Automating Reliable and Fault-Tolerant Design of LoRa-based IoT Networks

Xiaofan Yu*, Weihong Xu*, Ludmila Cherkasova† and Tajana Šimunić Rosing*
*Department of Computer Science and Engineering, University of California San Diego, La Jolla, USA
Email: {x1yu, wexu, tajana}@ucsd.edu
†Arm Research, San Jose, USA
Email: ludmila.cherkasova@arm.com

***Abstract*—Low-Power Wide-Area Networks (LPWAN) has recently been scaling rapidly, targeting at large-scale and low-power applications. LoRa and LoRaWAN have been adopted in many practical deployments. While the advantages of LoRa have been well-demonstrated, challenges in scalability and reliability impede LoRa networks from further expansion. Traditional deployment strategies for reliability and fault tolerance, which ensure that multiple networking paths are available, are not directly applicable because of LoRa's single-hop and Aloha medium-access design. In this paper, we study how to design LoRa networks in large regions so that their transmission reliability and fault tolerance against gateway failures and interference are met for LoRaWAN technology. We first introduce *m-gateway connectivity* to guarantee fault tolerance due to LoRa's unique properties. Next, we leverage state-of-the-art transmission reliability model based on estimated path loss from satellite maps. Combining the above two contributions, we formulate an Integer Nonlinear Program (INLP) that minimizes the number of gateways through strategic gateway placement and resource allocation. Constraints are imposed to achieve (i) fault tolerance, (ii) reliable transmission (i.e., satisfactory QoS), (iii) sufficiently long lifetime. Due to high complexity of INLP, we design a greedy heuristic, *RFT-LoRa,* to acquire a high-quality solution for larger size problems. Comprehensive evaluation is performed with ns-3 simulator using real-world datasets. The results demonstrate that RFT-LoRa enhances average packet delivery ratio by 10% - 54% over the existing heuristic under gateway failures and interference.**

***Index Terms*—IoT networks, LoRa, reliability, fault tolerance, gateway placement, resource allocation.**

I. Introduction

Recent years have witnessed explosive interest in Low-Power Wide-Area Networks (LPWAN) due to its remarkable ability in supporting large-scale and low-power Internet-of-Things (IoT) applications, e.g., smart cities [1], smart agriculture [2], and industrial IoT [3]. Ericsson mobility report anticipates 6.3 billion LPWAN IoT connections over the globe by 2026, which makes up 23% of the total connections [4]. LPWAN fills the technology gap between high bandwidth but limited range networks such as Wi-Fi/Bluetooth, or requiring high power as is the case in Cellular. LoRa (Long Range) is an open-source LPWAN technique that operates in sub-GHz ISM bands. Using Chirp Spread Spectrum (CSS) modulation, LoRa signals can be successfully decoded even if the received signal strength is below the noise floor. Compared to other LPWAN technologies, such as Sigfox [5] and NB-IoT [6], LoRa can be easily utilized by any individual due to unlicensed band usage, open-source infrastructure, and low installation cost. Meanwhile, an ecosystem of LoRa and LoRaWAN (i.e., an open protocol for LoRa) is scaling rapidly thanks to Semtech [7] and LoRa Alliance [8], which enables multi-city and nation-wide LoRa deployments such as The Things Network [9]. According to IoT Analytics, LoRaWAN is the most adopted LPWAN technology to date, representing more than 1/3 of all global deployments [10].

Although these deployments demonstrate LoRa's capabilities, there are still a lot of challenges preventing LoRa and LoRaWAN from a broader adoption, especially due to scalability and reliability issues [11]. Existing works in designing LoRa networks have studied assigning radio parameters to end devices to maximize Quality of Service (QoS), lifetime and energy efficiency [12]–[14]. Ousat *et al.* [15] were the first to jointly optimize gateway placement and parameters configuration in LoRa networks. Nevertheless, previous research overlooked the fault tolerance (or robustness) of LoRa networks upon failures. The sparse distribution of gateways and long communication links make LoRa networks susceptible to gateway failures and interference. In contrast to traditional wireless networks, LoRa uses a *single-hop broadcast* and *Aloha MAC protocol*. Therefore, traditional reliability-aware or fault-tolerant strategies through alternative path routing and scheduling [16], [17] are not directly applicable. The issues of reliability and fault tolerance in LoRa networks are key challenges for large-scale deployment of LoRa applications.

In this paper, we propose an end-to-end methodology for designing reliable and fault tolerant LoRa networks via strategic gateway placement and parameter configuration. The contributions of this paper are the following:

1) To the best of our knowledge, our work is the first one to offer an automated design of a fault-tolerant LoRa network that ensures its functionality in spite of gateway failures and interference under LoRaWAN, by introducing a new requirement of *m-gateway connectivity.*
2) In contrast to theoretical path loss models used in previous works [12]–[15], we leverage state-of-the-art transmission reliability models based on estimated path loss from remote sensing, which takes into account attenuating land cover effects [18].
3) We formulate an Integer Nonlinear Programming (INLP) minimizing the number of gateways given the device locations. The imposed constraints include: (i) fault tol-

erance, (ii) reliable transmission (i.e., satisfactory QoS), and (iii) long lifetime. We demonstrate the problem is NP-complete.
4) Due to the complexity in solving the INLP, we design a greedy heuristic named *RFT-LoRa*, that provides high-quality solutions in large-scale cases.
5) We validate the performance and desirable properties of LoRa networks in ns-3 simulator [19] with the state-of-the-art open-source LoRaWAN module [20]. The end device locations are configured using a real-world deployment from PurpleAir [21]. The results indicate that RFT-LoRa approximates well the optimal solution and enhances 10% - 54% on average packet delivery ratio (PDR) compared with existing heuristic under gateway failures and interference.

II. Related Work

A. Joint Gateway Placement and Resource Allocation

The problem of gateway (or base station) placement has been extensively studied in Cellular networks [22], ad-hoc networks [23], [24] and mesh networks [17], [25]. The majority of literature has applied Integer Linear Programming (ILP) or Mixed-Integer Linear Programming (MILP) to jointly optimize gateway placement with routing [26], link scheduling [27], or both [17]. Joint optimization is necessary for maximizing the overall performance which includes cost, coverage, QoS, lifetime, robustness, etc. [17], [22], [23] Due to the NP-hardness of the original formulation, multiple heuristics are proposed to find sub-optimal solutions in large-scale cases, e.g., clustering algorithms [28], genetic algorithms [29], population-based metaheuristics [24] and simulated annealing [30].

