An Efficient Heuristic LoRaWAN Adaptive Resource Allocation for IoT Applications

Jean Moraes*, Nagib Matni*, André Riker*, Helder Oliveira*, Eduardo Cerqueira*, Cristiano Both†, Denis Rosário*
*Federal University of Pará (UFPA), Belém – Brazil
†University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo – Brazil
Email: jean.anjos.moraes@itec.ufpa.br, cbboth@unisinos.br, {nagib, afr, heldermay, denis, cerqueira}@ufpa.br

***Abstract*—Long Range Wide Area Network (LoRaWAN) enables flexible long-range communication with low power consumption and low-cost design perspectives. However, 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 flexibility in the LoRaWAN transmission configurations allows higher management in the use of end-device parameters, which allows better resource utilization and improves network scalability. This paper proposes an adaptive solution to handle the define best LoRaWAN parameter settings to reduce the channel utilization and, consequently, maximize the number of packets delivered. Additionally, to validate our method, we formulated mixed-Integer linear programming and results compared to those given by the heuristics. Results provided by the heuristic are close to those provided by the MILP.**

***Index Terms*—LORA, LoRaWAN, IoT, MILP**

I. Introduction

Currently, Internet-of-Things (IoT) is expanding at a fast rate to provide connections for billions of IoT devices, where the everyday environment will soon have a large number of IoT devices per square meter [1]. Both Academy and Industry estimate that Internet would have approximately 500 billion IoT devices connected to Internet until 2030, where long-range transmission technologies will support 9% of these connections [2]. IoT technology changed society's behavior, generating a diversity of new applications due to heterogeneity requirements for IoT device communication, *e.g.*, low energy consumption, high coverage, Quality of Service (QoS) support, and massive Machine-Type Communication.

To achieve this growth, Long Range Wide Area Network (LoRaWAN) provides extended coverage to operate in unlicensed and pure implementation frequency ranges with low cost, low energy consumption, and flexible transmission rate. 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 causing a network overload and create the called hotspot problem, which results in signal interference and affects the QoS due to packet loss occasioned by collisions [3]. 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).

One of the main benefits of LoRaWAN is the flexibility concerning adaptively configurable radio-related parameters, which can increase or decrease the channel utilization on-demand, by means of resource allocation mechanisms. An efficient adaptive resource allocation mechanism must be adjusting on-the-fly 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, adaptive resource allocation based on a heuristic method is required the make-decisions computationally applicable in a LoRaWAN. The state-of-the-art focused on the resource allocation aims the optimization models using SF allocation [4], SF and Coding Rate (CR) allocation [5], and increase flow [6]. However, none of the works considers the device requirements for an efficient resource allocation on-the-fly adjust the configuration of radio related parameters to maximize channel utilization while minimizing collisions.

We introduce this paper a heuristiC fOr adaptive REsource alloCaTion on LoRaWAN for the IoT applications (CORRECT). It dynamically adjusts the LoRaWAN parameters to reduce interference and packets collision, and thus maximizes the channel utilization and increases the number of delivered packets. The heuristic chooses the settings based on the signal strength and distance between the device and gateway to provide the *tradeoff* between the increasing of the transmission range and the reduction of delay, energy, and interference. Furthermore, we introduce a Mixed Integer Linear Programming (MILP) called Optimization Model for LoRaWAN Resource AlloCation for IoT ApplicatiOns (MARCO) to compare the proposed heuristic with an optimal solution for resource allocation. Simulation results show that CORRECT increases in 11.8% the Data Extraction Rate (DER), reduces 3.153 times the packet's collision rate, and increases packet delivery ratio compared to the Adaptive Data Rate (ADR), the LoRaWAN default heuristic. Furthermore, the proposed heuristic has results close to the optimum model, being just 0.6% worse, considering DER and have 1.124 times more collisions.

This paper is organized as follows. Section II presents the state-of-the-art, which explores resource allocation in LoRaWAN. Section III introduces the proposed heuristic model, called CORRECT. Section IV explores the simulation model developed to evaluate CORRECT and obtained results. Finally, Section V concludes the work and introduces future directions.

II. Related Works

This section presents the most recent works which focus on LoRaWAN resource allocation. Furthermore, in each work, we discuss their advantages and disadvantages. Amichi *et*

978-1-7281-8086-1/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Southeast University. Downloaded on January 20,2022 at 07:57:51 UTC from IEEE Xplore. Restrictions apply.

al. [4] 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 reduce energy consumption according to the SF choice.

