

Received December 19, 2019, accepted February 5, 2020, date of publication February 11, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973252

Deriving and Updating Optimal Transmission Configurations for LoRa Networks

RUBEN M. SANDOVAL¹, DAVID RODENAS-HERRAIZ²,
ANTONIO-JAVIER GARCIA-SANCHEZ³, AND
JOAN GARCIA-HARO³, (Member, IEEE)

¹Biyectiva Technologies, 30353 Cartagena, Spain

²Ocado Technology, London EC1Y 8AF, U.K.

³Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain

Corresponding author: Ruben M. Sandoval (rubenmartinez@biyectiva.com)

This work was supported in part by the project AIM under Grant TEC2016-76465-C2-1-R (AEI/FEDER, UE), and in part by the regional projects e-DIVITA, under Grant 20509/PDC/18 (Proof of Concept, 2018). The work of Ruben M. Sandoval was supported by the Spanish MECED for an FPU under Grant FPU14/03424.

ABSTRACT LoRa-based networks exhibit good flexibility in terms of configurable parameters and adjustable modulation properties. Thanks to this, wireless nodes can be tuned to improve their communication behavior. In fact, optimal network-level transmission configurations (C_{opt}) can be derived in such a way that the global network performance is maximized. To derive this C_{opt} , one must know the radio-propagation behavior of each node beforehand. Traditionally, this has been pursued by using general, low-precision, propagation models due to the infeasibility (in terms of time and energy) of deriving each individual node propagation behavior. In this work we propose a straightforward bounding technique that reduces up to 73% the energy and time required to obtain the radio-propagation behavior of each individual node in the network, enabling the derivation of network-level optimal transmission configurations. Also, we provide mechanisms to keep this knowledge updated, swiftly reacting to changes in the environment and leading to network performance improvements of 15% when compared to traditional alternatives like LoRaWAN ADR. Furthermore, by means of a testbed we demonstrate that this mechanism can also provide resistance to Denial-of-service attacks. Finally, we incorporate the power consumption into the proposed C_{opt} formulation and provide a generalizable power-consumption determination methodology. This way we can limit the set of eligible transmission configurations to help extending LoRa network lifetimes more than 40%.

INDEX TERMS LoRa, optimization, IoT, networking, LPWAN.

I. INTRODUCTION

LoRa networks are conceived to consist of hundreds, or even thousands of devices. This large number of IoT nodes would unavoidably increase the level of interference and congestion present in the wireless band. In light of this, it is clear why it is important to intelligently handle spectrum congestion to avoid a degradation of the network performance [1], [2].

Thanks to the pseudo-orthogonal properties and versatility of the LoRa modulation [3], collisions (which conform the main form of destructive interference) can be greatly reduced by carefully deriving an optimal network-level configuration [4]. That is, each node would be assigned a transmission configuration in such a way that this pseudo-orthogonal nature of the LoRa modulation is exploited to reduce packet

collisions. Such a global configuration (denoted henceforth as C_{opt}) would maximize the network-level performance by providing each device with its own transmission configuration. In LoRa, transmission configurations are defined by three parameters: Transmission power –TXP–, Spreading Factor –SF– (the ratio between the symbol rate and chip rate), and Coding Rate –CR– (the ratio between the length of the packet and the length of the error-correction code) [5]. Note that we intentionally leave out the bandwidth –BW–, which is also configurable, because all LoRa nodes that belong to the same LoRaWAN network must use the same bandwidth. Furthermore, we consider that LoRa nodes randomly choose one of the available frequencies, which is a known and standard technique to further reduce collisions [6].

However, the specific radio propagation environment as well as the presence of other potentially interfering wireless devices –operating in the same frequency bands as

The associate editor coordinating the review of this manuscript and approving it for publication was Min Jia ¹.

LoRa— make the final optimal network-level configuration \mathbf{C}_{opt} strongly dependent on the particular deployment location. In this sense, it is crucial to have a method to characterize the environment and, ultimately, the quality of the wireless link. The result of this characterizing method would be a model (denoted as M) of the environment that would help us determine how transmitted signals behave under each transmission configuration and how much interference is already present so as to derive a more accurate \mathbf{C}_{opt} .

Furthermore, LoRa nodes are devices whose power consumption should be kept to a minimum as they are typically off-grid; that is, they are either battery-powered or relying upon harvesting energy from the medium. Depending on the specific application LoRa network operators may choose to restrict the eligible transmission configurations to increase power-efficiency of certain devices (by only allowing less power-consuming transmission configurations). From a methodological point of view, this can be accomplished by reducing (or penalizing) the set of eligible configurations per device when deriving \mathbf{C}_{opt} . To precisely determine which transmission configurations require the most energy, power consumption of LoRa nodes must be considered. Specifically, the power consumption derived from transmitting data packets under each transmission configuration must be calculated. Therefore, a consumption study is also needed to derive a power consumption model.

Finally, once we have a sound model of the power consumption of the device (used to select a set of eligible transmission configurations), and a model of how the environment behaves (M), \mathbf{C}_{opt} can be computed. However, environments and even wireless deployments rarely remain stable over time, and a way to maintain \mathbf{C}_{opt} and M updated is required as well. For that purpose, we also propose a trigger-based method whose main objective is to keep up with changes in the environment and network. By keeping track of the radio propagation conditions, our method is capable of detecting when the environment has changed enough so it is worth, in terms of energy and time, analyzing it again. By way of a simple test case, this trigger-based algorithm is also shown to provide resilience against Denial-of-Service (DoS) attacks.

A. CONTRIBUTIONS

The main contribution of this work is a hierarchical and structured methodology that is shown to increase performance of LoRa-based networks by:

- 1) Modeling the behavior of the radio propagation environment in which such a network is deployed.
- 2) Incorporating this knowledge in classical formulas for maximizing network performance.
- 3) Keeping the derived models up-to-date in an inexpensive and robust fashion.

Finally, by means of a testbed we showcase the application of the above methodology and prove it to outperform traditional approaches such as Adaptive Data Rate (the most popular algorithm for adapting transmission configurations in LoRa networks) [2].

The remainder of the paper is organized as follows. Related work is introduced in Section II. Section III presents a mathematical analysis of the network throughput. Next, in Section IV a thorough procedure to model power consumption is derived and discussed. The effects of packet losses in \mathbf{C}_{opt} is covered in Section V. In the next section, Section VI, we describe a methodology to keep \mathbf{C}_{opt} updated when the propagation environment changes. Then, Section VII, presents a set of experiments and, finally, conclusions are drawn in Section VIII.

