : | Packet Delivery Ratio |
| PRR | : | Packet Reception Ratio |
| RF | : | Radio Frame |
| RL | : | Reinforcement Learning |
| RSSI | : | Received Signal Strength Indicator |
| SF | : | Spreading Factor |
| ToA | : | Time on Air |
| UDP | : | User Datagram Protocol |

# 1 Introduction

In an age where technological innovation occurs at an unprecedented pace and market competition becomes intense, optimization of technologies assumes a critical role. In order to be able to perform optimization, it is necessary to deeply understand not only the structure and potential of available technologies but also their limitations. Such understanding can help improve the performance of technologies, and increase the quality of products depending on it.

The work described in this report is being developed within the Center of Electronics, Optoelectronics and Telecommunications (CEOT). One of the main objectives of CEOT is to develop technical competences and tools for the progress of the smart agriculture field (also called precision agriculture or agriculture 4.0). This requires the interleaving collaboration of multiple research fields spanning from Computer Science and Electronics, to Physics, Chemistry and Biology. The fruitful interdisciplinary collaboration between these fields will lead to the development of new technological processes able to optimize crops, increase fruit and vegetable quality and enable better operation of distribution channels. This objective is aligned with the Algarve region development plan that aims to equip the region with tools and technological capabilities for the improvement of agricultural economic output, while giving the region the ability to adapt to extreme weather patterns caused by climate change. Recent literature points out Long Range Wide Area Networks (LoRaWAN) as one of the most promising technologies for the monitorization of wide areas and remote farms. Research on how to optimize this technology to achieve its promised long range is already underway and mainly focuses on optimizing the PHY parameters (spreading factor, code rate and bandwidth) in such environments [14, 18, 28].

The increasing popularization of the Internet of Things (IoT) led to a focus on low-power technologies, and LoRa stood out as one of these technologies. LoRa is a long range protocol that allows the transmission of data through long distances with low energy consumption, over unlicensed spectrum, allowing to build networks that cover hectares of land. These features make this technology preferred in many agriculture studies. Despite the fact that LoRaWAN technology is highly regulated and solutions are readily available for end-user consumption, it has not yet spread or been massively adopted. This happens because its deployment still requires a lot of expertise to set up everything and, therefore, most of existing applications are only slowly materializing by the hands of specialized companies and amateur "makers community". Moving to new deployments requires previous feasibility studies, range tests, and so on. The massive adoption of LoRaWAN technology has also been delayed due to the unnormalized utilization of communication standards, multiple physical constraints imposed by practical scenarios and having to deal with a software development stack that is constantly evolving and being updated.

One of the goals of this dissertation is to evaluate the viability of using this communication technology in ongoing projects at CEOT. The successful deployment of a LoRaWAN network, subject to physical constraints (e.g. number of nodes, location of nodes and gateways, non line of sight (NLoS) communications, budget, etc) requires a arduous planning stage that includes market research, matching of technical characteristics of end nodes to the desired task and constraints, range coverage tests and optimization studies. These optimization studies can be done through simulation of the full LoRaWAN stack, and are essential not only to produce reliable deployments but also to help outline the workflow involved in practical scenarios.

*The remainder of this report is organized as it follows. Section 2 provides the objective of this thesis ; Section 3 provides more detail related to LoRa and LoRa related applications; Section 4 will discuss the mathematical model, simulation and practical use case. The results are discussed on Section 5 and it concludes on section 6.*

# 2 Research Question, Hypothesis and General Approach

## 2.1 Research Question

The large-scale deployment of Low Power Wide Area Networks (LPWAN) brings many challenges because radio resources are scarce and management becomes very challenging in practical networks. Therefore, procedures for an efficient and dynamic management of resources become necessary. In the case of LoRaWAN networks, an Adaptive Data Rate (ADR) mechanism is available that dynamically assigns transmission parameters to the end node to ensure that it works properly under different/changing conditions. [15].

In the literature where LoRa and link budget optimization are a common theme, optimization often comes down to either increasing the performance of the ADR algorithm or changing the architecture of the network. Architecture related proposals usually provide good results, as in [12], but are far from being user-friendly, which is what IoT thrives for (apart from low energy consumption). As for the ADR, and alternatives to this algorithm, the existing proposals have moderate success and are specific to the environments in which they have been applied. These include, for example, tweaking the default ADR algorithm using Reinforcement Learning (RL) such as Q-Learning, as in [9, 15]. These approaches proved to be successful, and have room for improvement, but optimization is done considering a single gateway. To fill this gap, multiple gateways are taken into account in this report and energy fairness is considered when placing such gateways. More clearly, the network lifetime is extended because there is an attempt to place gateways in a way that the depletion of device's battery is delayed as much as possible, and equally (or almost) for the various devices. Given the dynamics of the environments in question, it is envisaged that gateways may change place when needed, but only if this has a positive impact on extending the lifetime of the network, as it will be explained further in this work. Energy consumption is directly related with the Time on Air (ToA), which is the time required for a packet to travel from the end-node to the gateway, and this is Spreading Factor (SF) dependent. The adequate SF depends on the distance between endpoints and surrounding conditions, which may change over time.

The advantage of taking energy consumption fairness into account stands out when the network is placed in remote or hard to reach areas. The research work to be developed is expected to significantly improve cost and time associated with the recharge/replacement of device batteries because this will be done as few times as possible, and recharge/replacement operations can be adequately scheduled in order to avoid waste associated with multiple trips and any loss of data. Orchard monitorization is one of the CEOT's projects that fits this kind of deployment scenario.

Given a scenario where the devices are static and gateways are mobile, gateways can be re-positioned to ensure extending the lifetime of the network, and decisions must be able to be transcribed to reality. **Can energy consumption be reduced and fairness be ensured by simply re-positioning gateways?**

## 2.2 Hypothesis

As far as literature indicates, gateway mobility is not a new topic. There are many articles addressing network or gateway mobility for different purposes (e.g., data gathering), but most of them are based on communication protocols other than LoRa. LoRa related research has mostly focused on optimizing ADR parameters for specific architectures, or on enhancing ADR algorithm, without considering the whole network fairness. Regarding fairness, this has been a hot research topic for many years in networks, although it has not been in LoRa networks maybe due to the fact that this is a recent technology. As time passes and LoRa popularity increases, in particular in agriculture related activities (e.g., monitorization), it is expected that fairness will come into play. The hypothesis is that **fairness is expected to have a critical role in large scale scenarios and heterogeneous communication environments**. For this reason we believe it is the right time to tackle such topic, so that one can understand how large scale and heterogeneous communication environments can have coverage and lifetime improved. This allows deployments to be effective and additional research finds (from any application context) to emerge.

## 2.3 General Approach

Understanding the problem is undoubtedly an important step, and this allowed us to realize that the problem is relatively complex and needs a plan for its treatment. Such plan includes the following steps:

1. Outline of the optimization model. This step will lead to an understanding of the variables that are involved in the process, and their relationship. In the end the problem is broken down and understood in a deeper way.

2. Analysis of feasible LoRaWAN structure. Time has to be invested in becoming aware of what would be required in a real deployment.

3. Comparison of simulation environments where choices from optimization results could be evaluated, before moving into a real deployment. Discrete event simulators are usually the case, where real world systems can be modeled into a set of processes/modules that progress through time.

4. Move to a practical use case. Besides being important to test things in a real environment, this will allow us to develop a custom propagation error model, for incorporation into the simulation model, so that the real environment is represented more reliably. This requires architecting modules in Chirpstack, an open-source LoRa Wide Area Network server stack, so that measurements of interest are collected.

# 3 LoRa and LoRaWAN

## 3.1 Technical Overview

LoRa is a communication protocol for the creation of a long distance communication link between an end-device and a gateway. LoRa uses Chirp Spread Spectrum (CSS) modulation for low power consumption purposes, which provides a radio range larger than conventional modulations used in legacy systems, like Frequency Shifting Keying (FSK). This modulation technique uses the entire allocated bandwidth to broadcast a signal, making it robust to channel degradation mechanisms like multi-path fading, Doppler effect and in-band jamming interference.

LoRa chirps (also known as symbols) are the carrier of data. The Spreading Factor (SF) ends up controlling the chirp rate, and thus controls the speed of data transmission: lower SFs mean faster chirps, higher data transmission rates and shorter Time on Air (ToA) and, therefore, longer battery life because the transceivers are active for a shorter period; higher SFs provides higher receiver sensitivity. The duration of the symbol, denoted by $T$, depends on the $SF$ and bandwidth, denoted by $BW$, as follows [6]:

$$T = \frac{2^{SF}}{BW} \tag{1}$$

and the bit rate will be:

$$B = SF \times \frac{BW}{2^{SF}} \tag{2}$$

The robustness of the CSS signal allows LoRa packets to travel long distances in open fields. The most common range is 5 to 15 km under direct LoS. The range of a LoRa link highly depends on the environment and obstructions.

In the implementation of LoRa modulation, every 4 bits of data is encoded into 5, 6, 7, or 8 bits in total according to the forward error correction in use, which is selected by setting the Coding Rate (CR) to 1, 2, 3, and 4, respectively. So, the actual user data bit rate must be reduced by the factor $\frac{4}{4+CR}$.

The PHY header has 20 bits and is encoded with the most reliable code rate, while the rest of the frame is encoded with the code rate specified in the PHY header. The PHY payload carries Medium Access Control (MAC) layer information: MAC header, MAC payload and Message Integrity Check (MIC). MAC header defines protocol version, message type (data or management frame), whether it is transmitted in uplink or downlink, and whether it shall be acknowledged. At the MAC payload, there will be a frame header with information that is relevant to the uplink integrity, such as the device address (used during the joining process with the network), frame control (used as a frame counter to avoid frame duplication) and other options. The whole structure can be seen in Figure 1[6].

