Joint Slice-based Spreading Factor and Transmission Power Optimization in LoRa Smart City Networks

Samir Dawaliby, Abbas Bradai, Yannis Pousset

Version of Record: https://www.sciencedirect.com/science/article/pii/S2542660519301908
Manuscript_ac624bef13206bc9f5ee123bbaa72052

Joint Slice-based Spreading Factor and Transmission Power Optimization in LoRa Smart City Networks

Samir Dawaliby[1], Abbas Bradai, Yannis Pousset

XLIM Laboratory, 11 Boulevard Marie et Pierre Curie, Chasseneuil du Poitou, France

[a] *University of Poitiers*

Abstract

The fifth generation (5G) wireless networks is expected to support an all-connected world with a multitude internet of things (IoT) applications. To reach this goal, network slicing is adopted to provide flexibility in managing heterogeneous IoT networks. The focus of this paper is to implement an adaptive dynamic network slicing mechanism in a Lora-based smart city network using a maximum likelihood estimation. The latter avoids resource starvation and is combined with a slice-based optimization method that configures spreading factor and transmission power parameters in a way that maximizes the performance utility in each slice. Simulation results performed in realistic LoRa scenarios highlight the utility of our proposition in respecting defined quality of service (QoS) thresholds in terms of delay, throughput, energy consumption and improving reliability while providing a complete isolation between LoRa slices.

Keywords: Internet of Things (IoT), smart city, LoRa, slice optimization, resource allocation, dynamic network slicing

1. Introduction

By 2020, the fifth generation (5G) wireless networks will potentially support 50 billion devices in an all-connected world of humans and machines communicating through the internet [1]. This increasing number poses several challenges regarding wireless network management with each having various service requirements. Therefore, network flexibility is needed to be able to provide services like monitoring smart objects using real-time connectivity. To achieve this flexibility and improve the decision making in terms of resource allocation and parameters configuration, major technologies arised namely network functions virtualization (NFV) and software defined networking (SDN). With the development of the latter, network slicing is proposed as one of the most important technologies to reach this goal by using a collection of logical functions and separating the software-based control plane from the hardware-based data plane in future generation networks.

The random-based access nature in the internet of things (IoT), gives the motivation to investigate network slicing over LoRa technology with the objective to provide isolation between multiple virtual

Email addresses: samir.dawaliby@univ-poitiers.fr (Samir Dawaliby), abbas.bradai@univ-poitiers.fr (Abbas Bradai), yannis.pousset@univ-poitiers.fr (Yannis Pousset)
[1]Corresponding author

networks with each having specific QoS requirements to be created on a common LoRa physical device, being mutually instantiated on-demand and independently managed. Each slice suffers from performance degradation even in isolated networks due to the limited number of available channels in unlicensed bands of Long Range wide area network (LoRaWAN) [2]. This leads to new challenges in finding an efficient resource reservation method supported with a slice-based parameters configuration for LoRa devices (spreading factor (SF), transmission power (TP), bandwidth, etc.) to improve QoS of IoT devices and limit interference and collisions in LoRa network.

1.1. Related works

Many recent works focused on the importance of LoRaWAN as a candidate technology supported by many network operators even for mobile things [3] compared to other low power, wide area (LPWA) technologies like DASH7 [4], LTE-M [5] and NB-IoT [6]. Using its large coverage capability and low energy consumption, LoRa enables the opportunity to efficiently cover and support various smart city IoT applications. The performance of LoRaWAN has been intensively evaluated in the literature for IoT applications at the link and system level viewpoints [7] [8] and integrated in industrial IoT monitoring applications [9] [10]. Many research studies focused on proposing various SF configuration strategies over multiple network deployments [11] with the goal to overcome capacity limits [12] and to provide a trade-off solution that minimizes energy consumption while maximizing reliability [13]. Furthermore, authors showed the importance of configuring IoT devices with a proper combination between SF and TP parameters to improve scalability of LoRaWAN [14] and to avoid performance degradation and unfairness that happens in LoRa network if IoT devices configure SF and TP locally [15]. LoRa originally includes a link-based adaptation of SF and TP configurations using the Adaptive Data Rate (ADR) mechanism. Many works tried to propose modified and improved ADR algorithms with the goal to increase reliability and energy-efficiency without taking into consideration the possibility of intra-SF and inter-SF collisions [16] [17] [18]. The latter can be decreased with the knowledge of the entire network or by finding the optimum configuration after testing all combinations of LoRa parameters that respects specific thresholds [19]. However, this method is considered as time consuming because sometimes, achieving multi-objectives in terms of reliability and energy-efficiency does not always require tuning parameters, especially on IoT devices placed at the edge of their communication range [20]. In [21], the performance of the official ADR mechanism proposed by LoRa is evaluated and shows the impact of different configurable parameters in terms of slow convergence rate which introduces higher energy consumption and packet losses. This highlights the need for an optimization solution that can decently configure LoRa parameters and improve the performance of LoRaWAN.

Nowadays, guaranteeing service requirements in LoRaWAN with traffic slicing remains as open research issue [2]. Therefore, unlike the previous work, we aim in this article to extend network slicing in LoRaWAN [22] by considering smart city applications belonging to different QoS classes and to support the latter with a slice-based SF and TP configuration optimization which, to the best of our knowledge, has not been treated before by the research community.

1.2. Contributions and outlines

Our main contributions with respect to the surveyed literature are stated as follows:

1. We include QoS in LoRa, which was previously considered as a best effort technology, with the goal to test the flexibility that network slicing provides in terms of traffic management and QoS integration.
2. We propose an adaptive dynamic resource reservation algorithm where the bandwidth is efficiently reserved on each gateway separately based on a maximum likelihood estimation (MLE). The goal of this scheme is to avoid channels starvation of LoRa slices while considering the exact need of each slice starting by the one with the highest slicing priority.
3. We propose *TOPG* as a novel slicing optimization method that is based on Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Geometric Mean Method (GMM). The proposed method efficiently configures LoRa SF and TP parameters and improves the performance of each slice in terms of QoS, reliability and energy consumption.

The remainder of this paper is organized as follows. We devote Section II and III to respectively describe the LoRa system model and the network slicing problem established in this paper. Section IV presents the proposed slicing and optimization algorithm implemented over the LoRa module of NS3 simulator [23]. The performance evaluation of the algorithm and simulation results are analyzed and carried out through various scenarios in Section V. Finally, Section VI concludes the paper.

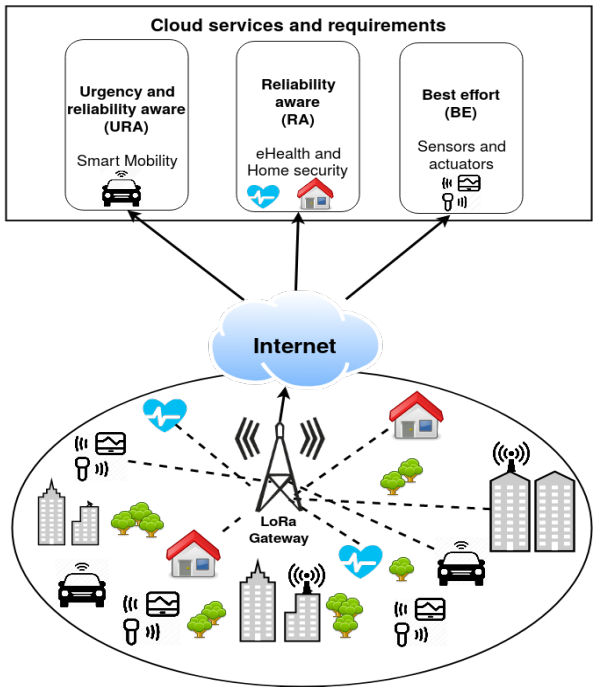


Figure 1: Smart City applications in LoRa-based network

2. Modeling slicing in Smart City Network

In smart city networks, the vision is to reach a high quality of life environment relying on data collected via connected objects and many sensors and actuators. **Fig. 1** illustrates a smart city

scenario enabled by IoT with various use cases for citizens in mobility, smart home, health and many other fields. However, due to the heterogeneity of these applications, a single smart city network is unable to support all of these traffic types within a network without compromising QoS for any of them. In case of accident, a connected vehicle should immediately communicate the information to the people involved and responsible of emergency situations. However, this information could be lost or arrived without respecting the required delay in urban cities.

