

Article

Improving Quality-of-Service in LoRa Low-Power Wide-Area Networks through optimized radio resource management

Eduardo Sallum ^{1*}, Nuno Pereira ², Mário Alves ² and Max Santos ¹

¹ Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa, Brazil; sallum@alunos.utfpr.edu.br (E.S.); maxsantos@utfpr.edu.br (M.S.)

² Politécnico do Porto - ISEP/IPP, Porto, Portugal; nap@isep.ipp.pt (N.P.); mjf@isep.ipp.pt (M.A.)

* Correspondence: sallum@alunos.utfpr.edu.br; Tel.: +55(42)98403-0302 (E.S.)

Abstract: Low Power Wide Area Networks (LPWAN) enable a growing number of Internet-of-Things (IoT) applications with large geographical coverage, low bit-rate and long lifetime requirements. LoRa (Long Range) is a well-known LPWAN technology which uses a proprietary *Chirp Spread Spectrum* (CSS) physical layer, while the upper layers are defined by an open standard - LoRaWAN. In this paper, we propose a simple yet effective method to improve the Quality-of-Service (QoS) of LoRa networks by fine-tuning specific radio parameters. Through a *Mixed Integer Linear Programming* (MILP) problem formulation, we find optimal settings for the *Spreading Factor* (SF) and *Carrier Frequency* (CF) radio parameters, considering the network traffic specifications as a whole, to improve the *Data Extraction Rate* (DER) and to reduce the *packet collision rate* and the energy consumption in LoRa networks. The effectiveness of the optimization procedure is demonstrated by simulations, considering realistic scenarios. In relation to the traditional LoRa radio parameter assignment policies, our solution leads to an average increase of 30% in DER, and a *number of collisions* 17 times smaller. In comparison to networks with dynamic radio parameter assignment policies, there is an increase of 10.5% and 4% of DER, and a *number of collisions* 13.5 and 7.5 times smaller than *equal-distribution* and *random* distribution, respectively. Regarding the *network energy consumption* metric, the proposed optimization obtained an average consumption 3.6 and 2.74 times lower than the *equal-distribution* and *random* dynamic allocation policies, respectively. Furthermore, we approach the practical aspects on how to implement and integrate the optimization mechanism proposed in LoRa, guaranteeing backward compatibility with the standard protocol.

Keywords: Internet of Things (IoT); Low-Power Wide Area Network (LPWAN); Mixed Integer Linear Programming (MILP); LoRaWAN; LoRa Simulator (LoRaSim); Open-Source; Optimization; Quality-of-Service (QoS); Data Extraction Rate; Packet Collision Rate; Energy Consumption; Energy Efficiency; Network Performance; Protocol Overhead; Performance Evaluation; Performance Improvement.

1. Introduction

We are at the dawn of the next generation of the Internet, which will be dominated by trillions of tiny computing devices embedded in everyday objects - the paradigm usually dubbed as Internet-of-Things (IoT). Many communication technologies can be used to interconnect IoT devices, e.g. short-range communication technologies such as Bluetooth, ZigBee and Z-Wave. However, such technologies face critical challenges in terms of energy, cost, and complexity when applications require covering large geographical areas and feature a large number of devices, due to the need for tricky

multi-hop routing in potentially harsh environments. Recent standards/technologies for Low-Power Wide-Area Networks (LPWAN) are an effective way to overcome such challenges and pave the way for new IoT applications that cover wide geographical areas over long distances. [1]. LPWAN technologies such as LoRa [2], SigFox [3], Weightless [4], WAVEIoT [5], and Wi-Fi HaLow [6] compete for an increasing market share among other IoT technologies, with a projected market worth of 65 Billion USD and smart gas and water metering applications occupying 20% of the LPWAN market by 2025 [7]. Figure 1 shows how LPWAN compares to other wireless technologies in terms of range and Bandwidth (BW).

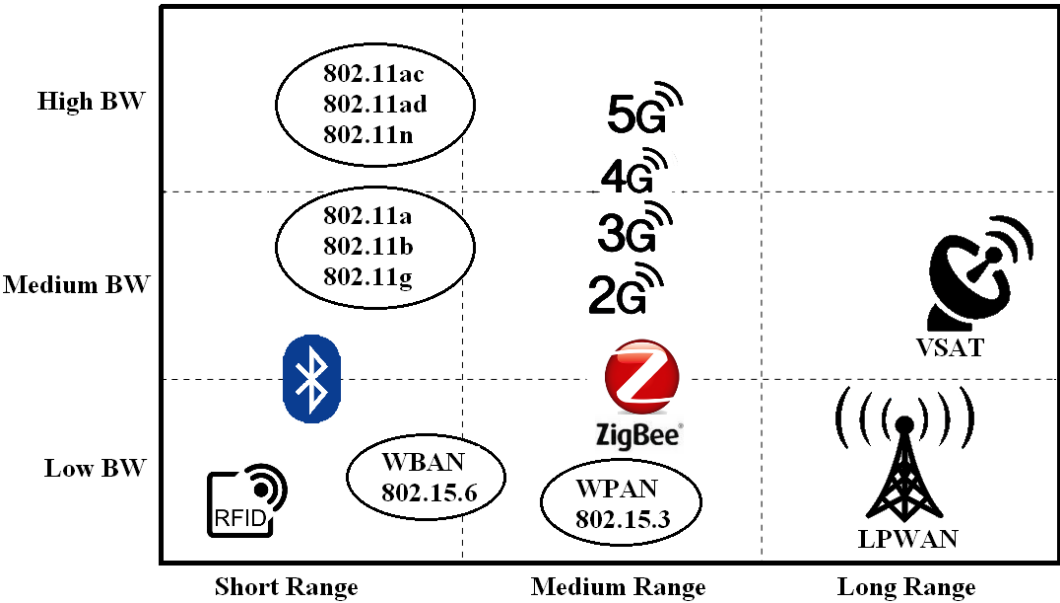


Figure 1. Mapping radio Coverage vs. Bandwidth of wireless technologies [8].

This work focuses on LoRa, a proprietary physical layer protocol that affords low-power and long-distance communication through Chirp Spread Spectrum (CSS) modulation. The main advantages of LoRa over other LPWAN technologies are: the open-source MAC protocol LoRaWAN specification, low-cost application availability, and community support [2]. e.g., “The Things Network”, a crowd-sourced community from 85 countries building a public and global IoT network based on LoRa [9].

The configuration of LoRa networks may be a challenging task, especially as the network scales up in the number of nodes. LoRa has some specific radio-related parameters that can be adjusted, such as *Carrier Frequency (CF)*, *Spreading Factor (SF)*, *Bandwidth (BW)*, *Transmission Power (TP)*, and *Coding Rate (CR)*. These parameters can be tuned at a device and/or network level to enhance overall network performance, namely reducing energy consumption, improving radio coverage, reducing radio interference and error rates. However, despite the increasing adoption of LoRa in IoT applications, the tools, methods, and models available to manage and optimize its performance are still scarce.

In this work, we propose a way to improve the Quality-of-Service (QoS) of LoRa networks, specifically to increase the *Data Extraction Rate (DER)* and to reduce the *number of collisions* and energy consumption, through the formulation of a *Mixed Integer Linear Programming* problem, which generates optimal settings for the *Spreading Factor* and *Carrier Frequency* parameters. The effectiveness of the proposed optimization procedure is demonstrated by simulations, considering realistic scenarios, showing that our solution performs better than standard and other benchmarking radio parameter assignment policies.

We identify the following relevant contributions:¹

- (i) Definition of a mathematical optimization formulation of the problem of assigning SF and CF parameters according to the traffic characteristics of LoRa End-devices;
- (ii) Definition of an approximation algorithm that leads to results very close to the optimal, but with an execution time that grows much slower as the number of End-devices in the network scales;
- (iii) Approach the practicalities of how to implement, integrate the approximation algorithm in LoRa and analyze the imposed overhead of the radio parameter assignment scheme to ensure compatibility with the standard protocol;
- (iv) Implementation of the proposed optimization algorithms and of other baseline policies (benchmarks to which our method is compared against in Section 5), and as well as of the ADR mechanism (not implemented in LoRaSim) in the LoRaSim open-source simulator, making our code available to the community [11].
- (v) Comparative performance study of 5 radio parameter assignment policies, showing the merit of the proposed parameter assignment methods against the benchmarks, with respect to relevant QoS metrics;

The remainder of the paper is structured as follows: Section 2 overviews the characteristics of the LoRa communication protocol that are most relevant within the context of this paper. In Section 3, we present a method to select radio parameters of LoRa networks, based on a formulation using *Mixed Integer Linear Programming*. Section 4 presents the Approximation Algorithm, which can efficiently produce results very close to the optimal. Section 5 outlines the simulation setup, the evaluation metrics, and elaborates on the comparative analysis performance of our parameter optimization method against relevant benchmark policies. Section 6 discusses relevant related work. Finally, Section 7 concludes the paper by summarising the main contributions and unveiling future research directions.

¹ This paper builds on our previous work, which has been reported in a paper currently under submission to an international conference [10]. In this context, we would like to provide the following clarifications to the Editor (and reviewers):

- This journal paper adds significant content and contributions to the previous one, namely:
 - (1) A more elaborated related work section, with 20 additional bibliographic references;
 - (2) A more comprehensive description of the characteristics of LoRa protocol that are most relevant within the context of this work;
 - (3) An analysis of the overhead imposed by the proposed radio parameter assignment schemes;
 - (4) A short description on how these schemes can be implemented, guaranteeing backward compatibility with the standard protocol and COTS LoRa devices;
 - (5) An implementation of the ADR mechanism in the LoRaSim simulator, and related evaluation;
 - (6) An extended set of experiments and related results, on the analysis of energy consumption of the different assignment policies.
- The conference paper had been submitted and accepted for publication at IEEE ISIE 2019, but due to bureaucratic problems related to specific funding to support the mission to the conference, it ended up not being published. Then we decided to resubmit it to IEEE ICIT 2020 (<https://www.itba.edu.ar/intranet/icit2020/>), which justifies it being still under reviewing. We can provide a copy to the editor/reviewers, if deemed necessary.