El-Aasser *et al.* [7] introduced two SF allocation heuristics, which adjust the SF service radius. However, setting SF based only on distance is not recommended as it depends on the signal strength received by the device. Caillouet *et al.* [8] designed an optimization model for LoRaWAN that optimize SF allocation to minimize collisions and maximize number of served nodes. Nevertheless, this model does not take into account channel allocation and the choice of SF to reduce energy consumption.

Sandoval *et al.* presented two Markov chains to model the Transmission Cycle (TC) and energy consumption [6]. Based on these models, the optimal solution finds TC that increases throughput while keeping power consumption below a threshold. The authors also present a solution for calculating a global network configuration that maximizes the throughput obtained analytically [5]. Such a proposal formulates a Markov Decision Process (MDP), in which the optimal update policy allows the system to maximize the accumulated network throughput for a short time interval.

It is possible to conclude based on the state-of-the-art that an efficient resource allocation must consider the reception power at the gateway and distance for adjusting on-the-fly radio-related parameters. To the best of our knowledge, this is the first attempt that considers a heuristic and an optimization model for resource allocation that configures radio related parameters to maximize channel utilization, minimizing collisions while considering signal strength using device position.

III. Adaptive Resource allocation for LoRaWAN

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

A. Network and System Model

Efficient resource allocation for LoRaWAN provides flexibility to adjust the radio parameters. In this sense, LoRaWAN supports different radio related parameters to provide a *trade-off* between the increasing of the transmission range and the reduction of delay, energy, and interference. Each packet can be transmitted with a different SF value, which can be defined as the ratio between the symbol and chirp rate, where higher SF values increase the sensitivity and radio range at the cost of an increase in Time on Air (ToA) and energy consumption to transmit a packet. Packet transmission with higher SF values takes much longer to send packets at lower rates, resulting in more collisions For instance, for the packet with a payload 20 bytes transmitted with SF 12 instead of 11, ToA increases from 659ms to 1318ms, respectively. Furthermore, a device using SF11 consumes ten folds more energy for transmission than when using SF7 [9]. Besides, SF's have an orthogonal design, enabling them to easily separated in the receptor and, avoiding collisions. Moreover, CF determines the transmission frequency, which can range from 137 MHz to 1020 MHz in 61 Hz steps.

In this paper, we assumed that a single gateway could simultaneously decode signals in all SF and CF. Even though LoRaWAN offers different possibilities to orthogonalize the transmissions, an efficient resource allocation for the uplink transmission, it is a challenging task. This challenge exists due to a higher probability of having IoT devices transmitting on the same SF and CF on dense LoRaWAN, as expected in future IoT scenarios. In this way, the LoRaWAN dense suffers higher packet loss caused by interference. Therefore, we introduce a resource allocation to adjust SF and CF parameters, maximizing channel utilization while minimizing interference and collisions.

For LoRaWAN modeling purposes, it is considered two device types: IoT devices and Gateway (GW). IoT devices are defined into classes that determine its functionality in communication: Class A, Class B, and Class C. In this implementation, we considered only Class A devices, which behaves by opening a window to wait for messages (downlink) only at the end of its transmission (uplink). In this context, the set of GWs must be deployed in the network area, based on a positioning algorithm to improve the application performance and reduce the deployed and maintained costs [10]. GW has a circular coverage area A with a radio range R_j, where there are a given number of devices N deployed following a uniform distribution. Each IoT device has a identification $i \in [1, N]$, and a tuple $D_i = (x_i, y_i, z_i)$ to represent its geographical coordinates. Moreover, the tuple $GW_j = (x_j, y_j, z_j, tx_j)$ represents the GW geographical coordinates and the transmission power tx_j for each GW with a identification $j \in [1, M]$. Furthermore, f denotes transmission frequency, and C means a constant with $3dB$ as a default value.