Table 1: IoT QCIs table [24] [25]

QCI	Slice Name	Resource Type	Priority	Packet Delay Budget (ms)	PER %	Example Services
71	URA	GBR	1	100	10^{-3}	Real time, smart mobility
72	RA	GBR	2	200	10^{-3}	Real time, eHealth and home security
73	BE	nGBR	3	300	10^{-6}	Delay tolerant, smart agriculture

In this work, the focus is on applying traffic slicing in smart city scenarios, virtually isolated, and with specific QoS thresholds. **Table 1** summarizes the key QoS requirements of three virtual slices defined in terms of guaranteed bit rate (GBR), slice priority, packet delay budget (PDB) and packet error rate (PER). Urgency and reliability-aware (URA) slice gives more importance for QoS and reliability and requires the highest slicing priority. The latter can be required by many IoT applications, i.e: surveillance, emergency alerting and smart mobility. Moreover, reliability-aware (RA) slice gives equal importance for applications requiring reliability and are less critical in terms of delay. Some examples regarding this slice can be summarized in live monitoring applications like health sensors and home security systems. The third and last slice denoted as best effort (BE) slice, gives the lowest priority for IoT members running applications with non-guaranteed data rate and delay-tolerant QoS requirements, i.e: smart metering and smart agriculture applications.

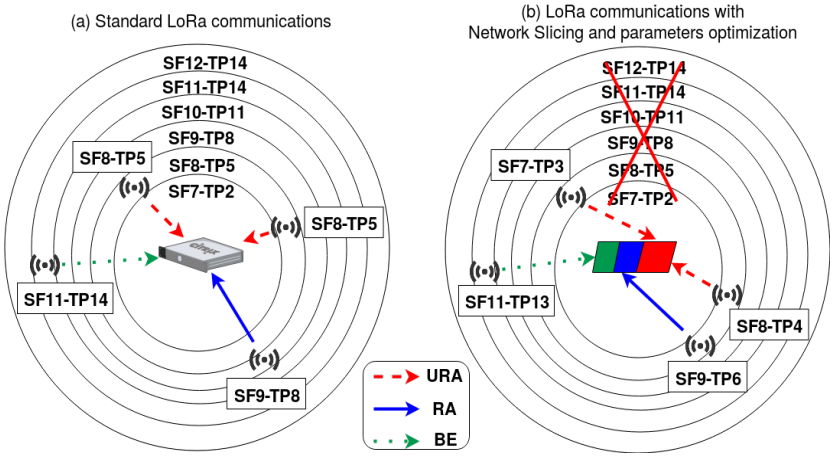


Figure 2: (a) Standard LoRa and (b) LoRa Network Slicing with parameters optimization

Fig. 2 illustrates how IoT devices are connected to a gateway in LoRa standard architecture (**Fig. 2a**) and configured with one of SF-TP combinations available with the ADR configuration. The server aims to increase both SF and TP values simultaneously to be able to decode packets at larger distance from the gateway. However, when network slicing is applied on a Lora gateway (**Fig. 2b**), ADR

mechanism becomes inefficient especially if the device in question belongs to a slice having specific QoS thresholds that need to be respected before reaching external LoRa servers through the internet. For example, with traffic slicing, the receipt of urgent communications is now guaranteed at the GW level. However, overestimating SF and TP configurations leads to an increase in energy consumption due to the longer activity time for an IoT device when uploading a packet with high SF configuration. Moreover, if a high SF is configured, achieved throughput may be lower than the one that needs to be guaranteed in the corresponding slice. Hence, for each slice, one should not be limited to the possible configurations proposed by LoRa ADR mechanism. This work enables the possibility to define specific slice-based SF and TP combination to be configured on an IoT device in a way that respects its QoS thresholds.

In this case of work, we assume that centralized LoRa servers are aware of the QoS required by each active device in the network in terms of delay, throughput and reliability. Moreover, LoRa servers are responsible on defining resource reservation strategies on LoRa gateways (GWs) and on configuring the devices with SF and TP parameters. Let N(V,K) be a directed LoRa network including V={S,M,C} components and consists of S LoRa servers, $M = \{m_1, ..m_{m'}.., m_M\}$ denotes the set of LoRa gateways and $C = \{c_1, ..c_{c'}.., c_C\}$ denotes the set of channels on each gateway. Let $K = \{k_1, ..k_{k'}.., k_K\}$ be the set of IoT devices connected to the gateways and belongs to the set of slices L. Each slice is defined based on delay, throughput and reliability requirements of IoT applications [24]. It is noteworthy that to improve communications in an IoT environment, multiple objectives should be reached. More precisely, we jointly consider in this work QoS, energy and reliability requirements as major key factors to optimize parameters configuration of an IoT device belonging to a slice with a specific slicing priority sp_l. On each LoRa gateway, a slicing rate is estimated based on the throughput required by the devices active in each slice l in order to define capacity c_l that needs to be reserved. Each gateway has a fixed number of C channels with $C_{l,m}$ the set of channels reserved for slice l on GW m. We search to jointly optimize QoS and network slicing energy efficiency by assigning slice members with the proper SF and TP configurations. However, solving this multi-objective problem is challenging. Therefore, the goal in this work is to optimize parameters selection after evaluating the cost and benefits in each slice. We added σ_1, σ_2 and σ_3 as constant variables to equally distribute the weight between objective functions and we introduced $\alpha_{k,l} \in \{0, 1\}$ and $\beta_{C_l,m} \in \{0, 1\}$ as two binary decision variables that respectively indicate the admission of device k to slice l and the reservation of a channel C_l on GW m.

3. Problem Description

Network slicing optimization in IoT is a twofold problem and involves: *1) Finding the best inter-slicing resources reservation strategy*; *2) Configuring each slice member with the optimum SF and TP parameters*. In this work, the goal is to optimize the global performance of each slice in terms of QoS, energy and reliability. This turns the second problem of finding the best SF and TP configuration for an IoT device into a multi-objective problem formulated as follows:

3.1. QoS in a LoRa slice

LoRa adopts Chirp Spread Spectrum (CSS) modulation which tansmits symbols by encoding them into multiple signals (chips) of increasing or decreasing radio frequencies making signals more robust to multi-path interference, Doppler shifts and fading [26]. Each device k adopts a specific SF configuration for information transmission. The choice of SF configuration is very crucial because the latter is directly related to throughput, range and transmission time. Moreover, the bandwidth is by definition the number of wave cycles per second, which in LoRa, represents the number of chips transmitted per second and is defined as the chip rate CR. Hence, with CSS, CR is always going to be numerically equal to the selected bandwidth of a LoRa channel and their symbols can be interchangeably used [27]. LoRa spreads 2^{SF} chips per symbol with SF being a discrete value that represents the number of bits per symbol and varies between 7 and 12 resulting a data rate computed as written in **Eq. 1** below:

$$r_{k,c} = SF.\frac{CR}{2^{SF}} = SF.\frac{b_c}{2^{SF}} \quad bits/s \tag{1}$$

where CR denotes the chip rate and $r_{k,c}$ the data rate achieved by a device k assigned to a channel c with specific bandwidth b_c configured with a specific SF configuration. Moreover, delay in LoRa is expressed following to **Eq. 2** below:

$$d_{k,c} = \frac{L}{r_{k,c}} \quad seconds \tag{2}$$

$d_{k,c}$ represents the transmission delay of a packet with a length of L bits transmitted by device k with a specific SF configuration. Hence, a higher SF configuration increases throughput and simultaneously decreases delay. Based on what was previously mentioned, we model in **Eq. 3** the QoS cost as:

$$\begin{aligned} &QoS_{k,c} = \overline{r_{k,c}} + (1 - \overline{d_{k,c}}) \\ &Maximize \sum_{k \in K} \alpha_{k,l} QoS_{k,c}, \forall l \in L, \forall c \in C_{l,m} \end{aligned} \tag{3}$$

where $QoS_{k,c}$ denotes the benefits that should be maximized for each slice members and respectively includes $\overline{d_{k,c}}$ and $\overline{r_{k,c}}$ normalized by dividing $r_{k,c}$ and $d_{k,c}$ values by the highest throughput and delay that can be achieved over a wireless LoRa link. The latter cannot exceeds thresholds defined for its specific virtual slice defined in **Table 1.**

3.2. Interference in a LoRa slice

The reason for loosing a packet uploaded by an IoT LoRa device is three-fold: 1) when a packet is received under-sensitivity if the transmitting device was out of range or configured with bad SF and TP values, 2) when packets are lost due to collisions that happens between two devices transmitting with the same spreading factor at the same time and 3) when a collision happens between two packets transmitted with different spreading factors leading to a potential loss due to inter-SF interference. The goal in this section is to reduce the probability of losing a packet. Let $PLR'_{k,c}$ be a binary variable that indicates:

$$PLR'_{k,c} = \begin{cases} 0 & \text{if device } k \text{ successfully reaches } c \in C_{l,m} \\ 1 & \text{Otherwise} \end{cases}$$