85 **2. LoRa Overview**

86 LoRa uses a radio frequency technology that allows communication over long distances,
87 with minimal energy consumption and a proprietary Chirp Spread Spectrum physical layer and
88 upper layers defined by an open standard - LoRaWAN. The End-devices can be sensors and
89 actuators, while gateways provide the radio connectivity to End-devices and deliver packets to
90 the network infrastructure composed of network servers and application servers. Network servers
91 are responsible for functions such as checking addresses of End-devices, checking received frames
92 for authenticity, handling acknowledgments, forwarding application payloads. Application servers
93 perform encryption/decryption of application payloads, manage user authorization, among other
94 related functions [9]. Figure 2 shows the LoRaWAN network architecture, composed by End-Devices,
95 gateways, network servers, and application servers.

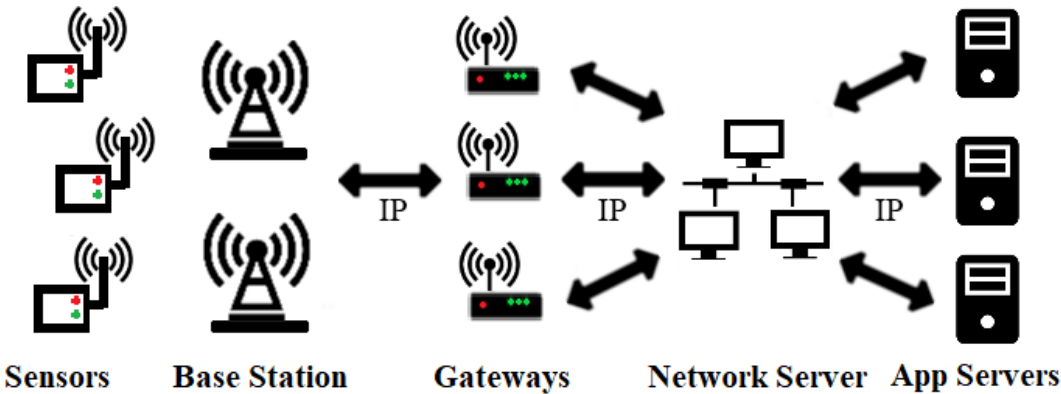


Figure 2. LoRaWAN network architecture.

96 Figure 3 shows the LoRaWAN communication stack or protocol stack, that means an
97 implementation of networking protocol suite that ensure that distribute applications can be
98 communicate in transparent manner.

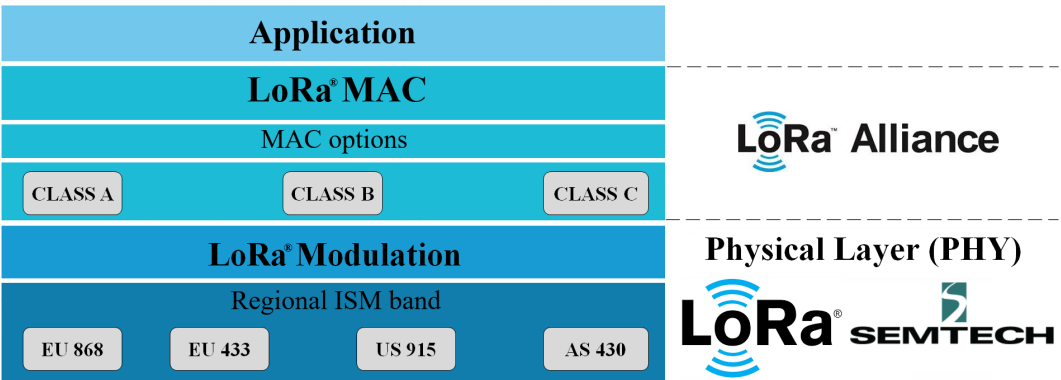


Figure 3. LoRaWAN Protocol Stack [12].

99 As shown in Figure 3, the top layer is the Application layer. The bottom layer is the Physical
100 Layer (PHY), as detailed in Section 2.1, responsible for LoRa modulation and ISM band definition,
101 which depends on the geographic region [2]. Above the PHY layer, is the Medium Access Control
102 sub-Layer (MAC), defined by the LoRa Alliance. To optimize energy consumption, LoRaWAN uses
103 a pure ALOHA [13] medium access control mechanism, encompassing three classes of End-devices:
104 Class A, Class B, and Class C, as detailed in Section 2.2.

2.1. LoRa Physical Layer

The physical layer LoRa implements CSS transmission method. The characteristics of LoRa are based on the following configurable parameters:

Bandwidth - BW: BW is the width of the frequencies in the transmission band. Higher BW values provide a higher data transfer rate and greater sensitivity to noise. The sets of BW in LoRa are 125 kHz, 250 kHz, and 500 kHz;

Spreading Factor - SF: SF is the ratio between the symbol rate and the chip rate, which can be in the range of 7 to 12. Higher SF increases the Signal to Noise Ratio, transmission range, and packet airtime, therefore it decreases the data rate. An important feature of LoRa is that SFs are orthogonal. If different SFs are used, the gateway can successfully decode multiple simultaneous data packets. LoRa modulation transmits the data at a chip rate equal to the programmed BW (chip-per-second-per-Hertz). The symbol rate and the bitrate are proportional to the BW. With CSS, each LoRa symbol is coded with a spreading code of 2^{SF} chips. Then, it takes 2^{SF} chips ($SF = SF_{bits} \times 2^{SF}$) to spread a symbol [14];

Carrier Frequency - CF: CF is the center frequency and can be programmed in 61 Hz steps in a range of 137 MHz to 1020 MHz;

Transmission Power - TP: Due to hardware limitations, the TP in a LoRa network can be configured in steps of 1 dB with a signal power between 2 dBm and 20 dBm, with a service level of 1% from 17 dBm [15];

Coding Rate: LoRa modulation adds Forward Error Correction (FEC), providing protection against transmission interference by encoding 4-bit data with 5-bit to 8-bit redundancies, allowing the receiver to detect and correct errors in the message. The CR values are 4/5, 4/6, 4/7, and 4/8, proportional to the FEC. LoRa devices with different CR can switch to communicate with each other through an explicit header stored in the packet header. Which means, if the code rate is denoted as $k = N$, where k represents useful information, and encoder generates N number of output bits, then $N - k$ will be the redundant bits. Higher CR values provide greater interference protection. However, it increases the air time. LoRa devices with different CR can switch to communicate with each other through an explicit header stored in the packet header [16].

The airtime of a LoRa transmission is computed according to the payload size and the combination of several radio parameters: SF, BW, and CR. These parameters can make the transmission time vary significantly. Table 1 exemplifies the different SNR, airtime, and bitrate resulting from the different SF at a fixed payload length of ten bytes, $BW = 125$ kHz, and $CR = 4/5$.

Table 1. Chirps, SNR, Airtime, and Bitrate according to parameter SF.

SF	Chirps / Symbol	SNR	Airtime ^a	Bitrate
7	128	-7.5	56 ms	5469 bps
8	256	-10	103 ms	3125 bps
9	512	-12.5	205 ms	1758 bps
10	1024	-15	371 ms	977 bps
11	2048	-17.5	741 ms	537 bps
12	4096	-20	1483 ms	293 bps

^a10 bytes per packet.

2.2. Class Transactions

The upper layer protocol, LoRaWAN, defines three classes of End-devices, with bidirectional communication, according to downlink latencies and power requirements. Class A devices have longer battery life because of higher latency. The downlink occurs within two windows, both with a specified delay, after an uplink transmission. Figure 4 shows a LoRaWAN Class A transaction.

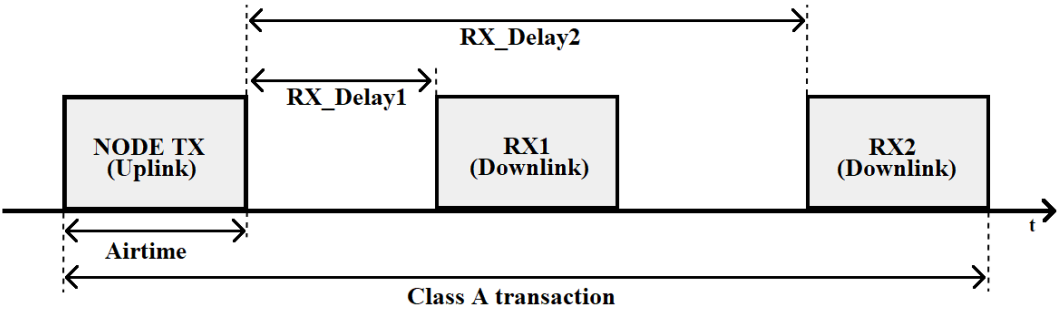


Figure 4. Class A transaction example.

141 Class B devices schedule downlink receptions from the base station at a pre-established period,
142 determining when applications can send control messages to the End-devices. In relation to Class A,
143 the Class B transaction has additional downlink windows, which occur at specific times following a
144 beacon, as illustrated in Figure 5.

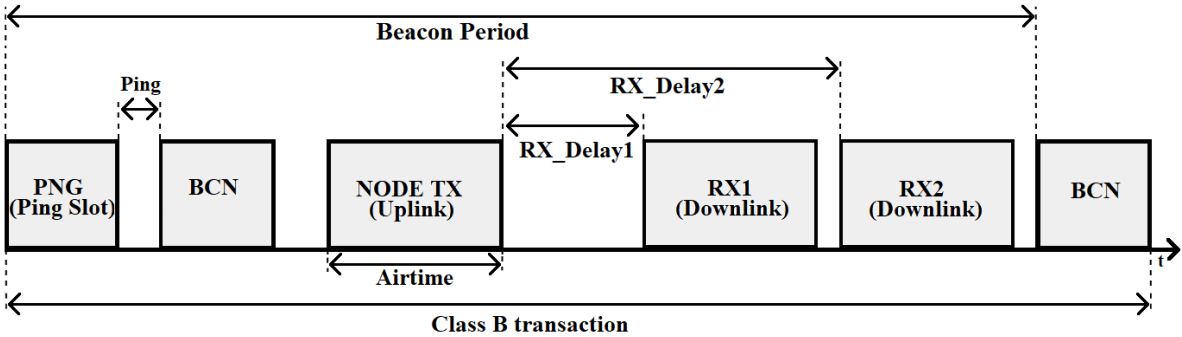


Figure 5. Class B transaction example.

145 Class C devices are grid-powered and always listening the transission medium and thus receive
146 downlink transmissions with the lowest latency [17]. As can be seen in the example shown in Figure 6,
147 it consists of receive windows that remain almost continuously open (only close during transmission).