II. RELATED WORK

A. SF ALLOCATION / GLOBAL CONFIGURATION

The problem of SF allocation in LoRa networks has been addressed from various perspectives. Some authors have focused on assuming mathematically tractable node distributions: homogeneous, Gaussian, or annular distribution [7], [8], or distributions based on Poisson processes [9]. This facilitates the mathematical analysis and lead to closed-form expressions when some propagation model (i.e. Rayleigh, Log-normal, etc.) are considered.

Targeting the problem mathematically has many benefits: closed-form expressions are ideal for optimizing network performance and with the availability of powerful network simulators like ns-3 [10] or Omnet [11], complex scenarios can be analyzed. However, in real life, nodes are not deployed following mathematical distributions. In fact, the underlying idea of the IoT is to provide computing and communication capabilities to those *things* that are in need of them; not to those placed on a convenient location. Also, one of the main disadvantages of such an approach is that it heavily relies on the Signal-to-Noise ratio (SNR) to derive the Packet Reception Ratio (PRR). This relation strongly depends on the particular LoRa radio transceiver in use [3], thus losing the generalizing capabilities of the mathematical models. Finally, mathematically modeling the distribution of nodes assumes that the deployment is immutable, and therefore, it intrinsically considers that it cannot change.

SF adaptation is addressed in [12], [13] in order to ensure fairness in terms of data rate; that is to make all nodes experience the same effective network data rate. Although in some cases this might be of interest, we believe that, in a general scenario, the heterogeneous nature of IoT must be regarded and the specific importance of each node must be individually considered. In this sense, we have included a term of *importance* into our formulation. This incorporates the idea that not all nodes should be treated equally. For example, in a Smart City, a node controlling a lamp post may not be as important as a node controlling a water pump, and therefore, more network resources should be devoted to the latter. Furthermore, works in [12] and [13] (and all the papers commented above) assume that nodes can only have a single transmission configuration (and thus, they optimize performance for such a configuration). However, in practice, nodes can alternate between many transmission configurations to,

in a global sense, have a mix of any arbitrary number of configurations. This enriches the performance characterization and allows for further performance improvement (as will be seen in the following sections).

B. MODELING NETWORK PACKET RECEPTION RATE

When it comes to estimate the PRR, most works [8], [13]–[15] oversimplified LoRa networks by assuming that, as long as the SNR is over a given threshold –which is made to depend on the SF–, all packets will be received, and hence, the PRR is 1. When the SNR drops below such a threshold, no packets are received, and as such, the PRR is zero. This *binary* PRR vs SNR relation is far from reality [4], [5] and leads to incorrect SF allocation.

Other more intricate approaches resorted to empirical analyses [4] to characterize the continuous and smooth relation between the PRR and the SNR. This represents a natural step forward in increasing the precision of LoRa simulations. Yet, the imperfect orthogonality of SFs is not considered in [4], hindering the applicability of it in high-precision modeling. Also, as regards increasing simulation accuracy, one should acknowledge that precise PRR vs SNR relations will, again, ultimately depend on the specific LoRa transceivers employed and the particular environment.

For the above reasons, in the next section we introduce the proposed approach: instead of assuming simplistic conditions (i.e. uniform node distribution or a predictable PRR vs SNR relation), we opt for characterizing the precise environment.

III. NETWORK THROUGHPUT

Modeling network performance is a popular task in research works on LoRa, mainly due to its potential to lay the foundations for a later performance optimization. Traditionally, this implies determining the Signal-to-Noise Ratio (SNR) and/or the Packet Reception Ratio (PRR) of each LoRa link. To this end, and as mentioned in previous paragraphs, most works resorted to: (i) assuming rather simple node distributions when modeling a LoRa-based network (e.g. uniform density, Gaussian distribution, etc.), (ii) implementing generic radio propagation models that do not consider the particularities of each environment like the log-normal shadowing path-loss models [16], or (iii) employing environment-agnostic *PRR vs SNR* models to determine the likelihood of receiving a particular packet (that is, determining the PRR by means of the perceived SNR, which effectively ignores packet collisions and other complex phenomena).

In contrast to these alternatives, we propose to incorporate in the performance-maximization process a previous step in which the specific propagation environment is characterized. The result of this characterization is the aforementioned model (M) which reflects the PRR for each and every transmission configuration –required to model the network performance as shown in the following paragraphs–. The reason why the PRR cannot be simply derived from the SNR (that is, the third alternative described above) is because each of the pseudo-orthogonal channels may behave differently

depending on whether the same pseudo-orthogonal channel is under used by other nodes of the same or different LoRa-based network. Furthermore, *PRR vs SNR* models depend on radio transceivers and may not be readily available for some particular LoRa nodes (or specific combinations of transmitter-receiver nodes).

That is, instead of employing a generic propagation model to derive the PRR value for each node (that is, compute the SNR based on the link distance and some pre-defined noise model), we first characterize the medium –obtain a model M – to determine the actual perceived PRR for each LoRa node. This way, the computed optimal global configuration is completely particularized to the specific medium in which nodes are deployed. Furthermore, we also allow to limit the eligible transmission configurations as per the power consumption restrictions of each individual node. Finally, we demonstrate that the process of acquiring M is worth doing it in terms of time and battery consumption.

As indicated in Section I, the final objective of this task is to maximize the network throughput by determining the optimal transmission configuration of each device. The configuration for a device i is a vector \mathbf{c}_i whose k -th entry indicates the percentage of packets that must be generated under the transmission configuration k . The global network configuration is a set whose i -th entry corresponds to \mathbf{c}_i , that is, $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N\}$ for a network of N devices. Note that, as previously indicated, a transmission configuration is a combination of a specific Spreading Factor, Coding Ratio, and Transmission Power value expressed in dBm. Although discussing how network throughput is precisely modeled is out of the scope of this work (interested readers are referred to [3]), suffices to say that the full context of each node i is considered: the length of generated packets L_i , the rate at which such packets are transmitted λ_i , the importance of the node in the network G_i (that is, “the assigned level of priority by the application”), the PRR attained for each k configuration PRR_i^k , and the probability of a packet collision under such a k configuration ϕ_i^k . Note that this packet collision probability ϕ_i^k can be derived from the study of the LoRa MAC protocol (see [17] for a more in-depth analysis).