This mainly depends on the sensitivity of the gateway that increases alongside an increase in SF configuration [23]. Based on random access formula [28], collisions happen on a gateway channel between two devices configured with the same spreading factor. The probability of the latter G_{SF} depends on the number of packets generated during the transmission of 1 packet with the same SF and is written in **Eq. 4** based on random access formula below:

$$PLR''_{k,c} = 1 - e^{-2G_{SF}} \tag{4}$$

Moreover, we also follow the assumptions in [23] where a packet should survive interference that comes from other LoRa transmissions. Signal-to-interference-plus-noise ratio (SINR) varies based on the adopted SF on each device. Each LoRa device experiences a SINR value computed based on the **Eq. 5** below:

$$SINR_{i,j} = \frac{P^{rx}_{n,i}}{\sigma^2 + \sum_{n \in \partial_j} P^{rx}_{n,j}} \tag{5}$$

where $P^{rx}_{n,i}$ is the power of the packet n under consideration sent by device with $SF = i$ and ∂_j a set of interfering packets with a common $SF = j$. Each element in the cochannel rejection below [29] denotes the minimum signal power margin $V_{i,j}$, with $i, j \in \{7, ..., 12\}$, that a packet sent with $SF = i$ must have in order to be decoded successfully over every interfering packet with $SF = j$. Hence, packet survives interference with all interfering packets if, considering all combinations of SF, a higher power margin value (dB) is satisfied than the corresponding co-channel rejection value.

$$\begin{array}{c} \\ SF_7 \\ SF_8 \\ SF_9 \\ SF_{10} \\ SF_{11} \\ SF_{12} \end{array} \begin{array}{c} \begin{array}{cccccc} SF_7 & SF_8 & SF_9 & SF_{10} & SF_{11} & SF_{12} \end{array} \\ \begin{pmatrix} -6 & 16 & 18 & 19 & 19 & 20 \\ 24 & -6 & 20 & 22 & 22 & 22 \\ 27 & 27 & -6 & 23 & 25 & 25 \\ 30 & 30 & 30 & -6 & 26 & 28 \\ 33 & 33 & 33 & 33 & -6 & 29 \\ 36 & 36 & 36 & 36 & 36 & -6 \end{pmatrix} \end{array}$$

Therefore, the probability of collisions $PLR'''_{k,c}$ that happens on a gateway channel between two devices configured with different spreading factors is also modeled as a binary variable if it satisfies interference thresholds:

$$PLR'''_{k,c} = \begin{cases} 0 & \text{if device } k \text{ survives interference} \\ 1 & \text{Otherwise} \end{cases}$$

We finally model in **Eq. 6**, the reliability cost of a link with the objective of minimizing the probability of loosing a packet due to interference or low channel sensitivity:

$$\begin{aligned} &PLR_{k,c} = PLR'_{k,c} + PLR''_{k,c} + PLR'''_{k,c} \\ &Minimize \sum_{k \in K} \alpha_{k,l} PLR_{k,c}, \forall c \in C_{l,m}, \forall l \in L \end{aligned} \tag{6}$$

3.3. Energy Consumption in a LoRa slice

Increasing the spreading factor reduces the transmitted data rate, decreases the strength of the signal and offers a better sensitivity at the gateway receiver following to **Eq. 7** below:

$$P^{tx}_{k,l,m} = \frac{P^{rx}_{k,l,m} L}{g^{rx}_{k,l,m} g^{tx}_{k,l,m} e^{\xi}} \tag{7}$$

where $P^{rx}_{k,l,m}$ and $P^{tx}_{k,l,m}$ denotes the received and transmitted power of an active device with a channel antenna gain expressed with $g^{rx}_{k,l,m}$ and $g^{tx}_{k,l,m}$ respectively. e^{ξ} is the lognormal shadowing component with $\xi \sim N(0, \sigma^2)$ and L is the path loss which depends on the distance between the transmitter and the receiver. The latter is adopted to evaluate the performance of LoRa devices in a dense environment and is expressed following to the **Eq. 8** below:

$$L = L_0 + 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) \tag{8}$$

where d the length of the path (m), n represents the path loss distance exponent, d_0 the reference distance (m) and L_0 the path loss at reference distance (dB). We assume in this paper two energy states for an IoT device which can be computed in active or in sleep mode. Accordingly, we compute the energy of a LoRa device during a slicing interval time following to **Eq. 9** with the objective of minimizing energy consumption in a LoRa slice without degrading QoS performance:

$$\begin{aligned} &E_{k,c} = P^{tx}_{k,l,m} T_{active} + P^{sleep}_{k,l,m} T_{sleep} \\ &Minimize \sum_{k \in K} \alpha_{k,l} E_{k,c}, \forall c \in C_{l,m}, \forall m \in M, \forall l \in L \end{aligned} \tag{9}$$

Due to the multi-objectivity of the problem, we search to find the optimum slicing strategy with the proper SF and TP configurations that simultaneously maximize QoS benefits of each slice and minimize energy and reliability costs without under-optimizing a function over another. This multi-objective problem is formulated subject to the constraints below:

$$C1 : \sum_{l \in L} \alpha_{k,l} = 1, \forall k \in K \tag{10a}$$

$$C2 : b_{l,m} \cap b_{l',m} = \emptyset, \forall l, l' \in L, \forall m \in M \tag{10b}$$

$$C3 : 0 \leq P_{k,c} \leq P^{max}_{k}, \forall c \in C_{l,m} \tag{10c}$$

$$C4 : \sum_{k \epsilon K} \alpha_{k,l} \beta_{c,m} R_{k,c} \leq R^{max}_{l,m}, \forall l \in L, \forall m \in M, \forall c \in C_{l,m} \tag{10d}$$

$$C5 : \alpha_{k,l} \in \{0, 1\}, \forall k \in K, \forall l \in L \tag{10e}$$

$$C6 : \beta_{c,m} = \begin{cases} 1 & \text{if channel belongs to channel } c \in C_{l,m}. \\ 0 & \text{Otherwise.} \end{cases} \tag{10f}$$

Knowing that multiple virtual network slices are built on top of a common physical gateway, (10a) ensures that each device should always choose exactly one and only network slice even if the latter was implemented on different physical gateways. Moreover, a perfect isolation is guaranteed in (10b) between two bandwidth parts assigned for two different slices regardless if the latter was reserved on the same or on two different gateways. The transmission power of each device is limited in constraint (10c). Furthermore, constraint (10d) guarantees the sum of uplink traffic sent by slice members which do not exceed the maximum data rate capacity of the slice that can be sent through each gateway. Constraint (10e) ensures binary association values of device k to slice l and constraint (10f) ensures binary reservation values of a channel $c \in C_{l,m}$ that belongs to slice l on a LoRa GW m.

4. The Proposed Slicing Algorithm

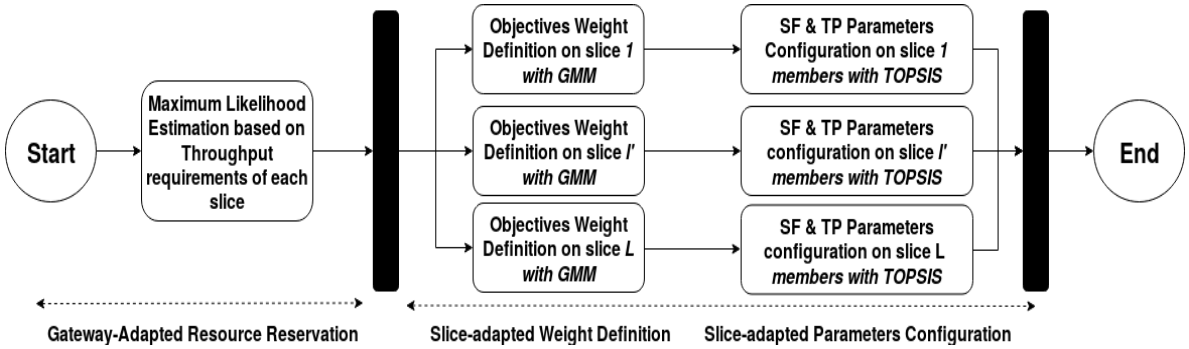


Figure 3: The Proposed Slicing and TOPG optimization algorithm

In this section, we expound the proposed slicing and configuration mechanism, illustrated in **Fig. 3**, that will optimize LoRa network slicing by catching up to the multi-objective optimization problem in finding the appropriate resource reservation and the best configuration to adopt for IoT devices. Network slicing virtually splits the network into various virtual networks that are isolated with each having heterogeneous degree of importance in terms of QoS, energy and reliability. The first problem appears in finding a decent slicing-strategy to split the physical network in a way that avoids resource starvation. To this manner, we propose to estimate and reserve appropriate channel radio resources by finding the maximum likelihood buffer demands for each slice starting by the one with the highest slicing priority. Next, GMM [30] is adopted to define the weights based on the objectives importance in each slice before being imported to a TOPSIS-based optimization [31] to find the best solution that maximizes utility requirements of the members of each slice.