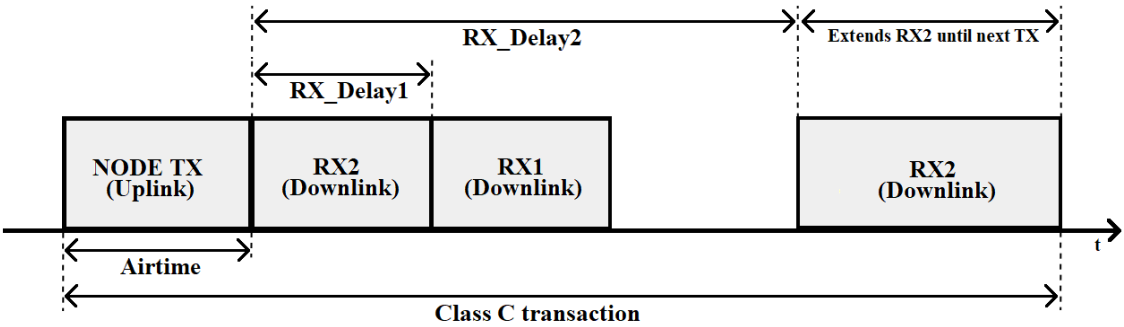


Figure 6. Class C transaction example.

148 2.3. Frame Structure

149 The preamble, in a range of 6 to 65535 symbols, initiates the LoRa packet structure. An optional
150 header, which describes the length and Forward Error Correction (FEC) rate of the payload, specifies a
151 16-bit Cyclic Redundancy Check (CRC). The header is transferred with a 4/8 FEC rate. After, there is
152 the payload in the packet structure [15]. Figure 7 shows the frame structure of LoRa.

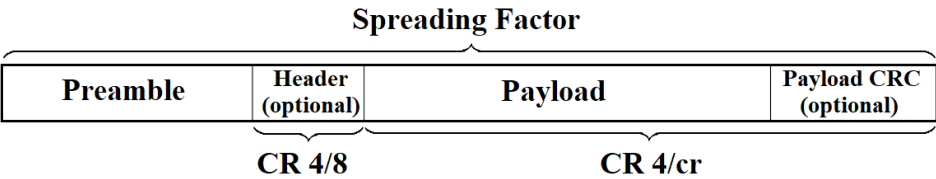


Figure 7. LoRa frame structure.

2.4. Adaptive Data Rate

The Adaptive Data Rate (ADR) mechanism is used to optimize the data transmission rate and the transmission power of the network nodes, in order to optimize network scalability and energy consumption. This mechanism runs asynchronously, with low complexity in the End-devices and with more complexity in the network server. The ADR should only be enabled by End-devices in stable RF conditions, and deactivated when the End-device detect unstable RF conditions. For instance, mobile End-devices must enable the ADR only when they are static for a certain period.

The appropriate data rate is determined by measurements of uplink messages based on the frame counter, Signal-to-Noise Ratio (SNR), and number of gateways [18]. As specified by the LoRa Alliance [2], after the ADR bit is set (to 1), the server analyses the 20 most recent uplinks. If the ADR bit is reset (to 0), the previous measurements are discarded and the measurements start again when the ADR is set again. If uplink messages are not received by the gateway, the ADR algorithm in the End-devices (ADR-NODE) increases the SF value of the subsequent uplink frame, thereby reducing the data rate and increasing the probability of reaching a gateway. Algorithm 1 shows the ADR-NODE algorithm.

Algorithm 1: ADR-NODE [19]

ADR_ACK_LIMIT ← 64

ADR_ACK_DELAY ← 32

ADR_ACK_CNT ← 0

if Uplink transmission then

ADR_ACK_CNT ← ADR_ACK_CNT + 1

167 if ADR_ACK_CNT == ADR_ACK_LIMIT then

Request response from server

if ADR_ACK_CNT ≥ ADR_ACK_LIMIT + ADR_ACK_DELAY then

node_SF ← node_SF + 1

if Downlink transmission received then

ADR_ACK_CNT ← 0

The server ADR algorithm (ADR-NET) increases the data rate, consequently decreasing the SF, and modifies the TP by measuring the SNR of the received frames, estimated based on the minimum SNR needed for the demodulation, and adjusted according to the specific margin of the device. The new parameters values calculated by the algorithm are sent to the End-device through a downlink frame, for being used in future transmissions [18] [19]. Algorithm 2 presents the ADR-NET algorithm.

Algorithm 2: ADR-NET [20]

Input : $dataRate$
 $txPower$
 SNR
 $deviceMargin$

Output: $desiredDataRate$
 $txPower$

$Link_{margin} \leftarrow SNR - \text{demodulationFloor}(dataRate)$
 $SNR_{margin} \leftarrow Link_{margin} - deviceMargin$
 $nStep \leftarrow SNR_{margin} / 3$
 $drId_x \leftarrow \text{getDataRateIndex}(dataRate)$
for $nStep > 0$ **and** $drId_x < ADR_{MaxDataRate}$ **do**
 $drId_x \leftarrow drId_x + 1$
 $nStep \leftarrow nStep - 1$
for $nStep > 0$ **and** $txPower > txPower_{min}$ **do**
 $txPower \leftarrow txPower - 3$
 $nStep \leftarrow nStep - 1$
for $nStep < 0$ **and** $txPower < txPower_{max}$ **do**
 $txPower \leftarrow txPower + 3$
 $nStep \leftarrow nStep + 1$

The ADR is an important mechanism for optimizing the data transfer rate, the airtime and the energy consumption. Understanding its operation is fundamental to make the optimization of radio parameters proposed in this work backward compatible with the standard LoRa protocol (therefore with ADR), as will be explained in Section 4.1. Next, Section 3 describes the MILP optimization problem.

3. MILP Optimization Problem

To increase the performance of the LoRa network through the optimization of radio parameters, we developed a formulation using *Mixed Integer Linear Programming*, a formal mathematical optimization framework. There are many objectives that can be selected to improve the performance of LoRa networks. e.g., reduce the individual node energy consumption, improve the reliability of data delivery, or improve overall throughput. For the purpose of this work, we aim at improving the amount of successfully delivered messages over a period of time by tuning the radio parameters of the LoRa devices in the network. Our performance problem for LoRa, modeled as a MILP problem, assigns values to the CF and SF parameters in order to improve the performance of the network. Section 3.1 presents general aspects of the MILP, Section 3.2 introduces the notation and model used. Finally, Section 3.3 presents our MILP problem formulation.

3.1. Background on Mathematical Optimization

Mathematical optimization has been used in a great number of fields such as in [21] for vehicle-control problems, in [22] for short-term scheduling of resource-constrained multistage flow-shop batch facilities, in [23] for short-term hydro scheduling and unit commitment with the head-dependent reservoir, and in [24] for aircraft planning with collision avoidance. Mathematical optimization is also used for automotive applications when optimizing extensibility [25] or task activation modes [26]. The problems can be expressed by a set of integer, binary and continuous design variables, design parameters, linear equalities and inequalities representing the constraints on the solution, and a linear objective function. The main advantage consists in the possibility of leveraging

a well-established theory and solution methods, which includes the availability of efficient solver engines with controllable solution accuracy and constrained computing time [23].

Linear Programming, (LP, also named by linear programming) is a mathematical method for solving and achieve the best outcomes of problems in a mathematical model in which requirements are describes by linear relationships. The variables belong to set of real values. For the problems in which variables are integers, the Mixed Integer Linear Programming, (MILP) is suitable to be applied. MILP is a linear program in which solving and achieve best outcomes that require to take integer values and are solved by using the same technology as integer programs. Solving the problem attempts to find the best solution for the objective function in the set of solutions that satisfies the constraints. a MILP optimization problem is given for a set of variables, objective function, and a set of constraints.

$$\begin{aligned} & \min_x cx \\ & \text{subject to:} \\ & Ax \leq b \\ & \text{where:} \\ & x \in Z^n \times R^p \end{aligned}$$

In Integer Programming Problem, all variables are limited to be integer. The set S of $x \in Z^n \times R^p$ which satisfy the linear constraints $Ax \leq b$ is the Feasible Set:

$$S = \{x \in Z^n \times R^p, Ax \leq b\}$$

An element $x \in Z^n$ is the feasible solution. The optimization problem finds a solution in the feasible set that yields the best objective value, a feasibility problem finds an element that satisfies all constraints and restrictions, i.e., find an element in the feasible set.

3.2. Notation and Model

Consider a wireless network system with a LoRa gateway and n End-devices. End-devices transmit with an average transmission rate λ packets/second with an average length of ξ bytes. The time to transmit a packet of length ξ is given by a function $airtime(\xi, cf)$ which computes the time to transmit the packet at a given CF (cf). For the purpose of this model, we assume a fixed BW and CR, which must be taken into account for the airtime computations.

In order to model our problem as an MILP problem, we define an array of tuples $R_{i \in \{1 \dots n\}} = (cf_{i,CF}, sf_{i,SF})$ for each End-device, where CF is a list of available CFs and SF , a list of available SFs. The lists of SFs and CFs as input allows to, e.g., reserve certain CF for other uses. An example input could be $CF = 868.1, 868.3, 868.5$ and $SF = 7, 8, 9, 10, 11, 12$ indicating that the algorithm can use 3 CFs (868.1, 868.3, 868.5) and SF 7 to 12 in each of these channels. The $(cf_{i,CF}, sf_{i,SF})$ tuples are arrays of booleans which indicate if an End-device i is using a given CF or SF. These are the decision variables of our problem and one of the constraints of our model must restrict the solutions to only having one value of each of the arrays set to true in each End-device, as each End-device can only have one CF and one SF value assigned.

The End-devices of the LoRa networks as $N = \{N_1, N_2, N_3 \dots N_n\}$ with CF and SF parameters communicates with gateway G . To increase network performance, are considered multi-indexes as energy consumption of the End-devices, network range, interference reduction, and errors rate. In this case, are assigned CF and SF to maximize success probability in a single gateway. As an input, to model the LoRa network as MILP, we are given variables $\forall_i \in \{1 \dots N\} CF_n, SF_n$ where N is equal to the number of End-devices.