Eq. 1 (again, taken from [3]) computes the total throughput of the network (Γ) for a given global configuration matrix \mathbf{C} by aggregating the throughput of all N nodes and all K configurations. Note that, by maximizing Γ –employing any off-the-shelf maximization algorithm– we can obtain the optimal global network configuration \mathbf{C}_{opt} .

$$\Gamma(\mathbf{C}) = \sum_{i=1}^N \sum_{k=1}^K \xi_i^k \cdot \lambda_i \cdot c_i^k \cdot L_i \cdot G_i \cdot PRR_i^k \cdot (1 - \overline{\phi_i^k}), \quad (1)$$

where $\xi_i^k \in \{0, 1\}$ indicates whether the transmission configuration k is allowed for device i (according to the power consumption restrictions imposed to node i) and $c_i^k \in [0, 1]$ indicates the percentage of packets that node i transmits under configuration k . Note that this approach is in stark contrast with traditional works where nodes could only have a

single transmission configuration. In Eq. 1, the fact that nodes can use multiple transmission configurations is considered. Furthermore, PRR_i^k , the packet reception ratio for a packet transmitted by node i under configuration k , depends on the specific radio propagation environment, and thus, this is the value to be found when modeling the environment (i.e. obtaining M).

Since to attain C_{opt} we need to know λ_i (the packet generation rate of each node), we can distinguish two different scenarios:

- 1) We have full control over all network devices and can know or directly set λ_i for all i . This is the simplest scenario. λ_i is known beforehand and it is simply fed into Eq. 1
- 2) Some λ_i are unknown. Not only does this imply that such a parameter must be estimated but also that it is subject to change at any point. Two sub-scenarios can be considered:
 - Some (or all) LoRa devices can set their own packet sending rate (e.g. nodes can increase their information report rate when some event must be reported). We have to estimate λ_i for such devices.
 - The network is composed of N_{own} owned devices and N_{ext} external devices with $N = N_{own} + N_{ext}$. Obviously, devices out of our control may have an unknown λ .

We propose to estimate unknown λ values by employing the Maximum-Likelihood Estimator (MLE) following the procedure described in [5]. This can be achieved by averaging the observed inter-arrival time of packets $E[T_i]$ and invert it, that is: $\lambda_i = \frac{1}{E[T_i]}$. Note that this assumes that an underlying Poisson distribution governs the packet generation process –which is a fairly common assumption in Low-Power Wide Area Networks (LPWAN) networks [4], [12], [13], [15], [18], [19]–. We extend the procedure of [5] by, instead of averaging all the observed inter-arrival times for a node i , considering a moving average that assigns exponentially larger weights to recent samples. This has the additional benefit of increasing the importance of more recent samples; thus responding faster to changes in λ_i (e.g. when an external node dramatically changes its packet-generation rate).

IV. MODELING POWER CONSUMPTION

As indicated in Eq. 1, one can restrict the set of eligible transmission configurations by setting any arbitrary ξ_i^k to zero. The main idea behind this is to prevent nodes from using a very power-demanding configuration that may lead to an early exhaustion of their batteries. However, to make a sensible selection of allowed transmission configurations, the power consumption of the deployed nodes must be known beforehand.

Following the methodology detailed in [20], we have employed the Monsoon Power Monitor [21] to precisely determine the current consumption of the PyCom LoPy 4.0 nodes [22], a very popular LoRa IoT node platform. Note that for our measurements, we exclusively tested the

TABLE 1. Current consumption for each state of a LoPy 4.0 node.

State Number	Description	Duration	Current Consumption
1	Wake Up	540ms	51.84mA
2	Radio Preparation	560ms	59.41mA
3	Crafting Packet	78.65ms	48.58mA
4	Transmission	See Table 6 in [20]	See Table 2
5	Wait 1st window	988.9ms	39.62mA
6	1st receive window	See Table 6 in [20]	54.01mA
7	Wait 2nd window	See Eq. 4 in [20]	39.62mA
8	2nd receive window	33.8ms	54.01mA
9	Post-processing	710ms	39.59mA
10	Turn off sequence	47.71ms	3.31mA
11	Sleep	Application dependent	$20 \cdot 10^{-6}$ mA

consumption of such a device, without any external sensors or additional boards attached to it. Table 1 specifies the obtained results. Note that, as each Transmission Power value (TXP) leads to a different power consumption, the 13 configurable values (from 2dBm to 14dBm in 1dBm steps) have been tested and the current consumption determined. Table 2 shows these values.

Using Eq. 2 below and Tables 1 and 2, one can compute the current consumption (per packet transmission), I_{pkt} , by simply aggregating the currents consumed in each of the eleven states detailed in Table 1 –denoted as I_i – multiplied by their respective duration (T_i). Finally, power consumption (in Joules) can be determined by multiplying the current consumed by the voltage with which the node is powered, that is, 3.3V.

$$I_{pkt} = \sum_{i=1}^{i=11} T_i \cdot I_i \quad (2)$$

TABLE 2. Current consumption vs transmission power. Note that transmission power is capped at 13dBm to avoid exceeding maximum legal radiated power in some parts of the world like Europe.

Transmission Power	Current Consumption
2dBm	76.01mA
3dBm	78.27mA
4dBm	80.59mA
5dBm	83.75mA
6dBm	85.53mA
7dBm	89.02mA
8dBm	93.20mA
9dBm	94.14mA
10dBm	101.35mA
11dBm	103.32mA
12dBm	106.54mA
13dBm	114.15mA
14dBm	114.15mA

To illustrate the importance of power consumption let us consider a simple scenario: for an 18-byte packet transmission with the largest Spreading Factor (SF12), the by-default Coding Rate (CR4/5), and TXP being set to 8dBm, the total power consumption is approximately 0.94J per packet (including the *waking up* and *going back to sleep* LoPy processes). Considering that nodes send (on average) one packet every hour and regular 3000mAh 3.7V Li-Ion batteries are used, the lifetime of the network would be around 3.45 years. When deriving C_{opt} , this value could be easily extended to 4.84 years (40% improvement) by, for example, limiting the

allowed SFs to just SF7, that is, $\xi_i^k = 1$ only for those k for which SF = SF7.

V. MODELING THE PRR

At this point, the only missing piece for the computation of C_{opt} is PRR_i^k , that is, a model M that indicates the expected PRR of a packet generated by node i under transmission configuration k .

Note that given the quasi-orthogonal characteristics of the SFs, the SNR may drastically vary for each SF. Therefore, the PRR under each SF must be individually obtained. Another interesting fact is that, due to the so-called capture effect [23], the PRR may rapidly decrease once the SNR drops below a certain level. Since this reduction cannot be easily anticipated (especially when third-party LoRa networks are deployed), different TXP values must be swept.