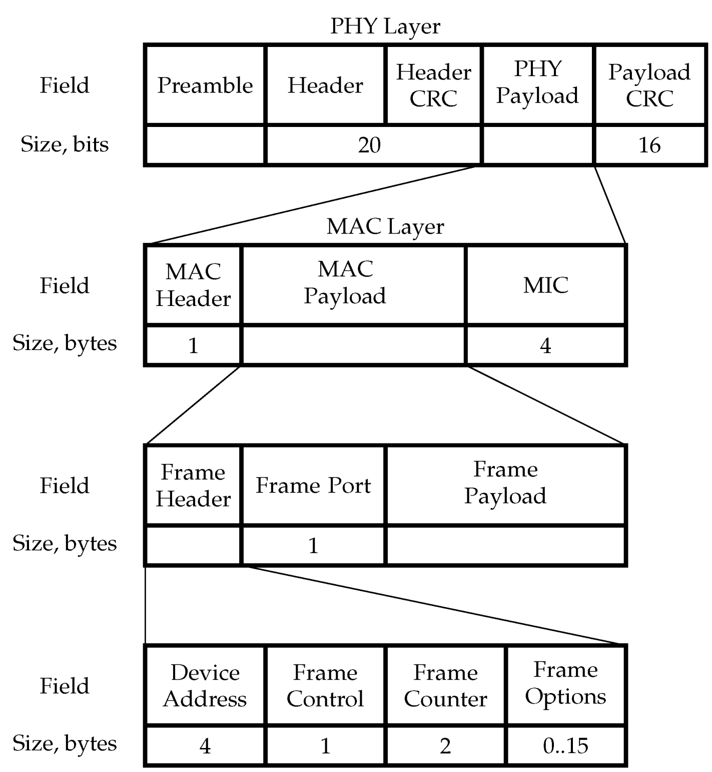


Figure 1: LoRa Packet Layers [7].

LoRa compliant networks follow a stars-to-stars architecture (see Figure 2) in which gateways relay messages between devices and a central Network Server (NS). By default, gateways communicate with an NS using IP and work as a bridge that converts the RF packets into IP packets, and vice-versa, allowing communication between end-devices and the NS [5].

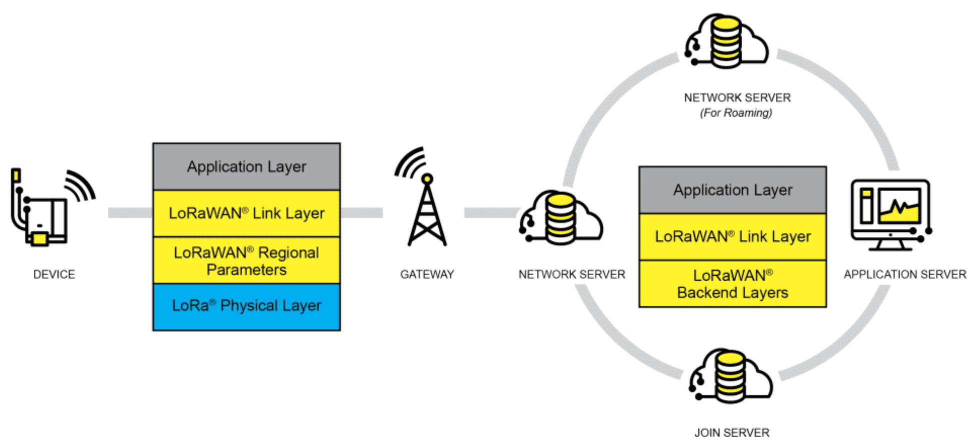


Figure 2: LoRaWAN architecture [5].

To make a star network viable the gateway must be able to span across a large number of end devices, so the network structure turns out to be critical for a gateway to achieve optimally. Firstly, all the heavy processing has been sent to the NS. Secondly, a duty cycle restriction is set for end-devices (in Europe it is 1%; more details in Section 1). Thirdly, multi-channel multi-modem transceiver gateways are used. The SF based modulation orthogonality allows a single channel to receive up to 8 simultaneous packets as long as different SF and bandwidth combinations are used (this is also known as datarate; see Figure 3). Therefore, a multi-channel gateway with 8 channels can receive up to 64 packets simultaneously.

| DataRate | Configuration | Indicative physical bit rate [bit/s] |
|---|---|---|
| 0 | LoRa: SF12 / 125 kHz | 250 |
| 1 | LoRa: SF11 / 125 kHz | 440 |
| 2 | LoRa: SF10 / 125 kHz | 980 |
| 3 | LoRa: SF9 / 125 kHz | 1760 |
| 4 | LoRa: SF8 / 125 kHz | 3125 |
| 5 | LoRa: SF7 / 125 kHz | 5470 |
| 6 | LoRa: SF7 / 250 kHz | 11000 |
| 7 | FSK: 50 kbps | 50000 |
| 8..14 | RFU | |
| 15 | Defined in LoRaWAN[1] | |

Figure 3: Data rates in Europe [26].

The ToA is also critical for the viability of LoRaWAN stars. It is well known that the more data the packet contains, the longer it takes to transmit. Also, the longer the device is awake then the more energy is spent. Looking at equation 3, where $PL$ is the packet payload/length, $H$ is the header implicit flag and $DE$ a low data rate optimization flag, it is clear that few factors play a big role in ToA: spreading factor, bandwidth, coding rate and the payload size. The first ones have already been discussed, and will be detailed in Section 3.5. Whereas these do not leave much space for improvement, given a distance and environment conditions, the payload size can in fact be reduced through encoding.

$$ToA = \frac{2^{SF}}{BW} \times [8 + \max(\left\lceil \frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)} \right\rceil \times (CR + 4), 0)] \tag{3}$$

The Cayenne Low Power Payload (CLPP) is currently the state of the art in payload size optimization. CLPP is compliant with the payload size restriction (51 bytes) and allows to send multiple sensor data at once. The structure of CLPP is pretty simple as it is shown in Figure 4. Each data channel uniquely identifies a sensor, so it is possible to have multiple sensors of the same type. As for the type it is a standardized number that defines the type of sensor that is being used (e.g., type number 67 is a temperature sensor) [22].

**Payload structure**

| 1 Byte | 1 Byte | N Bytes | 1 Byte | 1 Byte | M Bytes | ... |
|---|---|---|---|---|---|---|
| Data1 Ch. | Data1 Type | Data1 | Data2 Ch. | Data2 Type | Data2 | ... |

Figure 4: Cayenne LPP payload structure [22].

## 3.2 Device Classes: A, B and C

There are three classes of devices (A, B and C) that can be used in a LoRa network, and each one of them thrives in a specific situation. LoRa-based end devices operate differently depending on their class:

- Class A: This is the most popular among device classes. It supports bi-directional communication between the device and the gateway (see Figure 5). The uplink messages can be sent at any time. Devices open two short receive (downlink) windows (RX1 and RX2) after the uplink transmission is completed [23, 29]. Usually these kind of devices are used when there is no requirement for sending messages at predetermined times (e.g., environment monitoring). They usually have long intervals between uplinks, are battery powered and spend most of the time sleeping in order to optimize battery consumption.

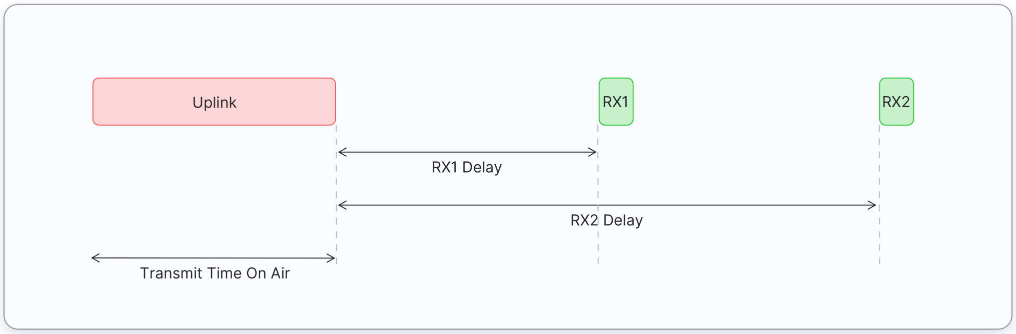


Figure 5: Bi-directional communication between device Class A and gateway [23].

- Class B: Can be seen as a Class A extension by adding a scheduled receive window for downlink messages from the network server (besides the two mentioned in Class A). Time-synchronized beacons transmitted by the gateway are used for devices to periodically open receiving window (known as "ping slot") 6. The Class B devices can also receive downlinks in the same way as Class A, after the uplink. These devices have lower latency than Class A because the receive window is predictable and configurable (not bounded to any uplink). This leads to an increase in battery consumption, as devices spend more time in active mode.

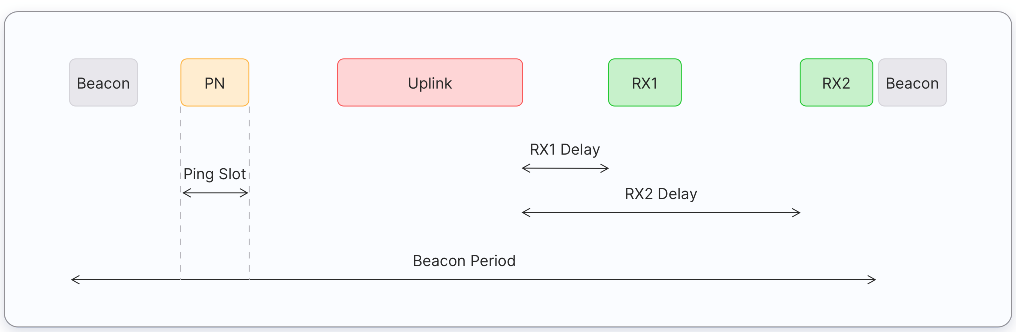


Figure 6: Bi-directional communication between device Class B and gateway [23].

- Class C: This class also extends Class A by keeping up the receive windows open all the time. This methodology allows for very low latency but the energy consumption is much higher than in class A, as devices are always active.

## 3.3 Packet Forwarders

A packet forwarder is a software module running at the gateway that forwards RF packets to an NS using the User Datagram Protocol (UDP), and also does the reverse process. The most popular one is the Semtech UDP version and for that reason it was chosen for this project. Alternatives to it include Semtech Basic Station and Chirpstack Concentratord.

The Semtech UDP Packet Forwarder adopts two protocols, one for uplink and another for downlink, and each protocol includes different types of messages.

### 3.3.1 Uplink protocol

The uplink protocol uses two types of message, the *PUSH_DATA* and *PUSH_ACK*. The first is used to forward RF packets or gateway updates while the second one is send whenever there is an acknowledgment requirement or whenever all the packets have been successfully received by the NS. The flow of the uplink operation can be seen in Figure 7.