B. Fault-Tolerant Networks Design

In terms of fault tolerence, existing works have invented strategies such as *k-coverage* and *m-connectivity*. *K-coverage* requires each target to be covered by at least k sensors, so that any failure of less than k sensors will not impede successful sensing [31]. An entire field of robust sensor deployment is brought up for robustly detecting targets given each sensor has a certain probability to fail [32]. *M-connectivity*, on the other hand, ensures at least m distinct networking paths for each node [31]. In this way, backup paths support redundancy and robust transmission upon link failures. For traditional Wi-Fi and Cellular networks, previous works have studied relay node placement [16] and gateway placement [17] to establish a robust routing tree. However, the above contributions are not applicable to LoRa networks due to *single-hop broadcast* and *Aloha MAC protocol*.

C. LoRa Networks Design

The QoS and lifetime of LoRa networks depend on device and gateway placement, and transmission parameters of all end devices. Most of the existing works had focused on assigning radio parameters among devices to improve QoS and/or network lifetime globally. Reynders *et al.* [12] aimed minimizing the maximal collision probability by deriving the percentage of nodes using each Spreading Factor. This approach serves as a foundation for their later contribution proposing a MAC-layer protocol named RS-LoRa [33] for scheduling uplink traffic and setting the radio parameters. Gao *et al.* [13] formulated a non-convex optimization problem maximizing the minimal energy efficiency among all end devices through resource allocation. Sallum *et al.* [14] solved MILP to find the optimal Spreading Factor and channel assignment regarding QoS. Besides, several other algorithms have been developed to adaptively change Spreading Factor according to the current performance or energy level [34], [35].

Ousat *et al.* [15] were the first to formulate an optimization problem of gateway placement and parameter configuration in LoRa networks. They formulated Mixed-Integer Nonlinear Programming (MINLP) to optimize energy efficiency and proposed a greedy heuristic to place a given number of gateways. Their method attempts to fairly allocate the gateway and radio resources without any performance or reliability guarantees. Additionally, they used theoretical log-normal path loss models and deployed one gateway in the feasible range of a device, both making their network unreliable in practice. In contrast, in our work we incorporate state-of-the-art path loss models and a novel strategy named *m-gateway connectivity*. Our goal is to ensure reliable transmission and fault tolerance in LoRa networks via strategic gateway placement and end device configuration. Note, that the proposed network-level optimizations in this paper are complement to hardware- and protocol-level techniques, e.g., Choir [36] and Charm [37].

III. Designing Reliable and Fault-Tolerant LoRa Networks

In this section, we first present key parameters of LoRa (Section III-A), then outline our major contributions. A novel strategy to design fault-tolerant LoRa networks is proposed in Section III-B. We also explain how to get a practical estimate of path loss based on remote sensing (Section III-C).

A. Key Parameters of LoRa

Key radio parameters of LoRa determine the transmission range, QoS and energy consumption of LoRa end devices in a complex manner. **Spreading Factor (SF)** refers to the number of bits that can fit into one chirp [38]. Intuitively, SF determines the slope of the chirp. Higher SF enhances the robustness to noise, lowers the sensitivity threshold for successful decoding, thus increases the effective transmission distance [13]. Meanwhile, switching to the next higher SF will double the transmission time and energy [38]. We summarize various characteristics of SF in Table I. **Channels** refer to the frequency bands used for signal transmission. The USA LoRaWAN frequency spectrum has 64 available uplink channels (125kHz for each) starting at 902.3 MHz [39]. **Transmission Power (Tx Power)** is the actual amount of radiated power that a transmitter produces, which affects the chance of surviving attenuation caused by the environment [38]. Other parameters such as code rate [38], are set per application, so we do not optimize for them here.

TABLE I: SNR threshold, sensitivity threshold, transmission time and data rate using different SF [13], [15]. The last two are obtained under 50 bytes payload.

SF	7	8	9	10
Sensitivity Threshold (dBm)	-123	-126	-129	-132
Transmission Time (ms)	98	175	329	616
Data Rate (kbps)	4.1	2.3	1.2	0.6

B. M-Gateway Connectivity

Traditional strategies to enhance network fault tolerance are based on introducing redundant nodes and paths in the network. With a traditional *m-connectivity*, a node will have multiple paths to one gateway through different relay nodes. However, such strategy does not apply to LoRa networks, where a single-hop broadcast is only permitted. Moreover, as a result of Aloha, carelessly placing redundant nodes might cause more collisions and degrade the QoS of LoRa networks.

In this paper, we introduce a new property of *m-gateway connectivity* as a guideline for placing devices and gateways in LoRa network for ensuring fault tolerance. If a LoRa end device meets *m-gateway connectivity*, it is capable of reaching m gateways, and the transmission reliability to the primary gateway exceeds a specified threshold. *M-gateway connectivity* leverages the latest LoRaWAN specification that LoRa end devices can switch to higher SFs and transmit farther distance when acknowledgements are lost [39]. Therefore, m gateways could be located within feasible ranges under various SFs. In the following lines, we use a simple example to demonstrate the benefits of using *m-gateway connectivity*.

Let us consider the deployment of two end devices e_1, e_2 and two gateways g_1, g_2 as shown in Figure 1. In a normal case, e_1 (or e_2) is capable of reaching g_1 (or g_2) using SF7. If g_1 fails or there is a strong interference around g_1, then e_1 will switch to SF8 automatically and will connect to g_2, which serves as a backup gateway for e_1. In this way, *2-gateway connectivity* is achieved for e_1 as g_1 and g_2 are placed within a feasible range of SF7 and SF8. Such design has multiple benefits:

- *Fault tolerance is improved* via backup gateways for e_1 during gateway failures and interference.
- *Unnecessary collisions are avoided.* If both g_1 and g_2 are located in SF7's range, packets of e_1 and e_2 may collide.
- *Savings on the total number of installed gateways and budget* by having multi-use of gateways. In Figure 1, if g_1 can also be reached by e_2 with a higher SF, we would achieve *2-gateway connectivity* for two end devices with only 2 deployed gateways (instead of 3 or 4 gateways required otherwise).

The mechanism of automatically increasing SF upon acknowledgement lost is enabled by LoRaWAN specification V1.0.4 [39]. Hence, *m-gateway connectivity* does not require changes on protocol or firmware, but only a careful design of gateway placement and parameter configuration.

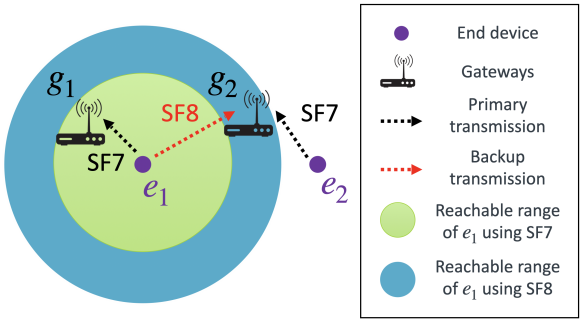

Fig. 1: An example of *2-gateway connectivity* setup on end device e_1.