233 3.3. MILP Problem Statement

Considering that SFs are orthogonal with each other in LoRa [27], we would like to maximize the probability of success (or minimize the collision probability) for each CF, SF pair. Therefore, the decision variables for each node have two Boolean arrays CF_n and SF_n that indicate the CF and SF of each N . e.g., the following arrays indicate that N_1 is using CF_7 and CF_1 :

$$\begin{aligned} SF_{1,7} &= 1 & CF_{1,0} &= 0 \\ SF_{1,8} &= 0 & CF_{1,1} &= 1 \\ SF_{1,9} &= 0 & CF_{1,2} &= 0 \\ SF_{1,10} &= 0 & CF_{1,3} &= 0 \\ SF_{1,11} &= 0 & CF_{1,4} &= 0 \\ SF_{1,12} &= 0 & CF_{1,5} &= 0 \\ & & CF_{1,6} &= 0 \\ & & CF_{1,7} &= 0 \end{aligned}$$

To express this in our MILP model, we define in Equation 1 the following objective function, which minimizes the difference between the utilization of each CF, SF pair, denoted as $U_{cf,sf}$, resulting in the minimum load for each CF, SF:

$$\begin{aligned} \sum_{cfi \in CF} \sum_{sfj \in SF} (U_{cfi,sfj} - U_{cfl,sfk}) \quad (1) \\ cfl : cfl \in CF \wedge cfi \neq cfl; \\ sfk : sfk \in SF \wedge sfj \neq sfk; \end{aligned}$$

, where:

$$U_{cfi,sfj} = N_{cfi,sfj} \times \text{airtime}(\xi, sfi) \times \lambda; \quad (2)$$

$$U_{cfl,sfk} = N_{cfl,sfk} \times \text{airtime}(\xi, sfk) \times \lambda$$

, and:

$$N_{cfi,sfj} = \sum_i^n cfi_{cfi} \times sf_{i,sfj}; \quad (2a)$$

$$N_{cfl,sfk} = \sum_i^n cfi_{cfl} \times sf_{i,sfk}$$

, subject to:

$$\sum_{cf \in CF} (cf_{i,cf}) = 1 \quad \forall i : 1..n \quad (2b)$$

, and:

$$\sum_{sf \in SF} (sf_{i,sf}) = 1 \quad \forall i : 1..n \quad (2c)$$

234 Equation 2 compute the utilization of the CF, SF pair, as defined by Equation 2a, considering the
235 number of End-devices assigned to that pair. The two constraints 2b and 2c limit the assignment CF
236 and SF such that each End-device is only assigned one CF and one SF.

Practical Issues: Equation 2a is the product of two binary variables, which is non-linear. This can however be made linear given:

$$z_{i,cf,sf} = cfi_{cf} \times sf_{i,sf} \quad (3)$$

, and the following additional constraints:

$$z_{i,cf,sf} \leq cf_{i,cf} \quad (3a)$$

$$z_{i,cf,sf} \leq sf_{i,cf} \quad (3b)$$

$$z_{i,cf,sf} \geq cf_{i,cf} + sf_{i,cf} - 1 \quad (3c)$$

This problem can be solved using one of the many state-of-the-art MILP Solvers. In our experiments, we used the CPLEX Optimizer [28].

4. Approximation Algorithm

The ADR mechanism, as described in Section 2.4, adapts the TP and SF parameters of each End-device based on the frame counter, SNR, and gateway number measurements. However, the performance of a LoRa network also depends on the occurrence of simultaneous transmissions with the same SF, determined through the two-dimensional coordinates of the End-devices and the LoRa network gateways.

To apply the results obtained through the MILP problem formulation, as described in Section 3.1, with backward compatibility in LoRa networks, we rely on the ADR mechanism to dynamically adjust the settings of End-devices. As indicated by the standard, only End-devices with a stable RF environment should enable ADR. Our model includes provisions to define a list of allowed SFs, which can be determined by measurements (which are part of the normal LoRa ADR procedure). This is backward compatible with existing LoRa Networks and only the LoRa Network Server needs to be updated to obtain the radio parameters according to our proposed method.

The time to solve MILP problems grows exponentially as the number of end devices increases. Due to scalability issues, we developed the Approximation Algorithm to provide parameter assignments (Algorithm 3). The approximation is based on a first-fit assignment of the utilization for each CF and SF combination.

Algorithm 3: Approximation Algorithm

Input : ξ , the average packet length in bytes
 λ , the average packet transmission rate
 Array CF of available CFs
 Array SF of available SFs, sorted by airtime

Output: Array R of (cf_i, sf_i) tuples with the CF and SF settings for node i

forall elements of CF **do**

forall elements of SF **do**

$U_{cf,sf} \leftarrow 0$

forall elements of V **do**

$R_i \leftarrow \min_{cf,sf}(U);$

$U_{R_i} \leftarrow U_{cf,sf} + \frac{\text{airtime}(\xi, sf_i)}{1/\lambda}$

$\triangleright (cf_i, sf_i)$ tuple

The Approximation Algorithm in Algorithm 3 receives as input the average packet length in bytes (ξ), and the average packet transmission rate (λ). It also receives two arrays with the available CF and SF, with the SF array being ordered from the SF with the smallest resulting airtime to the largest.

Algorithm 3 starts by creating an array of $U_{cf,sf}$ to hold the utilization of each (CF, SF) pair. Then, the algorithm assigns a CF and SF to each End-device by using a $\min_{cf,sf}(U)$ function which takes as input the array $U_{cf,sf}$ and returns a (cf_i, sf_i) tuple that will result in the lowest utilization increase. That is, it will return the CF, SF pair that is the $\min(U_{cf,sf} + \text{airtime}(\xi, sf)) \forall sf : sf \in SF$. Finally, the algorithm increases the utilization for that CF, SF pair.

4.1. Backward compatibility with LoRa

The LoRa Server project [29] provides a set of open source applications for building LoRaWAN networks. It is part of a larger project that encompasses a protocol packet forwarder broker for MQTT (LoRa Gateway Bridge), and a compatible application server. The mechanisms provided by LoRa Server allow users to manage the gateways in the LoRa network, the supported applications, and the devices associated with the applications [29].

Reviewing the documentation for sending LoRa Server uplink messages [30] [31], the way to implement the Approximation Algorithm with backward compatibility with the standard LoRa protocol and its mechanisms, e.g. ADR (as described in Section 2.4), is to implement the Approximation Algorithm at the LoRa application layer of LoRa and devices. Thus, the application layer determines the CF, SF pair individually for each node. The LoRa Server receives as input parameters the CF, SF pairs by the application server (that is running the Approximation Algorithm) and transmits to the End-devices in the downlink. Then, the node executes a corresponding application to configure the node parameters.

5. EVALUATION

To study the performance of different parameter selection strategies, including the ones assigned by solving the optimization problem presented in Section 3 and the algorithm in Section 4, we carried out a simulation study based on the tools developed in [15]. We aimed at studying the scalability of the network, namely analysing different performance metrics according to the number of End-devices and topologies. Building and analyzing such LoRa networks would be infeasible in practice, so we opted to base our study in simulation. Importantly, the assumptions and models of the LoRaSim simulation tool (that we have used) were validated by practical experiments.

5.1. Simulation Setup

The LoRa simulation tool, *LoRaSim* [15], allows to define model LoRa networks by setting the number of End-devices inserted in a two-dimensional area, average packet sending interval, the number of base stations and other radio parameters such as SF, CR, and BW. We executed multiple runs (each with a random uniform node distribution in space) with a simulated time of 1 year for all experiments.

In our experiments, we assume that a single base station can simultaneously decode concurrent signals on all orthogonal SF and BW settings. This assumption can be supported in practice with multiple LoRa chips, such as the Semtech SX1301 [32]. Table 2 presents the simulations parameters common to all simulation runs.

Table 2. Simulations parameters.

Parameter	Value
CR	4/5
Bandwidth	125 kHz
Sub-band	g and g1
Transmission power	14 dBm
Number of base stations	1
Transmission range	100 m
Packet size	20 bytes
Average packet transmission rate	16.6 minutes

5.2. Evaluation Metrics

We used three evaluation metrics to evaluate the performance of a LoRa network. The first is Data Extraction Rate, which evaluates the performance network-wide in a numerical range between 0 and 1, where in optimal network deployments the value is equal to 1. Equation 4 shows how the DER can be computed, where Nr represents the number of packets received, C the number of packet collisions, and Ns the total number of packets sent.

$$DER = \frac{Nr - C}{Ns} \tag{4}$$

The second is the *number of collisions*. Packet collisions occur when two or more network nodes attempt to send data simultaneously, resulting in collisions and possible loss of transmitted data, causing a negative effect on system performance. When two LoRa transmissions occur at the same time (perceived at the receiver), it is determined that the receiver can decode received packets simultaneously by analyzing CF, SF, energy and time conditions.

The last one is the *network energy consumption*, defined as the energy spent by the network to extract a message successfully, considering all network nodes. It depends on parameters such as SF, BW, CR, and TP. A low value of NEC means that the network parameters have been set efficiently [15].

5.3. Evaluation Results

We have performed experiments for the following parameter assignment policies as follows. The **min-airtime** is a default assignment used by LoRa End-devices which assigns a fixed CF and SF so that packets have the minimum air time. The **random** policy dynamically assigns CF, SF pairs randomly, aiming at reducing concurrent transmissions (that cause packet collisions). With a similar goal in mind, **equal-distribution** distributes the number of End-devices equally between CF, SF pairs. The **opt-problem** is the assignment resulting from solving the optimization problem presented using the CPLEX ILP solver (in Section 3). Finally, **approx-alg** is the result of using the Approximation Algorithm (in Section 4).

5.3.1. Analysis of network scalability of assignment policies

According to the ETSI EN300.220 standard, in Europe the max-duty-cycle for g (863.0 - 868.0 MHz) and g1 (868.0 - 868.6 MHz) bands is 1% [33]. Considering a network scenario with the configuration parameters in Table 2 and the max-duty-cycle of 1% for the sub-bands g and g1, Table 3 shows the maximum number of nodes that can be allocated for each assignment policy, resulting from our simulation analysis:

Table 3. Network scalability according to allocation policy.

Policy	Maximum number of nodes
equal-distribution	84 nodes
min-airtime	175 nodes
random	743 nodes in best-case
opt-problem	743 nodes
approx-alg	743 nodes

The results in Table 3 show that the policies that best scaled the network are **opt-problem** and **approx-alg**. This is due to the optimization to dynamically assign to each node the (CF, SF) pair leading to the shortest airtime, considering the use of sub-bands g and g1. As a result, the **opt-problem** and **approx-alg** policies allow the best scaling of the network, with a maximum number of 743 nodes.