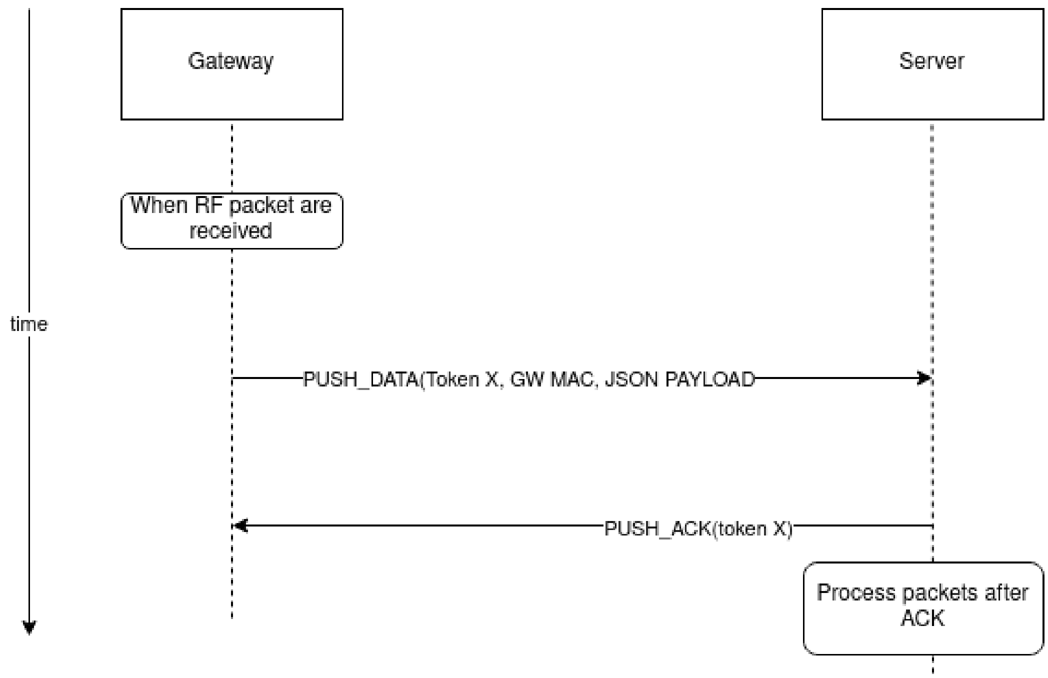


Figure 7: Uplink workflow between device and gateway. Image based on [19]

These messages are easily identifiable by their header format and size. As specified in Tables 1 and 2, the size of the *PUSH_DATA* header is 12 bytes while the size of the *PUSH_ACK* header is 3 bytes.

| Bytes | Function |
|---|---|
| 1 | Protocol version |
| 1-2 | Random token |
| 3 | PUSH_DATA identifier 0x00 |
| 4-11 | Gateway unique identifier (MAC address) |
| 12-end | JSON object |

Table 1: Push Data packet format. Table from [19]

.

| Bytes | Function |
|---|---|
| 0 | Protocol version |
| 1-2 | Same token as the PUSH_DATA packet to acknowledge |
| 3 | PUSH_ACK identifier 0x01 |

Table 2: Push Ack packet format. Table from [19]

.

The JSON object sent in *PUSH_DATA* messages can be of two types: 1) root object, which is the RF packet from an end-node (identified with the "rxpk" dictionary entry); 2) stat update from the gateway (identified with "stat"). The packet forwarder allows multiple JSON objects within the same object. The format adopted in each object type is shown in Tables 3 and 4.

| Name | Type | Function |
|---|---|---|
| time | string | UTC time of pkt RX, us precision, ISO 8601 "compact" format |
| tmms | number | GPS time of pkt RX, number of milliseconds since 06-Jan-1980 |
| tmst | number | Internal timestamp of "RX finished" event (32b unsigned) |
| freq | number | RX central frequency in MHz (unsigned float, Hz precision) |
| chan | number | Concentrator "IF" channel used for RX (unsigned integer) |
| rfch | number | Concentrator "RF chain" used for RX (unsigned integer) |
| stat | number | CRC status: 1 = OK, -1 = fail, 0 = no CRC |
| modu | string | Modulation identifier "LORA" or "FSK" |
| datr | string | LoRa datarate identifier (eg. SF12BW500) |
| datr | number | FSK datarate (unsigned, in bits per second) |
| codr | string | LoRa ECC coding rate identifier |
| rssi | number | RSSI in dBm (signed integer, 1 dB precision) |
| lsnr | number | Lora SNR ratio in dB (signed float, 0.1 dB precision) |
| size | number | RF packet payload size in bytes (unsigned integer) |
| data | string | Base64 encoded RF packet payload, padded |

Table 3: RXPX object format. Table from [19].

| Name | Type | Function |
|---|---|---|
| time | string | UTC "system" time of the gateway, ISO 8601 "expanded" format |
| lati | number | GPS latitude of the gateway in degree (float, N is +) |
| long | number | GPS latitude of the gateway in degree (float, E is +) |
| alti | number | GPS altitude of the gateway in meter RX (integer) |
| rxnb | number | Number of radio packets received (unsigned integer) |
| rxok | number | Number of radio packets received with a valid PHY CRC |
| rxfw | number | Number of radio packets forwarded (unsigned integer) |
| ackr | number | Percentage of uplink datagrams that were acknowledged |
| dwnb | number | Number of downlink datagrams received (unsigned integer) |
| txnb | number | Number of packets emitted (unsigned integer) |

Table 4: STAT object format [19].

#### 3.3.2 Downlink Protocol

The downlink protocol has similarities with the uplink one, and three different messages are used. Figure 8 shows how messages are exchanged.

The *PULL_DATA* is mostly used for the gateway to collect data (from the NS) that is ready to be sent as downlink to the device. The data exchange is initialized by the gateway because it might be impossible for the NS to send packets to the gateway (if gateway hidden behind NAT). When the gateway initializes the exchange, a data flow channel gets open and packets can flow in both directions. The gateway must periodically send *PULL_DATA* packets for the flow channel to stay open [19]. The structure of this packet type is shown next.

| Bytes | Function |
|---|---|
| 0 | Protocol version |
| 1-2 | Random token |
| 3 | PUSH_DATA identifier 0x02 |
| 4-11 | Gateway MAC address |

Table 5: Pull Data packet format. Table from [19].

The packet used to answer to *PULL_DATA* is the *PULL_ACK*, which also has

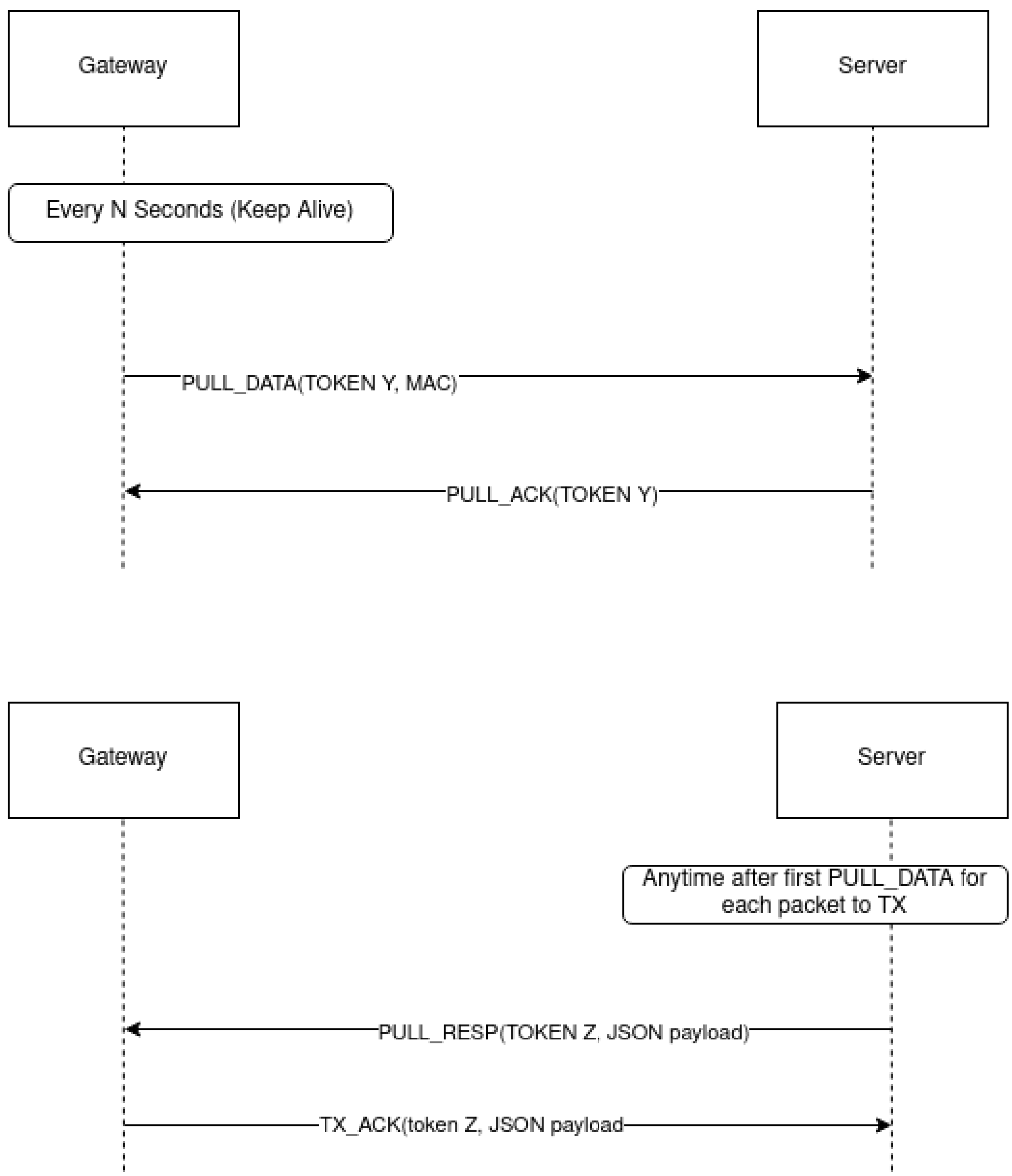


Figure 8: Downlink workflow between device and gateway [19].

the same structure as *PUSH_ACK*. It is also used to confirm that the flow channel is open and the server replies with a *PULL_RESP*.

| Bytes | Function |
|---|---|
| 0 | Protocol version |
| 1-2 | Same random token as PULL_DATA packet to acknowledge |
| 3 | PULL_ACK identifier 0x04 |

Table 6: Pull Ack packet format [19].

The *PULL_RESP* message is the reply sent from the NS to the gateway, and it includes data that must be sent to the device.

| Bytes | Function |
|---|---|
| 0 | Protocol version |
| 1-2 | Random token |
| 3 | PULL_RESP identifier 0x03 |
| 4-end | JSON object, starting with "{," and ending with "}," |

Table 7: Pull Resp packet format [19].