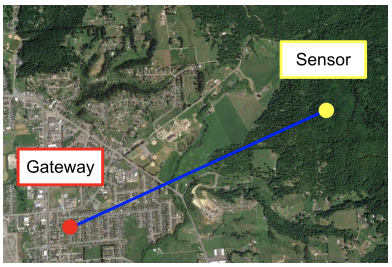

Fig. 2: An example of estimating path loss using remote sensing as in [18].

C. Estimating Path Loss with Remote Sensing

Path loss quantifies signal attenuation during propagation thus is key in evaluating transmission reliability. While previous works adopted Friis model [13] or log-normal model [15], these models are impractical since they fail to account for the irregular attenuation pattern brought by surrounding environment. Recent work by Lin *et al.* [18] proposed an accurate estimation of LoRa path loss using remote sensing. They used the log-normal model as a base model to estimate the large-scale path loss $PL(d)$ in dB:

$$PL(d) = \overline{PL(d_0)} + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + N_\sigma \qquad (1)$$

where d is the distance between LoRa node and gateway, $\overline{PL(d_0)}$ is the mean path loss at reference distance d_0, γ is the path loss exponent of the corresponding land type, and N_σ is a zero-mean Gaussian random variable with standard deviation of σ. Field experiments are performed to fit $\overline{PL(d_0)}, \gamma, \sigma$ of various land-cover types, including build-ups, forest, field, water and rangeland. The average fitting error between prediction and measurement of each category is less than 3 dB [18]. They further proposed an algorithm to estimate the path loss between any two locations after segmenting the land types from satellite map and aggregating the attenuation through each land type. For example, the propagation loss using LoRa from a sensor to a gateway shown in Figure 2 can be accurately predicted by summing up the attenuation through forest, field and build-ups.

In this paper, we leverage the remote sensing-based method in Lin *et al.* [18] to initialize the path loss matrix PL. PL_{ij}

is the path loss from end device $i \in N$ to candidate gateway $j \in G$, which is modeled as a Gaussian random variable:

$$PL_{ij} \sim \mathcal{N}\left(\overline{PL_{ij}}, \sigma^2\right), \tag{2}$$

where $\overline{PL_{ij}}$ is the expectation of path loss obtained from the algorithm with fitted land-cover parameters in [18].

IV. Problem Formulation

Combining *m-gateway connectivity* and the practical path loss estimation, we are able to formulate an optimization problem for gateway placement and end device configuration given device locations.

A. System Model

Let $\mathcal{N}$ denote a set of predetermined locations for the end devices. $\mathcal{G}$ is a candidate grid space for gateway placement. We use binary variables g_j to represent whether a gateway is placed at each grid point in $\mathcal{G}$:

$$g_j = \begin{cases} 1 & \text{if a gateway is placed at } j \\ 0 & \text{otherwise.} \end{cases} \qquad \forall j \in \mathcal{G} \tag{3}$$

For LoRa end devices, we assume a uniform sampling period T and each device initiates one uplink transmission per sample. There are three device classes in LoRaWAN [39]. In this work, we focus on *Class A* whereby each uplink transmission is followed by two downlink receive windows [40]. We require *confirmed* traffic (i.e., acknowledgement from gateway is required for each packet) because acknowledgements are the key in ensuring fault tolerance, which is explained in detail in Section III-B.

In our formulation, we associate the selection of SF, channel, Tx Power at each end device with an integer variable:

$$\begin{aligned} sf_i &\in \mathcal{SF}, \quad \forall i \in \mathcal{N} \\ ch_i &\in \mathcal{CH}, \quad \forall i \in \mathcal{N} \\ tp_i &\in \mathcal{TP}, \quad \forall i \in \mathcal{N} \end{aligned} \tag{4}$$

$\mathcal{SF}, \mathcal{CH}, \mathcal{TP}$ are candidate sets of available SF, channel and Tx Power respectively.

B. Optimal Problem Formulation

We formulate a problem as follows: given a set of end devices' locations, how to place gateway and allocate transmission parameters at each end device to minimize the total number of gateways, while ensuring: (i) fault tolerance against possible gateway failures and interference, (ii) reliable transmission (i.e., satisfactory QoS), (iii) sufficiently long lifetime. We summarize the important notations used in the paper in Table II.

Our complete Integer Nonlinear Programming is shown below as **(P)**. It minimizes the number of gateways as shown in Eq. (5a), which is equivalent to the minimum installation cost. Eq. (5b) handles the *m-gateway connectivity* requirement. Eq. (5c) states the QoS constraint. Eq. (5d) specifies the lifetime constraint in order to guarantee the longevity of each end device. We next discuss how we arrive at these expressions.

TABLE II: List of important notations in problem formulation.

Symbol	Meaning
$\mathcal{N}$	Set of given locations for LoRa end devices
$\mathcal{G}$	Set of grid locations to place LoRa gateways
PL	Path loss matrix for $\mathcal{N}$ and $\mathcal{G}$
σ	Average standard deviation of path loss from [37]
g_j	Binary variable of gateway placement at $j \in \mathcal{G}$
sf_i	Integer variable of SF at $i \in \mathcal{N}$
ch_i	Integer variable of channel at $i \in \mathcal{N}$
tp_i	Integer variable of Tx Power at $i \in \mathcal{N}$
c_{ij}^{ks}	Reacheability from device i to gateway j with SF k and Tx Power s
M	Redundant connectivity requirements
PDR_i	Packet delivery ratio of end device $i \in \mathcal{N}$
PDR_{th}	Predetermined PDR threshold at each end device
P_{ij}	Success deliver probability between $i \in \mathcal{N}$ and $j \in \mathcal{G}$
RSS_{ij}	Received signal strength at $j \in \mathcal{G}$ from $i \in \mathcal{N}$
SS_k	Sensitivity threshold under SF k (Table I)
N_{ij}	Number of devices reaching j and sharing the same resource with i
h_{ij}	Uncorrupted probability at i with other devices
T_k	Transmission time under SF k (Table I)
T	Uniform sampling interval at each LoRa end device
Pow_i	Average power consumption of end device $i \in \mathcal{N}$
E_{batt}	Capacity of batteries attached to end devices
L_i	Lifetime of end device $i \in \mathcal{N}$
L_{th}	Predetermined lifetime threshold at each end device

$$\textbf{(P)} \quad \min_{g,sf,ch,tp} \quad \sum_{j \in \mathcal{G}} g_j \tag{5a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{G}} g_j c_{ij}^{KS} \geq M, \quad \forall i \in \mathcal{N} \tag{5b}$$

$$PDR_i\left(g, sf, ch, tp\right) \geq PDR_{th}, \quad \forall i \in \mathcal{N} \tag{5c}$$

$$L_i\left(sf_i, tp_i\right) \geq L_{th}, \quad \forall i \in \mathcal{N} \tag{5d}$$