In the best-case scenario, the maximum number of nodes that can be allocated with the **random** policy is 743 nodes if the randomly generated CF and SF values for all nodes are the same as **opt-problem** and **approx-alg** policies (regardless of their ordering). However, the **random** policy allows only 6 nodes to be allocated in the worst-case scenario, which is all nodes allocated in SF12 using only one of the sub-bands (g or g1).

The **min-airtime** assignment policy allocates a maximum of 173 nodes, respecting the max-duty-cycle limit of 1%, with all nodes configured in SF7 using the g or g1 sub-band. Finally, the maximum number of nodes that can be allocated with the **equal-distribution** policy is 84 nodes.

5.3.2. DER rate

To better represent the difference between assignment policies, the network scenarios used in our simulations range from 50 nodes to 1500 nodes. Figure 8 shows the results of DER as a function of the number of End-devices.

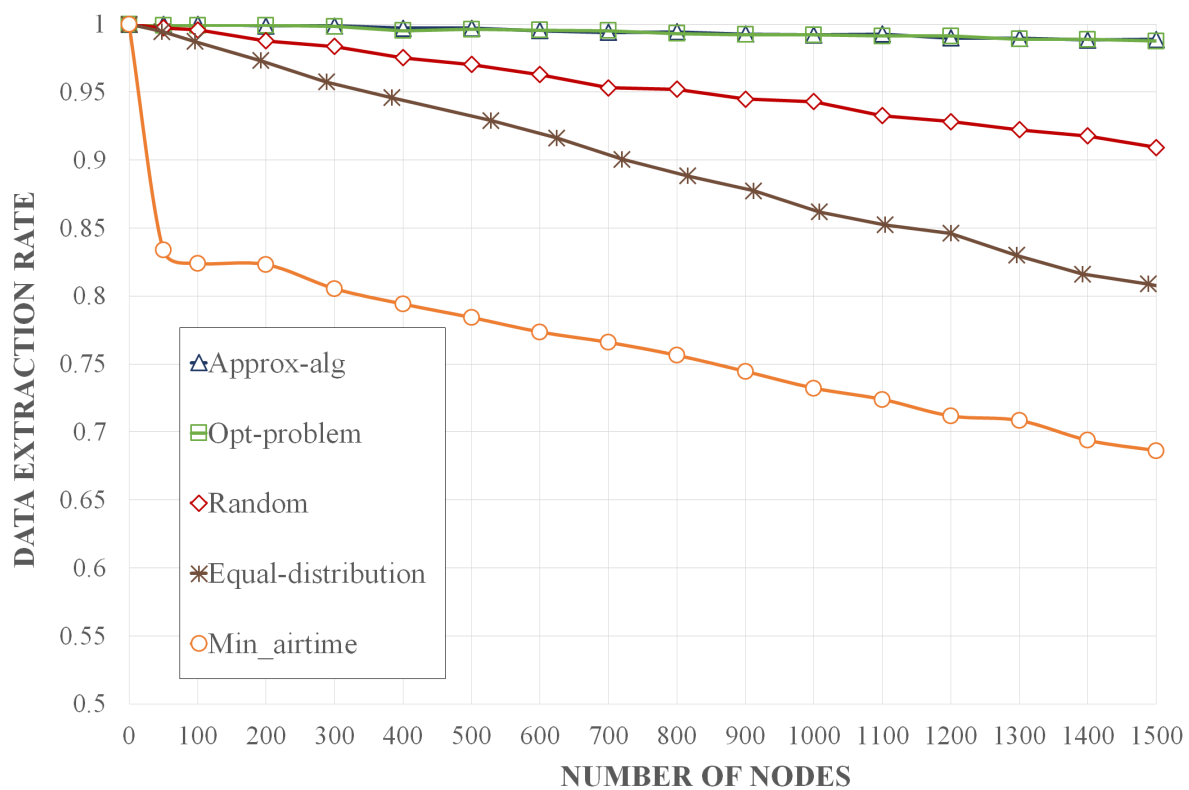


Figure 8. DER as a function of the number of nodes.

As shown in the graph of Figure 8, the policies **opt-problem**, represented by the green line, and **approx-alg** (navy blue line) exhibit the highest DER performance with an average increase of 30%, 10.5%, and 4% in relation to the **min-airtime** (orange line), **equal-distribution** (brown line), and **random** (red line), respectively. We note that the lines for **opt-problem** and **approx-alg** are similar, showing that the Approximation Algorithm gives results very close to the ones given by the optimization problem, at a fraction of the execution time.

5.3.3. Number of packet collisions

Figure 9 illustrates the *number of collisions* according to the number of nodes.

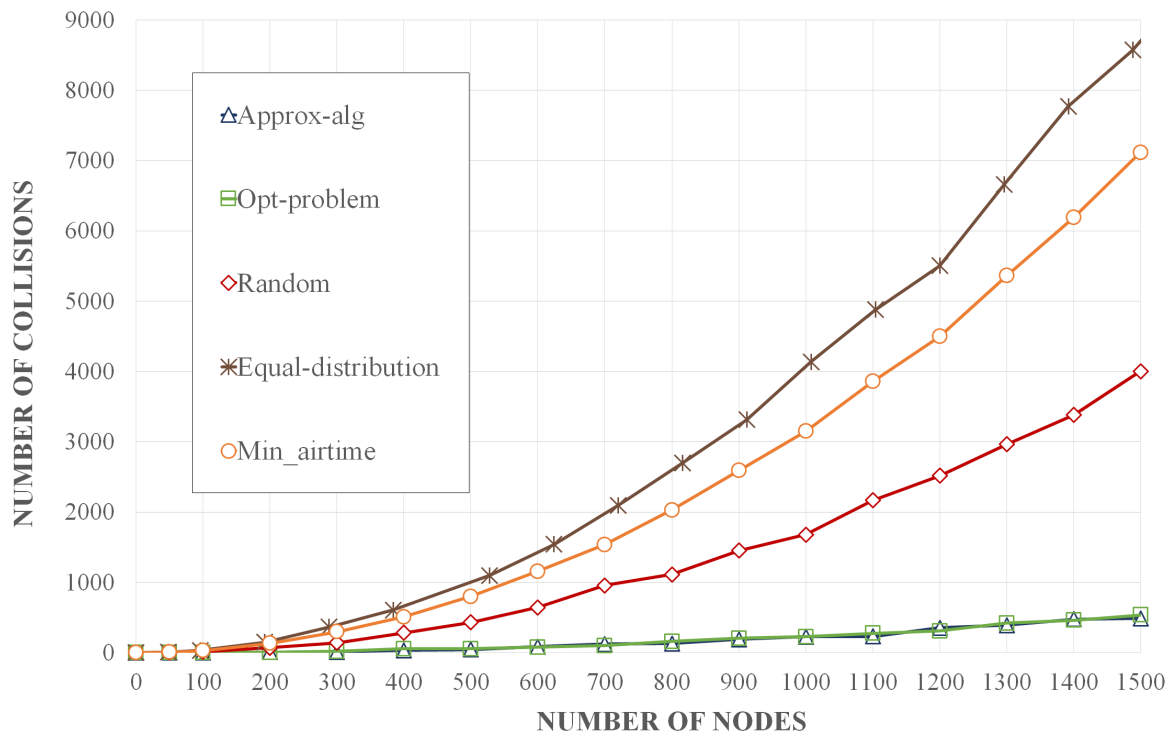


Figure 9. Number of collisions according to the number of nodes.

The results in Figure 9 indicates that **opt-problem** and **approx-alg** cause the lowest number of collisions, being the curves represented in the graph practically equivalent. The policies **equal-distribution**, **min-airtime**, and **random** lead to average collision rates of 17, 13.5, and 7.5 times higher, respectively, in relation to **opt-problem** and **approx-alg**.

5.3.4. Network energy consumption

Figure 10 demonstrates the energy consumption in mJ as a function of the number of nodes.

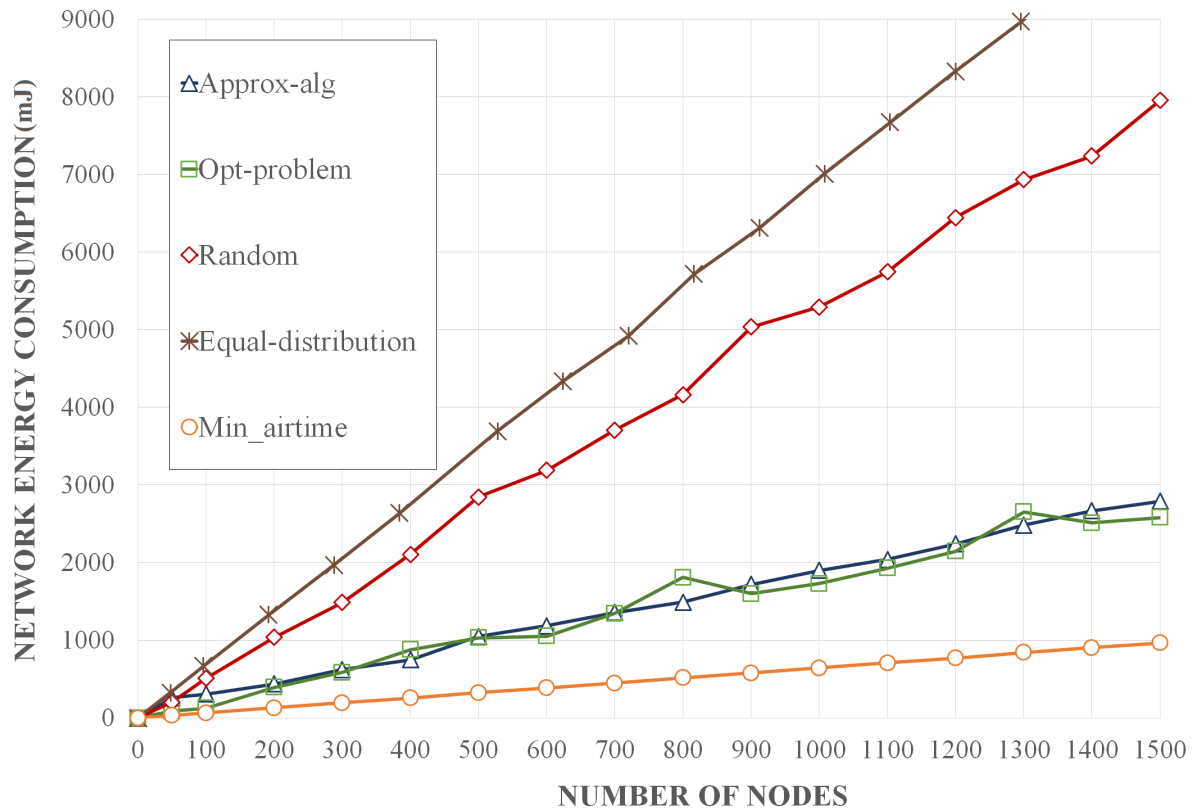


Figure 10. Energy consumption as a function of the number of nodes.