4.1. Dynamic MLE-based Inter-Slicing Algorithm

The physical capacity of LoRa GW radio resources is limited. We assume that the traffic that needs to be uploaded follows a Poisson distribution [32] knowing that the servers are aware of the amount of data stored in the buffer B_i of each slice member.

Lemma 1. *Let T_i be the throughput needed by each device $i, \forall i \in K_l$ captured at each slicing interval time and identified by a corresponding probability distribution. For a fixed physical capacity, the opti-*

mum slicing strategy is to virtually reserve resources for each slice based on the mean throughput of its members.

Proof: We consider T_i follows a Poisson distribution $\mathrm{P}(\lambda)$ where λ denotes the estimation parameter of throughput needed by device i assigned to slice $l, \forall i \in K_l$. Let $f(T_i|\lambda)$ be a probability density function similar to $L(\lambda|T_i)$ that represents the likelihood of λ given the observed throughput.

$$
\begin{aligned}
L(\lambda|T_1, T_2, ..., T_{K_l}) &= f(T_1|\lambda)f(T_2|\lambda)....f(T_{K_l}|\lambda) \\
L(\lambda|T_1, T_2, ..., T_{K_l}) &= \prod_{i=1}^{K_l} \frac{e^{-\lambda}\lambda^{T_i}}{T_i!} \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= log\left[\prod_{i=1}^{K_l} \frac{e^{-\lambda}\lambda^{T_i}}{T_i!}\right] \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= \sum_{i=1}^{K_l} log\left[\frac{e^{-\lambda}\lambda^{T_i}}{T_i!}\right] \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= \sum_{i=1}^{K_l} \big[log(e^{-\lambda}) + log(\lambda^{T_i}) - log(T_i!)\big] \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= \sum_{i=1}^{K_l} \big[-\lambda + T_i log\lambda - log(T_i!)\big]
\end{aligned}
$$

To find the maximum likelihood parameter, we apply the first derivative and solve it to zero.

$$
\begin{aligned}
\frac{\partial logL(\lambda|T_1, T_2, ..., T_{K_l})}{\partial\lambda} &= \sum_{i=1}^{K_l} \big[-1 + \frac{T_i}{\lambda}\big] \\
&= -K_l + \frac{\sum_{i=1}^{K_l} T_i}{\lambda} = 0 \\
\widehat{\lambda} &= \frac{\sum_{i=1}^{K_l} T_i}{K_l}, \forall i \in \{1, ..., K_l\}
\end{aligned}
$$

To prove that the $\widehat{\lambda}$ is the maximum value, we apply a second derivative as follows:

$$
\frac{\partial^2 logL(\lambda|T_1, T_2, ..., T_{K_l})}{\partial_\lambda^2} = -\frac{\sum_{i=1}^{K_l} T_i}{\lambda^2}, \forall l \in L
$$

The obtained result is always a negative number which indicates that $\widehat{\lambda}$ is maximum and the optimal parameter to consider. Hence, the best slicing decision is to consider the mean throughput $\widehat{\lambda}_l$ of slice l members $\forall l \in L$. However, slices are not equal in terms of priority. Therefore, GW resources will be dynamically allocated to the most urgent slice starting by the channel with the highest reliability. Let $\Theta_l = \widehat{\lambda}_l / \sum_{l=1}^{L} \widehat{\lambda}_l$ be the slicing rate based on which the algorithm reserves for each slice a capacity $c_{l,m} = c_m.\Theta_l, \forall l \in L$.

4.2. The Proposed TOPG Optimization Algorithm

After defining slicing objectives of each LoRa virtual slice, we next need to adapt the weight of every objective before optimizing SF and TP configurations in a way that best meets the requirements of the corresponding slice. To do this, we propose an optimization algorithm based on GMM and TOPSIS methods. Let A_l=$(a_{ij,l})_{n\times n}$ be a judgment matrix where $a_{ij,l} > 0$ and $a_{ij,l} \times a_{ji,l} = 1$, with n denotes the number of objectives compared in each judgment matrix for slice l. Based on the objective importance in each slice, a priority vector is derived for each slice denoted as $\psi_l = (\psi_{1,l}, \psi_{2,l}, ..., \psi_{(n-1),l}, \psi_{n,l})$, where $\psi_l \geq 0$ and $\sum_{i=1}^{n} \psi_i = 1$, from the decision matrix A_l. With GMM, weight configuration for each objective is defined as an objective function of the following optimization problem:

$$\begin{cases} Minimize \sum_{i=1}^{n} \sum_{j>i} [ln(a_{ij,l}) - (ln(w_{i,l}) - ln(w_{j,l}))]^2 \\ s.t. \quad w_{i,l} \geq 0, \sum_{i=1}^{n} w_{i,l} = 1, \forall l \in L \end{cases}$$

which have a unique solution and can be simply solved by the geometric means of the rows of each slice's decision matrix A_l:

$$w_{i,l} = \frac{\sqrt[n]{\prod_{j=1}^{n} a_{ij}}}{\sum_{i=1}^{n} (\sqrt[n]{\prod_{j=1}^{n} a_{ij}})} \tag{11}$$

After finding the objective weights for each slice, we import the weight vector of each slice into a decision matrix D_l, which consists of a set of possible alternatives A_x as follows:

$$D_l = \begin{array}{c} \\ A_1 \\ .. \\ .. \\ A_{m-1} \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} Alternatives & w_{1,l} & .. & w_{n-1,l} & w_{n,l} \end{array} \\ \begin{pmatrix} a_{1,1} & .. & a_{1,n-1} & a_{1,n} \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ a_{m-1,1} & .. & a_{m-1,n-1} & a_{m-1,n} \\ a_{m,1} & .. & a_{m,n-1} & a_{m,n} \end{pmatrix} \end{array}$$

where each value $a_{x,y}$ represents a parameter configuration of a device with $y \in \{1, 2, ..., n\}$ defines the objective and $x \in \{1, 2, ..., m\}$ denotes a combination of SF $i \in I = \{7, ..., 12\}$ and TP discrete values $j \in J = \{2, ..., 14\}$ in dBm among which LoRa servers need to assign the device with the best configuration based on W_l, the set of objectives weight values of the corresponding slice. TOPSIS method requires normalized values $\overline{a_{x,y}}$ in D_l with the goal is to find the alternative with the shortest distance from positive ideal solution and the one with the largest distance from the negative ideal solution.

$$\overline{a_{x,y}} = \frac{a_{x,y}}{\sqrt{\sum_{x=1}^{m} a_{x,y}^2}}, \quad with \quad x \in \{1, ..., m\}, y \in \{1, ..., n\} \tag{12a}$$

In other terms, the goal is to find the best configuration that maximizes QoS benefits and minimizes

the costs in terms of PLR and energy consumption. For each positive ideal solution A^+ and negative ideal solution A^-, normalized weight rating $v_{x,y}$ can be determined using the following equations:

$$v_{x,y} = w_{x,l}\overline{a_{x,y}}, \quad with \quad x \in \{1, ..., m\}, y \in \{1, ..., n\} \tag{12b}$$

$$A^+ = (v_1^+, v_2^+, ..., v_n^+) \tag{12c}$$

$$A^- = (v_1^-, v_2^-, ..., v_n^-) \tag{12d}$$

where V_y value results using equations

$$V_y^+ = \left\{ \max_x v_{x,y}, y \in Y_1; \min_x v_{x,y}, y \in Y_2 \right\} \tag{12e}$$

$$V_y^- = \left\{ \min_x v_{x,y}, y \in Y_1; \max_x v_{x,y}, y \in Y_2 \right\} \tag{12f}$$

where Y_1 and Y_2 respectively respect benefit and cost criterias. We calculate next the euclidean distance from the positive ideal solution and negative ideal solution of each alternative; respectively as follows:

$$d_i^+ = \sqrt{\sum_{j=1}^{n} (d_{i,j}^+)^2} \tag{12g}$$

$$d_i^- = \sqrt{\sum_{j=1}^{n} (d_{i,j}^-)^2} \tag{12h}$$

where $d_{x,y}^- = V_y^+ - v_{x,y}$, with $x = 1, ..., m$ and $d_{x,y}^- = V_y^- - v_{x,y}$, with $x = 1, ..., m$.

$$\zeta_x = \frac{d_x^-}{d_x^+ + d_x^-} \tag{12i}$$

We finally rank the configurations according to the relative closeness previously calculated and we assign each device with the configuration that provides the highest value ζ_x due to its closest position to the positive ideal solution.