M-Gateway Connectivity **Constraint:** The core idea of *m-gateway connectivity* is to provide the additional backup gateways in case of gateway failures and interference. We define a notation c_{ij}^{ks} as the feasibility of transmitting from device i to gateway j with SF k and Tx Power s.

$$c_{ij}^{ks} = \begin{cases} 1 & \text{if } P\left\{RSS(s, PL_{ij}) \geq SS_k\right\} \geq PDR_{th} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

RSS computes the received signal strength at the gateway from Tx Power and path loss, which is given by

$$RSS_{ij(\text{dBm})} = tp_{i(\text{dBm})} + G^{tx}{}_{(\text{dB})} + G^{rx}{}_{(\text{dB})} - PL_{ij(\text{dB})} \tag{7}$$

Here G^{tx} and G^{rx} are the transmitting and receiving antenna gains [41]. Without the loss of generality, we assume a worst-case scenario of $G^{tx} = G^{rx} = 0$ dB. Technically, a positive c_{ij}^{ks} as a parameter indicates that transmissions under such configuration can have a successful ratio higher than PDR_{th} if considering the sensitivity threshold SS_k. c_{ij}^{KS} with K, S as the maximal SF and Tx Power option respectively, evaluates whether device i can reach gateway j using the best possible parameters. In other words, c_{ij}^{KS} denotes the feasibility of gateway j being a backup gateway for device i. We compute the total backup connectivity by summing up c_{ij}^{KS} at all deployed gateways and require it to be at least M, which is how we arrive at the linear constraint in Eq. (5b).

QoS Constraint ensures the minimal packet delivery ratio (PDR) is met at each end device. High PDR indicates that the data collected by one end device can be reliably delivered to the network server. Given that the backhaul between gateways and network servers is usually stable, transmission from an end device is considered to be successful if the packet is received by at least one gateway [13]. Hence the PDR at end device $i \in \mathcal{N}$ can be computed as follows:

$$PDR_i = 1 - \prod_{j \in \mathcal{G}} (1 - g_j P_{ij}) \tag{8}$$

where P_{ij} stands for the probability that a successful transmission is established between device i and gateway j. Following existing models [13], [15], a successful transmission has to satisfy two conditions: (i) the received signal strength is higher than the sensitivity threshold, (ii) the signal is not corrupted due to collisions. Specifically, if a gateway is deployed at j, P_{ij} is estimated as follows:

$$P_{ij} = \Phi_{ij} \cdot h_{ij} \tag{9}$$

The first term Φ_{ij} refers to the probability that the received signal strength RSS_{ij} is above the sensitivity threshold. Given that path loss PL_{ij} follows Gaussian distribution (Eq. (2)), Φ_{ij} is explicitly written as CDF of a Gaussian distribution:

$$\begin{aligned} \Phi_{ij} &= P\left\{RSS_{ij} \geq SS_{sf_i}\right\} \\ &= P\left\{\mathcal{N}\left(\overline{PL_{ij}}, \sigma^2\right) \leq tp_i - SS_{sf_i}\right\} \\ &= \Phi(tp_i - SS_{sf_i}; \overline{PL_{ij}}, \sigma^2) \end{aligned} \tag{10}$$

where SS_{sf_i} is the sensitivity threshold depending on SF as shown in Table I.

The second term h_{ij} of Eq. (9) represents the probability that the signal is not corrupted in collisions. Here we focus on collisions among the same SF as they have higher SINR threshold compared to inter-SF collisions (we refer the readers to [42] for detailed discussions). We leverage the collision probability model in [15]. We define N_{ij} as the number of end devices using the same SF and channel resources as i and can reach gateway j, which also drives h_{ij} together with the transmission time T_{sf_i} as a function of SF. Here $[cond]$ is 1 when the inner condition $cond$ is met, otherwise 0.

$$N_{ij} = \sum_{i' \in \mathcal{N}} [sf_{i'} = sf_i] \cdot [ch_{i'} = ch_i] \cdot c_{i'j}^{sf_i tp_i} \tag{11a}$$

$$h_{ij} = \exp\left(-2T_{sf_i} N_{ij} / T\right) \tag{11b}$$

Combining Eq. (8), (9), (10), (11), we can calculate PDR_i given a complete gateway placement and device configuration. We require PDR at each end device i to exceed a predetermined threshold PDR_{th}, which transforms to Eq. (5c).

Lifetime Constraint: We adopt the model in [38] where power measurements are performed on an Arduino Uno-based platform [43] with Semtech's SX1276 chip [44]. The power model distinguishes the transmission, reception and sleep states, estimating the average power as follows:

$$Pow_i = \frac{1}{T}\left[E_{TX}(sf_i, tp_i) + E_{RX} + E_{sleep}(sf_i)\right]. \tag{12}$$

Suppose T is the uniform sampling interval. E_{TX} accounts for the transmission energy relying on the selections of SF and Tx Power:

$$E_{TX}(sf_i, tp_i) = \frac{T_{sf_i}}{PDR_i}\left[P_{MCU} + P_{TX}(tp_i)\right]. \tag{13}$$

P_{MCU} represents the power supply for an active MCU, while the power draw of LoRa chip during transmission is a function of the Tx Power, i.e., $P_{TX}(tp_i)$. For robustness, we require acknowledgements from gateways so the transmission time T_{sf_i} is divided by PDR_i to account for re-transmissions.

E_{RX} is a constant denoting the average energy in receiving and decoding acknowledgements. Note, that E_{RX} in each communication round varies according to SF selection, receive window, etc. Since the varying scale of E_{RX} is relatively small compared to E_{TX}, we leverage the estimated average reception energy value from [38]. The last term $E_{sleep} = (T - \frac{T_{sf_i}}{PDR_i})P_{sleep}$ estimates energy dissipation during sleep mode. P_{sleep} is the constant sleeping power of the whole system including MCU and LoRa chip. All parameters are obtained from [38].

Given the battery capacity E_{batt} and the predetermined lifetime lowerbound L_{th}, the constraint on lifetime L_i is expressed as follows and is identical to Eq. (5d):

$$L_i = E_{batt}/Pow_i \geq L_{th}. \tag{14}$$

C. Complexity Analysis

The formulation **(P)** contains $|\mathcal{G}| + 3|\mathcal{N}|$ integer variables, $2|\mathcal{N}|$ nonlinear non-convex constraints (Eq. (5c), (5d)) and $|\mathcal{N}|$ linear constraints (Eq. (5b)). Although optimal solvers exist for INLP, the proposed problem cannot be solved in polynomial time because the proposed problem **(P)** is NP-complete.