The distance between a given IoT device i and GW_j is computed based on the Euclidean Distance $dist_{D_i,GW_j}$, where $dist_{D_i,GW_j} \leq R_j$ defines that IoT device i is on the GW_j coverage. We consider the COST231 propagation model based on Okumura-Hata model, which is represented by $L_{b,i}$ in Eq. 1 [11]. In this context, $a(D_i.z, f)$ denotes the environment propagation for the urban environment computed based on Eq. 2.

$$L_{b,i} = 46.3 + 33.9\log_{10}(GW_j \cdot tx) - 13.82\log_{10}(GW_j \cdot z) + C \\ -a(D_i.z_n, f) + (44.9 - 6.55\log_{10}(GW_j \cdot z))\log_{10}(dist_{D_i,GW_j}) \tag{1}$$

$$a(D_i.z, f)\begin{cases} 8.29 \cdot (\log_{10}(1.54D_i.z))^2 - 1.1 & ,if \quad 150 \leq f \leq 200 \\ 3.2 \cdot (\log_{10}(11.75D_i.z))^2 - 4.97 & ,if \quad 200 < f < 1500 \end{cases} \tag{2}$$

The total power received ($P_{rx,j}$) by GW_j from the IoT device i is computed based on the sum of the device transmission power $D_i \cdot tx$ with the antenna gain GL, subtracting the propagation loss $L_{b,i}$, as shown in Eq. 3.

$$P_{rx,j} = D_i \cdot tx + GL - L_{b,i} \tag{3}$$

We define the tuple $L = (x_i, y_i, dist_{D_i,GW_j}, P_{rx,j})$ to have the Euclidean distance values and power received for each device. The device power received by a given GW_j ($P_{rx,j}$) is used to decide which is the minimal SF value to allow

Authorized licensed use limited to: Southeast University. Downloaded on January 20,2022 at 07:57:51 UTC from IEEE Xplore. Restrictions apply.

the communication between a given IoT device i and GW_j, since GW_j needs to receive a packet with receiver power $P_{rx,j}$ higher than the sensibility value for a given SF value [12]. The set of sensitivity is defined as $SF = \{sf_k | (k \in \mathbb{N}) \wedge (7 \leqslant k \leqslant 12)\}$, and the values for sf_k are shown in Table I.

TABLE I
SENSIBILITY FOR BANDWIDTH 125 KHZ

SF	sf_7	sf_8	sf_9	sf_{10}	sf_{11}	sf_{12}
Sensitivity (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 CRC. In this way, the time required to transmit a frame (ToA) depends on the Preamble length (T_{pream}) and the load duration (T_{load}), shown in Eq. 4.

$$ToA = T_{pream} + T_{load} \tag{4}$$

T_{pream} is computed by the sum of the preamble size (N_{pream}) with the mandatory preamble, *i.e.*, 4.25, which is then multiplied with the symbol duration (T_{simb}), as detailed in Eq. 5.

$$T_{pream} = (N_{pream} + 4.25) \cdot T_{simb^k} \tag{5}$$

T_{simb} is computed based on Seller [13], as shown in Eq. 6. As a result, a higher SF requires a longer T_{simb}, considering a constant bandwidth BW.

$$T_{simb^k} = \frac{2^{sf_k}}{BW}, \forall sf_k \in SF \tag{6}$$

T_{load} considers the load size (N_{load}) multiplied by T_{simb^k}, as shown in Eq. 7.

$$T_{load} = N_{load} \cdot T_{simb^k} \tag{7}$$

We computed N_{load} based on Eq. 8, 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, 1 otherwise. The implicit header reduces the packet size using predefined CR and the receiving check digit, Cyclic Redundancy Check (CRC), settings, where without it, the frame header would include these values. The DE value is set to 1 if data rate optimization DE is enabled.

$$N_{load} = 8 + max(ceil\left[\frac{(8PL - 4SF + 28 + 16CRC - 20IH)}{4(SF - 2DE)}\right] \cdot (CR + 4), 0) \tag{8}$$

We compute the CR value based on Eq. 9.

$$CR = \frac{4}{4+n}, n \in [1, 4] \tag{9}$$

We give priority to the use of some specific SF with a higher number of IoT devices to provide the *tradeoff* between maximizing channel utilization and reduce the interference, delay, and power consumption. This specification is needed because of increasing SF results in longer transmission times, increasing the collisions, while keeping the channel busy for a longer period of times [14]. Furthermore, the lowest SF value (*i.e.*, SF equals to 7) supports significantly more devices with lower interference compared to other SFs, due to the relation between the transmission rate and SF [14]. In this way, we obtain ToA for each SF denoted as $T_{sf_7}, T_{sf_8}, T_{sf_9}, T_{sf_{10}}, T_{sf_{11}}, T_{sf_{12}}$ based on Eqs. 4-9. We compute $Ratio_{sf_k}$ by dividing the sum of ToA to normalize the values.