Pseudo-code 1 Adaptive Slicing and (SF-TP) Configuration

Input : Capacities c_m; Number of slices L;
Set of Throughput Requirements T_l

begin
  Put slices in decreasing order based on priority sp_l
  for *each GW* m **do**
    for *each slice* $l \in L$ **do**
      Apply MLE based on the throughput required by slice l members in the range of GW m.
      Define slicing rate Θ_l.
      Reserve bandwidth capacity $c_{l,m}$.
    end
  end
  for *each GW* m **do**
    for *each slice* $l \in L$ **do**
      Apply GMM to define $W_{l,n}$ of each objective.
    end
  end
  Sort devices in $K_{l,m}$ based on urgency factor u_k.
  for *each device* $k \in K_{l,m}$ **do**
    Sort channels in $C_{l,m}$ based on link budget.
    for *each Channel* c *in* $C_{l,m}$ **do**
      if *config=false* **then**
        Apply TOPSIS to define (SF-TP) parameters: SF_k,TP_k=TOPSIS($w_{l,1}, .., w_{l,n}$)
      else
        config=*true*;
        Configure the device with SF_k and TP_k.
      end
    end
  end
end

Output: Set of resources reserved for each slice l.
(SF-TP) parameters configuration for each device k.

4.3. Complexity Analysis

We evaluate the complexity of the proposed algorithm briefly listed in **Pseudo-code 1** compared to other configuration methods implemented in this study. One primar method (*static*) is to statically configure all the devices with the same SF and TP configuration. The latter has a constant complexity of $O(1)$ due to its simplicity. Similarly, same complexity analysis is applied for dynamic random (DR) and dynamic adaptive (DA) methods because in DA, centralized LoRa servers assign a specific TP value based on the SF assigned for the device. The latter is determined based on the distance between the device and its closest GW. Whereas in DR, the controller randomly select SF and TP values for all IoT devices in the network. Moreover, the complexity of the proposed dynamic algorithm supported by ($TOPG$) is compared to the one supported by an optimal method (*optimal*). The latter includes a complete TOPSIS algorithm where all alternatives are tested with each including a different combination of SF and TP parameters. The complexity of the *optimal* algorithm is calculated as follows: an attribute normalization and weighting which result is $O(n^2)$, the algorithm complexity ranking which result is $O(1)$, the complexity of a positive-negative ideal solution and the distance to

alternative solutions is $O(n)$. Hence, the overall complexity of the *optimal* and the proposed $TOPG$ configuration is $O(n^2)$ [33]. However, instead of testing all possibilities of SF and TP configurations with the *optimal* algorithm, complexity is reduced in ($TOPG$) because the server reduces the search space to SF values that respect the guaranteed bit rate threshold. This reduces computation time without highly affecting QoS performance as will be shown in the following section.

5. Performance Evaluation

In uplink, centralized LoRa servers enable the opportunity to make efficient slicing decisions and optimum parameters configuration based on the knowledge of the data in the buffer of each LoRa device. We implemented our methods in the open source NS3 simulator [34] using LoRa model that was firstly developed by authors in [23]. For further implementation details, we refer the readers to LoRaWAN source code over github [35]. NS3 supports a protocol stack including PHY and MAC layers where packets are uploaded from IoT devices to LoRa gateways. The latter are connected to the network server in a point-to-point connection responsible of creating network slices and configuring IoT devices. The first section of **Table 2** gives a brief of LoRa parameters implemented in this work. Simulations are replicated 50 times with 95% confidence interval and are realized in realistic LoRa scenarios. We assume that devices are defining a random time for transmission but periodically uploading small packet payloads of 18 Bytes following to the work done in [26]. LoRa devices and gateways are both placed over a cell of 7.5 KM radius based on a uniform random distribution. Each device is configured with spreading factors that varies from 7 to 12 when uploading traffic to LoRa GWs. Each GW is characterized by 8 receiving channels wih each channel having a bandwidth of 125 kHz in the 867-868 MHz european sub-band.

The second section of **Table 2** summarizes LoRa energy model parameters. Based on the **Eq. 13** below, we seek to evaluate the energy consumed when we increase the number of LoRa devices in each slice.

$$E_{k,l,m} = \frac{p_i^{tx} + p_i^{rx}}{V + epa}.d_{tx/rx} \tag{13}$$

where $E_{k,l,m}$ is the energy consumed by an IoT device, V the LoRa supply voltage, *epa* the amplifier's added efficiency, d_{tx} the duration of transmission, p_i^{rx} the power of reception and p_i^{tx} the power of transmission that varies between 2 and 14 dBm depending on the configuration strategy adopted. We integrate an energy module for the LoRa module in NS3 similar to the one that already exists for Wifi and we applied energy parameters and the power model specified for LoRa in [26] and [36]. In the following, we start by a proof of isolation and we highlight the importance of finding proper SF-TP combination with a parameters study in which we focus on showing the impact of SF and TP on energy consumption, mean PLR and the percentage of devices that respected GBR and PDB.

5.1. Parameters Study

In this section, we investigate the performance of each slice when we put in place different SF-TP configuration strategies for a fixed number of 300 devices. We first study *static* configurations in which all devices in the cell are configured with one of the following SF-TP combinations (i.e.,

Table 2: Simulation Parameters

Simulation Parameters	
Simulation Time	600 seconds
Slicing Interval Time	50 seconds
Cell Radius	7.5 KM
Number of replications	50
LoRa devices and GWs distribution	Random Uniform
Propagation loss model	Log-distance
Bandwidth	125 kHz
Spreading Factor	{7,8,9,10,11,12}
Confidence intervals	95%
European ISM sub-band	863-870 MHz
Power Consumption Parameters [26] [36]	
Battery Maximum Capacity	950 mAh
LoRa Supply Voltage	3.3V
Amplifier Power's added Efficiency	10%
Connected (Tx/Rx-SF7 to SF12)	1.58 to 25.11 mW
Standby	0.09 mW
Sleep	0 mW

SF7-TP2, SF8-TP5, SF9-TP8, SF10-TP11, SF11-TP14 and SF12-TP14). Then, we study the impact of TP variation for *static* configuration compared to three types of dynamic configuration strategies namely, DR where each device randomly picks a SF and TP values, DA where each LoRa device dynamically adapts device parameters to one of the SF-TP configurations depending on the highest receiving power measured from the gateway and we compare them with $TOPG$ where dynamic slicing is supported with the proposed GMM and TOPSIS optimization.

Table 3: Parameters Study with static SF-TP configurations strategies

	Slice Name	Static					
		SF7-TP2	SF8-TP5	SF9-TP8	SF10-TP11	SF11-TP14	SF12-TP14
Devices that respect GBR (%)	Overall	2.9	6.21	14.65	23.08	0	0
Devices that respect PDB (%)	Overall	41.15	30.7	13.85	12.3	0	0
Mean Packet Loss Rate (%)	Overall	78.37	58.68	20.46	23.73	47.73	70.89
	URA Slice	6.94	6.80	10.23	3.33	5.33	5.94
	RA Slice	10.34	10.89	16.91	10.50	10.61	18.22
	BE Slice	82.71	82.31	72.87	86.16	84.07	75.84
Mean Energy Consumption (mJ)	Total	0.06	0.2	0.73	1.47	3.99	4.41
	URA Slice	0.01	0.04	0.16	0.28	0.67	0.74
	RA Slice	0.02	0.06	0.23	0.55	1.07	1.47
	BE Slice	0.03	0.1	0.35	0.64	2.26	2.21

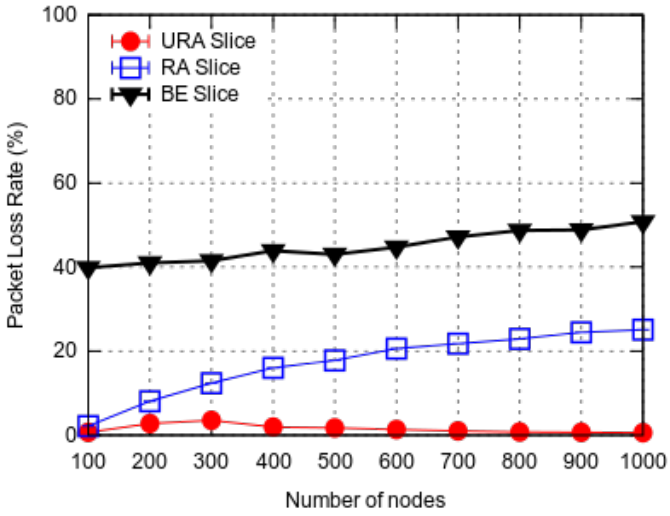


Figure 4: Proof of Isolation

5.1.1. Proof of Isolation

The very first step before investigating the strategies that can be used to configure SF and TP parameters is to prove the isolation concept between virtual slices in LoRa. Assuming that all devices are transmitting with the same DA configuration, we consider a single LoRa GW scenario in which we fix 20 LoRa devices for URA slice and we increase the number of devices. Therefore, all the devices that are left are assigned now to RA and BE slices. **Fig. 4** proves the isolation concept because when the number of devices increases in RA and BE slices, URA slice members were not affected and the percentage of PLR remained constant and nearly null whereas PLR increased in RA and BE slices in a more congested scenario.