First, **(P)** belongs to the class of NP. For any given assignment, we can verify whether it satisfies Eq. (5c), (5d) and (5b) in polynomial time. Then, we consider a relaxed version of **(P)** by setting $PDR_{th} = 0$, $L_{th} = 0$. Namely, we relax the nonlinear transmission reliability and lifetime constraints. The relaxed problem is equivalent to the minimum set cover problem whose goal is to find a cover for a given set of targets $\mathcal{N}$ with the minimum number of grid points from $\mathcal{G}$. The coverage provided by the grid is irregular in our case due to different land-cover types. Hence **(P)** is reducible to the well-known NP-complete problem of minimum set cover problem [45]. It is also clear that arbitrary instances of set covering can be encoded as an instance of **(P)**, and thus **(P)** is NP-complete.

V. Approximation Heuristic: RFT-LoRa

Since optimal solvers do not scale for large-size problems, we next propose a greedy heuristic, Reliable and Fault-Tolerant LoRa, namely *RFT-LoRa*, as outlined in Algorithm 1. RFT-LoRa greedily selects one location to place an additional gateway per each iteration, terminating once *m-gateway connectivity* is attained at every end device (line 4). To figure out which gateway candidate location is the best, RFT-LoRa

attempts to place a gateway at every unoccupied location (line 7), after which it greedily assigns SF, channel, and Tx Power to all end devices based on the current deployment (line 11). The benefit B of each assignment is evaluated based on the current values of (i) PDR, (ii) lifetime, and (iii) gateway connectivities:

$$B = \sum_{i \in \mathcal{N}} \omega_1 PDR_i + \omega_2 L_i \ + \min\left(Conn_i - M, 0\right). \quad (15)$$

The first two terms stress PDR and lifetime at each end device, i.e. PDR_i and L_i. The third term penalizes the gap between current gateway connectivities $Conn_i$ and the required level M. Note, that by upper bounding this term with 0, RFT-LoRa disregards extra gateway connectivities. Weight parameters ω_1, ω_2 enable a trade-off between PDR, lifetime and gateway connectivities. We set ω_1, ω_2 to small values so that the increased gateway connectivity plays the major role in greedy selection as long as transmission reliability and lifetime constraints are satisfied (line 22).

The key of the heuristic is the greedy device configuration procedure. Prior to the configuration, the end devices are sorted according to their path loss to the new gateway (line 8). End devices experiencing less path loss are given higher priority. The greedy configuration procedure starts with the minimum SF and channel, and the maximum Tx Power (line 23). The process ends once it finds a successful configuration that meets both PDR and lifetime constraints (line 24 - 27). Such mechanism ensures that we tend to allocate the best transmission resources to closer nodes.

Complexity Analysis: RFT-LoRa takes $O(M\left|\mathcal{N}\right|^3\left|\mathcal{G}\right|^2)$ to generate a result. Satisfying *m-gateway connectivity* at each end device takes $O(M\left|\mathcal{N}\right|)$ iterations (line 4). Traversing through each potential gateway location costs $O(|\mathcal{G}|)$ (line 6). RFT-LoRa attempts to configure all end devices located in the reachable range of the current gateway, which costs $O(|\mathcal{N}|)$. The `GreedyDevConfig` function steps through all resource combinations at each device (line 21), taking at most $|\mathcal{SF}| \cdot |\mathcal{CH}| \cdot |\mathcal{TX}|$ iterations, which is a constant under fixed $\mathcal{SF}, \mathcal{CH}, \mathcal{TX}$ sets. Computing PDR_i and L_i requires global evaluation that costs $O(|\mathcal{N}|\left|\mathcal{G}\right|)$. Combining the above components, we estimate the time complexity of RFT-LoRa at $O(M\left|\mathcal{N}\right|^3\left|\mathcal{G}\right|^2)$. Although RFT-LoRa is a polynomial-time algorithm, and therefore runs significantly faster than the optimal solver, its execution time increases dramatically for larger deployments. In the future, we plan to look into accelerating techniques for RFT-LoRa.

Algorithm 1: Greedy Heuristic for LoRa Deployment

Input: $\mathcal{N}, \mathcal{G}, PL, M$
Output: g, sf, ch, tp

1 $g_j \leftarrow 0,\ \forall j \in \mathcal{G}$
2 $sf_i \leftarrow -1, ch_i \leftarrow -1, tp_i \leftarrow -1,\ \forall i \in \mathcal{N}$
3 $Conn_i \leftarrow 0,\ \forall i \in \mathcal{N}$
4 **while** $\exists i \in \mathcal{N}, Conn_i < M$ **do**
5 $B_{best} \leftarrow -\infty$
6 **foreach** $j \in \mathcal{G}, g_j = 0$ **do**
7 $g_j \leftarrow 1$
8 $idx_{sorted} \leftarrow argsort(\{PL_{ij} \mid i \in \mathcal{N}\})$
9 **for** $i \leftarrow 1$ *to* $|\mathcal{N}|$ **do**
10 $i' \leftarrow idx_{sorted}[i]$
11 `GreedyDevConfig`$(g, sf, ch, tp, i', Conn_{i'})$
12 Update PDR and L globally
13 Update B_{new} according to Eq. (15)
14 **if** $B_{new} > B_{best}$ **then**
15 $B_{best} \leftarrow B_{new}$, $next \leftarrow j$
16 $g_j \leftarrow 0$
17 **if** $B_{best} = -\infty$ **then**
18 **break**
19 $g_{next} \leftarrow 1$
20 **return** g, sf, ch, tp
21
22 **Def** `GreedyDevConfig`$(g, sf, ch, tp, i, Conn_i)$**:**
23 **for** $sf_i \in \mathcal{SF}, ch_i \in \mathcal{CH}, tp_i \in \mathcal{TP}$ **do**
24 **if** $PDR_i(g, sf, ch, tp) \geq PDR_{th}$ *and* $L_i(sf_i, tp_i) \geq L_{th}$ **then**
25 $Conn_i \leftarrow Conn_i + 1$
26 Update sf_i, ch_i, tp_i
27 **break**
28 **return** $sf, ch, tp, PDR_i, L_i, Conn_i$

VI. Evaluation

A. Simulation Setup

We evaluate the performance of the following methods:

- **OPT**: The optimal solution to the proposed problem.
- **OPT$_\mathbf{r}$**: The optimal solution to the relaxed problem, where we convert the integer variables to continuous values.
- **EE-LoRa**: The energy efficiency-driven heuristic proposed in Ousat *et al.* [15]. EE-LoRa requires a user-specified gateway number as input which is set to the same as the output of our heuristic.
- **RFT-LoRa**: Our proposed heuristic.