Some NSs also require feedback by the gateway to make sure the message was successfully sent to the device, to avoid duplication. The *TX_ACK* is used for this purpose. An optional payload field can be included to give more detail about why the transmission failed.

| Bytes | Function |
|---|---|
| 0 | Protocol version |
| 1-2 | Same random token as PULL_RESP packet to acknowledge |
| 3 | TX_ACK identifier 0x05 |
| 4-11 | Gateway MAC address |
| 12-end | [optional] JSON object, "{," and ending with "}," |

Table 8: TX Ack packet format. Table from [19].

JSON objects in the previously mentioned Tables 7 must follow specific format, otherwise the packet will be rejected. The following data can be included in such object:

| Name | Type | Function |
|---|---|---|
| imme | bool | Send packet immediately (will ignore tmst & time) |
| tmst | number | Send packet on a certain timestamp value (will ignore time) |
| tmms | number | Send packet at a certain GPS time (GPS synchronization required) |
| freq | number | TX central frequency in MHz (unsigned float, Hz precision) |
| rfch | number | Concentrator “RF chain” used for TX (unsigned integer) |
| powe | number | TX output power in dBm (unsigned integer, dBm precision) |
| modu | string | Modulation identifier “LORA” or “FSK” |
| datr | string | LoRa datarate identifier (eg. SF12BW500) |
| datr | number | FSK datarate (unsigned, in bits per second) |
| codr | string | LoRa ECC coding rate identifier |
| fdev | number | FSK frequency deviation (unsigned integer, in Hz) |
| ipol | bool | Lora modulation polarization inversion |
| prea | number | RF preamble size (unsigned integer) |
| size | number | RF packet payload size in bytes (unsigned integer) |
| data | string | Base64 encoded RF packet payload, padding optional |
| ncrc | bool | If true, disable the CRC of the physical layer (optional) |

Table 9: Downlink object format. Table from [19].

## 3.4 Chirpstack

Chirpstack is an open-source LoRaWAN Network Server stack that provides the essential components to build user-friendly web interfaces for device management. Chirpstack APIs are provided for integration purposes, allowing LoRaWAN stack to be easily integrated with applications.

Chirpstack is composed of three modules that work together for a client to be able to communicate with end-devices. These operate as follows.

### 3.4.1 Gateway Bridge

The Chirpstack Gateway Bridge (CGB) is the module of Chirpstack LoRaWAN stack that converts the Packet Forwarder message payloads into a serialized data format (JSON or Protobuf) and publishes it in a Message Queueing Telemetry Transport (MQTT) server, to which the Network Server subscribes. The message will be published with a topic that indicates the source (by default the gateway MAC number and event type: up, down, status). This module can also be configured to support different types of packet forwarders.

### 3.4.2 Network Server

Once the message has been published in the MQTT server, the Chirpstack Network Server (CNS) (subscribing the topic) will read the message and confirm if it is a duplicate message. The duplication of LoRa frames can happen because a LoRa frame is broadcasted, allowing multiple gateways to read the frame for delivery to the CNS. In this case the CNS will keep the frame from the gateway that has the best connection with the device. The CNS also remembers gateway-device associations for downlink purposes. After dealing with frame duplication it will communicate with the Application Server through google Remote Procedure Call (gRPC).

The versatility of ChirpStack configuration is mostly present at the CNS. The CNS can, for example, be configured so that the connection with the user network is optimized. This can be impactful if the user truly understands how the implemented network works. For instance, when devices are too far away from the gateway the ToA is longer, and if the user is aware of the ToA then devices can be configured to open the RX window later (and not immediately after sending the packet). The receiving window can also become smaller because the user can predict when the packet will arrive to the device. As far as optimization is concerned, the CNS is also where the algorithm responsible for optimizing the LoRa MAC parameters is located. This algorithm is discussed in Section 3.5.

As previously stated, LoRa frames are broadcasted and gateways simply redirect them to the CNS. This means that the CNS can receive not only duplicate frames but also frames from devices that do not belong to its network. To address this issue, two key mechanisms have been created for network device validation.

The first one is Activation By Personalization (ABP) and is by far the simplest but also the most insecure mechanism. Basically the uplink contains a certain number of keys and if these match with the CNS ones, then the package is validated. The other mechanism is known as Over The Air Activation (OTAA). This technique is safer because it follows an handshake process with the server. In this process, both the server and device have a common key and whenever the device wants to deliver data to the NS, it negotiates a pair of keys (changes every session) that are used to validate the packets sent. Although this process is safer, it also takes longer due to key negotiation, specially in cases where the connection is poor.

### 3.4.3 Application Server

At last, there is the ChirpStack Application Server (CAS). It works mostly as a web interface for an easy set up of LoRaWAN network. It allows to configure the mentioned CNS settings and also to add gateways, devices and profiles associated with them. It has also an API to facilitate the integration of device data collection applications, like influxdb, or for data to be sent as a POST to the client application. a

## 3.5 Adaptive Data Rate

The Adaptive Data Rate (ADR) algorithm, as briefly mentioned in 2, uses the datarate and the Signal-to-Noite Ratio (SNR) from the last 20 packets to make decisions regarding datarate changes. This is a non optimal approach because of the time it takes to answer to environmental changes, and can be considered just a quick solution for datarate optimization. To fully understand this issue, a close look at the algorithm used in Chirpstack and The Things Network (TTN) is required [15]. Such algorithm is shown in Algorithm 1.

The idea behind ADR is very simple: if the link budget is high, the datarate can be increased and if the link budget is low, the datarate should be lowered. The algorithm will try to estimate the current SNR of the link, based on the last

20 uplink packets. Each uplink contains the frame counter and SNR. For each of this measurements, the best SNR value is calculated together with the margin (measured SNR minus the required SNR to demodulate a message given its data rate). Given the margin value, the *steps* are defined and the algorithm performs as follows: if $steps > 0$ then the ADR increases the datarate by $steps$ until the *maxDR* (defined at the Network/Application server) is achieved (if there are $steps$ left, the $TXPower$ will be lowered for the remaining steps, until 0); if $steps < 0$ then $TXPower$ increases [16].

For more harsh device localizations, this algorithm ends up having severe problems mainly due to the duty cycle constraints. The duty cycle, by definition, specifies the fraction of time that a resource is used. In this case the resources are the LoRa channels. Each region has a restriction for the usage of each channel. That is, a device can only occupy the channel for a certain amount of time, per *x* time (usually an hour). However, this is a per channel constraint so if the duty cycle is 10% and there are 10 channels, then the device can still achieve 100% up-time. As for Europe, the duty cycle is assigned to sub-band usage, instead of channels (see Table 10). A sub-band can include multiple channels, making it more restrictive.

| Sub-band | Frequency | Duty Cycle |
|---|---|---|
| g | 863.0 – 868.0 MHz | 1% |
| g1 | 868.0 – 868.6 MHz | 1% |
| g2 | 868.7 – 869.2 MHz | 0.1% |
| g3 | 869.4 – 869.65 MHz | 10% |
| g4 | 869.7 – 870.0 MHz | 1% |

Table 10: Europe sub-bands for LoRa [11].

Assuming each band has 1% of duty cycle restriction, and assuming that packet transmission time is 3 seconds, then the largest amount of packets a device can send in an hour (3600 seconds) is 36/3 = 12 packets per sub-band, which is 60 packets per hour considering all sub-bands in Table 10. So, if packets are

```
i ← 0
history[j] ← 0 for j  [0, 19]
offset = 10
threshold = -20, -17.5, -15, -12.5, -10.0, -7.5
DR ← -1, TX ← 1
Function ReceivePacket(mSNR, mDR)
    if DR = -1 then
        DR ← mDR
    end
    history[i] ← mSNR
    i ← i + 1
    if  i = 20 then
        AdjustADR
        i ← 0
    end
end
Function AdjustADR
    margin ← max(history) - threshold[DR] - offset
    steps ← round(margin/3)
    if steps > 0 then
        increase DR by steps until DR = maxDR
        decrease TX by remaining steps until TX = 0
    end
    else if steps < 0 then
        increase TX by steps until TX = 5
    end
end
```

**Algorithm 1:** Pseudo-code of the ADR in Chirpstack and TTN [15]

sent every minute, for example, then the device can change the data rate after 20 minutes (note, however, that data is usually sent more space apart to preserve battery). The problem is that in scenarios where the environment changes, and packet delivery ratio drops, this will take longer and collected measurements may no longer be useful (outdated). The response time of the ADR algorithm is too slow do deal with environment changes. A few approaches have been proposed to improve this. In [15] the authors mention that the ADR algorithm should consider the highest recorded SNR (over the last 20 packets) but this can result in a poor answer if that value is an outlier. For this reason the authors propose a mechanism to rule out the outliers, and consider also scalability. Although this is not a perfect solution, it does fix some of the problems that the standard ADR algorithm presents.

The ideal solution would be one that is able to learn immediate adjustments, according to some prediction, in order to avoid losses. This can be theoretically achieved using deep learning algorithms. This is not an easy task to achieve, but training and learning from the network/environment is expected to achieve a better answer than ADR. There is an early proposal using this approach in [9], where authors try to optimize a LoRaWAN environment using Q-Learning. Results show that Reinforcement and Deep Learning are strong candidates in replacing the ADR, as data rate is optimized more effectively. In [21] the authors propose a Chirpstack-based framework that allows these learning mechanisms to be integrated into the LoRaWAN stack. This is a key contribution for these kind of algorithms to be applied in real scenarios.

# 4 LoRa Gateway Placement Problem

The research work will be carried out in three parts: *i*) Development of an optimization model to plan for the best placement of gateways; *ii*) Simulation of a LoRaWAN, for evaluation of placement choices; *iii*) Physical deployment of a use case. These are discussed next.

## 4.1 Mathematical Optimization Model

### 4.1.1 Definitions and Notation

**Definition 1 (LoRa Bit Rate)** *LoRa modulation uses chirp spread spectrum signals to modulate data. The spreading factor determines the number of chirps contained in each symbol, given by $2^{SF}$. Therefore, $\frac{BW}{2^{SF}}$, where BW is the bandwidth, gives the symbol rate. Since the number of raw bits that can be encoded by a symbol is SF, and given a coding rate CR, the useful bit rate for a given SF will be $R_{SF} = SF \times \frac{BW}{2^{SF}} \times CR$.*

The bandwidth (BW) in LoRa can be 125kHz, 250kHz or 500kHz.

**Definition 2 (Transmission Duty Cycle - TDC)** *Ratio of the cumulated sum of transmission times per observation period. The maximum duty cycle ends up being the maximum percentage of time during which an end device can occupy a channel, per hour.*

**Definition 3 (Packet Reception Ratio - PRR)** *Probability of correct package reception, at a gateway $g \in \mathscr{G}$, assuming an average signal to noise ratio (SNR) for a particular distance between a device $d \in \mathscr{D}$ and the gateway, and assuming a certain SF for transmission.*

Besides the PRR, a no collision probability is also considered by many authors, as in [27]. The traditional ALOHA is usually the underlying medium access protocol.