The sum of ToA for each SF (T_{ToA}) is computed as follows:

$$T_{ToA} = \sum_{sf_k \in SF} T_{sf_k} \tag{10}$$

Initially, we compute the ratio between the ToA for each SF ($Ratio_{sf_k}$) with each SF (T_{ToA}) for the sum of ToA for all SF value, as shown Eq. 11.

$$Ratio_{sf_k} = \frac{T_{sf_k}}{T_{ToA}}, \quad \forall sf_k \in SF \tag{11}$$

Afterwards, we inverted the $Ratio_{sf_k}$ value according to Eq. 12, since higher ToA means worse network performance.

$$WeightedSum = \sum_{sf_k \in SF} \frac{1}{Ratio_{sf_k}} \tag{12}$$

Finally, dividing the normalized value for each SF, $Ratio_{sf_k}$, with the sum of it, *WeightedSum*, we compute the ratio of IoT devices from a LoRaWAN in each SF, as shown Eq. 13.

$$Priori_{sf} = \frac{Ratio_{sf_k}}{WeightedSum}, \quad \forall sf_k \in SF \tag{13}$$

In this way, we assign the SF value by guarantying that the selected SF value provides the packet reception at GW with enough power. Furthermore, we define the ratio of devices to be assigned for each SF value based on ToA, which gives priority to have more IoT devices in the lowest SF values.

B. MARCO

MARCO aims to maximize the channel utilization by minimizing collisions of the LoRaWAN through the adjust of SF and CF radio parameters. In general, MARCO considers a MILP to define the ideal settings of SF and CF parameters. The results of MARCO can be used as a benchmark to those achieved by other heuristics since MARCO represents the best SF and CF parameters configurations. The following variables are defined for the optimal resource allocation:

- $\vartheta_{i,sf,cf} \in \{0.1\}$: binary variable, where value of 1 means that IoT device i with spreading factor *sf* in the channel *cf* was chosen by the model, 0 otherwise;
- $\delta_{i,sf} \in \{0.1\}$: binary variable, where 1 denotes that an IoT device i has enough power for transmitting with spreading factor *sf*, 0 otherwise;
- λ: average transmission rate, measured in packets/second.

The MILP model to maximize the channel utilization by minimizing collisions by configuring the SF and CF parameters is formulated as in following. MARCO aims to use the LoRaWAN channel with the lowest possible cost based on the time needed to transmit a frame and the average transmission rate, as shown in Eq. 14. For this implementation it is configurated the Europe frequency plan which uses 8 available channels for uplink transmission, defined as $CF = \{cf_k | (k \in \mathbb{N}) \wedge (1 \leqslant k \leqslant 8)\}$ [15]. In this sense, MARCO computes the channel cost by $\left(T_{sf_k} \times \lambda\right)$, while the variable $\vartheta_{i,sf,cf}$ decides which SF (sf) and CF (cf) a given IoT device i will use. The constraint defined by Eq. 15 guarantees that the

Authorized licensed use limited to: Southeast University. Downloaded on January 20,2022 at 07:57:51 UTC from IEEE Xplore. Restrictions apply.

selected device i has enough power to use a given SF ($\delta_{d,sf}$), where each IoT device i compares each sf with the sensitivity values in Table I. The restriction introduced by Eq. 16 ensures that resource allocation has been made appropriately for all devices, and the number of devices is defined previously. The constraint defined by Eq. 17 ensures the amount of SF based on the priority vector T_{sf_k}. Finally, the restrictions introduced by Eqs. 18 and 19 perform channel allocation, considering the reduction of packet collisions on the same SF and channel.

$$\underset{U}{\text{Min}} \quad U = \sum_{i \in L} \sum_{sf \in SF} \sum_{cf \in CF} \vartheta_{i,sf,cf} \times \left(T_{sf_k} \times \lambda\right) \tag{14}$$

subject to:

$$\sum_{sf \in SF} \sum_{cf \in CF} \vartheta_{i,sf,cf} \times \delta_{d,sf} = 1, \quad \forall i \in L \tag{15}$$

$$\sum_{d \in L} \sum_{sf \in SF} \sum_{cf \in CF} \vartheta_{d,sf,cf} = N \tag{16}$$

$$\sum_{cf \in CF} \sum_{d \in L} \vartheta_{d,sf,cf} \times \delta_{d,sf} = N \times Priori_{sf}, \forall sf \in SF \tag{17}$$

$$\sum_{d \in L} \left(\vartheta_{d,sf,cf} - \vartheta_{d,sf,cf\text{-}c}\right) \leqslant 1, \quad \forall sf \in SF, \quad \forall cf \in CF, \forall c \in \{1..(cf-1)\} \tag{18}$$

