Journal of Signal Processing Systems (2022) 94:65–79
https://doi.org/10.1007/s11265-021-01678-8

Evaluation of an Adaptive Resource Allocation for LoRaWAN

Jean Moraes[1] · Helder Oliveira[1] · Eduardo Cerqueira[1] · Cristiano Both[2] · Sherali Zeadally[3] · Denis Rosário[1]

Received: 22 December 2020 / Revised: 1 April 2021 / Accepted: 25 June 2021 / Published online: 7 August 2021

Abstract

Long Range Wide Area Network (LoRaWAN) emerges to connect devices that require long-range and low-cost (bandwidth and power) communication services. In this context, the adoption of this technology brings new challenges due to the densification of IoT devices, which causes signal interference and affects the QoS directly. On the other hand, the LoRaWAN transmission configurations' flexibility allows higher management to use end-device parameters, allowing better resource utilization and improve network scalability. We evaluate an adaptive solution that defines the best LoRaWAN parameter settings to reduce the channel utilization and, consequently, maximize the number of packets delivered. Additionally, to validate the method, we used a mixed-integer linear programming solution and compared the results obtained with those given by the heuristics. The results achieved by the heuristics were very close to those provided by the optimal result, demonstrating the effectiveness of the heuristics.

Keywords Resource allocation · IoT · LoRaWAN · MILP

1 Introduction

Internet-of-Things (IoT) is expanding at a fast rate to provide connections for billions of devices [31]. The IoT technology has been changing social behavior through disruptive technologies in new verticals and applications, such as Smart farms/agriculture, home/building automation, and healthcare, to cite only a few examples of intelligent environment IoT verticals [23, 40, 43]. In this context, the everyday environment will soon have a large number of IoT devices per square meter [4]. For instance, both academia and industry estimate that the Internet would have approximately 500 billion IoT devices connected until 2030 [41]. Also, the number of 5G-connected IoT devices will reach 4.1 billion by 2024 [13]. IoT applications have different energy consumption requirements, coverage, Quality of Service (QoS), and massive Machine-Type Communication [9].

IoT applications could rely on different network technology for data dissemination. Specifically, short-range network technologies (*e.g.*, WiFi and Bluetooth) provide coverage area limited to a few meters and are positively affected by interferences. However, short-range networks need highly-dense deployment to achieve an expanded coverage area, increasing Capital expenditure (CAPEX) and Operational expenditure (OPEX). On the other hand, traditional cellular networks operating on licensed frequency bands can provide extended coverage from tens of hundreds of meters, supporting thousands of devices with high throughput. However, modulation complexity, along with medium access schemes, are highly energy demanding characteristics. To support the IoT growth, innovative long-range technologies, such as Low Power Wide Area Network

✉ Jean Moraes
jean.anjos.moraes@itec.ufpa.br

✉ Denis Rosário
denis@ufpa.br

Helder Oliveira
heldermay@ufpa.br

Eduardo Cerqueira
cerqueira@ufpa.br

Cristiano Both
cbboth@unisinos.br

Sherali Zeadally
szeadally@uky.edu

[1] Federal University of Pará (UFPA), Belém, Brazil

[2] University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil

[3] University of Kentucky, Lexington, KY, USA

(LPWAN) [21], is attracting attention from both academic and industry researchers. That is because of their promising capabilities in broad area connectivity and their operations on unlicensed frequency bands with the appropriate data rate, power consumption, and throughput tailored for many IoT applications [29].

Long-Range Wide-Area Network (LoRaWAN) technology is considered the most adopted LPWAN technology, promising ubiquitous connectivity for many IoT applications while keeping a simple network structure and management [26, 42]. In particular, LoRaWAN was developed by LoRa Alliance[5], *i.e.*, a non-profit association, which defined the higher layers and network architecture on top of the LoRa physical layer [14, 26]. The LoRa physical layer is a proprietary spread spectrum modulation based on the robust Chirp Spread Spectrum (CSS) by Semtech [24]. The LoRaWAN Alliance has more than 500 associated members, higher than 140 LoRaWAN deployments, and more than 130 Network Operators in different countries [44]. LoRaWAN provides extended coverage to operate in unlicensed and pure implementation frequency ranges with low cost, low energy consumption, and flexible transmission rate. Moreover, LoRaWAN must potentially support a large and varying number of IoT devices sending data to the application server through the same Gateway. This significant demand causes a network overload and creates the called hotspot problem, which results in signal interference and affects QoS due to packet loss caused by collisions [34]. For instance, a LoRaWAN gateway will be unable to correctly decode simultaneous signals sent by IoT devices using the same Spreading Factor (SF) on the same Carrier Frequency (CF).

[An efficient adaptive resource allocation mechanism must adjust on-the-fly the LoRaWAN radio-related parameters, such as SF and CF, to reduce the packet loss caused by interference based on current network conditions.] However, resource allocation holds several possibilities for configuring such parameters. In this way, a mathematical model developed through Mixed Integer Linear Programming (MILP) has great importance in formulating and presenting the optimal solution to maximize performance because it can optimize parameter settings. [Therefore, an adaptive resource allocation based on a heuristic method is required to make computationally-efficient decisions in LoRaWAN and to generate results close to an optimal solution.] State-of-the-art solutions that have focused on resource allocation have investigated the SF allocation [7], SF and Coding Rate (CR) allocation [36], and increasing the data flow [37]. However, none of the previous works considers the device requirements for an efficient resource allocation on-the-fly adjusting the configuration of radio related parameters to minimize channel utilization while minimizing collisions.

In this article, we introduce an extensive analysis of our proposed heuristiC fOr adaptive REsource alloCaTion on LoRaWAN for IoT applications (CORRECT) [30] to evaluate the impact of the radio configuration parameters in massive IoT scenarios in terms of QoS. We describe the CORRECT mechanism's operation, which dynamically adjusts the LoRaWAN parameters to reduce interference and packet collision, minimize the channel utilization, and increase the number of delivered packets. The heuristic chooses the settings based on the signal strength and distance between the device and the gateway to provide the tradeoff between increasing the transmission range and reducing the delay, energy, and interference. Furthermore, we describe a MILP called Optimization Model for LoRaWAN Resource AlloCation for IoT ApplicatiOns (MARCO) to compare the proposed heuristic with an optimal resource allocation solution. This article extends the previous work described in [30]. Its main research contributions include an extensive review of related works, a more detailed description of the mechanism, and an extensive evaluation of the impact of a resource allocation mechanism for LoRaWAN under a different network scenario. Specifically, using the CORRECT heuristic instead of the ADR heuristic, we obtained an improvement of up to 18% in PDR. For other heuristics, this difference is even more significant. The heuristic CORRECT yields up 18.70% higher throughput than Explora-AT; this difference increases even more than other heuristics.

The rest of the paper is organized as follows. Section 2 presents the state-of-the-art results which have explored resource allocation in LoRaWAN. Section 3 introduces the proposed heuristic model, called CORRECT. Section 4 explores the simulation model developed to evaluate CORRECT and present the results obtained. Finally, Section 5 concludes the work and introduces future research directions.

2 Related Works

This section presents the most recent works which focus on resource allocation for LoRaWAN. In each work, we discuss their advantages and disadvantages. Amichi et al. [7] formulated a nonlinear optimization of mixed integers considering the harmful effects of interference between SFs. The focus of the authors is to obtain a fair throughput and reduce energy consumption. Nevertheless, this model does not consider optimizing the choice of CFs and reduces energy consumption according to the SF choice.

El-Aasser et al. [18] proposed two SF allocation heuristics, which adjust the SF service radius, *i.e.*, the maximum distance that an SF can be assigned to an End-

Device (ED), ensuring the correct demodulation by the gateway. However, setting SF based only on distance is not recommended because it depends on the device's signal strength. In the same approach, Caillouet et al. [11] designed an optimization model for LoRaWAN that optimizes SF allocation to minimize collisions and maximize the number of served nodes. The optimization is modeled as an Integer Linear Programming (ILP) that considers a Rayleigh channel. Nevertheless, this model does not consider channel allocation and the choice of SF to reduce energy consumption. Analytical models have made progress in the last few years, *e.g.*, Sandovalet al. [37] presented two Markov chains to model the Transmission Cycle (TC) and energy consumption. Based on these models, the optimal solution finds TC that increases throughput while keeping power consumption below a threshold. Moreover, the authors present a solution for calculating a global network configuration that maximizes the throughput obtained analytically [36]. Such a proposal formulates a Markov Decision Process (MDP), in which the optimal update policy allows the system to maximize the accumulated network throughput in a short time interval.