For a given set of configurable SFs, CRs and TXPs, the total number of configurations is the Cartesian product of such parameters. Throughout the rest of the paper we consider SFs = {7, 8, 9, 10, 11, 12}, CRs = $\{\frac{4}{5}, \frac{4}{7}\}$, and TXPs = {2dBm, 6dBm, 10dBm, 14dBm}—we have restricted the CR values to those commonly considered in the IoT arena [4], [5] and TXP values to those enforced in LoRa nodes by the standard [24]–. Therefore, the total number of testable parameters K is 48 ($6 \times 2 \times 4$). The simplest and most direct way to evaluate PRR_i^k is making node i transmit a number of probe packets using configuration k . Then, the gateway will count the proportion of such packets that made it through the wireless channel to compute the PRR. Once this is accomplished, a new configuration k is chosen and this process is repeated until all K configurations have been swept. However, it is not intuitive to determine the number of packets that a LoRa node must send to attain a certain level of confidence over the measured PRR. To resolve this issue, we have used the formulas derived by N. Megill and M. Pavicic in [25]. Noting that every packet transmission is a realization of a Bernoulli process with success probability p (precisely, the PRR), the upper and lower bound of our p estimation are expressed as follows:

$$\begin{aligned} p_{lower} &= IB_{\frac{1}{2} \cdot (1-c)}^{-1}(m+1, n-m+1) \\ p_{upper} &= IB_{\frac{1}{2} \cdot (1+c)}^{-1}(m+1, n-m+1) \end{aligned} \quad (3)$$

where IB_x^{-1} is the inverse regularized incomplete beta function (see [25] for more details), $c \cdot 100\%$ is the desired confidence interval in percentage ($c = 0.9$ has been chosen for this study), m represents the positive cases (correctly received packets), and n the total tested events (total number of sent packets). From these formulas two facts can be stated: (i) the number of packets to be sent to reach a certain confidence level is determined by on the underlying *unknown* PRR of the channel. (ii) As expected, if a high confidence interval (c) or tighter bounds are required (i.e. the difference between p_{upper} and p_{lower}), a larger amount of probe packets is to be sent. In this work, we have opted for arbitrarily fixing the allowed uncertainty in terms of upper and lower bounds,

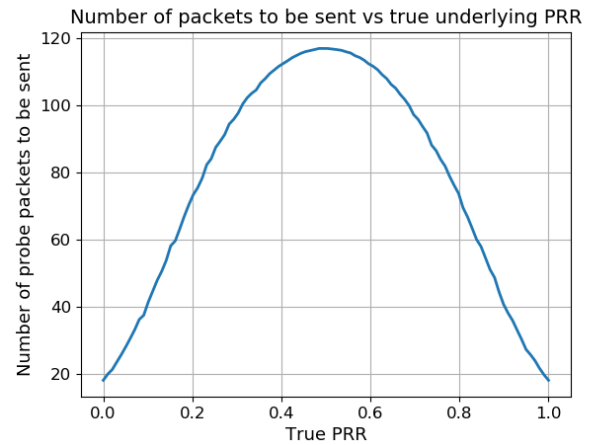


FIGURE 1. For a given confidence interval and p_{Δ} , the number of packets to be sent varies with the true underlying PRR.

that is $p_{\Delta} = p_{upper} - p_{lower} \leq 0.15$, in order to determine the number of packets to be sent. This value is a reasonable figure that ensures that, with 90% probability, the actual PRR will be within that 0.15 interval. Note that this implies that the actual number of packets to be sent to attain a certain confidence level ultimately depends on the underlying PRR of the channel. Fig. 1 shows the average required number of packets when the underlying PRR of the wireless channel varies between 0 and 1. There, one can see that when the PRR is around 0.5, six times more packets must be sent than when the PRR is 1 or 0.

The side benefit of determining the upper and lower bounds of the estimated PRR is that we can sweep all K configurations in such a way that the knowledge acquired from one configuration can be transferred to other configuration and, this way, further reduce the number of total transmissions. For instance, consider that for $k = (SF7, CR = \frac{4}{5}, 10dBm)$, it has been found that $p_{lower}^k = 0.4$ and $p_{upper}^k = 0.55$ (that is, with a confidence interval of 90%, the PRR lies within such an interval). When the transmission power is increased to $TXP = 14dBm$, the lower bound of the PRR for such a configuration must be larger than or equal to 0.4. This is so because of the monotonic increase of the PRR w.r.t. the SNR. Similarly, if the transmission power is decreased to $TXP = 6dBm$, the upper bound of the PRR must be smaller than or equal to 0.55. Furthermore, if a minimum acceptable PRR of 0.56 ($PRR_{min} = 0.56$) is desired for the communication, TXP values smaller than 10dBm will not be swept as PRR cannot do anything but to decrease or remain as is as the TXP decreases. With this *bounding* technique, the number of packets required to construct the PRR model is dramatically reduced—more information about this method can be found in the pseudo-code included in Section VIII–. Ultimately, this helps reducing the time and energy needed in such a process when compared to the classic approach of iteratively testing all K configurations (denoted as *brute-force* approach).

To evaluate the efficiency of our proposed algorithm, we have simulated 100 nodes running such a *bounding*

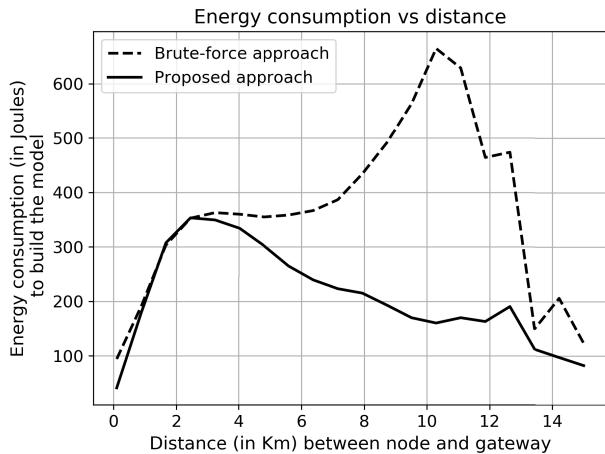


FIGURE 2. Energy consumed to construct M vs distance between nodes and gateway when the proposed (solid line) and brute-force (dashed line) techniques are employed –averaged for 100 nodes–. When using our approach, up to 73% less energy is consumed.

algorithm and the simple *brute-force* alternative (that is, testing all K configurations to obtain M). The distance between nodes and gateway has been varied between 0.1 to 15km to evaluate different realistic scenarios. The log-distance path-loss model proposed in [6] has been used to compute the perceived SNR of LoRa nodes. Finally, the power consumption required to construct the full PRR model has been computed according to Tables 1 and 2.

Fig. 2 shows the required energy to build the model M depending on the algorithm employed (i.e. the *bounding* or the *brute-force* algorithms). The main appreciated benefit of our proposal is the significant reduction in power consumption (of up to 73%) attained for larger distances. Furthermore, it is worth highlighting how constructing a full PRR model with the *bounding technique* consumes, in the worst case, 353 joules. This represents a 0.88% of the total capacity stored in a regular IoT battery (3000mAh @ 3.7V); thus making it a cost-effective way of increasing the precision of the derived C_{opt} (when it is compared to the alternative option of relying in environment-agnostic PRR models).