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

C. CORRECT

This subsection presents the resource allocation heuristic called CORRECT, which efficiently adjusts LoRaWAN radio parameters to maximize channel utilization while reducing the interference. The CORRECT heuristic receives as input the number of IoT devices N and also the number o available frequency channels CF according to the frequency plan of each region [15]. Based on Algorithm 1, CORRECT adjusts the SF and CF configurations on-the-fly for each IoT device in LoRaWAN with a lower computational cost.

Initially, it is calculated $Priori_{sf}$ according to the steps defined in Subsection III-A, which determines the fraction of the total of devices that must be allocated in each SF. Besides that, it is calculated $Quant_{sf}$ based on Eq. 20, which defines based on the number of IoT devices N the exact quantity of devices each SF can allocate. In this way, the CORRECT heuristic analyses if the device power is enough to transmit based on the sensitivity for each SF, as shown in Table I. Therefore, $Quant_{sf}$ checks if the current SF did not exceed the number of devices allowed for such SF, to ensure that most devices are distributed in the smaller SFs, which reduce ToA and consequently decrease collisions. Finally, CORRECT heuristic fairly distributes the number of IoT devices with a given SF value in each CF, assigning the carrier frequency with the lowest quantity of devices to ensure reducing channel overhead and thereby reducing packet collision.

$$Quant_{sf} = Priori_{sf} \times N \tag{20}$$

Algorithm 1: CORRECT

```
input : Number of devices N and number of channels CF.
output: sf and cf parameters for each IoT device.
initialization;
Calculate Priori_sf according to Eq. 13;
Calculate Quant_sf according to Eq. 20;
for sf ∈ {7,...,12} do
    if device has enough power for the sf then
        if devices in sf < Quant_sf then
            configure sf to the device;
            for cf ∈ {1,...,CF} do
                if devices in cf < Quant_sf / CF then
                    configure cf to the device;
                end
            end
            if device without CF configuration then
                configure the device to the channel cf with
                  the lowest use;
            end
        end
    end
end
```

IV. Evaluation

This section introduces the simulation evaluation of CORRECT heuristic, including scenario, metrics, methodology, and results. Specifically, we analyzed the performance of CORRECT and four other resource allocation heuristics with a different number of IoT devices concerning DER, the number of packet collisions, energy consumption, ToA, SF Distribution, and Fairness Index. Moreover, we analyzed MARCO model as a benchmark for the heuristics evaluated.

A. Methodology

We implemented the MARCO model in Optimization Programming Language (OPL) and solved using IBM CPLEX solver 12.6 on a computer with Intel (R) Xeon (R) Silver 4112 CPU @ 2.60GHz, 64 GB of RAM in the Ubuntu operating system Server. We set the CPLEX resolution time limit to 1h. Several experiments were carried out using the *LoRaSim* [16] simulator to evaluate the effectiveness of the CORRECT. For the *LoRaSim* scenario, we considered a GW with a radius of R_j of 1.5 km, and 100, 250, 500, 750, 1000, 2000 e 3000 IoT devices uniformly deployed in the simulation area transmitting data every 5 minutes with 20-byte packets. We assume CR $4/5$, BW of 125 kHz, and transmission power of 14 dBm.

We performed simulations with different resource allocation solutions for LoRaWAN. ADR heuristic adjusts SF, DR, BW, and CR based on the distance and physical obstacles in the transmission, which is implementation provided by *The Things Network* [17] network. The Minimum Airtime heuristic (*Min_Airt*) is a standard assignment used by IoT devices to assign a fixed CF and SF so that packets have the minimum ToA time. Equal distribution heuristic (*Eq_Distr*) distributes the number of IoT devices equally between the CF and SF values. Finally, MARCO and CORRECT both adjust the SF and CF values, as introduced in Section III.