The literature presents some strategies for resource allocation. Farhad et al. [20] proposed two resource allocation schemes to enhance the packet success ratio by lowering the impact of interference. The first scheme, called the channel-adaptive SF recovery algorithm, increments or decrements the SF based on the ED packets' retransmission, indicating the network's channel status. The second approach allocates SF to EDs based on ED sensitivity during the initial deployment. However, this work does not perform an intelligent configuration of CF, where, based on the presented interference model, it would cause significant losses due to the packet collisions on the same channel. Moreover, Khaled et al. [2] proposed a heuristic for data rate fairness among nodes within a LoRaWAN. In the first phase, it derives a fair data rate distribution along with all the devices in the LoRaWAN, which aims to address the unfair LoRaWAN characteristic when EDs are very close to the gateway or use lower SFs. The second phase performs transmission power allocation that seeks to mitigate the effect that harms the network. The second phase performs transmission power allocation that aims to mitigate the effect that harms the network. However, although the approach shows improvements in the Packet Delivery Ratio (PDR), this algorithm does not consider the CF allocation, which presents promising results in solving the problem of capture effect in network packets.

Zorbas et al. [46] proposed a resource allocation mechanism to improve the LoRaWAN potential by employing multiple communication parameters. They modeled the average success probability per set as a density function, analyzing intra-SF and inter-SF collisions. Each ED has different communication settings based on such a model in terms of Bandwidth (BW) and SF. In this sense, this work tries to assign most EDs on a given BW and SF to increase the packet delivery probability for each BW and the distance between the ED and a given gateway. However, the authors considered only the SF collisions without considering the interference when two packets have the same CF and SF. The algorithm does not cover all the packet loss possibilities. In a similar context, Babaki et al. [8] presented a solution to enhance the default resource allocation algorithm, *i.e.*, ADR, by dynamically designating the radio-related parameters, SF and TP, by applying the Ordering Weight Average (OWA) operator. This work aims to increase the network noise resilience and PDR in dense IoT scenarios recognizing the OWA decision-make nature and the Packet Loss Ratio (PLR) metric. However, this approach does not consider the packet collision as a fundamental problem affecting the network packet loss and achieving an optimal CF configuration on the devices.

Cuomo et al. [15] proposed two resource allocation solutions named ExploraSF and Explora-AT, which aim to optimize LoRaWAN execution by configuring the SF values for each ED based on the signal strength. Explora-SF attempts to equally allocate ED in the Gateway's (GW) radio area to all SF, limited by their Received Signal Strength Indicator (RSSI) values and appropriate thresholds. On the other hand, Explora-AT propagates impartial allotments of the ToA among the end nodes in the network, prioritizing the lower SFs to reduce collision probability. In this way, a higher signal strength leads to lower SF for a given ED, where it considers a limited number of ED in each SF based on ToA. Nevertheless, both Explora-SF and Explora-AT do not consider the CF allocation to further reduce packet loss when the packets have the same SF and CF configuration. From another perspective, Dawaliby et al. [16] presented a Software-defined Network-based solution for network slicing to optimize and manage a LoRaWAN. Each Gateway has multiple virtual slices, which must find the correct configurations of SF and transmission power that simultaneously increase the QoS and minimize cost energy consumption and package loss rate. However, this work adds a processing overhead to perform slicing over LoRaWAN, which reduces the network resources available and does not consider the CF (*i.e.*, channels) in decision making. Moreover, the authors did not analyze the computational cost of applying the proposed solution. Additionally, they did not evaluate important evaluation metrics for IoT applications.

Table 1 summarizes the main characteristics of the resource allocation mechanisms analyzed based on optimization goal, energy-efficiency, RSSI-awareness, and LoRaWAN radio parameters considered. Such characteristics significantly improve the system performance in terms

Table 1 Summary of analyzed resource allocation mechanisms for LoRaWAN

Resource allocation	Year	Optimization goal	Optimal	Energy	RSSI	Param.	
mechanisms			solution			SF	CF
Amichi et al. [7]	2019	Fair throughput and reduces energy	✓	✓		✓	
El-Asser et al. [18]	2018	PDR and throughpu				✓	
Caillouet et al. [11]	2019	Minimizes collisions and maximizes number of served nodes	✓			✓	
Sandoval et al. [36]	2019	Maximizes the throughput obtained analytically	✓				
Sandoval et al. [37]	2019	Increase throughput and reduces power consumption	✓	✓			
Farhad et al. [20]	2020	Enhances the packet success ratio by lowering the impact of interference		✓	✓	✓	
Khaled et al. [2]	2018	Addresses the unfair LoRaWAN characteristic		✓	✓	✓	
Zorbas et al. [46]	2018	Improves LoRaWAN capacity and reduces collisions		✓	✓		
Babaki et al.[8]	2020	Improves the noise resilience and PDR		✓		✓	
Cuomo et al. [15]	2017	Reduce collisions		✓	✓	✓	
Dawaliby et al. [16]	2019	Increases throughput reduce power consumption		✓		✓	
Moraes et al. [30]	**2020**	Improve scalability; channel optimization; MILP; heuristic	✓	✓	✓	✓	✓

of maximizing the QoS level by minimizing collisions. Based on our state-of-the-art analysis, we conclude that a resource allocation mechanism's adjustment of radio-related parameters shows promise for configuring such parameters. In this way, an optimization model is essential in formulating and achieving the optimal solution to maximize performance, whereas only a few works [7, 11, 36, 37] introduced an optimization model. Some studies [2, 7, 8, 15, 16, 20, 37] considered energy-efficiency by adjusting the radio-related parameters. Moreover, it is essential to estimate the signal strength based on the distance between the ED and the gateway to provide the tradeoff between increasing the transmission range and reducing the delay, energy, and interference. However, only a few works [2, 15, 20, 46] estimate the RSSI to adjust the radio-related parameters. Finally, none of the existing works efficiently combines SF and CF adjustment to improve the scalability, reliability, and energy-efficiency. To the best of our knowledge, only one study [30] has considered a heuristic and an optimization model for resource allocation that configures radio-related parameters to minimize channel utilization, minimizing collisions while considering RSSI based on the ED's position.

2.1 Research Contributions of this Work

We summarize the main research contributions of this work as follows:

- A MILP model optimally allocates the SF and CF LoRaWAN radio parameters through a MARCO channel optimization.
- A heuristic to efficiently on-the-fly configure the SF and CF parameters named CORRECT.
- An extensive evaluation of the impact of a resource allocation mechanism for LoRaWAN under a dense IoT scenario.

3 Adaptive Resource Allocation for LoRaWAN

In this section, we propose the optimal solution for resource allocation based on MILP, called MARCO. First, we describe a global view of network modeling. Next, we present the MARCO model to use as a benchmark resource allocation. Finally, we propose a heuristic to adaptively choose radio parameters based on the signal strength and distance between the device and the gateway, called CORRECT.

3.1 Network and System Model

For LoRaWAN modeling purposes, we consider two device types: $ed \in ED = \{1, 2, ..., n\}$ and $gw \in GW = \{1, 2, ..., m\}$. Each ED has an identification $i \in [1, n]$, and a tuple $ed_i = (x_i, y_i, z_i, tx_i)$ to represent its geographical coordinates and the transmission power tx_i. In contrast, the tuple $gw_j = (x_j, y_j, z_j)$ represents the geographical coordinates of a given gateway with an identification $j \in [1, m]$. In this context, we consider a set of gw_j deployed in the monitored area, which is deployed based on a positioning algorithm to improve the application performance and reduce the deployed and maintained costs [28]. Each gw_j has a circular coverage area A with a radio range R_j, where there is a set of ed_i deployed around the gw_j.

[LoRaWAN defines three classes of EDs at upper-layer protocols, namely A, B, and C [12].] These classes allow bidirectional communication to support different application requirements. In this article, we considered only Class A devices, which behave by opening a window to wait for messages (downlink) only at the end of its transmission (uplink) because it is the highest energy-efficient class.

In the LoRaWAN architecture, there is a single-hop communication between the ed_i and gw_j over several channels, forming a star network topology [27]. Each ed_i broadcasts messages for neighbor gw_j, which forwards the message to the application server through an existing IP network. LoRaWAN communication is bi-directional, although the uplink communication from ed_i to the network server is strongly favored as expected in many IoT applications. The network server implements the resource allocation mechanism, such as CORRECT, to return the configuration of radio-related parameters, *i.e.*, SF and CF, to each ed_i through the downlink communication, Figure 1 shows. Based on this architecture, LoRaWAN provides connectivity to ed_i deployed over a vast area by employing a control access mechanism with less complexity at the cost of low throughput [33].

We consider that each gw_j could simultaneously decode signals in all SF and CF because LoRaWAN offers different possibilities to orthogonalize the transmissions, *i.e.*, multiple ed_i use the same channel over other SF simultaneously without interference. However, an efficient resource allocation for the uplink transmission is still challenging due to a higher probability of having multiple ed_i transmitting on the same SF and CF on dense LoRaWAN, as expected in future IoT scenarios. In this way, a dense LoRaWAN suffers higher packet loss caused by interference.

Efficient resource allocation for LoRaWAN provides flexibility to adjust a set of radio parameters. In this context, LoRaWAN supports different configurable radio-related parameters to provide a tradeoff between increasing the transmission range and reducing delay, energy, and interference. For instance, each packet can be transmitted with a different SF value ($SF = \{sf_k | (k \in \mathbb{N}) \land (7 \leqslant k \leqslant 12)\}$), which defines how many chirps are sent per second by the data carrier. Higher sf_k values increase the sensitivity and radio range at the cost of growing Time on Air (ToA) and energy consumption to transmit a packet. In comparison, packet transmission with higher sf_k values takes much longer to send packets at lower rates, resulting in more collisions.