The results in the Figure 10 demonstrates that **equal-distribution** has a 3.73 times higher energy consumption rate than **opt-problem** and 3.6 greater than **approx-alg**. The **random** policy resulted in an average energy consumption 2.84 times and 2.74 times higher than **opt-problem** and **approx-alg**, respectively. The difference in the average energy consumption between the **opt-problem** and the **approx-alg** is 3.7%. Both obtained an average consumption 3 times greater in relation to the **min-airtime**. Using this policy, SF is set to SF7, which has the lowest energy consumption, as reported in Section 2. However, in **opt-problem** and **approx-alg** dynamic values of SF are assigned to the network nodes.

The radio parameter assignment policies proposed in this paper - **opt-problem** and **approx-alg**, proved to be better than policies **random**, **min-airtime**, and **equal-distribution** in relation to DER, *number of collisions*, and *network energy consumption*. The results show that **opt-problem** and **approx-alg** obtained high values of DER, above 0.98, for low, medium, and high-density scenarios. This demonstrates how the number of network nodes affects the performance of a LoRa network. The *number of collisions* was minimal in relation to **random**, **min-airtime**, and **equal-distribution**. Also, the energy consumption of the proposed optimization schemes is lower when compared to other methods of dynamic assignment of values of SF and CF: **random** and **equal-distribution**.

5.3.5. ADR analysis and comparison in dynamic assignment policies

As explained in Section 4.1, the optimization of LoRa network radio parameters proposed in this paper is compatible with the standard LoRa protocol, which includes the ADR mechanism. In this context, we implemented the ADR mechanism in the LoRaSim simulator, to be able to analyze its impact in the **random** and **approx-alg** dynamic assignment policies, using according to the DER, *number of collisions* and *network energy consumption* evaluation metrics.

To represent a more overloaded network scenario, keeping the network simulation settings in Table 2, the average packet transmission rate has been changed from 16.6 minutes to 1 minute.

Figures 11, 12 and 13 illustrate the DER rate, the number of collisions and the grid energy consumption, respectively, according to the number of nodes.

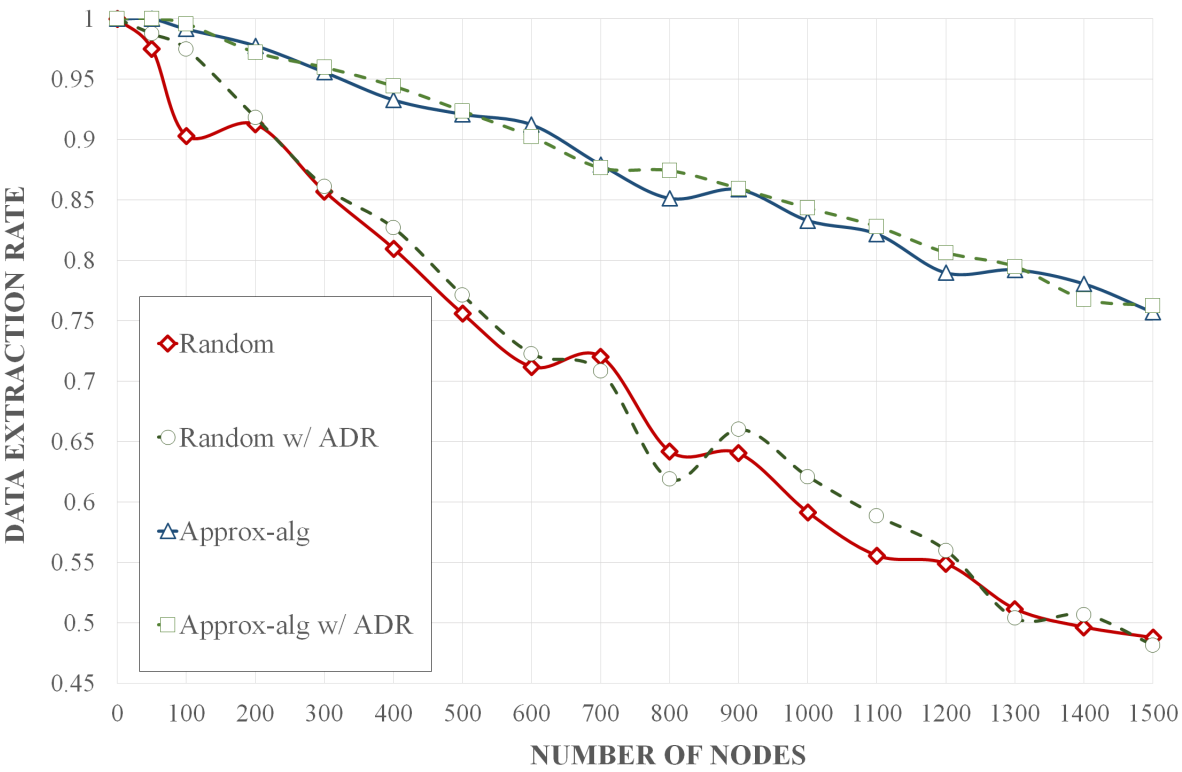


Figure 11. DER as a function of the number of nodes.

According to Figure 11, the improvement of the **random** policy with the ADR mechanism is 1.69% compared to without ADR. However, the difference in **approx-alg** was smaller, only 0.39%. Therefore, it was the **random** policy that obtained the highest gain of DER rate with ADR.

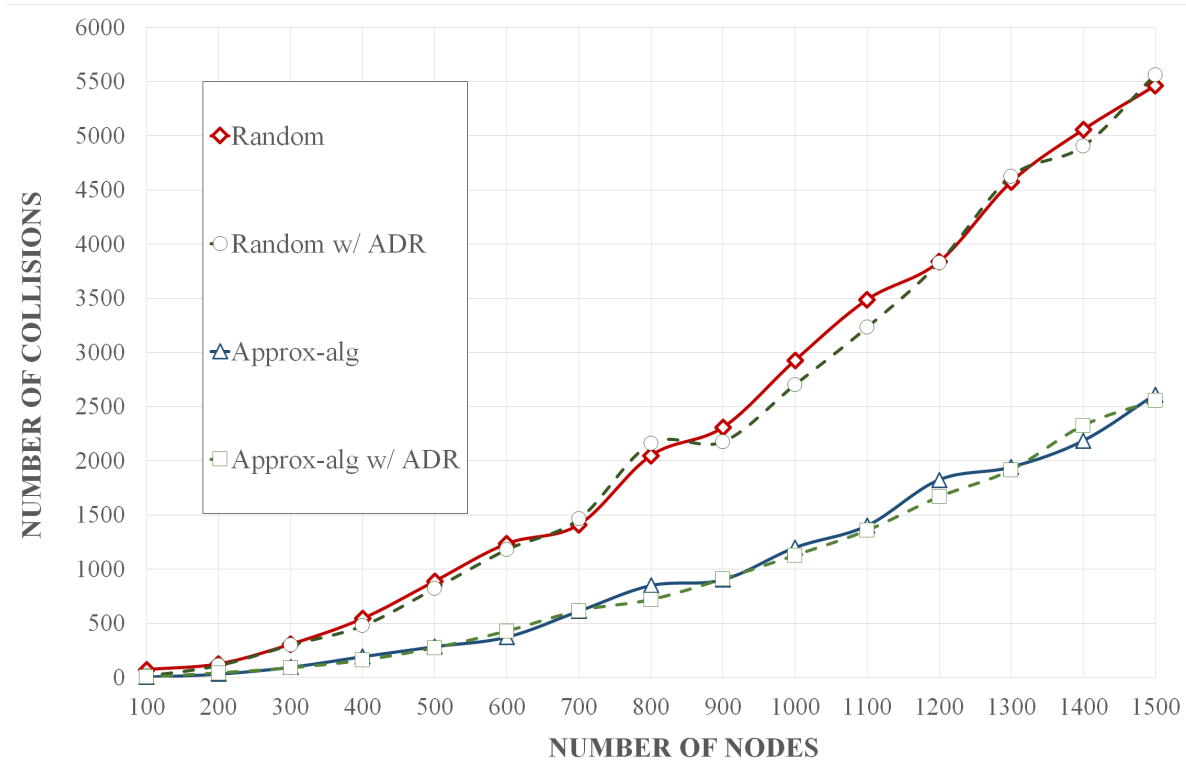


Figure 12. Number of collisions according to the number of nodes.

385 The graph in Figure 12 shows that the network packet collision rate decreased around 2.13% in
386 the **random** and 2.31% in the **approx-alg**.