5.1.2. Parameters study with Static SF-TP Configuration

Performance comparison between *static* configuration strategies is summarized in **Table 3** and evaluated in terms of QoS for a fixed packet transmission interval. When *static* configurations are adopted, all devices in the cell are configured with one of the following SF-TP combinations (i.e., SF7-TP2, SF8-TP5, SF9-TP8, SF10-TP11, SF11-TP14 and SF12-TP14). Results show that increasing the SF improves QoS metrics in terms of throughput and delay except for SF11 and SF12 where performance degrades tremendously. With high SF configurations, sensitivity is improved but the energy increases as well because with this configuration IoT devices occupy the spectrum for the longest time on air. This explains the increase in PLR and the probability that packets with same SF interfere upon transmission. However, with small SF configurations, energy is reduced with an improved QoS performance compared to high SF configurations. However, more than 50 % are lost due to lack of sensitivity, which means that a large number of packets are lost because they were not successfully received and decoded by the gateway.

Regarding QoS, increasing the SF reduces the throughput and increases the transmission delay.

This explains why the percentage of devices that respect PDB decreases due to the increase in transmission delay. However, knowing that throughput decreases when SF increases, it is noteworthy to mention that the percentage of devices that respect GBR is not affected and improves with SF. This is because a higher SF with higher TP helps more devices to deliver the required throughput while improving at the same time packets sensitivity. This clearly explains the low values in PLR and highlights the trade-off that some configurations deliver in terms of QoS, reliability and energy. Therefore, we pursue this study with $(SF9 - TP8)$ configuration due to its trade-off performance that this configuration provides between QoS, energy consumption and having the best overall PLR% between the ones simulated with *static* strategies.

Table 4: Complete Parameters Study with static and dynamic SF-TP configuration strategies

	Slice Name	Static-SF9					Dynamic		
		TP2	TP5	TP8	TP11	TP14	DR	DA	TOPG
Devices that respect GBR (%)	Overall	6	9.35	14.65	23.04	37.67	7.65	16.75	60.99
Devices that respect PDB (%)	Overall	0.45	1.7	13.85	19.32	29.86	76.3	94.8	85.73
Mean Packet Loss Rate (%)	Overall	61.77	45.96	20.46	12.3	9.59	20.86	4.37	12.26
	URA Slice	6.85	8.75	10.23	9.45	3.27	12.32	0.67	6.18
	RA Slice	15.54	16.00	16.91	15.24	5.84	23.69	0.97	11.13
	BE Slice	77.61	75.24	72.87	75.31	90.89	64	98.37	82.69
Mean Energy Consumption (mJ)	Total	0.18	0.37	0.73	1.46	2.91	3.53	1.04	1.8
	URA Slice	0.04	0.08	0.16	0.31	0.62	0.64	0.22	0.25
	RA Slice	0.06	0.12	0.23	0.46	0.92	1.1	0.33	0.49
	BE Slice	0.09	0.17	0.35	0.69	1.38	1.80	0.49	1.06

5.1.3. Parameters study with Dynamic SF-TP Configuration

After defining $(SF9 - TP8)$ as the best *static* configuration, we compare the latter to dynamic configurations. First, we highlight in this study the impact of increasing TP for *static* configurations before comparing its performance to DA, DR and the proposed $TOPG$ method. Based on the results shown in **Table 4** below, one can conclude the importance of efficiently identifying TP parameter due to its direct impact on QoS performance metrics. The results of each slice show the efficiency of URA compared to RA and BE slices in terms of reliability and energy consumption due to slicing priority consideration in MLE resource reservation mechanism. Increasing TP for SF9 configuration will increase packets arriving above sensitivity and improves the rate of devices that guaranteed delay and throughput on the expanse of energy consumption. This highlights the motivation for optimizing SF and TP parameters and the utility to sometimes increase TP for an IoT device, if the latter improves its QoS with respect to GBR and PDG thresholds defined for the slice it belongs to. Based on what was previously mentioned, IoT devices are configured with the highest TP for *static* $(SF9 - TP14)$ configuration because it gives the best QoS performance for LoRa slices. The latter will be evaluated in depth in the following section compared to DA, DR and the proposed $TOPG$ methods.

Regarding dynamic configurations, DA was the best strategy in terms of energy compared to DR and $TOPG$ because in DA, the centralized server dynamically configures LoRa devices with

one of the SF-TP combinations defined by LoRa. It measures the receiving power that a GW gets from the device depending on its position and configure the parameters accordingly. The advantages that dynamic configurations brings to LoRa are two-fold: first, depending on how far the device is from the gateway, a smaller distance requires a smaller SF configuration which also mean smaller TP and energy consumption. Secondly, the fact of adopting different SFs configuration reduces the the probability of collisions and the percentage of packets lost due to interference. However, similar to *static* configurations, *DA* is week in terms of QoS. This is also due to unefficient SF-TP distribution where it could be useful to improve QoS by keeping the same SF with higher TP value instead of increasing both SF and TP as it's done in *DA* method. Moreover, when devices are close to the gateway, it could be also interesting to reduce the TP to save energy without degrading QoS performance of IoT devices. *TOPG* results in **Table 4** clearly show the potential that this method brings and requires further evaluation in complete simulations due to the trade-off results that were achieved in terms of QoS, reliability and energy consumption.

5.2. Performance Evaluation of SF-TP Configurations

Following to previous simulation results, we focus in this section on evaluating the proposed *TOPG* configuration method that proved its worthiness for this study. We run now simulations starting by 100 devices over a network of four gateways managed by a centralized LoRa server and we increase the number of devices until the maximum number connected to a single gateway is reached and limited to 1000 devices, as shown in the scalability study in [37]. A load of one is emulated due to the legal duty-cycle limitations of 1% in the European region [38].

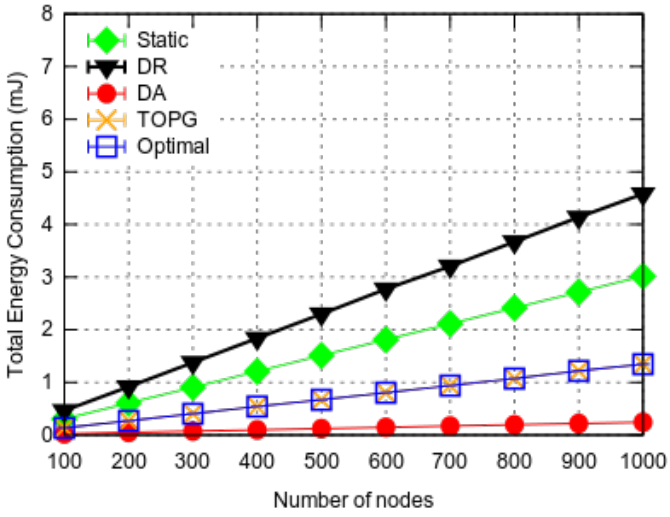
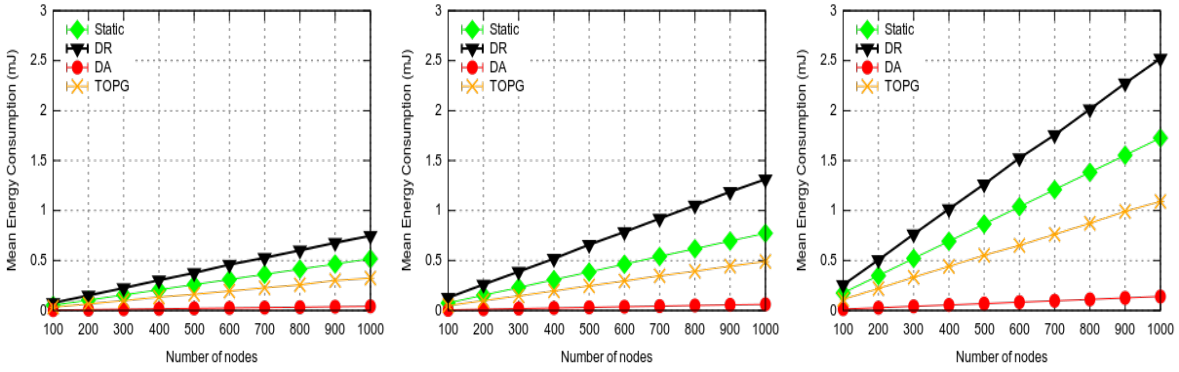


Figure 5: Total Energy Consumption Variation

5.2.1. Total Energy Consumption

In **Fig. 5**, when the number of devices increases, the total energy consumed increases as well regardless of the adopted SF-TP configuration. *DR* scored the highest energy consumption whereas

DA was the most energy efficient method because it configures for each device the minimum TP required. Moreover, it is normal that $TOPG$ algorithm consumes more energy because it configures SF and TP parameters while also considering QoS requirements of IoT devices in each slice. For further investigation, energy consumption is evaluated in each slice which also increased when the number of LoRa devices increases. In **Fig. 6a**, URA slice members scored the lowest energy consumption between the simulated slices. The reason for this result is due to the higher impact that SF parameter provides by letting IoT devices occupy the spectrum for a smaller duration of time even if configured with higher TP values. This also explains why even in RA and BE slices, $TOPG$ always had a higher energy consumption than DA and lower than DR and *static* configuration methods. RA and BE slice members consumed more energy compared to URA as shown in **Fig. 6b** and **Fig. 6c** respectively. This returns to GMM method that considers a slice-based configuration that gives higher importance for reliability and QoS in utility calculations. Hence, a higher weight is provided for QoS and reliability that forces delay-sensitive devices to take the most reliable gateway with the lowest SF-TP values compared to RA and BE slice members.