SF is a crucial parameter to increase QoS. For instance, ToA increases from 659ms to 1318ms for the packet transmitted with a payload of 20 bytes sent with sf_{12} instead of sf_{11}, respectively, which increases the collisions by keeping the channel busy during a more extended time. [Furthermore, the packet transmitted using sf_{11} consumes ten times more energy compared with using sf_7 [17].] Besides, SF's have a semi-orthogonal design, enabling them to separated into the receptor and avoiding collisions whether they conform to the interference model. Therefore, simultaneous transmissions must select different SFs and channels to avoid collisions.

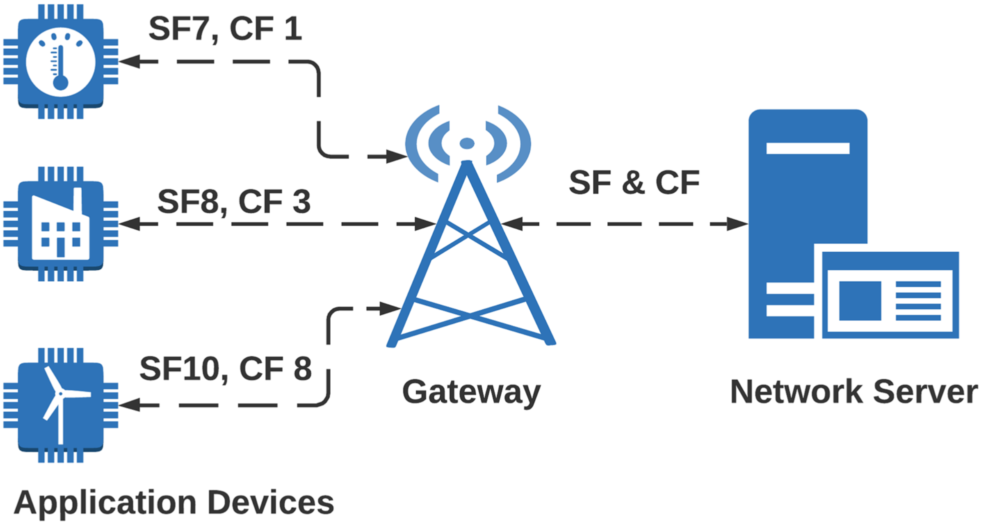


Figure 1 LoRaWAN architecture overview.

$CF = \{cf_f | (f \in \mathbb{F}) \wedge (1 \leqslant f \leqslant 8)\}$ is the center frequency for LoRaWAN communication, spread out on different frequency channels by implementing pseudo-random channel hopping [18]. The cf_f values depend on local frequency regulations, where LoRaWAN gateways typically support eight channels, while the ED usually supports at least 16 channels [18, 35]. It is worth pointing out that the number of channels could be designed to minimize the collision probability. We consider the European frequency plan with eight available channels for the uplink transmission. Furthermore, regional authorities regulate the maximum duty-cycle parameter defined as the maximum percentage of time an end-device can occupy a channel. For instance, if an end-device has a duty-cycle of 1% in the European band, the devices must remain for 100-times the last transmission duration before transmitting it again in the same channel [6]. We consider the European frequency plan with eight available channels for the uplink transmission and a duty-cycle of 1% for this article.

We consider the Long-distance propagation model $L_{Pl}(d)$, as shown in Eq. 1, which has good representation in a large number of indoor and outdoor LoRaWAN environments [19]. The $L_{Pl}(d_0)$ means the path loss at the reference distance d_0, γ denotes the loss exponent, and d denotes the Euclidean distance between a given ed_i and gw_j, which is subjected to $d \leq R_j$. The model parameters are derived from a regression or fitting curve over the measured data and depend on the environment. In this article, we consider γ as 2.32 and $L_{Pl}(d_0)$ as 128.95, which are based on the measures done in Oulu's city (Finland) [32].

$$L_{Pl}(d)[dB] = 10\gamma \log_{10}(d/d_0) + L_{Pl}(d_0) \tag{1}$$

[The total received power ($P_{rx,j,i}$) is used to measure the signal strength of a packet received by a gw_j from the ed_i.] It is computed based on the sum of the device transmission power $D_i \cdot tx$ with the antenna gain GL, subtracting the propagation loss $PL(d)$, as shown in Eq. 2.

$$P_{rx,j,i} = D_i \cdot tx + GL - L_{Pl}(d) \tag{2}$$

[We define the tuple $M = (x_i, y_i, z_i, d, P_{rx,j})$ to have the ED coordinates (x_i, y_i, z_i), Euclidean distance d between a given ed_i and gw_j, and power received $P_{rx,j}$. Specifically, the device power $P_{rx,j,i}$ is used to decide the minimal sf_k value to allow the communication between a given ed_i and gw_j, because gw_j needs to receive a packet with receiver power $P_{rx,j,i}$ higher than the sensibility value for a given SF value [39]. Table 2 shows the sensitivity S_{sf_k} value for each sf_k for a bandwidth of 125 kHz. Therefore, we assign an ED for each sf_k value by guaranteeing that the selected sf_k value provides the packet reception at gw_j with enough power $P_{rx,j,i}$.]

Table 2 Sensitivity value for each sf_k for bandwidth 125 kHz

SF	sf_7	sf_8	sf_9	sf_{10}	sf_{11}	sf_{12}
S_{sf_k} (dBm)	–125	–128	–131	–134	–136	–137

A LoRaWAN packet consists of a combination of non-modulated and modulated chirps. The non-modulated defines the preamble and the Start Frame Delimiter (SFD), while the modulated defines the payload and Cyclic Redundancy Check (CRC). In this way, the time required to transmit a frame over the air (*i.e.*, ToA) depends on the Preamble length (T_{pream}) and the load duration (T_{load}), shown in Eq. 3.

$$ToA = T_{pream} + T_{load} \tag{3}$$

The T_{pream} is computed by the sum of the preamble size (N_{pream}) with the mandatory preamble, *i.e.*, 4.25, multiplying with the symbol duration (T_{simb}), as detailed in Eq. 4.

$$T_{pream} = (N_{pream} + 4.25) \cdot T_{simb^k} \tag{4}$$

The T_{simb} is computed based on Seller et al. [38], as shown in Eq. 5. As a result, a higher SF requires a longer T_{simb}, considering a constant BW.

$$T_{simb^k} = \frac{2^{sf_k}}{BW}, \forall sf_k \in SF \tag{5}$$

The T_{load} is computed based on the load size (N_{load}) multiplied by T_{simb^k}, as shown in Eq. 6.

$$T_{load} = N_{load} \cdot T_{simb^k} \tag{6}$$

We computed N_{load} based on Eq. 7, where PL denotes the packet size, IH means the implicit header, DE represents the data rate optimization. Specifically, IH is 0 if the header is enabled, one otherwise. The implicit header reduces the packet size using predefined CR and the receiving check digit CRC, settings, where the frame header would include these values without it. The DE value is set to 1 if the data rate optimization DE is enabled.

$$\begin{aligned} N_{load} &= 8 + max(ceil\left[\frac{(8PL - 4sf_k + 28 + 16CRC - 20IH)}{4(sf_k - 2DE)}\right] \\ &\quad \cdot (CodingRate + 4), 0) \end{aligned} \tag{7}$$

We compute the Coding Rate value based on Eq. 8.

$$CodingRate = \frac{4}{4 + CR}, CR \in [1, 4] \tag{8}$$

Table 3 describes the main symbols used in this article.

3.2 MARCO

Resource allocation holds several possibilities for configuring SF and CF radio parameters. In this way, MARCO considers a mixed-integer linear optimization formulation to define the ideal configurations of the SF and CF parameters to minimize the channel utilization, which reduces the number of collisions. In general, MARCO take into account a MILP to define the ideal settings of SF and CF parameter, as it considers variables that are not integers. The results of MARCO can be used as a benchmark to those achieved by other heuristics because MARCO represents the best SF and CF parameters configurations.

To optimally allocate the EDs in each sf_k, we assign more ED in some specific sf_k to provide the tradeoff between minimizing channel utilization and reducing the interference, delay, and power consumption. This definition is because the lowest sf_k value (*i.e.*, SF equals 7) supports significantly more devices with lower interference than other SFs, which is explained by the relation between the transmission rate and SF [45]. Moreover, increasing the sf_k value results in longer transmission times, increasing the collisions while keeping the channel busy for a more extended period [45].