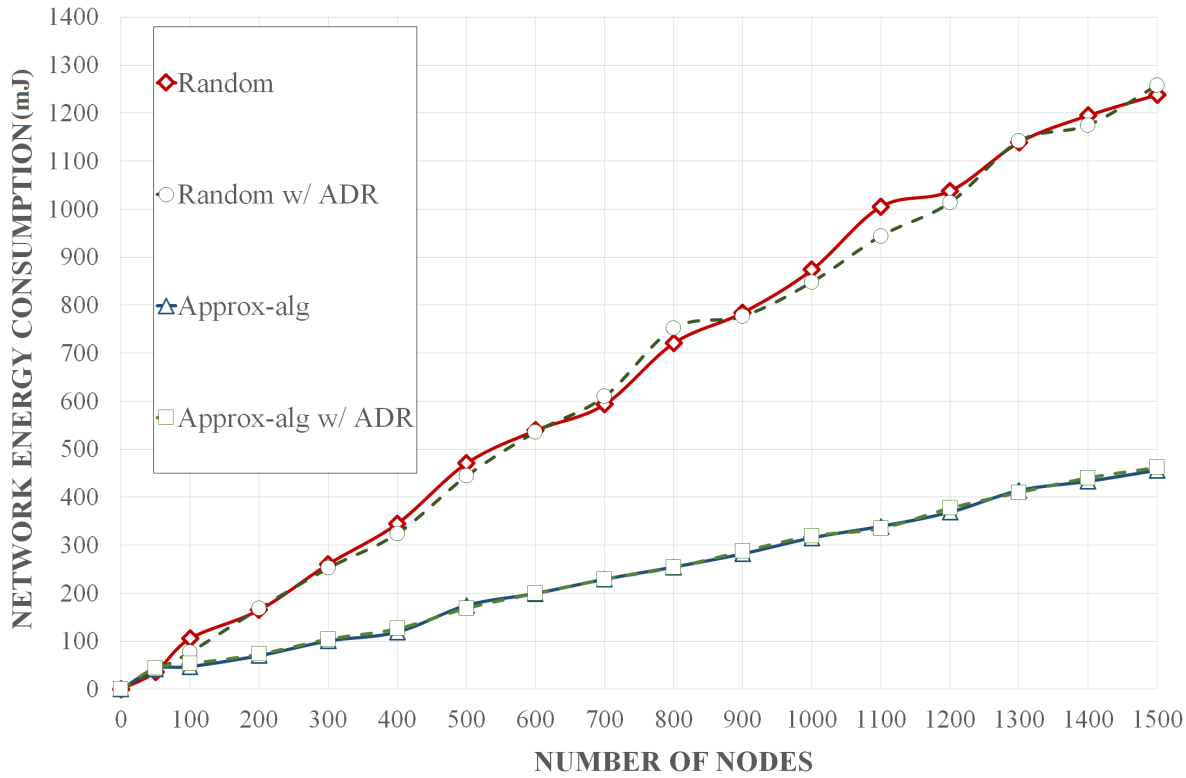


Figure 13. Energy consumption as a function of the number of nodes.

Figure 13 shows that average network power consumption dropped 1.48% in the **random** and 1.06% in **approx-alg**. Overall, our results demonstrate the importance of the ADR mechanism in the LoRa protocol and the effectiveness of the proposed radio assignment policies.

5.3.6. Analysis of overheads inferred by the optimization proposed in the standard LoRa protocol

The microcontroller in a LoRaWAN terminal processes sensor data and interfaces with the radio chip to transmit data over the network. The microcontroller must have enough memory for radio chip drivers, sensor drivers, and application code. Based on the recommended requirements for a microcontroller in a LoRaWAN terminal using the Semtech SX1267x radio chip: Microcontroller 16 bit or 32 bit CPU, 16 KB RAM and 256 KB flash memory [34], we analyze the possible overheads for End-device communication with the Approximation Algorithm application running on the LoRa server caused by additional programming code:

- (i) Computation Time: The Approximation Algorithm (Algorithm 3) has a linear complexity time $O(n) = 111n + 57$, in the worst-case. Thus, since the Approximation Algorithm is expected to run in the LoRaWAN Application Layer and End-devices with a time complexity that is similar to the ADR mechanism (Algorithm 1 and Algorithm 2), our optimization scheme causes no significant computation (time) overhead neither in the End-devices nor in the LoRa server.
- (ii) Storage space: The most significant part of the (optimization) algorithm runs in the LoRa server, accounting approximately 70 lines of code, while the extra code in the End-devices is around 60 lines. Overall, the implementation of our algorithm takes less than 20 kB of storage space (4 kB in average, 20 kB worst-case, considering different situations), which is not significant considering that most Commercial Off-The-Shelf nodes have at least 128 kB of programming/non-volatile memory (minimum requirements for a LoRaWAN microcontroller [34]);
- (iii) Energy consumption: The proposed optimization scheme achieves lower network energy consumption compared to other dynamic allocation policies. The way the Approximation Algorithm assigns End-devices pairs of (CF, SF) , ordered from lowest to highest airtime, results in an improved network scalability (maximum number of nodes) ratio for lower power consumption. Regarding the **min-airtime** policy, which has shorter airtime due to using the SF parameter fixed in SF7, the simulation results show that the Approximation Algorithm consumed on average 3x more energy.
- (iv) Communications: Considering with Approximation Algorithm input an array of valid (SF, CF) pairs, End-devices only receive the (SF, CF) parameter values once (generated by the Approximation Algorithm). Our simulations show that the proposed optimization has lower network packet send rate, as well as number of collisions, compared to the assignment policies **min-airtime**, **equal-distribution** and **random**. Therefore, the proposed optimization method causes no additional communication overhead.

6. Related work

Network performance and scalability are key factors for LoRa network design. These factors are influenced mainly by the correct choice of radio parameters and by environmental conditions [17]. Several research papers defined a possible formal approach, through the analysis and comparison of performance and scalability of LoRa networks, for the problem of parameter selection. In this section, we outline some relevant works that address these aspects.

The performance analysis of long-range transmission and channel attenuation, made by Petajajarvi [35], concludes that in the best scenario, a single gateway configured using SF 12, 125 MHz BW, and data rate of 1.8 kbps, covers an area of 30 km in peripheral urban environments. Additionally, this paper concludes that low values of SF are recommended for urban scenarios due to less interference caused by the Doppler effect. In other work, Petajajarvi [36] analyzed the performance of LoRa in three experiments. The first and second experiments were performed in a LoRa End-device

under the Doppler frequency shift. With TP of 14 dBm and SF12, at least 60% of the packets are received from 30 km distance on the water. The third experiment was performed on an End-device, mounted in a car, configured with a TP of 14 dBm and SF of 12. This work concludes that in a mobile scenario, the LoRa communication worsens in a displacement speed of End-devices around 40 km/h.

In an indoor environment, Neumann [37] did a performance analysis of the LoRa network with the parameters throughout RSSI, SNR, packet loss, packet error, power consumption, and delay to verify how the average current consumption of one End-device impacts in the performance of LoRa network with one device and one gateway. This study concludes that the data rate affects the packet loss. The scalability of LoRaWAN networks evaluated in [38] in ns-3 simulator, which the error model is based on interference among various concurrent transmissions, concludes that increasing the gateway density improves the negative effect of restrained downstream in Packet Delivery Ratio of upstream messages. Bakov [39] measured the limits of a LoRaWAN network, which is about 0.1 51-byte (Frame Payload) in a network with 3 main channels and 6 data rates. That corresponds to 5000 End-devices, each node generates 2 messages per day. A performance analysis conducted in [40] of LoRaWAN End-device using the metrics uplink, throughput, and transmission time concluded that in terms of LoRaWAN scalability, millions of devices can communicate with a reduced transfer.

Vatcharatiensakul [41] in a LoRaWAN performance evaluation carried out in internal and external environments concluded that the LoRa signal range is affected by the properties of the antennas such as antenna gain, directional and antenna height. Yousuf [13] measured how internal and external urban environments affect the LoRa signal. In an internal environment of a seven-story building, there was a minimal packet drop. The external coverage depends on the environment. In this experiment performed in a range of 4.4km, there was 15% of packet drops and the packet size alter the signal range.

Toussaint [42] evaluated over-the-air performance in a Markov chain model in different traffic conditions, duty cycles, and channel availability. The expected delay and energy consumption of the LoRa network depends on the number of channels, the number of sub-bands, and gateway parameters. Feltrin [43] characterized experimentally from the link-level viewpoint and evaluated through simulations the capacity of a LoRaWAN gateway to provide communications in large rural environments. In this scenario, a gateway covering an area of 1 km², the maximum node density is 185 End-devices/hectare. However, this gateway in an area of 46.5 km², the maximum density decreased to 1 node/hectare. To maximize the node density in this large area, 4 gateways provides 40 End-devices/hectare.

The work of Cattani [44] presents an analysis of the performance of the LoRa in relation of different configurations of PHY and environmental conditions. The analysis done in a scenario where the network nodes are at the communication limit concludes that it is more efficient to use the faster PHY configuration and higher transmission power than slower configurations that maximize the quality of the link. Because the faster PHY setting provides 100 times faster effective bit rate than the slower setting scenario, at the cost of a 10% lower average packet reception rate. Also, the work concludes that environmental factors such as temperature and humidity impact the rate of reception of packets and the intensity of the received signal. In a controlled environment with a temperature of 15 °C, an optimal link with 100% Packet Reception Rate becomes unusable at 60 °C. Over this temperature range, the received signal strength is reduced by 6 dBm, being 1 dBm per 10 °C.

In [45] was analyzed and compared the performance of LoRa networks with a proposed MAC layer two-step lightweight scheduling in ns-3 simulator. The results conclude that in a scenario with one gateway and 1000 End-devices, the proposed MAC layer reduces the Packet Error Ratio around 20%. In the performance analysis of LoRa network performed in [46], in an Additive white Gaussian noise (AWGN) and a frequency selective channel of the Frequency Shift Chirp Modulation (FCSM) and the Frequency-Shift keying modulation (FSK), concludes that the performance of the FCSM and FSK in an AWGN channel are equivalent. Although, in a frequency selective channel, the FCSM has a superior performance. In [47] are made a proposal and schema validation to maximize the average Packet

Success Probability through SF parameters using MATLAB simulator on Massive Connectivity (MC), respectively. The projected scheme is better to all others MC in relations of Monte Carlo simulations and analysis. e.g., in a 2000 End-devices LoRa network, the method provides stable 810 End-devices, an increase of 22% in MC over the Equal-Area-Based scheme.

Differently from the studies presented, this work proposes to optimize the performance of LoRa networks through the computation of SF and CF parameters, comparing and analyzing in realistic application scenarios with the standard LoRa assignment policy and other dynamic radio parameter assignment policies.

7. CONCLUSIONS AND FUTURE WORK

We have proposed a simple yet efficient methodology to improve the performance of LoRa networks by fine-tuning its SF and CF radio parameters, through a *Mixed Integer Linear Programming* optimization approach. This enables the LoRa network designer to choose the best network configuration. Importantly, these parameter assignment policies are backward compatible with the LoRa standard protocol, meaning that they can be implemented in comercial-off-the-shelf LoRa devices.