VI. KEEPING C_{opt} UPDATED

C_{opt} must react to changes in the channel and environment. This is accomplished by two different processes: (i) maintaining our knowledge of the channel/environment updated. That is, ensuring that PRR_i^k and λ_i reflect the reality of the propagation medium. (ii) Recomputing C_{opt} and updating the configuration of nodes when a sufficiently large change is detected.

Fortunately, our knowledge of PRR_i^k can be kept up-to-date inexpensively by exploiting the operational natural of LPWANs. By using the *counter* field present in the header of any sent packet, the gateway can easily determine the number of packets lost for any node i . If the observed PRR does not fall within the pre-determined PRR bounds (p_{lower} , p_{upper}) –for the configurations in use–, we can state that

the environment has changed –with a confidence interval $c \cdot 100\%$ –. Under such circumstances, the following process is executed for such a node i :

- 1) Following Section V, update our knowledge of the transmission configurations in use. That is, update $PRR_i^k \forall k \mid c_i^k \neq 0$ –this is described in detail in the pseudo-coded included in Section VIII–.
- 2) Recompute C_{opt} .
- 3) If the new vector c_i contains transmission configurations for which the PRR information is not updated, update them and go to step 2.
- 4) Once this process is finished, inform node i of its new transmission configuration c_i .

Similarly, for any change in the estimation of λ_i , we can compute the new $C_{opt-new}$ and, if the throughput under this new configuration, $\Gamma(C_{opt-new})$, is sufficiently larger than the throughput under the old configuration, $\Gamma(C_{opt-old})$, we proceed to update the configuration of nodes. This difference in throughputs (denoted as Γ_{Δ} hereinafter) can be arbitrarily set such as a balance between spending to much time updating and keeping nodes updated is found. For our experiments (presented in Section VII), we empirically found that a relative increase of 5% in Γ_{Δ} attains that trade-off.

VII. EXPERIMENTS

To evaluate the proposed solution we have implemented:

- The algorithms for constructing the PRR models for LoPy 4.0 LoRa devices as well as the necessary server-side code to command & control such devices (through gateways). All LoPy 4.0 LoRa nodes transmit to a single gateway device from The Things Network [26]. All transmissions are transferred to the Things Network backend and stored in a database for further processing RSSI and SNR values.
- The algorithms for estimating λ_i and computing C_{opt} .
- The algorithms for determining when to update C_{opt} and spreading the updated configuration to LoRa nodes.

All the pertaining code has been made available to the public domain in [27].

We have deployed 5 devices in an indoor environment (within the Engineering Department of Cambridge University) to test the performance of the proposed solution. Four of these five devices are assumed to be under our control and a fifth device is taking the role of an interfering third-party network.

Given the reduced number of LoRa devices available, we have forced nodes to always transmit in a single frequency channel (instead of randomly using one of the 8 available channels) and to adopt transmitting rates larger than those normally found in LPWAN: this effectively puts the network in a condition much more similar to that found in more crowded LoRa deployments.

A. OBTAINING THE PRR MODEL

The four devices under our control were set out to derive their respective PRR models employing the approach presented

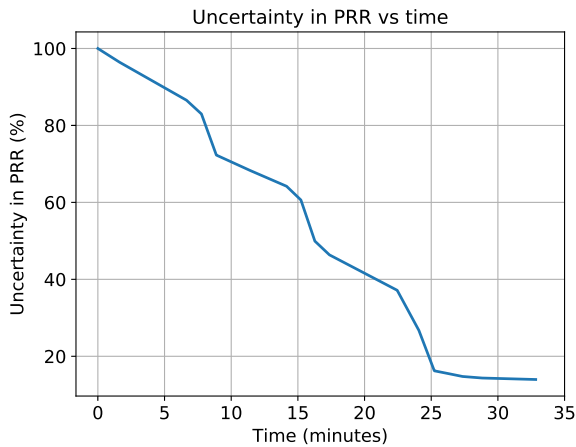


FIGURE 3. Evolution of the average uncertainty for the worst case scenario.

in Section V. In the worst case scenario, that is, for the node placed the farthest from the gateway, Fig. 3 illustrates how the average uncertainty in the PRR for all K configurations decreases as the exploration of the different parameters takes place. This average uncertainty is computed as follows: $\frac{100}{K} \sum_{k=1}^K p_{upper}^k - p_{lower}^k$.

When the uncertainty for all configurations is below 15% (the value chosen for p_{Δ}), and thus, the average uncertainty for the PRR reaches such a value, the process ends. Since the number of packets to be sent depends on the underlying PRR for each configuration, the process for obtaining the full PRR model (M) takes a variable amount of time for each node –depending on its radio propagation conditions–. Nevertheless, using the approach explained in Section V, the number of transmitted packets is not large and therefore, the process only takes around 20 to 30 minutes to complete –for our experiments the worst case scenario took 35 minutes–. To put this into perspective, one may consider that, by coupling the low-power design of LoRa networks with the use of truly optimal transmission configurations C_{opt} , LoRa nodes can last more than 4 years. Therefore, the acquisition of an accurate PRR model represents less than 0.0015% of total node’s lifetime. This demonstrates again that the derivation of the PRR model (M) is a worthy investment in terms of time and energy.

B. KEEPING C_{opt} UPDATED

Once the PRR model is acquired, C_{opt} can be computed and disseminated. However, and as indicated in previous sections, it is important to keep our knowledge about the environment updated. In this experiment we have tested an abrupt increase of the interference generated by an external third-party LoRa network.

We have done so by letting the fifth *interfering* node drastically increase the rate at which it generates packets (λ_5) –from a packet every two minutes, $\lambda = \frac{1}{120}$, to a packet every 3 seconds, $\lambda = \frac{1}{3}$. As this node is supposed to belong to an external network, we have no control over it and thus we have to continuously estimate λ_5 (denoted as $\overline{\lambda_5}$).

When interference created by this node becomes too high, a new C_{opt} is computed and disseminated. In a potentially realistic scenario an increase in interference levels may come from a neighboring network or from a potentially malicious attacker. In the latter, this could represent a DoS (Denial-of-Service) attack, which would aim at producing unsustainable packet collisions at a specific SF. By quickly detecting a decrease in the PRR and recomputing C_{opt} , nodes can minimize the impact of this kind of attacks.

This DoS resistance is achieved by making the new transmission configurations of LoRa devices avoid the more interfered SFs. This is true even when such SFs are the most convenient ones (either because they provide a larger PRR or the least time-on-air for packets). This is not necessarily the case for the classic Adaptive Data Rate (ADR) algorithm of LoRa networks [2], and we have demonstrated the superiority of our approach by running the same experiment when the ADR is activated and configured with its default values.