**Definition 4 (Feasible Spreading Factors)** *A spreading factor belongs to the set of feasible spreading factors of device $d \in \mathscr{D}$ for communication with location $l$, denoted by $\mathscr{S}_d^l$, if and only if it can be used for $d$ to communicate with location $l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g$.*

**Definition 5 (Most Critical Device)** *Assuming $\mathscr{L}^g$ to be the set of possible locations for a gateway $g \in \mathscr{G}$, the most critical device in the coverage area of $g \in \mathscr{G}$, when placed in location $l \in \mathscr{L}^g$, is given by $\Delta_g^l = \arg\max_{d \in \mathscr{C}_g^l}\{B_d^l\}$, where $B_d^l$ is the relative battery consumption of device $d \in \mathscr{D}$ when sending a packet to location $l$, and $\mathscr{C}_g^l = \{d \in \mathscr{D} : R_{d,s^*}^l \times PRR_{d,s^*}^l \times N_{d,s^*}^l \geq R_{d,s'^*}^{l'} \times PRR_{d,s'^*}^{l'} \times N_{d,s'^*}^{l'}, \forall l' \neq l\}$ is the set of devices that are expected to adjust their SF to $s^*$ (optimal SF) for communication with gateway $g$ at location $l$.*

The $R_{d,s^*}^l$, $PRR_{d,s^*}^l$ and $N_{d,s^*}^l$ are the bit rate, PRR and the no-colision probability when device $d$ is communicating to location $l$ using SF $s^*$, the optimal SF assigned by the ADR mechanism.

**Definition 6 (LoRa Gateway Placement Problem - LGP Problem)** *Given a set of end node devices $\mathscr{D}$ and a set of gateways $\mathscr{G}$, find the places for gateways that lead to a* ***fair minimization of energy depletion*** *in critical devices (considering a set of packets to be sent) while also ensuring that: i) all devices are covered and; ii) device transmission does not violate the TDC. More formally, let us assume that $\chi^U = \{\chi_1, \chi_2, ..., \chi_{|\chi^U|}\}$ is the universe set of all feasible gateway-place assignments. Let us also consider a cost function $f : \chi^U \rightarrow \Re^+$ defined by:*

$$f(\chi_i) = \arg\max_{<g,l> \in \chi_i}\{\frac{P_{\Delta_g^l} \times L_{\Delta_g^l}}{R_{\Delta_g^l,s^*}^l \times PRR_{\Delta_g^l,s^*}^l \times N_{\Delta_g^l,s^*}^l} \times B_d^l\} \tag{4}$$

*where $P_d$ is the number of packets per TDC to be sent by device $d$, $L_d$ is the average packet length, and the device being considered is the most critical one. Then the most energetically fair gateway placement is given by:*

$$\chi_i^* = \arg\min_{\chi_i \in \chi^U} \{f(\chi_i)\} \tag{5}$$

*A gateway-place assignment is considered to be feasible if all devices are covered and no device transmission violates the TDC.*

That is, from all possible gateway-place assignments, the one that provides the lowest upper bound on depletions at critical devices is the one that should be selected.

#### 4.1.2 Optimization Problem Formulation

Let us assume the following known information:

| | |
|---|---|
| $\mathscr{D}$ | Set of LoRa communicating devices, where $d \in \mathscr{D}$ denotes a specific device. |
| $\mathscr{G}$ | Set of available LoRa gateways, where $g \in \mathscr{G}$ denotes a specific gateway. |
| $B_d^l$ | Relative battery consumption of device $d \in \mathscr{D}$, when communicating with a gateway at location $l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g$, considering the time required for the transmission of all packets; $0 \leq B_d^l \leq 1$. |
| $\mathscr{S}$ | Set of SF-CR configurations, where $s \in \mathscr{S}$ denotes a specific configuration. |
| $\mathscr{L}^g$ | Set of possible locations for gateway $g \in \mathscr{G}$. |
| $\mathscr{C}_l$ | Set of covered devices when location $l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g$ is in use. |
| $\mathscr{S}_d^l$ | Set of SFs that can be used for device $d \in \mathscr{D}$ to communicate with a gateway in location $l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g$, $\mathscr{S}_d^l \subseteq \mathscr{S}$. |

Let us also consider the following variables:

| | |
|---|---|
| $\sigma_d^l$ | One if device $d \in \mathscr{D}$ is communicating with location $l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g$; zero otherwise. |
| $\varphi_d^{g,l}$ | One if $d \in \mathscr{D}$ is the most critical device, from all devices covered by gateway $g \in \mathscr{G}$ placed at location $l \in \mathscr{L}^g$; zero otherwise. |
| $\phi^{g,l}$ | One if gateway $g \in \mathscr{G}$ is to be placed at location $l \in \mathscr{L}^g$; zero otherwise. |
| $\Pi$ | Most difficult transmission conditions among all critical devices (upper bound). |

The LGP problem can be solved using the following objective function:

– Objective function:

$$\text{Minimize } \Pi \tag{6}$$

The following set of constraints must be fulfilled:

– Allocation of gateways to places and covering of all devices:

$$\sum_{\{l \in \mathscr{L}^g\}} \phi^{g,l} = 1, \forall g \in \mathscr{G} \tag{7}$$

$$\sum_{\{g \in \mathscr{G}\}} \sum_{\{l \in \mathscr{L}^g : d \in C_l\}} \sigma_d^l = 1, \forall d \in \mathscr{D} \tag{8}$$

$$\sigma_d^l \leq \sum_{\{g \in \mathscr{G} : l \in \mathscr{L}^g\}} \phi^{g,l}, \forall d \in \mathscr{D}, \forall l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g : d \in \mathscr{C}_l \tag{9}$$

$$\begin{aligned} &(R_{d,s^*}^l \times PRR_{d,s^*}^l \times N_{d,s^*}^l) \times \sigma_d^l \geq \\ &\geq (R_{d,s'^*}^{l'} \times PRR_{d,s'^*}^{l'} \times N_{d,s'^*}^{l'}) - \Theta \times (1 - \sigma_d^l), \\ &, \forall d \in \mathscr{D}, \forall l, l' \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g : d \in \mathscr{C}_l \wedge d \in \mathscr{C}_{l'} \end{aligned} \tag{10}$$

where Θ is a big value, required for constraints to hold true regardless of the gateway location a device is communicating with. Constraints (7) place gateways at one of the allowed locations, Constraints (8) ensure that all devices are covered. Constraints (9) ensures that communication with a location occurs only if there is a gateway placed in there. Constraints (10) ensure that devices communicate with the gateway location providing the best conditions.

– Most critical device depletion:

$$\sum_{\{d\in\mathscr{D}:d\in\mathscr{C}_l\}} \varphi_d^{g,l} = \phi^{g,l}, \forall g \in \mathscr{G}, \forall l \in \mathscr{L}^g \tag{11}$$

$$\varphi_d^{g,l} \leq \sigma_d^l, \forall g \in \mathscr{G}, \forall l \in \mathscr{L}^g, \forall d \in \mathscr{D} \tag{12}$$

$$\begin{aligned} B_d^l \times \varphi_d^{g,l} \geq B_{d'}^l \times \sigma_{d'}^l - \Theta \times (1 - \varphi_d^{g,l}), \\ , \forall g \in \mathscr{G}, l \in \mathscr{L}^g, \forall d, d' \in \mathscr{C}_l \end{aligned} \tag{13}$$

where Θ is a big value, required for constraints to hold true regardless of a node being considered critical or not, which must hold in mathematical optimization models. Constraints (11) and (12) determine the critical device per gateway cover, while Constraints (13) ensure that it is the one with higher relative energy consumption.

$$\begin{aligned} \Pi \geq \varphi_d^{g,l} \times \frac{P_d \times L_d}{R_{d,s*}^l \times PRR_{d,s*}^l \times N_{d,s*}^l} \times B_d^l, \\ , \forall g \in \mathscr{G}, l \in \mathscr{L}^g, \forall d \in \mathscr{D} \end{aligned} \tag{14}$$

Constraints (14) determine the most difficult transmission conditions among all critical devices.

– Non-negativity assignment to variables:

$$\varphi_d^{g,l}, \phi^{g,l}, \sigma_d^l \in \{0,1\}; \Pi \in \Re^+. \tag{15}$$

This optimization problem can be solved using packages like CPLEX or Gurobi, [10, 13], which find the optimal solution given an instance of the problem.

## 4.2 LoRaWAN Simulation

Simulators are a popular tool to understand the behavior of systems over time. In this research the goals are to explore the benefits of an adequate gateway placement and understand how taking fairness into account, and possibly gateway reallocation, can make the difference over time. The network scenario will be CEOT's LoRaWAN monitorization system, which will be deployed in an orchard. Such scenario is quite common as a data collection network in agriculture. The overall system includes around 25 static devices and one gateway (in the first place), which can be placed anywhere or within the range of a defined place. For this to be accomplished, a few simulators were considered for analysis.

Due to the increasing popularity of LoRa over the last few years, the number (and quality) of LoRa simulators has been increasing, while catching up the latest (more mature) protocols. Before going deeply into the evaluation, one has to set up a list of requirements that will serve as reference.

- Must have device energy profile.
- Must support multiple gateways.
- Must support 8 channel gateways.
- Must support 863-870MHz rules
- Must include ADR.

- Should support LoRa channel semi-orthogonality.

- Open-source, preferably.

It is important to mention that the community support and documentation will be key point when deciding for right simulator.

In [20] the authors list the most popular LoRaWAN simulators and explain with detail their design requirements and limitations. Although this list is outdated, it gives a good idea of the reasoning behind choosing one of them. The article delves into the features of each simulator but fails to demonstrate their support (community and documentation). The LoRa functionalities in each simulator, mentioned in [20], are discussed next.

- LoRaSim: It is a discrete-event simulator implemented using SimPy. The user can simulate a network of $N$ nodes and $M$ sinks, spaced either randomly or in a 2D grid space. It provides two evaluation metrics: Data Extraction Rate (DER) and Network Energy Consumption (NEC), whose output is related with the network as a whole.

- LoRaWANSim: It is an expansion of LoRaSim, adding features like bidirectional communication and perfect SF orthogonality assumption. However, it is not Open Source. To ensure the accuracy of their simulation model, the authors first conducted experiments in an RF shielded lab to assess the impact of two concurrent LoRa signals. The built simulator is not a system level simulator and focuses only in uplink traffic considering a single EU gateway network.