In this context, initially we obtain ToA for each sf_k based on Eq.3–8, which is denoted as T_{sf_7}, T_{sf_8}, T_{sf_9}, $T_{sf_{10}}$, $T_{sf_{11}}$, $T_{sf_{12}}$. Based on the T_{sf_k} values, it is possible to define the maximum number of ED for each sf_k, which is fundamental to the optimization process. In this sense, we need to estimate the overall ToA in the channel (T_{ToA}) as follows:

$$T_{ToA} = \sum_{sf_k \in SF} T_{sf_k} \tag{9}$$

We compute the ratio between the ToA for each sf_k ($Ratio_{sf_k}$) by dividing the sum of ToA to normalize the values, as shown in Eq. 10. It aims to obtain the fraction of each ToA concerning the total in the system.

$$Ratio_{sf_k} = \frac{T_{sf_k}}{T_{ToA}}, \quad \forall sf_k \in SF \tag{10}$$

We inverted the $Ratio_{sf_k}$ value according to Eq. 11 because of higher ToA means worse network performance.

$$WeightedSum = \sum_{sf_k \in SF} \frac{1}{Ratio_{sf_k}} \tag{11}$$

We divide the normalized value for each SF ($Ratio_{sf_k}$) with the sum of it ($WeightedSum$), resulting in the ratio of ED for each sf_K, as shown in Eq. 12. Hence, we define the ED ratio to be assigned for each sf_K value based on ToA, which gives priority to have more ED in the lowest sf_k value.

$$Priori_{sf} = \frac{Ratio_{sf_k}}{WeightedSum}, \quad \forall sf_k \in SF \tag{12}$$

Table 3 Main symbols used in our model

Symbol	Description
n	number IoT devices
m	number of GWs
tx_i	IoT device power transmission
ed_i	actual IoT device
i	IoT device index
gw_j	actual GW
j	GW index
R_j	GW radius range
A	circular coverage area
sf_k	actual SF
k	SF index
N	number SF channels
cf_f	actual CF
f	CF index
F	number of CF channels
$L_{Pl}(d)$	log-distance path loss value
$L_{Pl}(d_0)$	path loss in the reference distance
d_0	reference distance
γ	loss exponent
d	euclidean distance between a IoT device ed_i and a GW gw_j
$P_{rx,j,i}$	power received in the GW gw_j from the IoT device ed_i
GL	antenna gain
S_{sf_k}	sensibility value in the SF sf_k
T_{pream}	preamble lengh
T_{load}	load duration
N_{pream}	preamble size
T_{simb^k}	symbol duration in the SF sf_k
N_{load}	load size
PL	packet size
IH	implicit header
DE	data rate optimization
CRC	cyclic redundancy check
CR	coding rate value
T_{sf_k}	ToA in the SF sf_k
$\vartheta_{i,sf,cf}$	binary variable to check if the ed_i with sf_k in the cf_f was chosen by the model
$\delta_{i,sf}$	binary variable to check if the ed_i has enough power for transmitting the packet with the selected sf_k
λ	average transmission rate
U	channel utilization

Finally, we compute the exact number of EDs each sf_k ($Quant_{sf}$), determined based on Eq. 13.

$$Quant_{sf} = Priori_{sf} \times n, \quad \forall sf_k \in SF \tag{13}$$

Besides, the following variables are defined to guarantee the optimal settings of SF and CF parameters:

- $\vartheta_{i,sf,cf} \in \{0.1\}$: binary variable, where a value of 1 means that ed_i with sf_k in the cf_f was chosen by the model, 0 otherwise;
- $\delta_{i,sf} \in \{0.1\}$: a binary variable, where 1 denotes that ed_i has enough power for transmitting the packet with the selected sf_k value, 0 otherwise;
- λ: average transmission rate, measured in packets/second.

The MILP model aims to minimize the channel utilization U and collisions by configuring the SF and CF parameters. MARCO uses the LoRaWAN channel with the lowest possible cost based on the time needed to transmit a frame $T_s f_k$ and the average transmission rate λ, as shown in Eq. 14. Specifically, MARCO computes the channel cost by $\left(T_{sf_k} \times \lambda\right)$, while the variable $\vartheta_{i,sf,cf}$ decides which sf_k and cf_f values a given ed_i will use.

[The constraint, defined by Eq. 15, guarantees that the selected sf_k and cf_f values $(\delta_{d,sf})$ provide packet transmissions with enough power. This behavior is because the gw_j needs to receive a packet with receiver power $P_{rx,j,i}$ higher than the sensibility S_{sf_k} for a given sf_k value [39].] We consider the sensitivity S_{sf_k} values introduced in Table 2. The restriction introduced by Eq. 16 ensures that resource allocation has been made appropriately for all EDs, and the number of EDs is defined previously. The constraint defined by Eq. 17 establishes the ratio of ED to be assigned for each sf_K value based on ToA, which gave priority to have more ED in the lowest sf_k value. Finally, the restrictions introduced by Eqs. 18 and 19 perform channel allocation, considering the reduction of packet collisions on the same SF and CF.

$$\min_{U} \quad U = \sum_{i \in L} \sum_{sf_k \in SF} \sum_{cf_f \in CF} \vartheta_{i,sf,cf} \times \left(T_{sf_k} \times \lambda\right) \qquad (14)$$

subject to:

$$\sum_{sf_k \in SF} \sum_{cf_f \in CF} \vartheta_{i,sf,cf} \times \delta_{d,sf} = 1, \quad \forall i \in n \qquad (15)$$

$$\sum_{i \in L} \sum_{sf_k \in SF} \sum_{cf_f \in CF} \vartheta_{d,sf,cf} = n \qquad (16)$$

$$\sum_{cf_f \in CF} \sum_{i \in L} \vartheta_{d,sf,cf} \times \delta_{d,sf} = n \times Priori_{sf}, \forall sf_k \in SF \qquad (17)$$

$$\sum_{i \in L} (\vartheta_{d,sf,cf} - \vartheta_{d,sf,cf\text{-}c}) \leqslant 1, \quad \forall sf_k \in SF, \quad \forall cf_f \in CF, \forall c \in \{1..(cf_f - 1)\} \qquad (18)$$

$$\sum_{d \in L} (\vartheta_{d,sf,cf} - \vartheta_{d,sf,cf+c}) \leqslant 1, \quad \forall sf_k \in SF, \quad \forall cf_f \in CF, \forall c \in \{1 \cdots (8 - cf_f)\} \qquad (19)$$

3.3 CORRECT

This subsection presents CORRECT, a resource allocation heuristic to efficiently adjust on-the-fly SF and CF to minimize the channel utilization while reducing the interference. Unlike an optimization model, this approach uses mathematical approximations to allocate EDs in the available SF and CF, with a lower computational cost. This procedure makes it possible to implement the algorithm internally to the reduced system of an ED. [The complexity of the CORRECT heuristic is analyzed as follows. In worst case the complexity of CORRECT is $O(n * |SF| * (|CF| + (|CF| - 1)))$. Given the cardinality of the set SF $(|SF|)$ and the set CF $(|CF|)$ equal to 6 and 8, respectively, the number of CFs and SFs can be considered constant values. Hence, the CORRECT heuristic algorithm's complexity is then O(n), where it depends only on the number of EDs n.] In its operation, the CORRECT heuristic receives as input the number of EDs n. Based on Algorithm 1, CORRECT adjusts the SF and CF configurations for each ED in a LoRaWAN network.

Initially, we compute the sum of ToA for each SF (T_{ToA}) based on Eq. 9 to estimate the overall ToA in the channel. Moreover, CORRECT computes the fraction each SF ToA has to T_{ToA} $(Ratio_{sf_k})$ based on Eq. 10. Next, CORRECT inverted the $Ratio_{sf_k}$ value according to Eq. 11 because a higher ToA means worse network performance. Then, CORRECT computes the ratio of IoT devices from a LoRaWAN for each sf_k, as shown Eq. 12, by dividing the normalized value for each SF, $Ratio_{sf_k}$, with the sum of it, *WeightedSum*. Finally, based on the fraction of devices per sf_k obtained, we calculate $Quant_{sf}$ based on Eq. 13, which defines the exact number of EDs on each SF.

In this way, the CORRECT heuristic analyzes if the ED power is enough to transmit based on the sensitivity S_{sf_k} (Line 9). [In this context, we need to estimate the received power $P_{rx,j,i}$ in the gateway gw_j based on Eq. 2 and compare it with the current SF sensitivity S_{sf_k} based on Table 2.] After that, it checks if the number of EDs assigned to the current SF $(\sum ed_i \in \{sf_k\})$ did not exceed the number of devices allowed for such sf_k $(Quant_{sf})$, as shown in Line 10, and assigns the current SF to the device ed_i. This analysis helps ensure that most EDs are distributed in the smaller sf_k, reducing ToA and consequently decreasing collisions. In addition, it checks if the number of EDs assigned to the current CF $(\sum ed_i \in \{cf_f\})$ did not exceed the number of devices allowed for such cf_f $(\frac{Quant_{sf_k}}{CF})$, as shown in Line 13, and assigns the current CF to the device ed_i. Finally, the CORRECT heuristic checks if devices remain without CF configuration, comparing the number of EDs assigned to the current SF ($\sum ed_i \in \{sf_k\}$) and the sum of the number of devices allocated in each cf_f ($\sum_{cf_f \in CF} \sum ed_i \in \{cf_f\}$), and in case there are EDs without configuration, we configure each one ed_i in a cf_f. This step helps reduce the channel overhead, thereby reducing packet collision.