We implement RFT-LoRa and EE-LoRa with Python 3.7. The optimal problem is modeled in MATLAB R2020b. The original INLP is solved by BONMIN [46] which uses the Branch-and-Bound algorithm, while the relaxed NLP is solved by SNOPT [47] employing a sparse sequential quadratic programming (SQP) algorithm. Finally, we evaluate our design in ns-3 simulator [19] augmented with the state-of-the-art open-source LoRaWAN module [20]. We use the packet tracker in ns-3 to evaluate the packet delivery ratio (PDR) at each end device. The same lifetime model as in our formulation is implemented and used to predict the lifetime. To assess the fault tolerance of our design, we modify the PHY level of the gateway modules in ns-3 to randomly switch down

TABLE III: Parameter settings in evaluation.

Param.	Value	Param.	Value	Param.	Value
T	20 min	E_{batt}	3 Ah	L_{th}	2 years
PDR_{th}	0.8	E_{RX}	0.005 J	P_{MCU}	23.48 mW
P_{sleep}	100 μW	σ^2	100.0724	ω_1, ω_2	10^{-4}

0/1/2 gateway(s) and add interference at gateways. We set $\mathcal{SF} = \{7, 8, 9, 10\}$ to match the available SF in USA [39]. For channels, we assume 8 uplink channels are available for our application, i.e., $\mathcal{CH} = \{0, ..., 7\}$, leaving the rest channel resources for other users. Tx Power is selected from typical settings of Semtech's SX1276 chip [44], i.e., $\mathcal{TP} = \{2, 5, 7, 11, 14, 20\}$ (dBm). Other important parameter settings are reported in Table III. The source code is available online[1]. Experiments are done on a desktop with Intel Core i7-8700 CPU at 3.2 GHz and 16 GB RAM.

We conduct comprehensive experiments *on three scenarios*:

- **Comparison of optimal and heuristic algorithms on small, 400km^2, and medium, 900km^2, areas:** We compare optimal, OPT, and relaxed optimal problems, OPT$_\mathbf{r}$ under various settings on a smaller field with 20 km×20 km area. Through this we show that their solutions match well. We then evaluate OPT$_\mathbf{r}$, RFT-LoRa and EE-LoRa on a medium size field of 30 km×30 km, with 25 candidate gateways and 30 randomly distributed end devices. Log-normal path loss model is used for these experiments.
- **RFT-LoRa vs. EE-LoRa with PurpleAir [21] dataset:** We test RFT-LoRa and EE-LoRa on a real-world deployment from PurpleAir [21] with 264 air-quality sensors spread over a 100 km×60 km field in Southern California, USA. There are 216 candidate gateway grid points equally distribute over the field, with 6 km distance between adjacent locations. The path loss matrix is computed based on remote sensing and land-cover segmentation. The fault tolerance of both heuristics upon gateway failures and interference are evaluated in ns-3.
- **Scalability study on 2,500km^2 area:** We provide data on performance scalability of RFT-LoRa and EE-LoRa over a large region covering 50 km×50 km area, containing 64 candidate gateways and randomly distributed end devices. The number of devices are selected from $\{100, 200, 500, 1K, 2K, 5K\}$. The path loss matrix is computed using the log-normal propagation model.

B. Results

1) Optimality comparison on small and medium problems: The output of OPT and OPT$_\mathbf{r}$ optimization for a smaller area of 400km^2 is listed in Table IV. The first column shows parameter settings for each optimization case. For example, $(16, 5, 1)$ indicates 16 candidate gateways, 5 end devices and *1-gateway connectivity*. After relaxing to continuous variables, the resulting gateway number of OPT$_\mathbf{r}$ is not an integer value, so we use it as a lower bound. OPT$_\mathbf{r}$ is 212x - 1142x faster than OPT while still providing accurate results, which can be attributed to SNOPT's effectiveness in large-scale nonlinear optimization. Unfortunately, OPT does not scale to larger size problems due to its long runtime.

[1]https://github.com/Orienfish/robust-lora

TABLE IV: Comparison of OPT and OPT$_\mathbf{r}$ on 400km^2

Setting	OPT		OPT$_\mathbf{r}$	
$(\lvert\mathcal{G}\rvert, \lvert\mathcal{N}\rvert, M)$	Gateway #	Time (Sec)	Gateway #	Time (Sec)
$(16, 5, 1)$	2	1910	2.0	4.1
$(16, 5, 2)$	4	2314	4.0	2.5
$(16, 5, 3)$	8	2444	8.0	3.0
$(16, 7, 1)$	3	5334	3.0	25.1
$(9, 5, 1)$	2	8910	1.35	7.8

TABLE V: Comparison results on 900km^2

Method	Gateway #	Time (Sec)	PDR (Avg/Max/Min)	Lifetime (Yrs) (Avg/Max/Min)
OPT$_\mathbf{r}$	**2.92**	2953	**0.90/1.0/0.76**	2.2/2.4/1.9
EE-LoRa	3	**0.3**	0.68/0.84/0.55	2.2/2.6/1.6
RFT-LoRa	3	1.8	0.87/1.0/0.73	2.3/2.6/1.8

In a medium size area of 900km^2, the average number of gateways to deploy and the execution time of OPT$_\mathbf{r}$, RFT-LoRa and EE-LoRa over the 5 trials are summarized in Table V. The right two columns in Table IV show the PDR and the lifetime distribution among all end devices when applying the generated deployments under one random initialization in the ns-3 simulator. RFT-LoRa ends up with the closest integer to the lowerbound given by OPT$_\mathbf{r}$ as gateway number. With similar PDR and lifetime, RFT-LoRa is 1640x faster than OPT$_\mathbf{r}$. The baseline EE-LoRa takes less than a second to finish in this example, but results in the worst PDR as EE-LoRa only proportionally distributes the available resources without performance guarantees. In fact, RFT-LoRa trades off the runtime for a quality of the solution, as we show below. All methods attain similar lifetime patterns that are close to the 2-year threshold. Clearlyy, from these results it can be seen that RFT-LoRa provides a good approximation of the relaxed optimal bound.