- LoRaMatlab: It is a closed-source simulator. Assumes perfect SF orthogonality, considers duty cycle restrictions and matches the SF used for uplink traffic to the downlink feedback traffic.

- NS3 LoRaWAN: It is an NS3 module that can simulate (with high detail) a LoRa network. It is capable of studying multi-gateway networks and bi-directional traffic, and supports device energy profiles. The simulator's error model was derived from the base-band simulations of a LoRa transceiver over an Additive White Gaussian Noise (AWGN) channel. It is important to mention that, as far as the documentation indicates, it is possible to build our own error models. This will allow us to build an error model (for signal loss) based on the measurements taken from CEOT's orchard.

- LoRaEnergySim: It is a simulator that focuses on energy consumption, and supports both the ADR scheme and downlink messages. The simulator supports two channel models, namely a long-distance model with the shadowing and a COST 231 model. Also assumes perfect SF orthogonality and has bases on the gateway model of the WiMOD IC880A gateway.

- OMNeT++ Flora: It has a model focused purely on LoRa, known as FLORA, that can simulate the PHY and MAC layers. It also supports bidirectional communication and can simulate the back-haul network. Flora can also perform energy efficiency simulations.

Besides comparing the features of each simulator, the author in [20] also specify the scenarios for which the simulators perform the best. The scenarios include smart city, smart cold chain and environmental monitoring. As for the first and second scenarios, NS3 and OMNeT++ are the most complete, as these allow multiple gateways and custom built error models to be included. The authors also mention that the NS3 weakness is the fact that it doesn't provide an energy profile for the devices, although that has been implemented recently. Regarding the 3rd scenario, the authors defend that LoRaSim is the most recommended due to the fact that it supports downlink traffic.

The author in [20] point out that the most useful general simulators are OMNeT++ and NS3. The performance of these simulators was tested in [17, 25, 30] and for most cases NS3 presents the highest performance. NS3 is in fact more complete, faster, detailed and has more support from the community (see Figure 9), but OMNeT++ is easier to work with, has better graphical interface and documentation. Although both options are great, OMNeT++ was chosen due to the extensive and detailed tutorials and documentation when compared to NS3. The graphical interface also played an important role because it helps us understand how the network is flowing in a simulation environment.

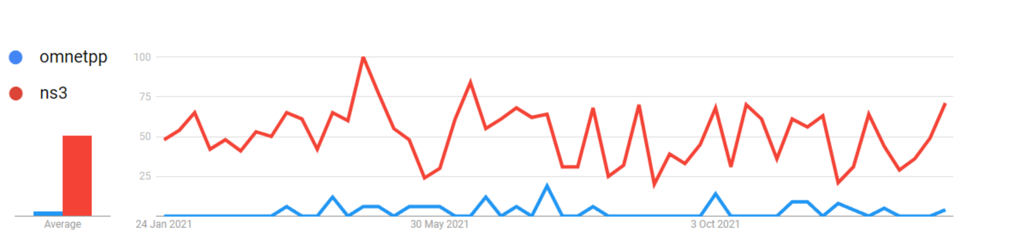


Figure 9: OMNeT++ and NS3 trend on google in the last 12 months. Information obtained using [1]

## 4.3 Practical Use Case

The objective of this research is to find the best approach for the gateway positioning in a LoRa network that includes multiple static end nodes, and transcribe the obtained results to an existing and functional network.

The physical network to be deployed by CEOT follows the stars-to-starts topology and currently has 25x Dragino Temperature & Humidity sensors, 1 gateway (RAK 7244 WisGate Developer D4), and a server that hosts the ChirpStack modules mentioned in 3.4. Two data related softwares were included: *i*) influxdb, which is a well established framework in the IoT area that allows to collect data and store it; *ii*) application developed in CEOT so that data can be visualized in real time [2].

The devices are scattered through the orchard as shown in Figure 11, and with a maximum distance of roughly 100 m between gateway and furthest device. Before the network deployment, measurements were done in site to access the influence of tree canopies' scattering/absorption and distance in the radio signal quality. Different quality parameters (e.g. SNR, PDR, etc) where measured as a function of distance between end-nodes and gateway [8]. Based on these measurements we found that at a distance show that at a distance of 70-100m the ADR algorithm tends to use datarate 0, while for 20m distance the datarate 5 is used. This means that the battery of device 23 discharges roughly 8 times faster than device 3. Given this steep gradient in battery consumption, energy efficiency planning is expected to bring significant benefits in improving the quality of the network.

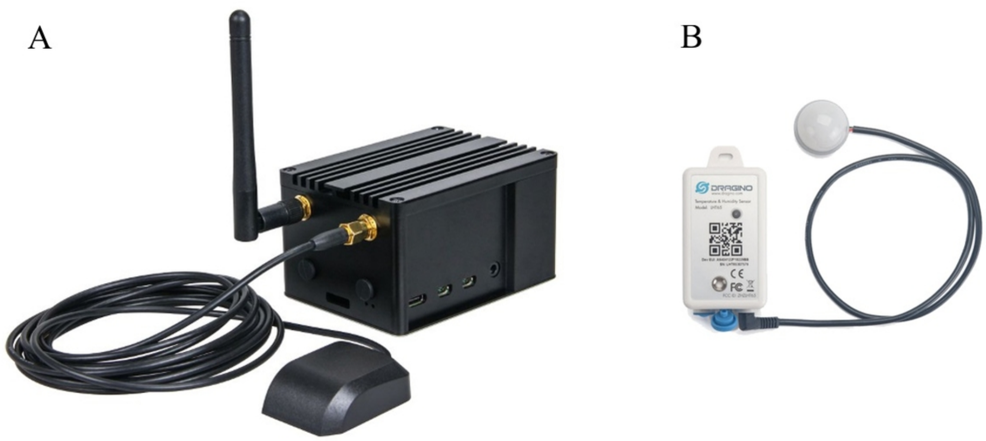


Figure 10: A - RAK 7244 gateway and B - LHT65.

#### 4.3.1 Dragino Temperature & Humidity

The Dragino Temperature and & Humidity (LHT65) module is an environment module that comes with a built-in SHT20 Temperature & Humidity sensor and has a external sensor that allows external sensors to be connected, such as an illumination sensor. This device works both as a data logger, recording up to 3200 environment measurements and also as a wireless sensor for network applications. The battery of this device is a 2400mAh non chargeable battery that is ratted for up to 10 years of utilization under optimal conditions (LoS communications and sparse uplink rates).

Built-in temperature sensor information:

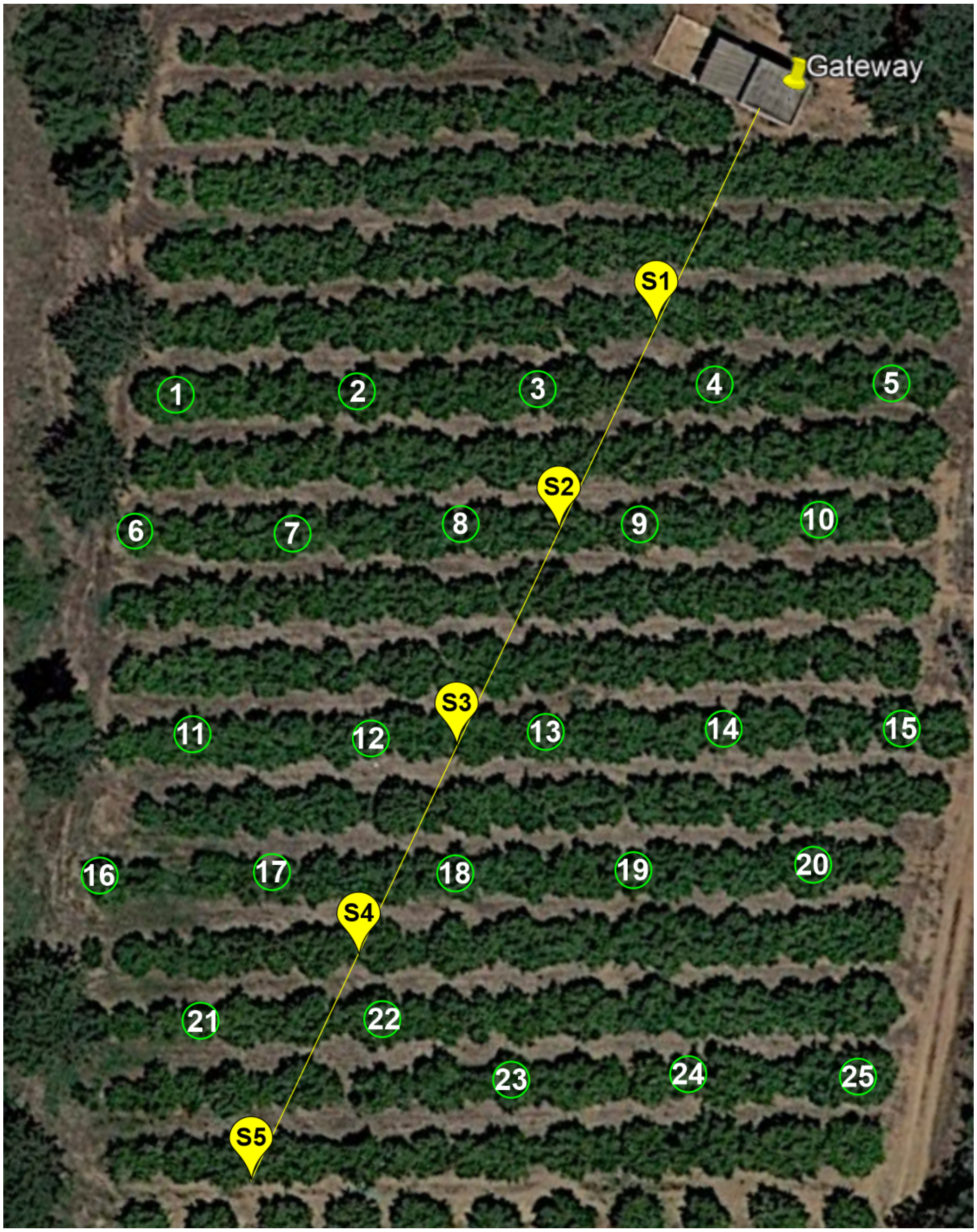


Figure 11: LoRaWAN network visualization. The green-tree shaped markers indicate the position of the LHT65 sensors while the yellow markers indicate the position of the preliminary communication's quality measurements

- Accuracy tolerance : Typ $\pm$ 0.3 °C
- Long term drift: $<$ 0.02 ° C/yr
- Operating range: -40 $\sim$ 125 °C

Built-in humidity sensor information:

- Resolution: 0.04 % RH
- Accuracy tolerance: Typ $\pm$ 3 % PH
- Long term drift: $<$ 0.02 °C/yr
- Operating range: 0 $\sim$ 96 % PH

External illumination sensor information:

- Resolution: 1 lx
- Range: 0-65535 lx
- Operating range: -40 °C $\sim$ 85 °C

With this device setup, the power consumption of the device is 4$\mu$A in idle mode and 130mA when transmitting at maximum power. Using the Dragino energy consumption prediction spreadsheet [3], it is possible to estimate how long the device will survive, given datarate and uplink intervals. Table 11 shows the device's theoretical battery expectation assuming an uplink interval of 20 minutes. As shown, the life expectation is highly affected by the datarate, with a difference as high as 86% between using DR=5 and DR=0.

| datarate | life expectancy (years) |
|---|---|
| 0 | 1.7 |
| 1 | 2.9 |
| 2 | 5.3 |
| 3 | 7.7 |
| 4 | 10.2 |
| 5 | 12.5 |

Table 11: Battery lifetime expectation for a LHT65 device at different datarates.