Algorithm 1 CORRECT.

```
input : Number of devices $n$.
output: $sf_k$ and $cf_f$ parameters for each $ed_i$.
Initialization;
Compute $T_{ToA}$ according to Eq. 9;
Compute $Ratio_{sf_k}$ according to Eq. 10;
Compute $WeightedSum$ according to Eq. 11;
Compute $Priori_{sf}$ according to Eq. 12;
Compute $Quant_{sf}$ according to Eq. 13;
for $ed_i \in \{1, \dots, n\}$ do
    for $sf_k \in SF$ do
        if $P_{rx,i} < S_{sf_k}$ then
            if $\sum ed_i \in \{sf_k\} < Quant_{sf_k}$ then
                $ed_i \leftarrow sf_k$;
                for $cf_f \in CF$ do
                    if $\sum ed_i \in \{cf_f\} < \frac{Quant_{sf_k}}{CF}$ then
                        $ed_i \leftarrow cf_f$;
                    end
                end
                if
                $\sum ed_i \in \{sf_k\} > \sum_{cf_f \in CF} \sum ed_i \in \{cf_f\}$
                then
                    for $cf_f \in CF$ do
                        $ed_i \leftarrow cf_f$;
                    end
                end
            end
        end
    end
end
```

4 Evaluation

This section presents an evaluation of the CORRECT heuristic using various simulation tests and performance metrics. We describe the simulation environment and the performance metrics used. Specifically, we analyzed the performance of CORRECT and four other resource allocation heuristics with different numbers of EDs using performance metrics such as PDR, the number of interfered packets, throughput, and transmission delay. Moreover, we analyzed the MARCO model as a benchmark performance for the heuristics we have evaluated.

4.1 Simulation Environment

We implemented the MARCO optimization model using the Optimization Programming Language (OPL) and the IBM CPLEX solver 12.6 on a computer equipped with an Intel (R) Xeon (R) Silver 4112 CPU @ 2.60GHz, 64 GB of RAM running the Ubuntu operating system server. The CPLEX resolution time limit was set to 1h, but all scenarios considered the convergence of MILP before reaching the limit. For example, MARCO converged in 34 minutes, on average, for the scenario with 3000 devices. Besides, we performed several simulation experiments using *NS-3* [22], an open-source discrete-event network simulator, targeted primarily for research and educational use to evaluate the effectiveness of the evaluated resource allocation heuristics. [We deployed one gateway at the center of a two-dimensional scenario with 6x6 km, with a radius of R_j of 2.5 km, which consists of the most challenging scenario due to the higher load of packet interference and packet loss compared with a multi-gateway scenario. Moreover, different numbers of EDs are deployed around the gateway, from 500 to 10000, as expected in massive IoT deployments.] Table 4 summarizes the main simulation parameters.

[We conducted 33 simulations with different randomly generated seeds by the simulator's default pseudo-random number generator (MRG32k3a). The results obtained show the values with a confidence interval of 95%. Each simulation evaluates the performance of different resource allocation solutions for LoRaWAN, namely ADR, Min_ToA, Explora-SF, Explora-AT, CORRECT, and MARCO optimization model.] Precisely, the ADR heuristic adjusts SF and the transmission power based on the distance and physical obstacles in the transmission, which is the implementation provided by *The Things Network* [1]. The Minimum Airtime heuristic (*Min_ToA*) is a standard assignment used

Table 4 Main simulation parameters

Parameters	Value
Region	EU863-870
CR	4/5
BW	125 kHz
Transmission power	14 dBm
Number of GWs	1
Number of EDs	{500, 1000, 2000, 3000, 5000, 7500, 10000}
Average transmission time	5 min
Packet Length	20 bytes
Simulation time	1 hour

by an ED to assign a fixed SF so that packets have the minimum ToA time,*i.e.*, sf_7. The EXPLoRa-SF heuristic distributes the number of EDs equally among the available SF to reduce packet collision [15]. The Explora-AT heuristic distributes the IoT devices along with the SF according to a priority in the lower SFs based on the ToA related to each SF [15]. Finally, MARCO and CORRECT both adjust the SF and CF values[30].

4.2 Performance Metrics

We considered four metrics to evaluate the heuristic for resource allocation of LoRaWAN, namely, PDR, number of interfered packets, network throughput, and transmission delay. ***PDR*** $\in [0, 1]$ is an effective way to analyze network deployment as a whole, *i.e.*, in an optimal network deployment with a minimum number of collisions and losses, *e.g.*, the value is close to 1 [10]. PDR is computed from the total number of received packets and the total number of transmitted packets over the network.

We define ***Interference*** as soon as packets overlap each other in the receptor using the same SF and CF, configuring the called capture effect, *i.e.*, the intra-SF collision. [The inter-SF collision is configured when two packets overlap, using different SF and the same CF. In these two contexts, the Interference causes a low Signal-to-Interference-plus-Noise Ratio (SINR).] [Based on the interference model proposed in [25], the SINR anticipated in a gw_j is computed by Eq. 20.] Where Tx_{packet_0} is the referenced packet transmission power, σ^2 is the variation between the measured path loss $L_{Pl}(d)$ and an expected path loss $EL_{Pl}(d)$, and $\sum_{packet=1}^{N_{Packets}} Tx_{packet}$ is the accumulative transmission power of the interfered packets. Besides, considering the SINR matrix (21), the element T_{sf_i,sf_j} is the margin (in dB units) to consider a packet lost due to interference. For example, suppose the computed SINR of a packet transmitted in sf_7 and a packet transmitted in sf_8 is below -24 dB. In that case, it is configured double packet loss due to collisions, which worsens the system's performance.

$$\text{SINR} = \frac{Tx_{packet_0}}{\sigma^2 + \sum_{packet=1}^{N_{Packets}} Tx_{packet}} \tag{20}$$

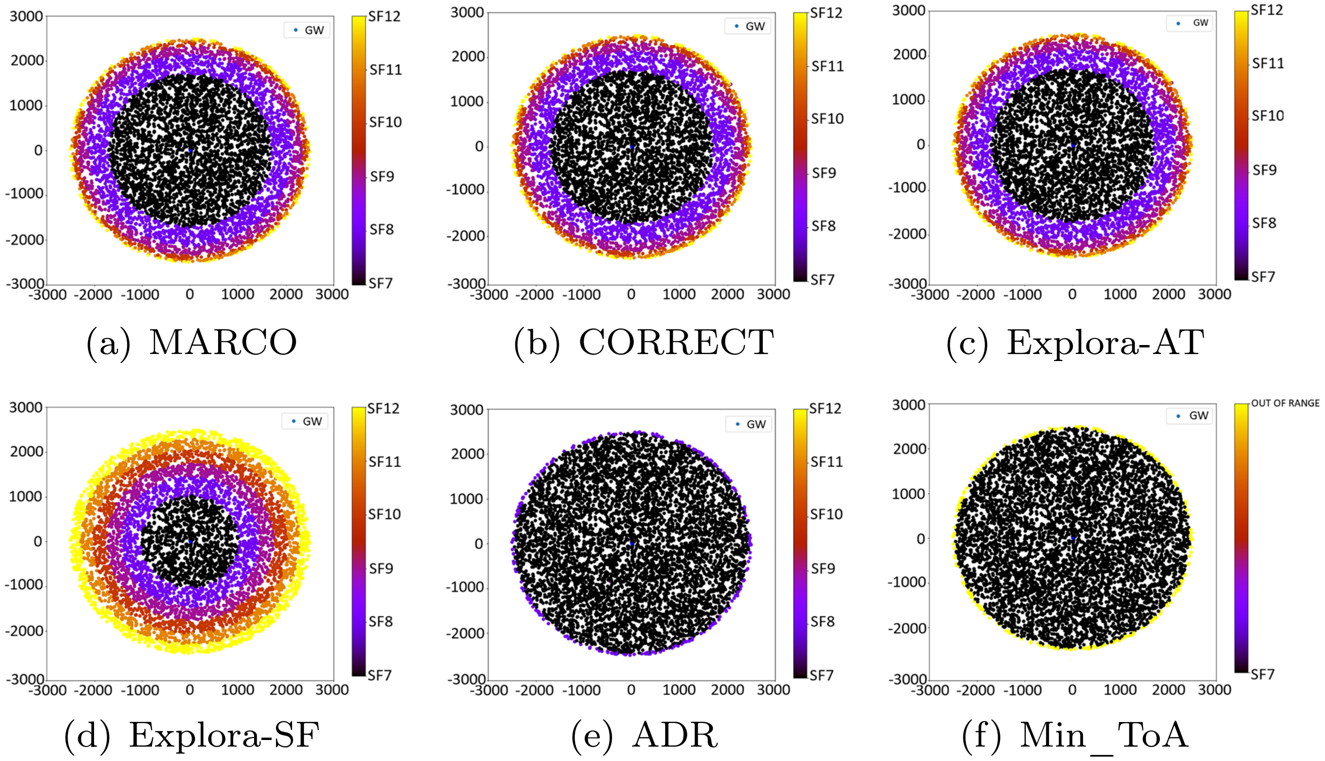
$$\boldsymbol{T} = \begin{array}{c} \\ sf_7 \\ sf_8 \\ sf_9 \\ sf_{10} \\ sf_{11} \\ sf_{12} \end{array} \begin{array}{c} \begin{array}{cccccc} sf_7 & sf_8 & sf_9 & sf_{10} & sf_{11} & sf_{12} \end{array} \\ \left(\begin{array}{cccccc} 6 & -16 & -18 & -19 & -19 & -20 \\ -24 & 6 & -20 & -22 & -22 & -22 \\ -27 & -27 & 6 & -23 & -25 & -25 \\ -30 & -30 & -30 & 6 & -26 & -28 \\ -33 & -33 & -33 & -33 & 6 & -29 \\ -36 & -36 & -36 & -36 & -36 & 6 \end{array} \right) \end{array} \tag{21}$$

We also consider the ***Network Throughput*** as the capacity of all the EDs to send any amount of data to the GW over time. In this work, the throughput is measured in Kbps, and we consider the simulation time as the measuring range. Finally, ***Delay*** is computed as the time spent between sending the packet from the physical layer of the IoT device through the radio channel until the GW's physical layer successfully receives it.