We used six metrics to evaluate the heuristic for resource allocation of LoRaWAN, namely, DER, number of collisions, ToA, energy consumption, number of SF allocation, and fairness index. DER evaluates performance in a numerical range

Authorized licensed use limited to: Southeast University. Downloaded on January 20,2022 at 07:57:51 UTC from IEEE Xplore. Restrictions apply.

between 0 and 1 computed according to Eq. 21, where value is equal to 1 means optimal network deployments. For the calculation of DER, it is required to calculate the number of received packets P_r, the number of packet collisions Col, and the number of sent packets P_{total}. DER does not capture the performance of an individual device. Furthermore, it analyzes the deployment of the network as a whole, in which DER is severely affected by the number of collisions [16].

$$DER = \frac{P_r - Col}{P_{total}} \tag{21}$$

We define collisions as soon as packets overlap each other in the receptor using the same SF and CF, causing interference and packet loss, worsen the system's performance. The frequency collision can be defined when the difference between the frequency of two packets is under a threshold established by the bandwidth according to the LoRa module used. The SF collision can be established by the equality of SF used for transmitted two packets. Moreover, when the difference of the power of the transmissions is relatively low, a power collision is caused, and GW for default keeps listening to two alternately, so it does not demodulate any of the transmissions.

We compute ToA based on Eq. 4, enabling us to analyze the packet transmission time. The energy consumption is defined as the energy used by the IoT device to extract a message successfully. The fairness index is used to demonstrate how well distributed the devices are in each SF. Besides that, to know the total time spend of each SF, this metric takes into account the multiplication of the amount of IoT devices in each SF N_{sf} by ToA for the respective SF using the vector ToA_{SF}, demonstrated in Subsection III-A. The fairness index is computed according to Eq. 22.

$$FairnessIndex = \frac{(\sum_{sf=7}^{12} N_{sf} \times T_{sf})^2}{6 \times \sum_{sf=7}^{12} (N_{sf} \times T_{sf})^2} \tag{22}$$

B. Results

Fig. 1(a) shows the number of collisions per node for different numbers of devices on the network for the evaluated resource allocation models. By analyzing the results, it is possible to conclude that the number of collisions per node for CORRECT heuristic is pretty close to MARCO model, being 10% higher since it has more collisions by frequency, because of in some scenarios, CORRECT has a more significant difference in devices per CF than MARCO. The similar performance of CORRECT heuristic compared to MARCO model is due to CORRECT heuristic efficiently adjusts SF and CF parameters. Specifically, it assigns the SF value by guarantying that the selected SF value provides packet reception at GW with enough power. It also defines the ratio of devices to be assigned for each SF value based on ToA, which gives priority to have more IoT devices in the lowest SF values. On the other hand, the number of collisions per device for the CORRECT is about 3 times less than ADR and *Min_Airt*. For instance, ADR and *Min_Airt* methods experienced 28.781 and 30.737 collision per node, respectfully, while CORRECT has approximately 9.45 packets collision in the scenario with 3000 devices. This behavior occurs because ADR and *Min_Airt* methods assigns the same SF for all IoT devices, and thus thus GW will be unable to correctly decode the simultaneous signals sent by the different devices using the same SF on the same channel. Finally, *Equal_Distr* method equally distributes the number of IoT devices along the available SF, and thus there are many IoT devices with higher SFs. This increases ToA, and also the packet collisions mainly in SFs 11 and 12, since occupy the channel longer increase the collision probability [8].

Fig. 1(b) shows DER for different numbers of devices on the network for CORRECT, MARCO, ADR, *Min_Airtime*, and *Equal_Distr* resource allocation. From the results, it is observed that CORRECT performs 11.8% better in terms of DER compared to ADR and *Min_Airt*, and MARCO,*i.e.*, optimal solution, is 0.6% better than CORRECT heuristic. In other words, CORRECT provides results remarkably close to the best solution available. Hence, DER results also confirms the benefits of CORRECT heuristic to better use channel with lower number of collisions and higher delivery probability since CORRECT efficiently adjusts SF and CF values.