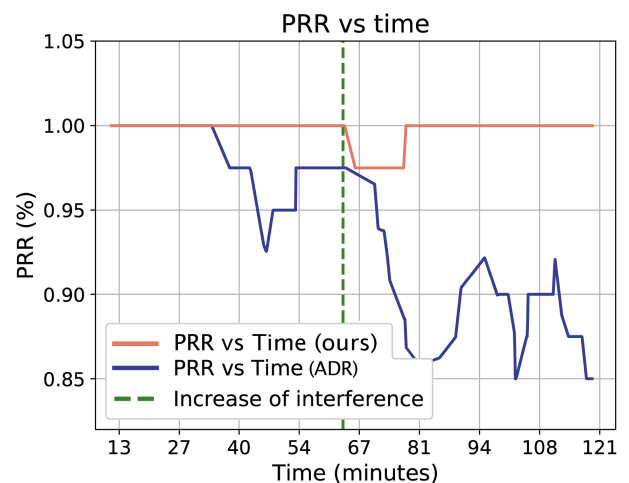


FIGURE 4. Average PRR of the network over time when the proposed adaptation mechanism is employed (red) and the ADR mechanism is used (blue). The green dashed line indicates the beginning of the interference.

Fig. 4 represents the attained PRR vs time (in seconds) for the two different approaches. The blue line depicts the evolution of the PRR when the aforementioned ADR algorithm whereas the red one uses our approach. It is worth noting that, for these experiments, no power restrictions were applied. That is $\xi_i^k = 1$ for all k and i .

The first noticeable fact to underline is that, prior to the increase of interference (marked with a green dashed vertical line), the PRR of both approaches is high. However, even in a relatively collision-free environment, ADR tends to underestimate the collision likelihood. In fact, during the whole experiment, ADR algorithm kept nodes transmitting at $k = \left(SF7, CR = \frac{4}{5}, 14dBm \right)$. This leads to a non-negligible packet collision probability which ultimately undermines the PRR (as shown in Fig. 4). For the first and second halves of the experiment (that is, before and after the increase in the interference, when λ_5 takes the values $\frac{1}{120}$ and $\frac{1}{3}$

TABLE 3. Distribution of transmission of configurations C_{opt} for the first half of the experiment.

Node	Configuration	Percentage
1	SF7, CR= $\frac{4}{5}$, 14dBm	18.3%
	SF8, CR= $\frac{4}{5}$, 14dBm	41.4%
	SF7, CR= $\frac{4}{7}$, 14dBm	40.3%
2	SF7, CR= $\frac{4}{5}$, 14dBm	24.4%
	SF8, CR= $\frac{4}{5}$, 14dBm	19.6%
	SF7, CR= $\frac{4}{7}$, 14dBm	44.2%
	SF10, CR= $\frac{4}{7}$, 14dBm	11.3%
3	SF7, CR= $\frac{4}{5}$, 14dBm	22.1%
	SF8, CR= $\frac{4}{5}$, 14dBm	40.4%
	SF7, CR= $\frac{4}{7}$, 14dBm	37.5%
4	SF9, CR= $\frac{4}{5}$, 14dBm	63.4%
	SF10, CR= $\frac{4}{5}$, 14dBm	20.4%
	SF10, CR= $\frac{4}{7}$, 14dBm	16.2%
5	SF7, CR= $\frac{4}{5}$, 14dBm	100%

TABLE 4. Distribution of transmission of configurations C_{opt} for the second half of the experiment.

Node	Configuration	Percentage
1	SF8, CR= $\frac{4}{5}$, 14dBm	62.0%
	SF8, CR= $\frac{4}{7}$, 14dBm	38.0%
2	SF8, CR= $\frac{4}{5}$, 14dBm	30.2%
	SF10, CR= $\frac{4}{5}$, 14dBm	20.3%
	SF11, CR= $\frac{4}{5}$, 14dBm	31.3%
	SF10, CR= $\frac{4}{7}$, 14dBm	18.0%
3	SF8, CR= $\frac{4}{5}$, 14dBm	48.3%
	SF9, CR= $\frac{4}{5}$, 14dBm	33.3%
	SF9, CR= $\frac{4}{7}$, 14dBm	18.3%
4	SF9, CR= $\frac{4}{5}$, 14dBm	61.3%
	SF10, CR= $\frac{4}{5}$, 14dBm	19.3%
	SF9, CR= $\frac{4}{7}$, 14dBm	19.4%
5	SF7, CR= $\frac{4}{5}$, 14dBm	100%

respectively), the distributions of transmission configurations (i.e. the computed C_{opt}) are indicated in Tables 3 and 4. Note that each configuration of C_{opt} indicates the percentage of packets generated under such combination of SF, CR, and TXP. For instance, node 1, in the first half of the experiment, will generate 18.3% of its packets using the following configuration: SF7, CR = $\frac{4}{5}$, and TXP=14dBm.

At some random time –marked with a green dashed vertical line– the fifth node increases its packet-generation rate; thus, increasing the interference to our deployed network.

This produces a reduction in the PRR of the network under study for both the ADR and our proposed mechanism. However, with our solution the new configuration is spread throughout the network when λ_5 changes enough to make $\Gamma(C_{opt_old})$ differ more than 5% from $\Gamma(C_{opt_new})$ (the predefined margin). This happens at around the second 4000th and helps restore the PRR to its original 100% value (see red line in Fig. 4). However, when using the ADR algorithm, there is no way to alleviate congestion in SF7 and thus, the PRR drops considerably (see blue line in Fig. 4).

With our proposed approach, these differences in PRR (of up to 15%) lead to an effective increase of Γ –the actual measure of network performance– of 15% (as the rest of parameters of Eq. 1 remain the same).

VIII. CONCLUSION

LPWAN-based IoT networks like LoRaWAN are becoming more and more crowded, making the optimal radio-resource allocation a pressing need. Thanks to the ability of LoRa nodes to individually adapt their transmission parameters (like Spreading Factor, Coding Rate, or Transmission Power), network operators can derive globally optimal network configurations by carefully crafting individual node transmission settings.

In order to derive this optimal network-wise configuration (C_{opt}), the performance of each individual transmission configuration must be determined. To achieve high accuracy levels, instead of assuming simplistic propagation models or unrealistic node distributions, we propose a mechanism by which LoRa nodes can characterize the propagation environment in which they are deployed. After deriving how the environment impacts on each transmission configuration, devices can obtain a truly optimal C_{opt} –which fully regards the specifics of each radio propagation environment.