4.3 Results

SF is the leading radio parameter that can be adjusted to influence the radio range, ToA, collisions, and energy consumption. In this sense, Figure 2 represents a snapshot of the SF distribution for the different evaluated models or heuristics when 7500 EDs are deployed around a GW, wherein the behavior is quite similar to other numbers of EDs deployed around a GW. By analyzing the results, we can conclude that both MARCO and CORRECT have similar SF distribution, showing that there is a similar SF allocation caused by the use of $Priori_{sf}$ defined in Eq. 12, as Fig. 2a and b show. CORRECT defines ED ratio to be assigned for each SF value based on ToA, which gives priority to have more EDs in the lowest SF values. For instance,CORRECT assigned 47% of ED to $sf7$, 25.8% to sf8, and so on. CORRECT also assigns the SF value by guaranteeing that the selected SF value provides packet reception at the gateway with enough RSSI. *Min_ToA* assigns all ED in the lowest SF, *i.e.*, sf_7, as Fig. 2f shows, so that maximum data rate with minimum ToA can be achieved. ADR assigns many EDs in the lowest SF, *i.e.*, sf_7, and Fig. 2e shows. Therefore, the amount of SF is similar for *Min_ToA* and ADR, due to the extensive GW coverage range, changing a small portion, which is in sf_8. The SF configuration *Min_ToA* leads to 3.8% of EDs which are out of the range of the gateway. EXPLoRa-SF uniformly distributes EDs among the available SF values to reduce packet collision, as Fig. 2d shows. Finally, EXPLoRa-AT computes a fairness ToA distribution to normalize the ToA for all SF configuration, as Fig. 2c shows. Thus EXPLoRa-AT prioritizes the lower SFs to reduce the total network ToA.

Figure 3 shows PDR for different numbers of EDs on the network for CORRECT, MARCO, ADR, *Min_ToA*, EXPLoRa-SF, and EXPLoRa-AT resource allocation. In this way, simulating a real scenario considering an imperfect SF orthogonality, occurs a higher number of packet loss due to collisions [11]. This problem can be seen by the higher drop in performance in all algorithms with device densification. The CORRECT heuristic provides results remarkably close to the best solution available (*i.e.*, MARCO optimization model). Specifically, CORRECT reduced the PDR by 3.5% in the worst case compared to



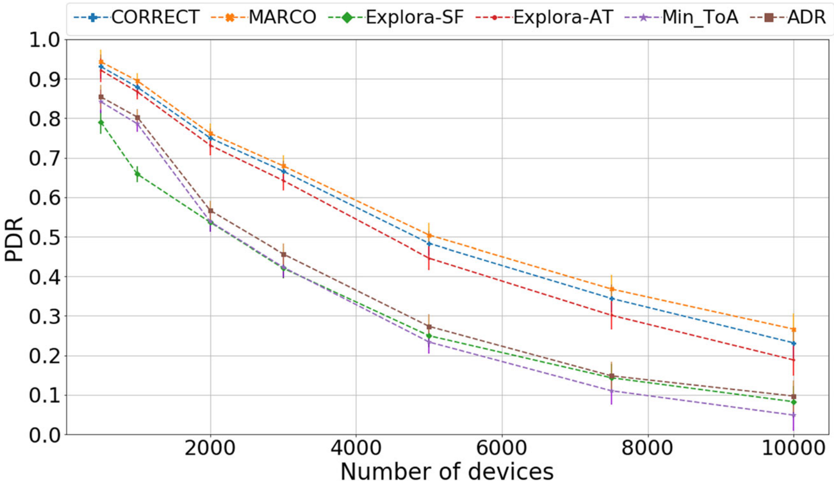
(a) MARCO (b) CORRECT (c) Explora-AT

(d) Explora-SF (e) ADR (f) Min_ToA

Figure 2 SF value for each ED according to each resource allocation mechanism for a scenario with 10000 EDs.

MARCO. This result occurs because CORRECT heuristic efficiently adjusts SF and CF parameters. Specifically, CORRECT assigns the SF value by guaranteeing that the selected SF value provides packet reception at the GW with enough received power. Moreover, it defines the number of devices assigned for each SF value based on ToA, which gives priority enables more ED in the lowest SF values. We also observe that CORRECT performs 4.3%, 14%, 13%, and 18%, better in terms of PDR compared to EXPLoRa-AT, EXPLoRa-SF, *Min_ToA*, and ADR, respectively. The *Min_ToA* and ADR have a poor PDR performance because both set most EDs in the sf_7. This behavior leads to high chances of packet collisions and a low transmission range, resulting in packet losses, especially in networks with high EDs. For instance, the GW will be unable to correctly decode the simultaneous signals sent by the different EDs using the same SF on the same channel. On the other hand, by uniformly distributing EDs in each SF, EXPLoRa-SF assigns many ED in higher SFs. This issue leads to packet collisions probability mainly in sf_{11} and sf_{12} because EXPLoRa-SF occupies the channel longer [11]. Finally, EXPLoRa-AT reduces packet collisions by

Figure 3 PDR according to number of EDs for each resource allocation mechanisms.



balancing the ToA of packets in each SF and decreases packet losses when using lower SF values. EXPLoRa-AT also assigns the SF based on ToA, but EXPLoRa-AT has PDR results 4.3% lower compared to CORRECT because the latter avoids collisions with an efficient CF allocation. Therefore, the PDR results confirm the CORRECT resource allocation heuristic benefits to using the channel with a higher delivery probability, even considering an imperfect SF orthogonality, because CORRECT efficiently adjusts the SF and CF values.

Figure 4 shows the overall throughput (in Kbps) for different numbers of EDs on the network for the evaluated resource allocation models. By analyzing the results, we observe that the throughput increases when the number of EDs increases, regardless of the resource allocation mechanism. This behavior is because the throughput is the sum of the throughput of each ED in the network. However, the packets lost due to collisions considering an imperfect SF orthogonality affects all mechanisms [25]. In the densest scenario, with 10.000 devices, the higher number of packets lost impairs the performance. Moreover, in all algorithms, throughput stabilizes and tends to stop growing after a certain point. This behavior is due to the network's duty cycle limitation limiting the rate of data exchanged on the channel in high device deployments [3]. [Let us analyze the scenario with higher packet losses, *i.e.*, 10.000 EDs deployed around the GW; we note that the CORRECT heuristic yield result 5.35% closer compared to MARCO, with 556.3 and 586.09 Kbps, respectively. Explora-AT also assigns the SF based on ToA, leading to Throughput results of 23% compared to CORRECT because CORRECT avoids collisions due to the efficient allocation of CF. EXPLoRa-SF yields similar results compared to ADR, *i.e.*, 198.70 Kbps, which is 179.96% worse than CORRECT. This behavior occurs because EXPLoRa-SF has many EDs transmitting in higher SFs, which occupy the channel longer, while *Min_ToA* and ADR allocate most EDs in the sf_7.. The SF allocation performed by *Min_ToA* leads to worse Throughput results, *i.e.*, 117.12 Kbps, which is 374.98% worse compared to CORRECT because of the high number of collisions. Specifically, using the CORRECT heuristic instead of the ADR heuristic, up to 18% improvement in PDR is obtained. The heuristic CORRECT yields up 23% lower throughput compared with the Explora-AT.]