Simulation results show that our methodology optimizes the assignment of CF and SF pairs with an average increase of 30% of DER in relation to the standard LoRa assignment, which assigns fixed CF and SF values between the End-devices so that the packets have the minimum air time. In comparison to networks where CF and SF pairs are dynamically assigned by the gateway, there is an increase of 10.5% and 4% of DER in relation to the egalitarian and random distribution, respectively. Furthermore, our method leads to a *number of collisions* that is 17, 13.5, and 7.5 times smaller than egalitarian, standard LoRa and random distribution, respectively. In relation to the average energy consumption, the scenario with the standard LoRa assignment, whose SF value set at 7, obtained an energy consumption 3 times lower than the proposed optimization. However, our optimization achieved a 3.73 and 2.84 times lower energy consumption than egalitarian and random distribution, respectively.

Furthermore, we are engaged to extend it for the physical experiments in short scale through a computer tool that allow and ensure developers make designs that best attend the application requirements.

Abbreviations

The following abbreviations are used in this manuscript:

ADR	Adaptive Data Rate
BW	Bandwidth
CF	Carrier Frequency
CPU	CPU – Central Processing Unit
CR	Coding Rate
CSS	Chirp Spread Spectrum
DER	Data Extraction Rate
FEC	Forward Error Correction
IoT	Internet of Things
LoRa	Low Range
LoRaSim	LoRa Simulator
LPWAN	Low Power Wide Area Network
QoS	Quality of Service
MAC	Medium Access Control
MILP	Mixed Integer Linear Programming
PHY	Physical Layer
SF	Spreading Factor
SNR	Signal to Noise Ratio
TP	Transmission Power

References

- Angrisani, L.; Arpaia, P.; Bonavolontà, F.; Conti, M.; Liccardo, A. LoRa protocol performance assessment in critical noise conditions. 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), 2017, pp. 1–5. doi:10.1109/RTSI.2017.8065952.
- Committee, L.A.T.; others. LoRaWAN 1.1 Specification. *LoRa Alliance Stand* **2017**, 1.
- Sigfox. Sigfox, the world’s leading IoT services provider. <https://www.sigfox.com/en>, 2019.
- SIG, W. What is Weightless - Weightless. <http://www.weightless.org/about/what-is-weightless>, 2019.
- WAVIoT. WAVIoT LPWAN – Low-power long-range (LPWAN) solutions for IoT and M2M. <https://waviot.com/>, 2019.
- Alliance, W.F. Wi-Fi HaLow. <https://www.wi-fi.org/discover-wi-fi/wi-fi-halow>, 2019.
- Insights, G.M. Global LPWAN Market Size worth over \$65 Bn by 2025. <https://www.gminsights.com/pressrelease/lpwan-market>, 2019.
- Egli, P.R. LPWAN Low Power Wide Area Network: Overview of emerging technologies for low power wide area networks in internet of things and m2m scenarios. http://indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf, 2015.
- Industries, T.T. Building a global open LoRaWAN network. <https://www.thethingsnetwork.org/>, 2019.
- Sallum, E.; Pereira, N.; Alves, M.; Santos, M. Performance optimization on LoRa networks through assigning radio parameters. Under submission to 2020 IEEE International Conference on Industrial Technology (ICIT). IEEE.
- Sallum, E.; Pereira, N.; Alves, M.; Santos, M. LoRa_optimization. https://github.com/esallum/LoRa_optimization, 2019.
- Components, M. Tecnologia IoT LoRa. <http://amr.sk/?s=tech-lora>, 2016.
- Yousuf, A.M.; Rochester, E.M.; Ghaderi, M. A low-cost LoRaWAN testbed for IoT: Implementation and measurements. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018, pp. 361–366. doi:10.1109/WF-IoT.2018.8355180.
- Noreen, U.; Bounceur, A.; Clavier, L. A study of LoRa low power and wide area network technology. 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2017, pp. 1–6. doi:10.1109/ATSIP.2017.8075570.
- Bor, M.C.; Roedig, U.; Voigt, T.; Alonso, J.M. Do LoRa Low-Power Wide-Area Networks Scale? Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems; ACM: New York, NY, USA, 2016; MSWiM ’16, pp. 59–67. doi:10.1145/2988287.2989163.

16. Navarro-Ortiz, J.; Sendra, S.; Ameigeiras, P.; Lopez-Soler, J.M. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things. *IEEE Communications Magazine* **2018**, *56*, 60–67. doi:10.1109/MCOM.2018.1700625.
17. Raza, U.; Kulkarni, P.; Sooriyabandara, M. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys Tutorials* **2017**, *19*, 855–873. doi:10.1109/COMST.2017.2652320.
18. Industries, T.T. LoRaWAN Adaptive Data Rate. <https://www.thethingsnetwork.org/docs/applications/mqtt/api.html#uplink-messages>, 2019.
19. Slabicki, M.; Premsankar, G.; Di Francesco, M. Adaptive configuration of lora networks for dense iot deployments. NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2018, pp. 1–9.
20. Network, T.T. The Things Network Stack. <https://github.com/TheThingsNetwork/ttn>, 2019.
21. Earl, M.G.; D'Andrea, R. Iterative MILP methods for vehicle-control problems. *IEEE Transactions on Robotics* **2005**, *21*, 1158–1167. doi:10.1109/TRO.2005.853499.
22. Méndez, C.; Henning, G.; Cerdá, J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Computers Chemical Engineering* **2001**, *25*, 701–711. doi:https://doi.org/10.1016/S0098-1354(01)00671-8.
23. Borghetti, A.; D'Ambrosio, C.; Lodi, A.; Martello, S. An MILP Approach for Short-Term Hydro Scheduling and Unit Commitment With Head-Dependent Reservoir. *IEEE Transactions on Power Systems* **2008**, *23*, 1115–1124. doi:10.1109/TPWRS.2008.926704.
24. Richards, A.; How, J.P. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), 2002, Vol. 3, pp. 1936–1941 vol.3. doi:10.1109/ACC.2002.1023918.
25. Zhu, Q.; Yang, Y.; Natale, M.; Scholte, E.; Sangiovanni-Vincentelli, A. Optimizing the Software Architecture for Extensibility in Hard Real-Time Distributed Systems. *IEEE Transactions on Industrial Informatics* **2010**, *6*, 621–636. doi:10.1109/TII.2010.2053938.
26. Zheng, W.; di Natale, M.; Pinello, C.; Giusto, P.; Vincentelli, A.S. Synthesis of Task and Message Activation Models in Real-Time Distributed Automotive Systems. 2007 Design, Automation Test in Europe Conference Exhibition, 2007, pp. 1–6. doi:10.1109/DATE.2007.364573.
27. Voigt, T.; Bor, M.; Roedig, U.; Alonso, J. Mitigating Inter-network Interference in LoRa Networks. Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks; Junction Publishing: USA, 2017; EWSN & #8217;17, pp. 323–328.
28. IBM. CPLEX Optimizer. <https://www.ibm.com/analytics/cplex-optimizer>, 2019.
29. Server, L. The LoRa Server project. <https://www.loraserver.io/overview/>.
30. Industries, T.T. API Reference. <https://www.thethingsnetwork.org/docs/applications/mqtt/api.html#uplink-messages>, 2019.
31. Alliance, L. LoRa Server. <https://github.com/brocaar/loraserver>, 2019.
32. Semtech. SX1301 Datasheet. <https://www.semtech.com/uploads/documents/sx1301.pdf>, 2017.
33. Industries, T.T. Duty Cycle for LoRaWAN Devices. <https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html>, 2019.
34. Semtech. MCU Requirements for LoRaWAN. https://www.semtech.com/uploads/documents/AN1200.28_MCU_Requirements_for_LoRaWAN_V3.pdf, 2017.
35. Petäjäjarvi, J.; Mikhaylov, K.; Roivainen, A.; Hanninen, T.; Pettissalo, M. On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. 2015 14th International Conference on ITS Telecommunications (ITST), 2015, pp. 55–59. doi:10.1109/ITST.2015.7377400.
36. Petäjäjarvi, J.; Mikhaylov, K.; Pettissalo, M.; Janhunen, J.; Linatti, J. Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. *International Journal of Distributed Sensor Networks* **2017**, *13*, 1550147717699412.
37. Neumann, P.; Montavont, J.; Noël, T. Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study. 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2016, pp. 1–8. doi:10.1109/WiMOB.2016.7763213.
38. den Abeele, F.V.; Haxhibeqiri, J.; Moerman, I.; Hoebeke, J. Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3. *IEEE Internet of Things Journal* **2017**, *4*, 2186–2198. doi:10.1109/JIOT.2017.2768498.

39. Bankov, D.; Khorov, E.; Lyakhov, A. On the Limits of LoRaWAN Channel Access. 2016 International Conference on Engineering and Telecommunication (EnT), 2016, pp. 10–14. doi:10.1109/EnT.2016.011.
40. Mikhaylov, K.; Petaejaervi, J.; Haenninen, T. Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology. European Wireless 2016; 22th European Wireless Conference, 2016, pp. 1–6.
41. Vatcharatiansakul, N.; Tuwanut, P.; Pornavalai, C. Experimental performance evaluation of LoRaWAN: A case study in Bangkok. 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2017, pp. 1–4. doi:10.1109/JCSSE.2017.8025948.
42. Toussaint, J.; Rachkidy, N.E.; Guitton, A. Performance analysis of the on-the-air activation in LoRaWAN. 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016, pp. 1–7. doi:10.1109/IEMCON.2016.7746082.
43. Feltrin, L.; Buratti, C.; Vinciarelli, E.; Bonis, R.D.; Verdone, R. LoRaWAN: Evaluation of Link- and System-Level Performance. *IEEE Internet of Things Journal* **2018**, *5*, 2249–2258. doi:10.1109/JIOT.2018.2828867.
44. Cattani, M.; Boano, C.; Römer, K. An experimental evaluation of the reliability of lora long-range low-power wireless communication. *Journal of Sensor and Actuator Networks* **2017**, *6*, 7.
45. Reynders, B.; Wang, Q.; Tuset-Peiro, P.; Vilajosana, X.; Pollin, S. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. *IEEE Internet of Things Journal* **2018**, *5*, 1830–1842. doi:10.1109/JIOT.2018.2815150.
46. Vangelista, L. Frequency Shift Chirp Modulation: The LoRa Modulation. *IEEE Signal Processing Letters* **2017**, *24*, 1818–1821. doi:10.1109/LSP.2017.2762960.
47. Lim, J.T.; Han, Y. Spreading Factor Allocation for Massive Connectivity in LoRa Systems. *IEEE Communications Letters* **2018**, *22*, 800–803. doi:10.1109/LCOMM.2018.2797274.