2) RFT-LoRa vs. EE-LoRa with PurpleAir [21] dataset: In this experiment, we test RFT-LoRa and EE-LoRa on an area covering $6,000km^2$. End device locations are initialized based on the PurpleAir deployment [21]. The remote sensing-based model is used to configure the path loss matrix. We run ns-3 simulations for various levels of gateway failures and interference. We incorporate our remote sensing-based path loss model into EE-LoRa to equalize the comparison between the two models. Without that, the generated deployment by EE-LoRa would be unusable due to its PDR being close to zero.

The transmission reliability and lifetime under various levels of gateway failures and interference are shown in Figure 3. RFT-LoRa deploys 6, 9, 12 gateways under *1, 2, 3-gateway connectivity* for the PurpleAir setup. The baseline is EE-LoRa with gateway numbers set to 6, 9, 12 to match RFT-LoRa in

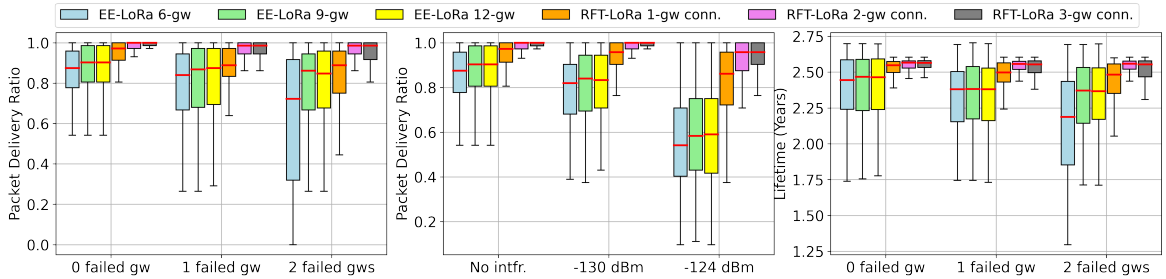

Fig. 3: ns-3 simulation results. Left: Distribution of PDR under various levels of gateway failures. Middle: Distribution of PDR under various levels of interference. Right: Distribution of lifetime under various levels of gateway failures.

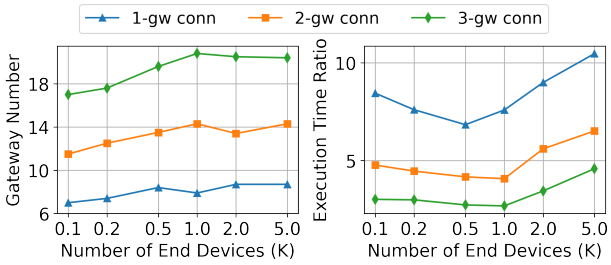

Fig. 4: Left: Average placed gateway number of RFT-LoRa. Right: Average execution time ratio between RFT-LoRa and EE-LoRa.

each comparison. RFT-LoRa enhances PDR by 10% - 54% on average as compared to EE-LoRa. The benefits of *m-gateway connectivity* for fault tolerance of the network are clear when gateway failures and interference are injected. When failures or interference occur and a lot of incoming packets are corrupted, the end device automatically switches to the higher SF. With RFT-LoRa, additional backup gateways can be reached, so the robustness is enhanced. Impressively, RFT-LoRa is able to maintain the average PDR above 0.8, the predetermined PDR_{th}, across all cases. In spite of the same gateway number, EE-LoRa has much lower PDR. With strong interference of -124 dBm, the average PDR of all EE-LoRa's deployments drops by as much as 60%, while RFT-LoRa retains robustness.

The lifetime distribution among all devices in the gateway-failure scenario is presented in Figure 3 (right). We omit the plot of lifetime distribution in the interference scenario since it has a similar pattern. Higher PDR leads to longer lifetime in devices with the *confirmed* transmissions. We can observe 5% - 18% longer and more stable lifetime on average with RFT-LoRa over its EE-LoRa counterpart. RFT-LoRa does not optimize the lifetime specifically, but instead focuses on meeting the 2-year lifetime constraint. In summary, ns-3 evaluations show RFT-LoRa's excellent ability to improve the fault tolerance while ensuring transmission reliability and lifetime, when incorporated into LoRaWAN.

3) Scalability on large problems: Figure 4 (left) shows the number of placed gateways with RFT-LoRa when varying the number of end devices in a large-scale deployment. We take the average gateway number after 10 random end-device initializations. RFT-LoRa places nearly constant number of gateways to cover the given area under *1-* and *2-gateway connectivity*, even though the number of end devices increases on the log scale. The sudden gateway increase at 500 and 1K devices under *3-gateway connectivity* indicates a saturated capacity with our configuration. We also notice that RFT-LoRa requires only 1.2x - 1.5x more gateways instead of 2x when switching from *1-* to *3-gateway connectivity*, resulting in large cost savings during the deployment. These savings come from leveraging backup gateways as the primary gateways for other end devices. In practice, the user can identify different *m-gateway connectivity* requirements at various locations and apply our methodology to trade-off fault tolerance and installation costs.

Figure 4 (right) depicts the execution time ratio between RFT-LoRa and EE-LoRa under the same gateway number. RFT-LoRa takes generally longer to finish compared with EE-LoRa. However, EE-LoRa comes with no performance guarantees while RFT-LoRa instead ensures PDR and lifetime constraints especially under potential gateway failures and interference (as shown in the previous experiment). RFT-LoRa takes at most 38 minutes for a large configuration of 5,000 devices, while EE-LoRa takes only 10 minutes. As RFT-LoRa is run only once prior to deployment, it is reasonable to invest a bit more time for a better performance to ensure a more reliable deployment.

VII. Conclusion

In this paper, we study the problem of automating the design of a robust and fault-tolerant LoRa network leveraging current LoRaWAN. We propose *m-gateway connectivity* as a novel strategy to provide fault-tolerance in LoRa networks. We use the remote sensing technology to obtain a more accurate estimate of path loss in the network. An INLP is rigorously formulated to minimize the number of gateways through gateway placement and resource allocation, while achieving (i) fault tolerance, (ii) reliable transmission, (i.e., satisfactory QoS), (iii) sufficiently long lifetime. Due to the complexity in solving INLP, we provide a greedy heuristic, RFT-LoRa, to get high-quality solutions. Comprehensive evaluations show that RFT-LoRa is a good approximation of the optimal solution. It improves the PDR by 10% - 54% on average over the state of the art design by Ousat *et al.* [15] when gateway failures and interference occur.

ACKNOWLEDGMENTS

This work was originated and largely completed while X. Yu worked at Arm Research during the summer internship of 2020. T. Rosing and X. Yu are partially supported by Semiconductor Research Corporation task #2805.001, and in part by National Science Foundation under Grants #1911095, #1826967, #1730158, #1527034, #2100237, #2112167, #2003279.