Figure 5 shows the number of interference packets, *i.e.*, collisions, for different numbers of EDs on the network for the evaluated resource allocation models. It is essential to highlight that we defined interference as soon as packets overlap each other in the same CF receptor with the same SF and differents SFs regarding the SINR threshold based on matrix (21). In this context, interference has a high probability of packet loss, worsening the system's performance. By analyzing the results, it is possible to conclude that the number of packet collisions for the CORRECT heuristic is close to the MARCO model, being 4.74% higher as it has more collisions due to the packets with the same frequency. This behavior occurs because, in some scenarios, CORRECT has a more significant difference in the number of EDs per CF than MARCO. The similar performance of the CORRECT heuristic compared to the MARCO optimization model is because the CORRECT heuristic efficiently adjusts the SF and CF parameters. In contrast, the number of packet collisions for CORRECT is about 5.34% lower than EXPLoRa-AT, 16.24% lower than EXPLoRa-SF, 14.92% lower than ADR, and 19.24% lower than *Min_ToA*. Specifically, in a scenario with 10.000 EDs deployed around the GW, Explora-SF, ADR, and *Min_ToA* methods experienced 97381, 110058, 108351, and 114142 packet collisions, respectively, while CORRECT has approximately 92180 packets collisions. EXPLoRa-AT also assigns the SF based on ToA. Explora-AT has PDR results 5.34% lower than CORRECT because the latter avoids collision with an efficient CF allocation. This behavior occurs because ADR and *Min_ToA* methods

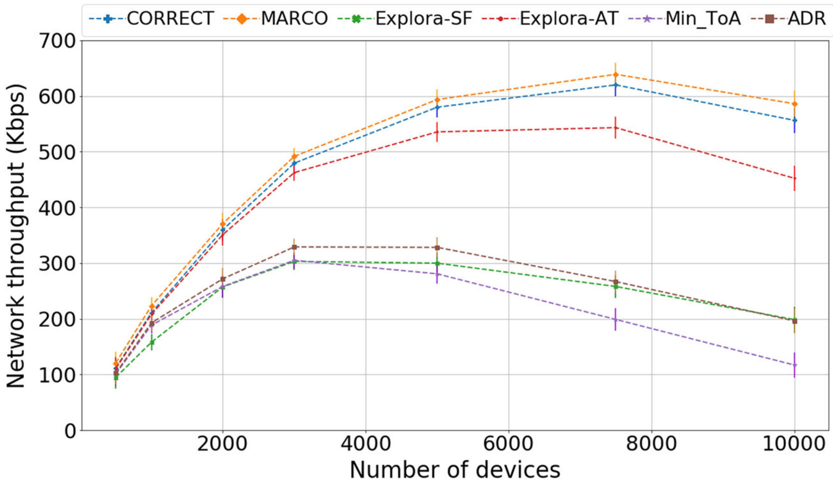


Figure 4 Overall network throughput according to number of EDs for each resource allocation mechanisms.

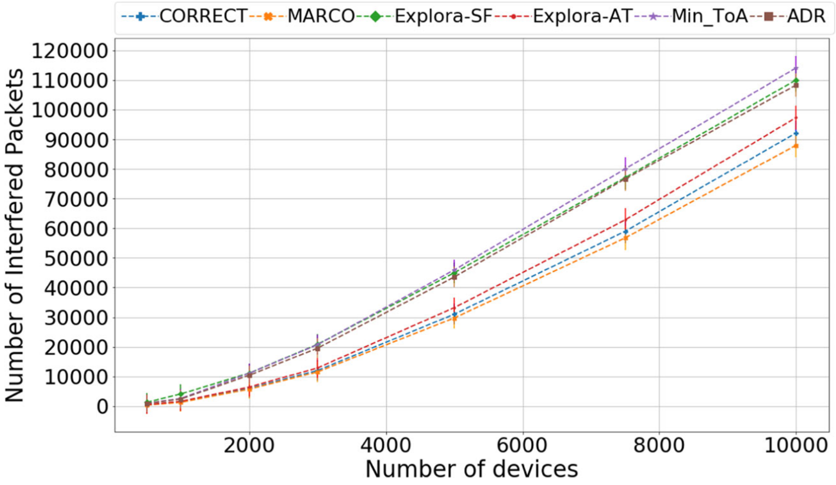


Figure 5 Number of interfered packets according to number of EDs for each resource allocation mechanisms.

assign many EDs in the same SF. Thus, the GW cannot correctly decode the simultaneous signals sent by the different devices using the same SF on the same CF. Finally, the EXPLoRa-SF method equally distributes the number of EDs along with the available SF. Thus there are many EDs with higher SFs, which increase the packet collisions mainly in SFs 11 and 12; once it occupies the channel longer, the collision probability increases [11].

Figure 6 shows the delay results for different numbers of EDs for the evaluated models or heuristics. As expected, ADR and *Min_ToA* delivered the packets with the lowest delay values because they assign the lowest SF values for all EDs, resulting in shorter transmission times. For instance, the ToA increases from 659 ms to 1318 ms for a packets transmitted with sf_{12} instead of sf_{11}, respectively [17]. It is worth mentioning that MARCO and CORRECT have similar delay performance results because the CORRECT heuristic can efficiently adjust the SF and CF values to provide identical performance results compared to MARCO. On the other hand, EXPLoRa-SF delivered packets with the highest delay performance, which is explained by the fact that it mainly considers high SF values, resulting in longer ToA than lower SF. EXPLoRa-AT has a similar SF distribution to MARCO and CORRECT, leading to a similar ToA. However, once the packets are lost due to collisions, the network performance in terms of QoS is affected. Moreover, as the definition, the delay metric only computes the correctly received packets by the GW transmitted from the EDs. Therefore, the packets configured in higher SFs, as of sf_{10}, sf_{11}, or sf_{12}, suffer more collisions because it occupies the channel longer. The collision probability increases [11], causing the average delay to be affected mainly by lower SF packets as the scenario densifies. Consequently, all the algorithms have

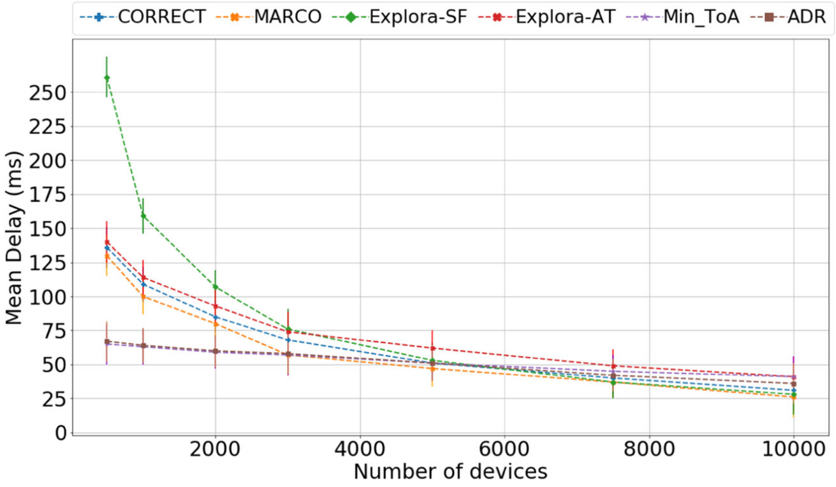


Figure 6 Delay per device according to number of EDs for each resource allocation mechanisms.

Table 5 Performance comparison among evaluated resource allocation mechanisms

Mechanisms	PDR	Throughput	Collisions	Delay
MARCO	63.11%	432.15 Kbps	27620.71	68.14 ms
CORRECT	61.2%	416.85 Kbps	28868.71	74.28 ms
EXPLoRa-AT	58.55%	380.45 Kbps	30688.28	81.85 ms
ADR	45.7%	241.09 Kbps	37396.85	54.42 ms
EXPLoRa-SF	41.17%	224.32 Kbps	38492.85	103 ms
Min_ToA	42.61%	207.12 Kbps	39355.57	54 ms

mostly considered packets received from lower SFs such as sf_7 and sf_8, making larger deployments, such as 10,000 EDs, even delay behavior.

We found that CORRECT demonstrated an efficient performance in delivering high PDR values from our performance evaluation analysis. Consequently, a high throughput due to SF and CF's smart allocation reduces most packets interference even in a semi-orthogonal SF scenario. It considers duty cycle limitations, as shown in interference results, while offering a lower delay. Table 5 summarizes the average performance results from all deployment formats for the evaluated resource allocation mechanisms.

5 Conclusion

Resource allocation is a crucial aspect of LoRaWAN, especially as scalability grows. This article evaluates a heuristic for resource allocation for LoRaWAN, called CORRECT. The evaluated heuristic adjusts the LoRaWAN SF and CF parameters to reduce the channel utilization, packet collisions and, consequently, maximize the number of packets delivered. The results obtained through simulations have shown that the CORRECT heuristic provides results close to optimal obtained by the MARCO model to use the channel, improving the allocation of LoRaWAN parameters to reduce collisions and improve the performance of the system as a whole. Specifically, using the CORRECT heuristic instead of the ADR heuristic, an improvement of up to 18% in PDR is obtained. For other heuristics, this difference is even more significant. As for the network throughput, the heuristic CORRECT heuristic yields up 18.70% improvement compared to the EXPLoRa-AT; this difference increases even more when compared with other heuristics. In the future, we plan to develop a machine learning model that considers SF, ToA, and CF to improve resource allocation.

Acknowledgments We thank the anonymous reviewers for their valuable comments which helped us improve the quality, content, and presentation of this paper. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, and also by the Brazilian National Council for Research and Development (CNPq).