(a) URA Slice (b) RA Slice (c) BE Slice

Figure 6: Mean Energy Consumption in each slice with different SF-TP configurations

5.2.2. Packet Loss Rate

In this section, packet loss rate for each configuration algorithm is evaluated. Results shown in **Fig. 7** prove the efficiency of the proposed optimization method in reducing PLR compared to static and dynamic configuration strategies. With *static* ($SF9 - TP14$) configuration, PLR was highly affected, continuously increased and more than 30 % of packets were lost due to congestion. However, the worst result was scored with DR method where IoT devices lost approximately 40 % of their packets due to wrong configurations that lead to intra-SF and inter-SF collisions. The *optimal* configuration had the lowest PLR percentage between the simulated strategies but with higher complexity compared to the proposed $TOPG$ configuration. This puts the latter as a trade-off solution between performance and computation time. Moreover, we also look towards mean PLR results in URA, RA and BE slices illustrated in **Fig. 8a**, **Fig. 8b** and **Fig. 8c** respectively. Here, URA and RA slice members requiring urgent and reliable communications are more prioritized in terms of resource reservation than the best effort slice resulting lower PLR regardless of the method adopted for SF-TP configuration. This

returns to the efficiency of the estimation method that avoids resource starvation and dynamically reserves physical channels on LoRa gateways following to the throughput requirements of each slice members. Additionally, the efficiency of the proposed configuration method can also be concluded which gave the lowest PLR with *TOPG* with a rate that did not bypass 20% in URA and RA slices and 30% in the BE slice.

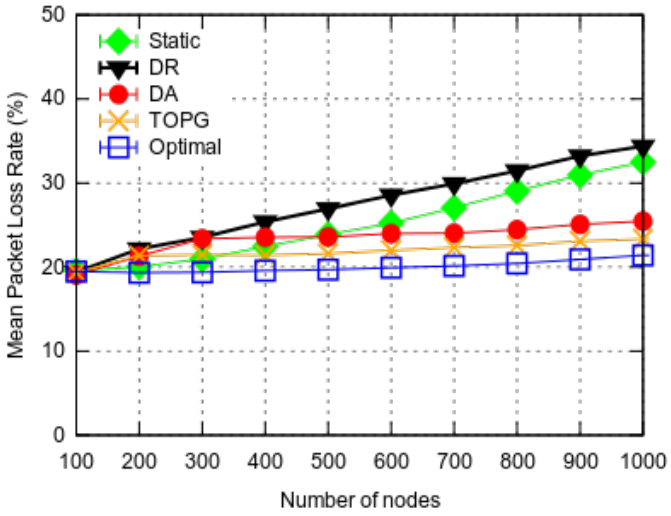
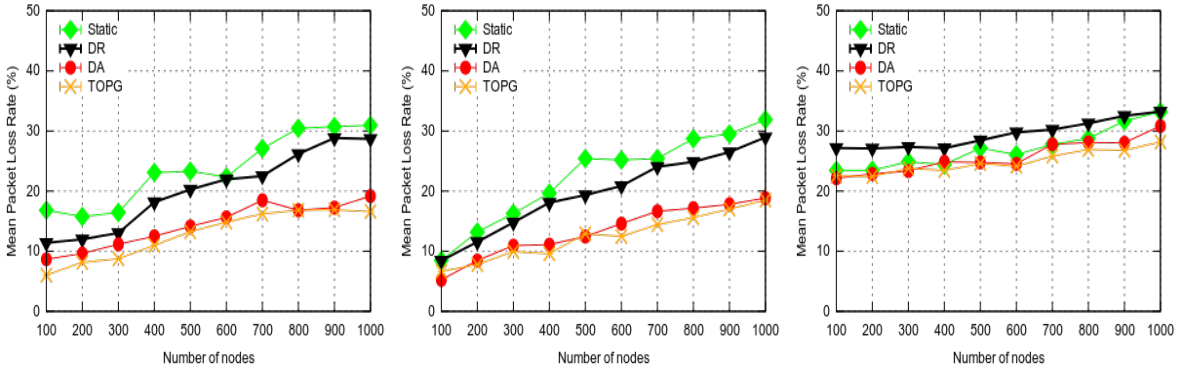


Figure 7: Packet Loss Rate Variation



(a) URA Slice (b) RA Slice (c) BE Slice

Figure 8: Mean Packet Loss Rate in each slice with different SF-TP configurations

5.2.3. Percentage of Unserved devices in Delay

In **Fig. 9**, *static* configuration had the worst results with 15% of devices that did not respect their delay thresholds. Results of *DR* and *static* were nearly similar and return to the random configuration that did not take into consideration the link quality neither QoS requirements of IoT devices. With *DA* configuration, IoT devices had much better results compared to the previous configurations with

a rate that did not exceed 10% of the devices violating their delay thresholds. However, unlike *DA* where the controller jointly increases or decreases SF-TP combination for an IoT device, *TOPG* algorithm searches for the best SF-TP combination based on the objectives and the weight defined by GMM method. *TOPG* sometimes modify TP for a device instead of increasing both SF and TP parameters like in the case of *DA* configuration. This explains the improvement and the decrease in the percentage of devices that violated their PDB in *TOPG* with less computation complexity than the *optimal* configuration.

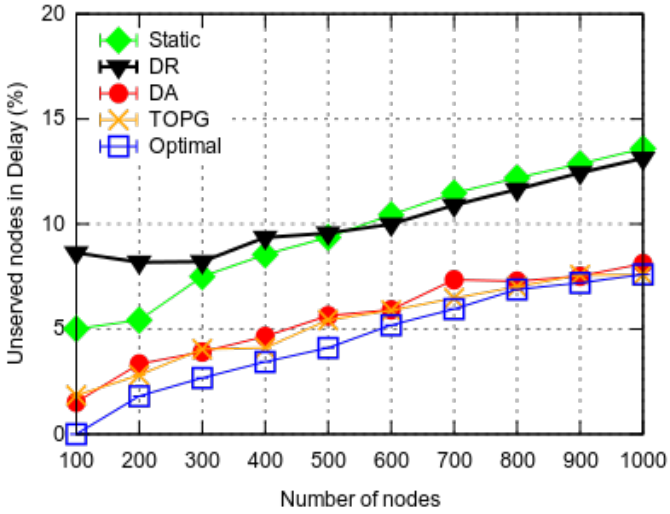


Figure 9: Percentage of Unserved nodes in Delay

5.2.4. Percentage of Unserved devices in Throughput

Further improvement in throughput is achieved in **Fig. 10** where *TOPG* came as the second best configuration method behind the *optimal* algorithm. The former scores nearly similar results with less computation time. With both *TOPG* and *optimal* algorithms, the rate of devices that did not guarantee their throughput did not exceed 30% even in a very congested scenario. This mainly highlights the efficiency of reducing alternatives in TOPSIS instead of testing all SF and TP combinations. Moreover, *static* and *DR* configurations had the worst results with a rate that exceeded 50% of the devices that violated the GBR defined in each slice. With *DA*, smaller SF values provide an achievable throughput that can be sometimes very high compared to the one that needs to be guaranteed. This is also true with smaller SF parameters where in both cases, an IoT device with *DA* configuration is assigned a specific TP for each SF parameter. However with the proposed algorithm, *TOPG* provides the guaranteed throughput with an efficient SF or TP variation. With *TOPG* optimization, a proper SF and TP combination is found that guarantees throughput while saving lots of energy for each slice members.