REFERENCES

[1] Semtech, "Smart cities transformed using lora technology," Nov. 2016. [Online]. Available: https://www.semtech.com/uploads/technology/LoRa/Semtech_SmartCitiesTransformed_WhitePaper_FINAL.pdf

[2] Semtech, "Revolutionizing smart arguiculture using semtech's lora technology," Oct. 2017. [Online]. Available: https://www.semtech.com/uploads/technology/LoRa/WP-SEMTECH-LORA-SMART-AGRICULTURE.pdf

[3] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund, "Using lora for industrial wireless networks," in *IEEE International Workshop on Factory Communication Systems (WFCS)*. IEEE, 2017, pp. 1–4.

[4] Ericsson, "Ericsson mobility report," Nov. 2020. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2020

[5] J. C. Zuniga and B. Ponsard, "Sigfox system description," *LPWAN@ IETF97, Nov. 14th*, vol. 25, 2016.

[6] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "Nb-iot system for m2m communication," in *IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2016, pp. 1–5.

[7] "Semtech." [Online]. Available: https://www.semtech.com/

[8] "LoRa Alliance." [Online]. Available: https://lora-alliance.org/

[9] "The things network." [Online]. Available: https://www.thethingsnetwork.org/

[10] IoT Analytics, "Lpwan market report 2020-2025," Jan. 2020. [Online]. Available: https://iot-analytics.com/product/lpwan-market-report-2020-2025/

[11] B. Ghena, J. Adkins, L. Shangguan, K. Jamieson, P. Levis, and P. Dutta, "Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity," in *Intl. Conf. on Mobile Computing and Networking*, 2019.

[12] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[13] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in lora networks," in *International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 788–798.

[14] E. Sallum, N. Pereira, M. Alves, and M. Santos, "Performance optimization on lora networks through assigning radio parameters," in *IEEE Intl. Conf. on Industrial Technology (ICIT)*, 2020.

[15] B. Ousat and M. Ghaderi, "Lora network planning: Gateway placement and device configuration," in *IEEE International Congress on Internet of Things (ICIOT)*. IEEE, 2019, pp. 25–32.

[16] I. Khoufi, P. Minet, and A. Laouiti, "Fault-tolerant and constrained relay node placement in wireless sensor networks," in *IEEE Intl. Conf. on Mobile Ad Hoc and Sensor Systems (MASS)*, 2016, pp. 127–135.

[17] K. Gokbayrak, "Robust gateway placement in wireless mesh networks," *Computers & Operations Research*, vol. 97, pp. 84–95, 2018.

[18] Y. Lin, W. Dong, Y. Gao, and T. Gu, "Sateloc: A virtual fingerprinting approach to outdoor lora localization using satellite images," in *ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2020, pp. 13–24.

[19] "ns-3: a discrete-event network simulator for internet systems." [Online]. Available: https://www.nsnam.org/

[20] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of lorawan performance under different parameter settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, 2019.

[21] PurpleAir LLC, "Purpleair: Real time air quality monitoring." [Online]. Available: https://www2.purpleair.com/

[22] Y. Liu, W. Huangfu, H. Zhang, H. Wang, W. An, and K. Long, "An efficient geometry-induced genetic algorithm for base station placement in cellular networks," *IEEE Access*, vol. 7, pp. 108 604–108 616, 2019.

[23] I. Gravalos, P. Makris, K. Christodoulopoulos, and E. A. Varvarigos, "Efficient gateways placement for internet of things with qos constraints," in *IEEE Global Communications Conference (GLOBECOM)*, 2016.

[24] C. Zhao and et al., "Maximizing lifetime of a wireless sensor network via joint optimizing sink placement and sensor-to-sink routing," *Applied Mathematical Modelling*, vol. 49, pp. 319–337, 2017.

[25] D. Benyamina, A. Hafid, and M. Gendreau, "A multi-objective optimization model for planning robust and least interfered wireless mesh networks," in *IEEE Global Telecommunications Conference*, 2008.

[26] S. P. Tirani and A. Avokh, "On the performance of sink placement in wsns considering energy-balanced compressive sensing-based data aggregation," *J. of Network and Computer Applications*, vol. 107, 2018.

[27] F. Li, Y. Wang, X.-Y. Li, A. Nusairat, and Y. Wu, "Gateway placement for throughput optimization in wireless mesh networks," *Mobile Networks and Applications*, vol. 13, no. 1-2, pp. 198–211, 2008.

[28] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway placement optimization in wireless mesh networks with qos constraints," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, 2006.

[29] A. M. Ahmed and A. H. A. Hashim, "A genetic approach for gateway placement in wireless mesh networks," *Intl. Journal of Computer Science and Network Security (IJCSNS)*, vol. 15, no. 7, p. 11, 2015.

[30] M. R. Girgis, T. M. Mahmoud, B. A. Abdullatif, and A. M. Rabie, "Solving the wireless mesh network design problem using genetic algorithm and simulated annealing optimization methods," *International Journal of Computer Applications*, vol. 96, no. 11, 2014.

[31] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks," *Computers & Electrical Engineering*, vol. 56, 2016.

[32] C. Yang, K.-W. Chin, Y. Liu, J. Zhang, and T. He, "Robust targets coverage for energy harvesting wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5884–5892, 2019.

[33] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of lorawans through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, 2018.

[34] F. Cuomo et al, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," in *IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications*, 2017.

[35] R. Kerkouche, R. Alami, R. Féraud, N. Varsier, and P. Maillé, "Node-based optimization of lora transmissions with multi-armed bandit algorithms," in *2018 25th International Conference on Telecommunications (ICT)*. IEEE, 2018, pp. 521–526.

[36] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *The Conference of the ACM Special Interest Group on Data Communication*, 2017.

[37] A. Dongare and et al., "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in *ACM/IEEE Intl. Conf on Information Processing in Sensor Networks (IPSN)*, 2018.

[38] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 2, pp. 1–35, 2019.

[39] LoRa Alliance, "LoRaWAN Specification V1.0.4," Oct. 2020. [Online]. Available: https://lora-alliance.org/resource_hub/lorawan-104-specification-package/

[40] ——, "A Technical Overview of LoRa and LoRaWAN," Nov. 2020. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf

[41] S. Demetri, M. Zúñiga, G. P. Picco, F. Kuipers, L. Bruzzone, and T. Telkamp, "Automated estimation of link quality for lora: A remote sensing approach," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2019, pp. 145–156.

[42] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of lora networks in a smart city scenario," in *2017 IEEE International Conference on communications (ICC)*. ieee, 2017, pp. 1–7.

[43] "Arduino uno rev3." [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3

[44] "Semtech sx1276 chip." [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276

[45] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[46] B. Pierre and et al., "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, 2008.

[47] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, 2005.