#### 4.3.2 LSE01

The Dragino LSE01 is a LoRaWAN Soil Moisture & EC sensor module that is designed to measure the soil moisture of saline-alkaline and loamy soil. This is quite similar to the LHT65, works as a data logger and as a wireless sensor for network applications [4]. The device specifications are the following.

| Parameter | Soil Mosture | Soil Conductivity | Soil Temperature |
|---|---|---|---|
| Range | 0-100.00% | 0-20000$\mu$s/cm (25°C)(0-20.0EC) | -40.00°C∼85°C |
| Unit | V/V% | $\mu$S/cm | °C |
| Resolution | 0.01% | 1 $\mu$S/cm | 0.01°C |
| Accuracy | ∼3% (0-53%) ∼5% (>53%) | 2%FS | -10°C∼50° : < 0.3°C All other: <0.6°C |
| Measure Method | FDR, with temperature & EC compensate | Conductivity, with temperature compensate | RTD, and calibrate |

Table 12: LSE01 device specifications.

This device is powered by a 8500mA Li-SOCI2 battery that is designed to make it autonomous for more than 10 years. Using the same spreadsheet from Dragino, with the same uplink intervals as the LHT65, it is possible to obtain the battery lifetime expectation for this device, given the datarate. Table 13 shows that the highest battery lifetime expectation is lower than LHT65 but it suffers less energy consumption from datarate changes than LHT65, lasting 45% less when changed from datarate 5 to 0. The reason why the datarate has lower influence here is the fact that the sensors from this device consumes more energy while measuring, making the datarate change less relevant in total.

| datarate | life expectancy (years) |
|---|---|
| 0 | 3.3 |
| 1 | 4.1 |
| 2 | 5.1 |
| 3 | 5.5 |
| 4 | 5.8 |
| 5 | 6 |

Table 13: Battery lifetime expectation for a LHT65 device at different data rates.

#### 4.3.3 RAK 7244C WisGate Developer D4

This gateway consists of a Raspberry Pi 4, a RAK2245 Pi HAT, GPS and 4G/LTE communication module and a heat sink for better performance and thermal heat dissipation. The RAK 2245 Pi HAT uses a SX1301 RF front-end chip from Semtech that is a LoRa processing engine able to receive up to 8 LoRa packets, sent using different spreading factors on different channels. Information can be summarized as follows:

- Full LoRaWAN Stack support (version 1.0.2)
- Supports for 8 channels and SF (SF7-SF12)
- Frequency band support for multiple regions including EU433 and EU868
- Tx Power: 27 dBm Max
- RX Sensitivity: -139dBm

# 5 Analysis of Results

## 5.1 Scenario Setup

The architecture implemented follows the one proposed by the LoRa Alliance. The Flora module for OMNet, which allows the simulation of end-to-end LoRa networks, was used to build the intended simulation model. This module assumes a network structure that follows the LoRa Alliance proposal, which means that by default it implements a stars-to-stars network topology having devices, gateways and a central network server (see Figure 12), where each component has a lot of customization options. This module also includes energy consumption models, which can be changed. For the implementation of our simulation model the default energy consumption values were used (see Table 14), as there is no information available regarding the real energy consumption of the devices used for practical deployment.

The structure of the network first took into consideration the shape of the practical deployment field. Orange trees have very dense canopies (with high water concentrations in the leaves) leading to significant signal scattering and absorption. The high values of attenuation that were experimentally measured in the orchard could not be reproduced by the software's Oulu and Okumura-Hata path loss model for a simulated orchard with the same dimensions. In order to reproduce the attenuation measured, we assumed a simulated environment 10 times the size of the real case. By doing so, we were able to simulate considerable power consumption differences between end-nodes at different distances of the gateway. The whole network structure is as follows:

- Orange grove of 45.6-hectares of square-shaped field with 25 sensing devices;
- Orange trees are equally spaced, each having off-the-shelf temperature and luminosity sensors, mounted inside the tree's canopy, to evaluate the impact of local conditions on fruit development;

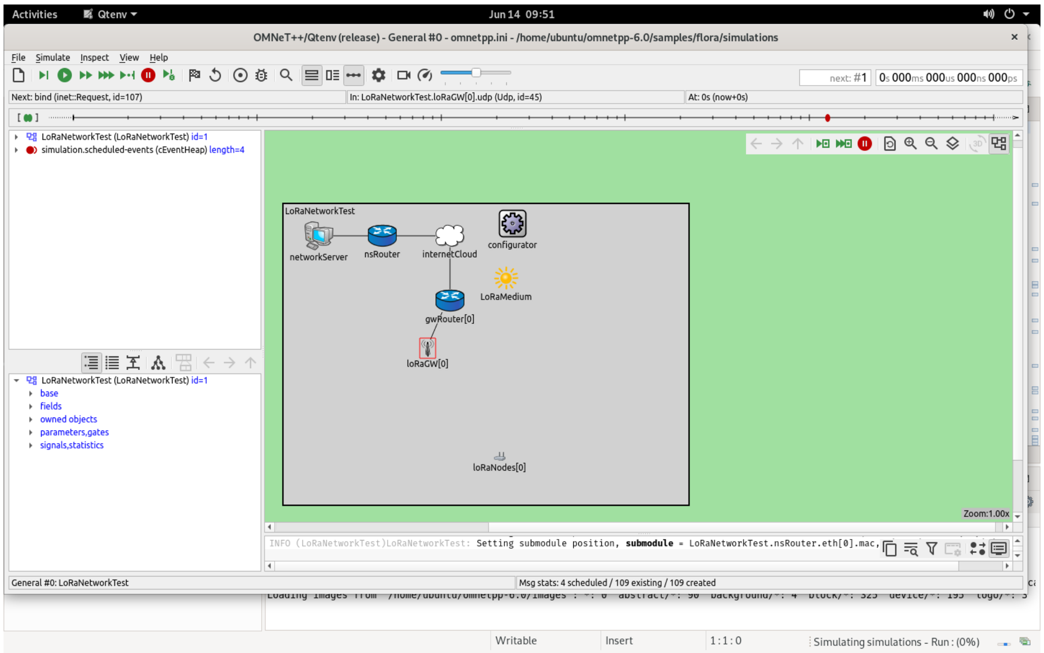


Figure 12: LoRaWAN network implementation on OMNet using Flora module.

- Each device is 150m horizontally and 190m vertically away from each other
- 2 gateways and 5 different feasible locations for both gateways.
- Oulu path loss model, using $n = 2.32$, $B = 128.95$, $\sigma = 7.8$ and antenna gain of 3dBi, similarly to [24].
- ADR algorithm for SF optimization;
- Energy model provided by Flora, following values in Table 14.

## 5.2 Results Discussion

The first step is to use the simulation model to collect $R^l_{d,s^*}$, $PRR^l_{d,s^*}$ and $N^l_{d,s^*}$, which depend on path loss conditions, and $B^l_d$, resulting from the energy model in use, $\forall l \in \bigcup_{\{g \in \mathscr{G}\}} \mathscr{L}^g$ and $\forall d \in \mathscr{D}$. This information is input information to the

| Mode | Power Consumption (W) |
|---|---|
| Off | 0 |
| Sleep | 0.001 |
| Switching | 0.002 |
| Receiver, idle | 0.002 |
| Receiver, busy | 0.005 |
| Receiver, receiving | 0.01 |
| Receiver, receiving preamble | 0.01 |
| Receiver, receiving header | 0.01 |
| Receiver, receiving data | 0.01 |
| Transmitter, idle | 0.002 |
| Transmitter, transmitting | 0.1 |
| Transmitter, transmitting preamble | 0.1 |
| Transmitter, transmitting header | 0.1 |
| Transmitter, transmitting data | 0.1 |

Table 14: Flora Energy Consumption Model.

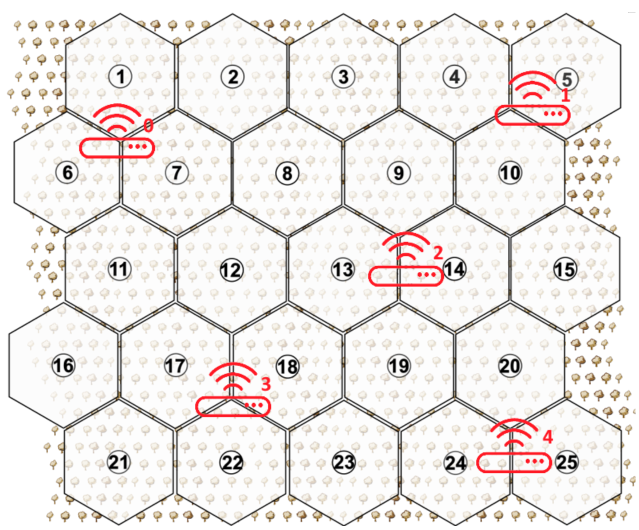


Figure 13: Schematic representation of the gateways and end node locations.

optimization model. Such characterization of transmission conditions and energy consumption, from every device towards every possible gateway location, is done considering the transmission of 10000 packets per device.

The next step is to determine the optimal gateway placement using the mathematical optimization model, and then run the simulation considering the placement of gateways found by the optimization model. The gateways viable positions were randomly selected, and these can be seen in Table 15.

| Identification | x(m) | y(m) |
|---|---|---|
| 0 | 70 | 150 |
| 1 | 600 | 40 |
| 2 | 360 | 560 |
| 3 | 210 | 600 |
| 4 | 560 | 740 |

Table 15: Possible Gateway Locations.