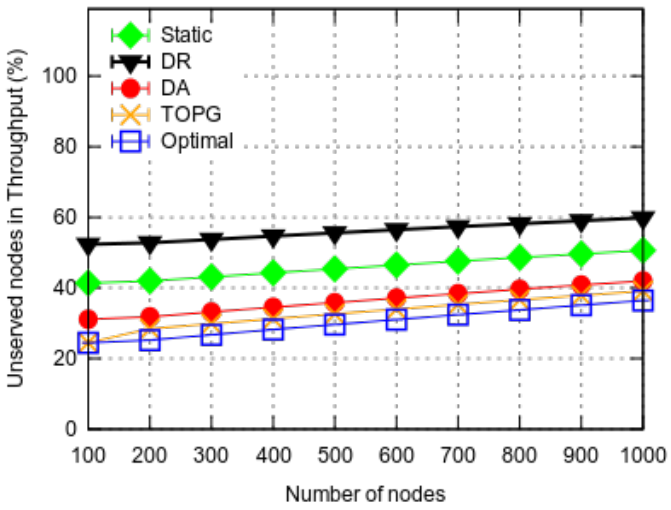


Figure 10: Percentage of Unserved nodes in Throughput

6. Conclusion

LoRa is emerging as one of the best technologies enabling LPWA for IoT networks. The goal of this paper is to evaluate how network performance can be improved in a Smart City environment in terms of QoS, energy and reliability. Therefore, we implement network slicing in LoRa to increase flexibility in the network and to enable the opportunity to consider QoS while optimizing LoRa configurable parameters. Moreover, we propose a novel configuration method that gives a trade-off solution to the multi-objective problem in a slice based manner. The proposed optimization algorithm consists of a dynamic network inter-slicing algorithm based on a maximum likelihood estimation and a slice-based SF and TP configuration which includes a combination between GMM and TOPSIS optimization algorithms. Numerical results show that our proposition outperformed static and dynamic configuration strategies, highlighted the efficiency in providing LoRa slices with a dynamic slice-based configuration and improved the performance in terms of reliability and the percentage of devices that satisfied their throughput and delay requirements.

References

1. Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., Henry, J.. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. Cisco Press; 2017.

2. Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., Watteyne, T.. Understanding the limits of lorawan. *IEEE Communications magazine* 2017;55(9):34–40.

3. Ayoub, W., Samhat, A.E., Nouvel, F., Mroue, M., Prévotet, J.C.. Internet of mobile things: Overview of lorawan, dash7, and nb-iot in lpwans standards and supported mobility. *IEEE Communications Surveys & Tutorials* 2018;.

4. Weyn, M., Ergeerts, G., Wante, L., Vercauteren, C., Hellinckx, P.. Survey of the dash7 alliance protocol for 433 mhz wireless sensor communication. *International Journal of Distributed Sensor Networks* 2013;9(12):870430.

5. Dawaliby, S., Bradai, A., Pousset, Y.. In depth performance evaluation of lte-m for m2m communications. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*. IEEE; 2016:1–8.

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

7. Ayele, E.D., Hakkenberg, C., Meijers, J.P., Zhang, K., Meratnia, N., Havinga, P.J.. Performance analysis of lora radio for an indoor iot applications. In: *2017 International Conference on Internet of Things for the Global Community (IoTGC)*. IEEE; 2017:1–8.

8. Daud, S., Yang, T.S., Romli, M.A., Ahmad, Z.A., Mahrom, N., Raof, R.A.A.. Performance evaluation of low cost lora modules in iot applications. In: *IOP Conference Series: Materials Science and Engineering*; vol. 318. IOP Publishing; 2018:012053.

9. Navarro-Ortiz, J., Sendra, S., Ameigeiras, P., Lopez-Soler, J.M.. Integration of lorawan and 4g/5g for the industrial internet of things. *IEEE Communications Magazine* 2018;56(2):60–67.

10. Luvisotto, M., Tramarin, F., Vangelista, L., Vitturi, S.. On the use of lorawan for indoor industrial iot applications. *Wireless Communications and Mobile Computing* 2018;2018.

11. Ochoa, M.N., Guizar, A., Maman, M., Duda, A.. Evaluating lora energy efficiency for adaptive networks: From star to mesh topologies. In: *Wireless and Mobile Computing, Networking and Communications (WiMob),*. IEEE; 2017:1–8.

12. Varsier, N., Schwoerer, J.. Capacity limits of lorawan technology for smart metering applications. In: *Communications (ICC), 2017 IEEE International Conference On*. IEEE; 2017:1–6.

13. Le, X.C., Vrigneau, B., Gautier, M., Mabon, M., Berder, O.. Energy/reliability tradeoff of lora communications over fading channels. In: *2018 25th International Conference on Telecommunications (ICT)*. IEEE; 2018:544–548.

14. Petäjäjärvi, J., Mikhaylov, K., Pettissalo, M., Janhunen, J., Iinatti, J.. Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage. *International Journal of Distributed Sensor Networks* 2017;13(3):1550147717699412.

15. Reynders, B., Meert, W., Pollin, S.. Range and coexistence analysis of long range unlicensed communication. In: *Telecommunications (ICT), 2016 23rd International Conference on*. IEEE; 2016:1–6.

16. Kim, D.Y., Kim, S., Hassan, H., Park, J.H.. Adaptive data rate control in low power wide area networks for long range iot services. *Journal of computational science* 2017;22:171–178.

17. Slabicki, M., Premsankar, G., Di Francesco, M.. Adaptive configuration of lora networks for dense iot deployments. In: *16th IEEE/IFIP Network Operations and Management Symposium (NOMS 2018)*. 2018:1–9.

18. Reynders, B., Meert, W., Pollin, S.. Power and spreading factor control in low power wide area networks. In: *Communications (ICC), 2017 IEEE International Conference on*. IEEE; 2017:1–6.

19. Bor, M., Roedig, U.. Lora transmission parameter selection. *13th International Conference on Distributed Computing in Sensor Systems (DCOSS)* 2017;.

20. Cattani, M., Boano, C.A., Römer, K.. An experimental evaluation of the reliability of lora long-range low-power wireless communication. *Journal of Sensor and Actuator Networks* 2017;6(2):7.

21. Li, S., Raza, U., Khan, A.. How agile is the adaptive data rate mechanism of lorawan? *arXiv preprint arXiv:180809286* 2018;.

22. Dawaliby, S., Bradai, A., Pousset, Y.. Adaptive dynamic network slicing in lora networks. *Future Generation Computer Systems* 2019;98:697–707.

23. Magrin, D., Centenaro, M., Vangelista, L.. Performance evaluation of lora networks in a smart city scenario. In: *Communications (ICC), 2017 IEEE International Conference on*. IEEE; 2017:1–7.

24. Al-Shammari, B.K., Al-Aboody, N., Al-Raweshidy, H.S.. Iot traffic management and integration in the qos supported network. *IEEE Internet of Things Journal* 2018;5(1):352–370.

25. ETSI, T.. 103 467 v1.1.1:" speech and multimedia transmission quality (stq). *Quality of Service aspects for IoT* 2018;.

26. Blenn, N., Kuipers, F.. Lorawan in the wild: Measurements from the things network. *arXiv preprint arXiv:170603086* 2017;.

27. Semtech, . Lora modulation basics. 2015. URL: `https://www.semtech.com/uploads/documents/an1200.22.pdf`.

28. Tanenbaum, A.S.. a d. wetherall. *Computer networks 5th ed Boston: Pearson* 2011;.

29. Goursaud, C., Gorce, J.M.. Dedicated networks for iot: Phy/mac state of the art and challenges. *EAI endorsed transactions on Internet of Things* 2015;.

30. Yadav, A., Jayswal, S.. Using geometric mean method of analytical hierarchy process for decision making in functional layout. *International Journal of Engineering Research and Technology (IJERT)* 2013;2:5.

31. Shih, H.S., Shyur, H.J., Lee, E.S.. An extension of topsis for group decision making. *Mathematical and Computer Modelling* 2007;45(7-8):801–813.

32. Sopin, E., Ageev, K., Markova, E., Vikhrova, O., Gaidamaka, Y.V.. Performance analysis of m2m traffic in lte network using queuing systems with random resource requirements. *Automatic Control and Computer Sciences* 2018;52(5):345–353.

33. Hamdani, , Wardoyo, R.. The complexity calculation for group decision making using topsis algorithm. In: *AIP Conference Proceedings*; vol. 1755. AIP Publishing; 2016:070007.

34. Ns-3.29 documentation. 2019.

35. Magrin, . An ns-3 module for simulation of lorawan networks. 2019. URL: `https://github.com/signetlabdei/lorawan`.

36. Bouguera, T., Diouris, J.F., Chaillout, J.J., Andrieux, G.. Energy consumption modeling for communicating sensors using lora technology. In: *2018 IEEE Conference on Antenna Measurements & Applications (CAMA)*. IEEE; 2018:1–4.

37. Haxhibeqiri, J., Van den Abeele, F., Moerman, I., Hoebeke, J.. Lora scalability: A simulation model based on interference measurements. *Sensors* 2017;17(6):1193.

38. Augustin, A., Yi, J., Clausen, T., Townsley, W.M.. A study of lora: Long range & low power networks for the internet of things. *Sensors* 2016;16(9):1466.