To reduce the energy and time employed in deriving this environment characterization model M , we further propose a *bounding* technique that allows us to reduce in more than 73% the energy required –when compared to more straightforward approaches–. Also, we complement the model acquisition mechanism with a way of keeping this environment knowledge updated. We proposed a trigger-based method by which, when the environment behavior deviates from what it is expected, complementary information is gathered in an as-needed basis.

All the techniques proposed here are implemented and tested in a real LoRa deployment and the code and tools made publicly available. Results in real scenarios demonstrate that, not only are these environment models inexpensive to be constructed (requiring just 0.88% of the total node energy and 0.0015% of its expected lifetime), but thanks to our proposal, we can also significantly increase the network Packet Reception Rate by 15%, correspondingly improving the throughput metric by 15% –when compared to traditional methods like LoRaWAN ADR–.

APPENDIX

In this Section the pseudo-code of the different algorithms is listed. Note that the code is also fully released and made publicly available at [27].

Algorithm 1 Computation of PRR Model for Each Node i

```

1: Input: Set of SF values to test SFs,
2: set of CR values to test CRs,
3: set of TXP values to test TXPs,
4: confidence interval  $c$ ,
5: maximum bounds margin:  $p_{\Delta}$ ,
6: min acceptable PRR:  $PRR_{min}$ 
7: Output:  $PRR_{model}^k$  for  $k$  in SFs×CRs×TXPs
8:  $PRR_{model}^k = \text{NaN}$  for all  $k$  in SFs×CRs×TXPs
9:  $p_{lower}^k = 0$  for all  $k$  in SFs×CRs×TXPs
10:  $p_{upper}^k = 1$  for all  $k$  in SFs×CRs×TXPs
11:  $k_{next} \leftarrow \text{GetNextConfig}(p_{lower}, p_{upper}, PRR_{model})$ 
12:  $PRR_{min}, p_{\Delta}, SFs, CRs, TXPs$ 
13: if  $k_{next} \neq \emptyset$  then
14:   Order node sweep parameters  $k_{next}$ 
15:   for each received packet do
16:     determine  $m, n$  with the counter field of the
       packet
17:     compute  $p_{lower}^k, p_{upper}^k$  with Eq. 3
18:     if  $p_{upper}^k - p_{lower}^k \leq p_{\Delta}$  then
19:        $PRR_{model}^k \leftarrow \frac{m}{n}$ 
20:       BoundLowerBounds(SF, CR, TXP,  $p_{lower}$ ,
        $p_{upper}$ , SFs, CRs, and TXPs)
21:       BoundUpperBounds(SF, CR, TXP,  $p_{lower}$ ,
        $p_{upper}$ , SFs, CRs, and TXPs)
22:       go to 12

```

Algorithm 2 Determine the Next Configuration to Test for the Bounding Technique

```

1: procedure GetNextConfig
2:   Input:  $p_{lower}, p_{upper}, PRR_{model}$ 
3:   SFs, CRs, and TXPs
4:   maximum bounds margin  $p_{\Delta}$ ,
5:   min acceptable PRR:  $PRR_{min}$ 
6:   Output: a combination of SF, CR, and TXP to test
       next
7:    $max_{margin} = p_{\Delta}$ 
8:    $k_{next} = \emptyset$ 
9:   for SF=7 to SF=12 do
10:    for all CR in OrderOuterInner(CRs) do
11:      for all TXP in OrderOuterInner(TXPs) do
12:         $k \leftarrow (SF, CR, TXP)$ 
13:        if  $p_{upper}^k - p_{lower}^k \geq max_{margin}$  and
        $p_{upper}^k \geq PRR_{min}$  then
14:           $max_{margin} \leftarrow p_{upper}^k - p_{lower}^k$ 
15:           $k_{next} \leftarrow k$ 
16:   return  $k_{next}$ 

```

Algorithm 3 Bound Upper Bounds

```

1: procedure BoundUpperBounds
2:   Input: SF, CR, TXP,  $p_{lower}, p_{upper}$ , SFs, CRs, and
       TXPs
3:    $k \leftarrow (SF, CR, TXP)$ 
4:   for all CR' in CRs do
5:     for all TXP' in TXPs do
6:       if TXP'  $\leq$  TXP then
7:         if CR'  $\leq$  CR then
8:            $k' \leftarrow (SF, CR', TXP')$ 
9:            $p_{lower}^{k'} \leftarrow \min(p_{lower}^{k'}, p_{lower}^k)$ 
10:           $p_{upper}^{k'} \leftarrow p_{upper}^k$ 

```

Algorithm 4 Bound Lower Bounds

```

1: procedure BoundLowerBounds
2:   Input: SF, CR, TXP,  $p_{lower}, p_{upper}$ , SFs, CRs, and
       TXPs
3:    $k \leftarrow (SF, CR, TXP)$ 
4:   for all CR' in CRs do
5:     for all TXP' in TXPs do
6:       if TXP'  $\geq$  TXP then
7:         if CR'  $\geq$  CR then
8:            $k' \leftarrow (SF, CR', TXP')$ 
9:            $p_{lower}^{k'} \leftarrow p_{lower}^k$ 
10:           $p_{upper}^{k'} \leftarrow \max(p_{upper}^{k'}, p_{upper}^k)$ 

```

Algorithm 5 Reorder a Set in an Out-In Fashion (e.g. Transforms [1, 2, 3, 4, 5] to [1, 5, 2, 4, 3])

```

1: procedure OrderOuterInner
2:   Input: Set S
3:    $L \leftarrow \text{len}(S)$ 
4:    $M \leftarrow \lceil \frac{L}{2} \rceil$ 
5:    $new\_order \leftarrow \emptyset$ 
6:   for  $i = 1$  to  $M$  do
7:      $new\_order \leftarrow new\_order + S[i]$ 
8:      $new\_order \leftarrow new\_order + S[(L - i + 1)]$ 
9:   return  $L$  first elements of  $new\_order$ 

```

REFERENCES

- [1] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in LoRa networks," 2016, *arXiv:1611.00688*. [Online]. Available: <http://arxiv.org/abs/1611.00688>
- [2] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," in *Proc. NOMS-IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2018, pp. 1–9.
- [3] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Optimizing and updating LoRa communication parameters: A machine learning approach," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 884–895, Sep. 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8758936/>
- [4] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8090518/>
- [5] R. M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, and T. M. Chen, "Optimal policy derivation for transmission duty-cycle constrained LPWAN," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3114–3125, Aug. 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8354856/>