References

1. Adaptive Data Rate (2019). https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate.html.
2. Abdelfadeel, K. Q., Cionca, V., & Pesch, D. (2018). Fair adaptive data rate allocation and power control in lorawan. In *IEEE 19Th international symposium on" a world of wireless, mobile and multimedia networks"(woWMom)* (pp. 14–15). IEEE.
3. Adelantado, F., & et al (2017). Understanding the limits of loraWAN. *IEEE Communications magazine*, *55*(9), 34–40.
4. Akpakwu, G. A. et al. (2018). A survey on 5G networks for the Internet of Things: Communication technologies and challenges. *IEEE Access*, *6*, 3619–3647.
5. Alliance, L. (2015). White paper: A technical overview of LoRa and lora WAN. The LoRa Alliance: San Ramon, CA, USA, pp. 7–11.
6. Alliance, L. (2017). Lorawan 1.1 regional parameters technical specification.
7. Amichi, L., et al. (2019). Joint allocation strategies of power and spreading factors with imperfect orthogonality in LoRa Networks. arXiv:1904.11303.
8. Babaki, J., Rasti, M., & Aslani, R. (2020). Dynamic spreading factor and power allocation of lora networks for dense iot deployments. In *2020 IEEE 31St annual international symposium on personal, indoor and mobile radio communications, pp. 1–6.*
9. Bockelmann, C. et al. (2016). Massive machine-type communications in 5G: physical and MAC-layer solutions. *IEEE Communications Magazine*, *54*(9), 59–65. https://doi.org/10.1109/MCOM.2016.7565189.
10. Bor, M. C., Roedig, U., Voigt, T., & Alonso, J.M. (2016). Do lora low-power wide-area networks scale? In *Proceedings of the 19th ACM international conference on modeling, analysis and simulation of wireless and mobile systems* (pp. 59–67).
11. Caillouet, C. et al. (2019). Optimal SF Allocation in loraWAN considering physical capture and imperfect orthogonality. In *Globecom, 2019.*
12. de Carvalho Silva, J., et al. (2017). LoraWAN – A low power WAN protocol for Internet of Things: A review and opportunities. In *2nd international multidisciplinary conference on computer and energy science* (pp. 1–6). IEEE.
13. Cerwall, P. et al. (2015). Ericsson mobility report. On the Pulse of the Networked Society. Hg. v Ericsson.
14. Croce, D. et al. (2018). Impact of LoRa imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters*, *22*(4), 796–799.
15. Cuomo, F., Campo, M., Caponi, A., Bianchi, G., Rossini, G., & Pisani, P. (2017). Explora: Extending the performance of lora by suitable spreading factor allocations. In *2017 IEEE 13Th international conference on wireless and mobile computing, networking and communications (wimob)* (pp. 1–8). IEEE.
16. Dawaliby, S., Bradai, A., & Pousset, Y. (2019). Adaptive dynamic network slicing in lora networks. *Future Generation Computer Systems*, *98*, 697–707.
17. Duda, A., & Heusse, M. (2019). Spatial issues in modeling loraWAN capacity. In *ACM MSWIm* (pp. 191–198).
18. El-Aasser, M., Elshabrawy, T., & Ashour, M. (2018). Joint spreading factor and coding rate assignment in lorawan networks.

In *Global conference on internet of things (GCIot)* (pp. 1–7). IEEE.
19. El Chall, R., Lahoud, S., & El Helou, M. (2019). Lorawan network: Radio propagation models and performance evaluation in various environments in lebanon. *IEEE Internet of Things Journal*, *6*(2), 2366–2378.
20. Farhad, A., Kim, D. H., & Pyun, J.Y. (2020). Resource allocation to massive internet of things in lorawans. *Sensors*, *20*(9), 2645.
21. Fernandez-Garcia, R., & Gil, I. (2017). An alternative wearable tracking system based on a Low-Power Wide-Area network. Sensors 17(3).
22. Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C., & Kopena, J. (2008). Network simulations with the ns-3 simulator. *SIGCOMM Demonstration*, *14*(14), 527.
23. Kulatunga, C., Shalloo, L., Donnelly, W., Robson, E., & Ivanov, S. (2017). Opportunistic wireless networking for smart dairy farming. *IT Professional*, *19*(2), 16–23.
24. Lyu, J., Yu, D., & Fu, L. (2019). Achieving Max-Min Throughput in LoRa Networks. arXiv:1904.12300.
25. Magrin, D., Centenaro, M., & Vangelista, L. (2017). Performance evaluation of lora networks in a smart city scenario. In *2017 IEEE International conference on communications (ICC)* (pp. 1–7). ieee.
26. Matni, N., Moraes, J., Oliveira, H., Rosário, D., & Cerqueira, E. (2020). Lorawan gateway placement model for dynamic internet of things scenarios. *Sensors*, *20*(15), 4336.
27. Matni, N., Moraes, J., Pacheco, L., Rosário, D., MayOliveira, H., Cerqueira, E., & Neto, A.J.V. (2020). Experimenting long range wide area network in an e-health environment: Discussion and future directions. In *Roceedings of the 16th international wireless communications mobile computing conference (IWCMC* (p. 2020). Cyprus: Limassol.
28. Matni, N., Moraes, J., Rosário, D., Cerqueira, E., & Neto, A. (2019). Optimal gateway placement based on fuzzy c-means for low power wide area networks. In *2019 IEEE Latin-american conference on communications (LATINCOM)* (pp. 1–6). IEEE.
29. Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2018). *A comparative study of lpwan technologies for large-scale iot deployment*. South Korea: ICT Express.
30. Moraes, J., Matni, N., Riker, A., Oliveira, H., Cerqueira, E., Both, C., & Rosário, D. (2020). An Efficient Heuristic loraWAN Adaptive Resource Allocation for IoT Applications. In *25th IEEE symposium on computers and communications (ISCC)* (pp. 1–6). IEEE.
31. Mota, R., Riker, A., & Rosário, D. (2019). Adjusting group communication in dense internet of things networks with heterogeneous energy sources. In *Proceedings of the 11th Brazilian Symposium on Ubiquitous and Pervasive Computing. SBC, Porto Alegre, RS, Brasil.*
32. Petajajarvi, J., Mikhaylov, K., Roivainen, A., Hanninen, T., & Pettissalo, M. (2015). On the coverage of lpwans: range evaluation and channel attenuation model for lora technology. In *2015 14th international conference on ITS telecommunications (ITST)* (pp. 55–59). IEEE.
33. Qadir, Q. M., Rashid, T. A., Al-Salihi, N. K., Ismael, B., Kist, A. A., & Zhang, Z. (2018). Low power wide area networks: A survey of enabling technologies, applications and interoperability needs. IEEE Access.
34. Raza, U. et al. (2017). Low power wide area networks: an overview. *IEEE Communication Surveys & Tutorials*, *19*(2), 855–873.
35. Sallum, E., Pereira, N., Alves, M., & Santos, M. (2020). Improving quality-of-service in lora low-power wide-area networks through optimized radio resource management. *Journal of Sensor and Actuator Networks*, *9*(1), 10.
36. Sandoval, R. M., Garcia-Sanchez, A. J., & Garcia-Haro, J. (2019). Optimizing and updating lora communication parameters: a machine learning approach. *IEEE Transactions on Network and Service Management*, *16*(3), 884–895.
37. Sandoval, R. M. et al. (2019). Performance optimization of LoRa nodes for the future smart city/industry. *EURASIP Journal on Wireless Communications and Networking*, *2019*(1), 1–13.
38. Seller, O. B. A. (2017). Wireless communication method. US Patent 9,647,718.
39. SX1276, L. (2019). 77/78/79 datasheet. Online: http://www.semtech.com/images/datasheet/sx1276.pdf.
40. Taneja, M., Jalodia, N., Malone, P., Byabazaire, J., Davy, A., & Olariu, C. (2019). Connected cows: Utilizing fog and cloud analytics toward data-driven decisions for smart dairy farming. *IEEE Internet of Things Magazine*, *2*(4), 32–37.
41. Tran, C., & Misra, S. (2019). The Technical Foundations of IoT. *IEEE Wireless Communications*, *26*(3), 8–8.
42. Vejlgaard, B. et al. (2017). Interference impact on coverage and capacity for low power wide area IoT networks. In *proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2017)* (pp. 1–6). IEEE.
43. Visconti, P., de Fazio, R., Velázquez, R., Del-Valle-Soto, C., & Giannoccaro, N.I. (2020). Development of sensors-based agri-food traceability system remotely managed by a software platform for optimized farm management. *Sensors*, *20*(13), 3632.
44. Yegin, A., Kramp, T., Dufour, P., Gupta, R., Soss, R., Hersent, O., Hunt, D., & Sornin, N. (2020). Lorawan protocol: specifications, security, and capabilities. In *LPWAN Technologies for iot and m2m applications* (pp. 37–63). Elsevier.
45. Yousuf, A. M. et al. (2018). Throughput, coverage and scalability of LoRa LPWAN for internet of things. In *IEEE/ACM 26Th international symposium on quality of service* (pp. 1–10).
46. Zorbas, D., Papadopoulos, G. Z., Maille, P., Montavont, N., & Douligeris, C. (2018). Improving lora network capacity using multiple spreading factor configurations. In *proceedings of the 25th International Conference on Telecommunications (ICT 2018)* (pp. 516–520).