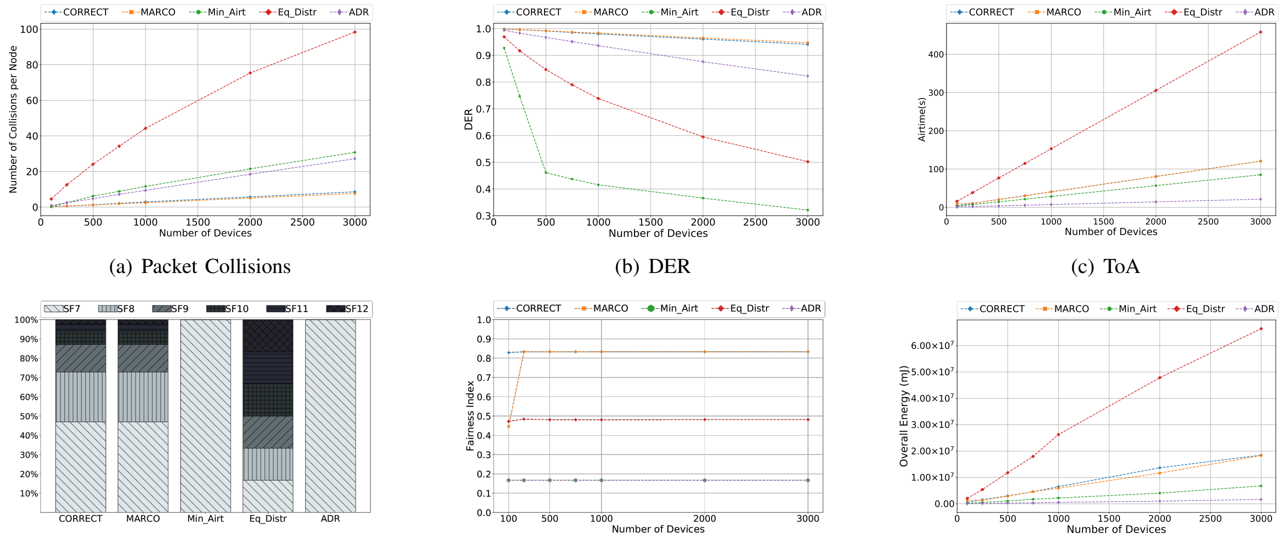
Fig. 1(c) shows ToA results for different numbers of devices for the evaluated model or heuristics. It is important to mention that MARCO and CORRECT have the same ToA performance. This behavior happens because CORRECT heuristic is able to efficiently adjust the SF and CF values to provide similar performance compared to MARCO. On the other hand, ADR and *Min_Airt* delivered the packets with the lowest ToA values, since they assign the lowest SF values for all IoT devices resulting in shorter transmission times. For instance, ToA increases from 659ms to 1318ms for a packet transmitted with SF 12 instead of 11, respectively [9]. *Equal_Distr* delivered packets with highest ToA performance, which is explained by the fact that it mainly considers high SF values, resulting in longer ToA than lower SF.

Fig. 1(d) shows the SF distribution for a scenario with 3000 devices for the different models or heuristics. Both MARCO and CORRECT have similar SF distribution, showing that there is a resembling SF allocation caused by the use of $Priori_{sf}$ defined in Eq. 13. For ADR and *Min_Airt*, the amount of SF is occasionally identical, changing only the CF allocation. This behavior occurs due to both selects the lowest SF value to deliver packets with lowest ToA. Finally, *Equal_Distr* equally distributes the number of IoT devices between the available SF values.

Fig. 1(e) demonstrates the Fairness Index in terms of ToA for different numbers of devices on the network. From the results, it is observed that all the evaluated heuristics and model maintain constant behavior in most of the scenarios. This behavior happens because for a high number of devices is maintained the SF distribution. In this sense, it is observed that CORRECT and MARCO present the fairest approach concerning the other methods because they perform a regular distribution of SF, prioritizing which have less ToA. However, unlike CORRECT, MARCO does not have a well-defined priority of SF for small numbers of devices, being unfair for this scenario. Besides that, ADR and *Min_Airt* have lower values because they concentrate all devices in SF7, having a higher total expenditure.

Fig. 1(f) illustrates the total energy consumed in mJ, depending on the different number of devices. Analyzing the results, it can be seen that ADR has the lowest energy consumption regardless of the number of devices in the network

Authorized licensed use limited to: Southeast University. Downloaded on January 20,2022 at 07:57:51 UTC from IEEE Xplore. Restrictions apply.