- [6] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pet-tissalo, "On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology," in *Proc. 14th Int. Conf. ITS Telecommun. (ITST)*, Dec. 2015, pp. 55–59.
- [7] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6. [Online]. Available: <https://lirias.kuleuven.be/handle/123456789/577142> and <http://ieeexplore.ieee.org/document/7996380/>
- [8] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris, "Improving LoRa network capacity using multiple spreading factor configurations," in *Proc. 25th Int. Conf. Telecommun. (ICT)*, Saint-Malo, France, Jun. 2018, pp. 3–8.
- [9] J.-T. Lim and Y. Han, "Spreading factor allocation for massive connectivity in LoRa systems," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 800–803, Apr. 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8268120/>
- [10] N. Consortium. (2014). *ns-3*. [Online]. Available: <https://www.nsnam.org/>
- [11] F. Chen, I. Dietrich, R. German, and F. Dressler, "An energy model for simulation studies of wireless sensor networks using OMNeT++," *PIK-Praxis der Informationsverarbeitung und Kommunikation*, vol. 32, no. 2, pp. 133–138, Jan. 2009. [Online]. Available: <https://www.degruyter.com/view/j/piko.2009.32.issue-2/piko.2009.0023/piko.2009.0023.xml>
- [12] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair adaptive data rate allocation and power control in LoRaWAN," 2018, *arXiv:1802.10338*. [Online]. Available: <http://arxiv.org/abs/1802.10338>
- [13] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/8115779/>
- [14] M. Luvisotto, F. Tramarin, L. Vangelista, and S. Vitturi, "On the use of LoRaWAN for indoor industrial IoT applications," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, 2018. [Online]. Available: <https://www.hindawi.com/journals/wcmc/2018/3982646/>
- [15] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, "LoRa throughput analysis with imperfect spreading factor orthogonality," 2018, *arXiv:1803.06534*. [Online]. Available: <http://arxiv.org/abs/1803.06534>
- [16] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Improving RSSI-based path-loss models accuracy for critical infrastructures: A smart grid substation case-study," *IEEE Trans Ind. Informat.*, vol. 14, no. 5, pp. 2230–2240, May 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/8114199/>
- [17] C. Caillouet, M. Heusse, and F. Rousseau, "Optimal SF allocation in LoRaWAN considering physical capture and imperfect orthogonality," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–8. [Online]. Available: <https://hal.inria.fr/hal-02267218>
- [18] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of LoRaWAN channel access," in *Proc. IEEE 18th Int. Symp. A World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2017, pp. 1–3. [Online]. Available: <http://ieeexplore.ieee.org/document/7974300/>
- [19] A. Azari and C. Cavdar, "Self-organized low-power IoT networks: A distributed learning approach," Jul. 2018, *arXiv:1807.09333*. [Online]. Available: <http://arxiv.org/abs/1807.09333>
- [20] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, p. 2364, Oct. 2017. [Online]. Available: <http://www.mdpi.com/1424-8220/17/10/2364>
- [21] MSON. (2018). *High Voltage Power Monitor-Monsoon Solutions*. [Online]. Available: <https://www.msoon.com/>
- [22] Pycom. (2016). *LoPy IoT Devices*. [Online]. Available: <https://pycom.io/hardware/lopy-specs/>
- [23] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of LoRaWAN channel access with capture effect," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5. [Online]. Available: <http://ieeexplore.ieee.org/document/8292748/>
- [24] L. Alliance. *A Technical Overview of LoRa and LoRaWAN*. Accessed: Feb. 8, 2020. [Online]. Available: https://www.tuv.com/media/corporate/products_1/electronic_components_and_lasers/TUeV_Rheinland_Overview_LoRa_and_LoRaWANtmp.pdf
- [25] N. D. Megill and M. Pavicic, "Estimating Bernoulli trial probability from a small sample," 2011, *arXiv:1105.1486*. [Online]. Available: <http://arxiv.org/abs/1105.1486>
- [26] G. Wienke and J. Stokking. (2017). *The Things Network*. [Online]. Available: <https://www.thingsnetwork.org>
- [27] R. M. Sandoval. (2019). *Code and Scripts*. [Online]. Available: <https://github.com/rubmarsan/loragateway-opt>



RUBEN M. SANDOVAL received the M.S. degree in telecommunication engineering and the Ph.D. degree from the Universidad Politécnica de Cartagena (UPCT), Cartagena, Spain, in 2014 and 2019, respectively. Since 2019, he has been holding the position of CTO at Bijectiva Technologies, a Startup focused on the applicability of machine learning techniques to image and video analysis. His research interests include machine learning, artificial vision, and the applicability of both paradigms to manage critical environments.



DAVID RODENAS-HERRAZ received the degree in telecommunications engineering from the Universidad Politécnica de Cartagena (UPCT), Spain, in 2010, and the Ph.D. degree (*cum laude*) in information and communication technologies from UPCT, in 2014. From 2014 to 2019, he was a Research Associate with the University of Cambridge, U.K. Since 2019, he has been a Wireless Software Engineer with Ocado Technology, U.K. His main research interests span low-power short and the long-range wireless IoT technologies, OFDMA-based technologies, embedded networked sensor systems, and wireless communications for networked robotics applications.



ANTONIO-JAVIER GARCIA-SANCHEZ received the M.S. degree in industrial engineering from the Universidad Politécnica de Cartagena (UPCT), Spain, in 2000, and the Ph.D. degree from the Information and Communication Technologies Department (DTIC), UPCT, in 2005. Since 2001, he has been with DTIC, UPCT. He is currently an Associate Professor with UPCT and also the Head of DTIC. He has coauthored more than 90 conference and journal articles, 40 of them indexed in the Journal Citation Report (JCR). He has been the main head in several regional/national/EU research projects. He has belonged to more than 30 conferences/workshops as a TPC staff and to eight as a member of the organizing committee. His main research interests are in the areas of wireless sensor networks (WSNs) and the Internet of Things, streaming services, performance evaluation of communication networks, smart data processing, and nanocommunications. He is a Reviewer of notable communication journals listed in the ISI-JCR, highlighting the IEEE INTERNET OF THINGS JOURNAL, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and *Computer Networks*.



JOAN GARCIA-HARO (Member, IEEE) received the M.S. and Ph.D. degrees in telecommunication engineering from the Universitat Politècnica de Catalunya, Spain, in 1989 and 1995, respectively. He was a Visiting Scholar with Queen's University, Kingston, Canada, from 1991 to 1992, and also with Cornell University, Ithaca, USA, from 2010 to 2011. He is currently a Professor with the Universidad Politécnica de Cartagena. He has authored or coauthored more than 80 journal articles mainly in the fields of switching, wireless networking, and performance evaluation. He also received an Honorable Mention for the IEEE Communications Society Best Tutorial paper Award, in 1995. He served as an Editor-in-Chief for the IEEE GLOBAL COMMUNICATIONS NEWSLETTER, included in the *IEEE Communications Magazine*, from April 2002 to December 2004. He has been a Technical Editor of the *IEEE Communications Magazine*, from March 2001 to December 2011.