The results given by the optimization model are then compared against other placements. Table 16 summarizes the obtained results, where the placement resulting from the optimization model (and impact of that choice) is displayed in bold. Results were similar for a coverage cutoff of 0% (all devices covered by all gateway locations, when in use) and 10% (a device not able to transmit at least 10% of its packets, towards a given gateway location, is considered uncovered).

Devices end up transmitting a different amount of packets, and for this reason we cannot look at energy consumption in an isolated way. For this reason the results regarding energy consumption per packet are the ones included in Table 16. Results show that the optimization model ends up being capable of finding the best places for gateways, when compared with other possible locations. This is because it was able to select one of the gateway positioning combinations presenting the lowest value in the “Largest” column, meaning that the worst energy consumption per packet is minimized. This solution is one of the fairest solutions because minimizing such upper bound (worst energy consumption ) ends up balancing energy consumption among devices, extending network lifetime. The average energy consumption per packet is also one of the lowest, ensuring energy saving

| Gateways positions | Energy Consumption / Packet Data Rate: Average | Standard Deviation | Lowest | Largest |
|---|---|---|---|---|
| 0 1 | 0,41 | 0,23 | 0,073 | 0,86 |
| 0 2 | 0,25 | 0,12 | 0,07 | 0,62 |
| **0 3** | **0,28** | **0,13** | **0,08** | **0,53** |
| 0 4 | 0,29 | 0,13 | 0,08 | 0,56 |
| 1 2 | 0,29 | 0,16 | 0,07 | 0,70 |
| 1 3 | 0,27 | 0,14 | 0,07 | 0,53 |
| 1 4 | 0,39 | 0,20 | 0,07 | 0,73 |
| 2 3 | 0,28 | 0,17 | 0,07 | 0,66 |
| 2 4 | 0,36 | 0,21 | 0,07 | 0,87 |
| 3 4 | 0,40 | 0,25 | 0,08 | 0,96 |

Table 16: Energy Consumption per Packet (2 GWs).

in general.

These results allows us to conclude that basing the decision on the most critical nodes, one per coveraged range, and make placements that lead to the minimization of the most difficult transmission conditions among critical devices, is adequate for such kind of deployment where in the long-run different devices will be communicating in parallel, with the same gateway or different gateways, and sharing the spectrum. Any attempt to minimize the sum of device's energy consumption, or maximize the overall throughput, would not be be appropriate in this context.

Relocating gateways in already built networks may not be an easy task, due to monitoring and optimization requirements, but we are likely to see environments that depend on this being built around such networks (e.g. using semi-mobile gateways). Such kind of optimization allows for better data acquisition and, therefore, better quality of the final product. In addition, an optimized energy consumption, particularly considering fairness, allows for more autonomous environments because device changes will occur less frequently. It is in these situations that the proposal made will shine.

# 6 Conclusion

The use of IoT in agriculture, for monitoring and data collection, has been growing in the last years thus pushing development towards new technologies of lower energy consumption, while ensuring high area coverage. LoRa is included in these new technologies.

To further improve the quality of LoRa networks this work studied an approach that aims to optimize the gateway distribution through the field of the most critical devices thus ensuring that the whole network remains fully operational over longer periods. The results show that the proposed optimization model proposed is in fact adequate for planning the gateway location in a LoRaWAN network. While this type of approach requires a lot of information, the location of gateways is typically limited in already built environments. In new environments, where monitoring and optimization play a significant role, these will be built around network constraints that contemplate the location of gateways. In new deployments this proposal is expected to shine due to the ability to define the best positions for the gateways, those that do not compromise the packet delivery ratio of devices while ensuring energy depletion fairness to increase network lifetime.

Planning is done offline, for any real scenario, and applied when appropriate, allowing prior validation of gateways placement. Such pipeline can also be used to anticipate any gateway rearrangement need to ensure the extension of the lifetime of the network, being only required to change the input information. This allows us to conclude that there is in fact a place for this kind of approach in a real deployment scenarios.

## Resulting Publications

- Bruno Mendes, Dário Passos and Noélia Correia, "On the Optimization of LoRaWAN Gateway Placement in Wide Area Monitoring Systems", International Federation for Information Processing (IFIP) International Internet

OF Things (IOT) Conference, October 2022.

- Bruno Mendes, Dário Passos and Noélia Correia, “Coverage Characterization of LoRaWAN Sensor Networks for Citrus Orchard Monitoring”, International Young Engineers Forum on Electrical and Computer Engineering (YEF-ECE), July 2022

- Bruno Mendes, Shani du Plessis, Dário Passos and Noélia Correia, “On the Integration of Transmission Optimization Components in LoRaWAN Stack”, International Conference on Communication and Intelligent Systems (ICCIS), December 2021.

# References

[1] URL: https://trends.google.com/trends/explore?q=omnetpp,ns3.

[2] URL: https://iotdash.ceot.tech/.

[3] URL: https://www.dragino.com/downloads/downloads/LoRa_End_Node/Battery_Analyze/DRAGINO_Battery_Life_Guide.pdf.

[4] URL: https://www.dragino.com/downloads/downloads/LoRa_End_Node/LSE01/LoRaWAN_Soil_Moisture_%5C%26_EC_Sensor_UserManual_v1.5.pdf.

[5] LoRa Alliance. *What is LoRaWAN® Specification*. Oct. 2021. URL: https://lora-alliance.org/about-lorawan/.

[6] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. "LoRaWAN Modeling and MCS Allocation to Satisfy Heterogeneous QoS Requirements". In: *Sensors* 19.19 (2019). ISSN: 1424-8220. DOI: 10.3390/s19194204. URL: https://www.mdpi.com/1424-8220/19/19/4204.

[7] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. *Lorawan modeling and MCS allocation to satisfy heterogeneous QoS requirements*. Sept. 2019. URL: https://www.mdpi.com/1424-8220/19/19/4204/htm#.

[8] Noélia Correia Bruno Mendes Dário Passos. "Coverage Characterization of LoRaWAN Sensor Networks for Citrus Orchard Monitoring". In: *International Young Engineers Forum on Electrical and Computer Engineering*. 2022.

[9] Rodrigo Carvalho, Faroq Al-Tam, and Noélia Correia. "Q-Learning ADR Agent for LoRaWAN Optimization". In: *2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. 2021, pp. 104–108. DOI: 10.1109/IAICT52856.2021.9532518.

[10] IBM ILOG Cplex. "V12. 1: User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53 (2009), p. 157.

[11] *Duty cycle*. URL: https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/.

[12] Rui Fernandes, Miguel Luís, and Susana Sargento. "Large-Scale LoRa Networks: A Mode Adaptive Protocol". In: *IEEE Internet of Things Journal* 8.17 (2021), pp. 13487–13502. DOI: 10.1109/JIOT.2021.3064932.

[13] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2022. URL: https://www.gurobi.com.

[14] Sebastián Gutiérrez et al. "Smart Mobile LoRa Agriculture System based on Internet of Things". In: *2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*. 2019, pp. 1–6. DOI: 10.1109/CONCAPANXXXIX47272.2019.8977109.

[15] Vojtěch Hauser and Tomáš Hégr. "Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure". In: *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*. 2017, pp. 85–90. DOI: 10.1109/FiCloud.2017.47.

[16] The Things Industries. *Adaptive Data Rate*. URL: https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/.

[17] Muhammad Amir Khan, Halabi Hasbullah, and Babar Nazir. "Recent open source wireless sensor network supporting simulators: A performance comparison". In: *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*. 2014, pp. 324–328. DOI: 10.1109/I4CT.2014.6914198.

[18] Sungwook Ko et al. "LoRa network performance comparison between open area and tree farm based on PHY factors". In: *2018 IEEE Sensors Applications Symposium (SAS)*. 2018, pp. 1–6. DOI: 10.1109/SAS.2018.8336763.

[19] Lora-net. *PACKET_FORWARDER/PROTOCOL.TXT at master · Lora-net/packet_forwarder*. URL: https://github.com/Lora-net/packet%5C_forwarder/blob/master/PROTOCOL.TXT#L358.

[20] Jaco M. Marais, Adnan M. Abu-Mahfouz, and Gerhard P. Hancke. "A Review of LoRaWAN Simulators: Design Requirements and Limitations". In: *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. 2019, pp. 1–6. DOI: 10.1109/IMITEC45504.2019.9015882.

[21] Bruno Mendes et al. "Framework for the Integration of Transmission Optimization Components into LoRaWAN Stack". In: *Communication and Intelligent Systems*. Ed. by Harish Sharma et al. Singapore: Springer Nature Singapore, 2022, pp. 421–431. ISBN: 978-981-19-2130-8.

[22] myDevicesIoT. *cayenne-docs/LORA.md at master · myDevicesIoT/cayenne-docs*. URL: https://github.com/myDevicesIoT/cayenne-docs/blob/master/docs/LORA.md.

[23] The Things Network. *Device Classes*. URL: https://www.thethingsnetwork.org/docs/lorawan/classes/.

[24] Juha Petajajarvi et al. "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology". In: *2015 14th International Conference on ITS Telecommunications (ITST)*. 2015, pp. 55–59. DOI: 10.1109/ITST.2015.7377400.

[25] Atta ur Rehman Khana, Sardar M. Bilalb, and Mazliza Othmana. *A Performance Comparison of Network Simulators for Wireless Networks*. 2013. arXiv: 1307.4129 [cs.NI].

[26] *RF Wireless World*. URL: https://www.rfwireless-world.com/Terminology/LoRaWAN-Spreading-factor-Range-and-Data-Rate.html.

[27] Antonio-Javier Garcia-Sanchez Ruben M. Sandoval and Joan Garcia-Haro. "Optimizing and Updating LoRa Communication Parameters: A Machine Learning Approach". In: *IEEE Transactions on Network and Service Management* 16.3 (2019).

[28] Daeun Yim et al. "An experimental LoRa performance evaluation in tree farm". In: *2018 IEEE Sensors Applications Symposium (SAS)*. 2018, pp. 1–6. DOI: 10.1109/SAS.2018.8336764.

[29] Daeun Yim et al. "An experimental LoRa performance evaluation in tree farm". In: *2018 IEEE Sensors Applications Symposium (SAS)*. 2018, pp. 1–6. DOI: 10.1109/SAS.2018.8336764.

[30] Anis Zarrad and Izzat Alsmadi. "Evaluating network test scenarios for network simulators systems". In: *International Journal of Distributed Sensor Networks* 13.10 (2017), p. 1550147717738216. DOI: 10.1177/1550147717738216. eprint: https://doi.org/10.1177/1550147717738216. URL: https://doi.org/10.1177/1550147717738216.