(a) Packet Collisions

(b) DER

(c) ToA

(d) SF Distribution

(e) Fairness Index

(f) Energy Consumption

Fig. 1. Simulation Results for Different Number of Devices

since it strategically configures some devices for smaller SFs, with energy consumption being inversely proportional to the increase in SF. For instance, a device using SF 11 consumes 10 times more energy for a transmission than when using SF 7 [9]. In this sense, the *Min_Airt* has the second-lowest consumption, as it has a fixed SF setting at the lowest values for all devices. On the other hand, MARCO and CORRECT appear as the third and fourth most efficient performance respectively, being the CORRECT approximately 1.167 more costly, since they distributes the devices proportionally even if it is possible to allocate with an SF that consumes less energy, but this affects the network's DER. Therefore, CORRECT gain in the performance of the network as a whole, but loses for having a higher energy consumption.

V. CONCLUSION

Resource allocation is a crucial aspect of LoRaWAN, especially as scalability grows. This paper introduces a heuristic and a model for resource allocation for LoRaWAN, called CORRECT and MARCO, respectively. The proposed heuristic adjusts the LoRaWAN SF and CF parameters to reduce the channel utilization, packets collision and, consequently, maximize the number of packets delivered. The results obtained through simulations showed that the CORRECT heuristic provides results close to optimal obtained by MARCO model to use of the channel, improving the allocation of LoRaWAN parameters to reduce collisions and improve the system as a whole. Specifically, using the CORRECT heuristic instead of the ADR heuristic, an improvement of up to 11.8% in DER is appreciated; for other heuristics, this difference is even more significant. As for the number of collisions, the heuristic CORRECT show up 3.153 times better, compared to the one used by LoRaWAN. Regarding the energy consumption of the network, despite the heuristic CORRECT having the best DER, it still has the fourth lowest energy consumption.

REFERENCES

[1] G. A. Akpakwu *et al.*, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[2] "MYTHINGS next-gen wireless technology for the industrial internet of things," https://behrtech.com/lpwan-technology/, accessed: 2020-01-20.

[3] U. Raza *et al.*, "Low power wide area networks: An overview," *IEEE Communication Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[4] L. Amichi *et al.*, "Joint Allocation Strategies of Power and Spreading Factors with Imperfect Orthogonality in LoRa Networks," *arXiv preprint arXiv:1904.11303*, 2019.

[5] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Optimizing and updating lora communication parameters: A machine learning approach," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 884–895, 2019.

[6] R. M. Sandoval *et al.*, "Performance optimization of LoRa nodes for the future smart city/industry," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–13, 2019.

[7] M. El-Aasser *et al.*, "Joint Spreading Factor and Coding Rate Assignment in LoRaWAN Networks," in *IEEE GCIoT*. IEEE, 2018, pp. 1–7.

[8] C. Caillouet *et al.*, "Optimal SF Allocation in LoRaWAN Considering Physical Capture and Imperfect Orthogonality," in *Globecom, 2019*.

[9] A. Duda and M. Heusse, "Spatial issues in modeling LoRaWAN capacity," in *ACM MSWiM*, 2019, pp. 191–198.

[10] N. Matni *et al.*, "Optimal Gateway Placement Based on Fuzzy C-Means for Low Power Wide Area Networks," in *LATINCOM*, 2019, pp. 1–7.

[11] E. Harinda *et al.*, "Comparative Performance Analysis of Empirical Propagation Models for LoRaWAN 868MHz in an Urban Scenario," in *5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 154–159.

[12] L. SX1276, "77/78/79 datasheet," *Online: http://www. semtech. com/images/datasheet/sx1276. pdf*, 2019.

[13] O. B. A. Seller, "Wireless communication method," May 9 2017, uS Patent 9,647,718.

[14] A. M. Yousuf *et al.*, "Throughput, coverage and scalability of LoRa LPWAN for internet of things," in *IEEE/ACM 26th International Symposium on Quality of Service*, 2018, pp. 1–10.

[15] L. A. T. C. R. P. Workgroup, "Rp002-1.0.0 lorawan® regional 49 parameters," *Online: https://lora-alliance.org/sites/default/files/2020-01/rp_2-1.0.0_final_release.pdf*, 2019.

[16] M. C. Bor *et al.*, "Do LoRa low-power wide-area networks scale?" in *ACM MSWiM*. ACM, 2016, pp. 59–67.

[17] "Adaptive Data Rate," https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate.html, 2019.

Authorized licensed use limited to: Southeast University. Downloaded on January 20,2022 at 07:57:51 UTC from IEEE Xplore. Restrictions apply.