A Frequency-Based Intelligent Slicing in LoRaWAN with Admission Control Aspects

Ghina Dandachi, Yassine Hadjadj-Aoul

A Frequency-based Intelligent Slicing in LoRaWAN with Admission Control Aspects

Ghina Dandachi *Centre Inria Rennes - Bretagne Atlantique*
Rennes, France
ghina.dandachi@inria.fr
Yassine Hadjadj-Aoul *Centre Inria Rennes - Bretagne Atlantique*
Rennes, France
yassine.hadjadj-aoul@irisa.fr

***Abstract*—The significant deployment of LoRaWan networks is increasingly questioning its ability to handle massive numbers of IoT devices and its ability to support service differentiation. The few existing attempts to implement service differentiation suffer from a lack of scalability and do not meet the qualitative criteria of the services, since without admission control there is no way to restrain the devices from transmitting. In this paper, we present a scalable probabilistic approach that not only enables an efficient sharing of LoRaWan access networks between different services/slices, but more importantly allows achieving the objectives of the supported services through the integration of an admission control. Since the derivation of devices' repartition probabilities is a very complex problem, we propose an evolutionary algorithm to derive them efficiently. The obtained results clearly show the ability of the proposed solution to efficiently utilize the scarce radio resources, while achieving the qualitative objectives of the prioritized services.**

***Index Terms*—Slicing, LoRaWAN, Admission Control, Genetic Algorithm**

I. Introduction

Low Power Wide Area Network (LPWAN) is one of the technologies needed to cater for the evolution of the Internet-of-Things (IoT). This technology is low cost, energy-efficient and could interconnect thousands of geographically dispersed end devices running on batteries. The interconnection between these devices is made possible by different technologies, such as Long Range Wide Area Network (LoRaWAN), SIGFOX, and NB-IoT. Particularly, LoRaWAN is the most widely adopted due to its simplicity, openness, and cost-effectiveness.

With the emergence of network slicing in 5G/6G networks [1], [2], end-to-end slicing of IoT networks, and particularly LoRaWAN networks, is becoming a must. Indeed, the slicing of IoT networks would allow not only to have a support of multiple services within the same infrastructure, but more importantly to have some guarantees, in such best effort networks.

The slicing of LoRaWAN networks mainly consists in partitioning the substrate network into multiple virtual networks. While the virtualization of the core network does not present any particular difficulty, in the age of virtualization and services' cloudification, it is quite different for the access networks. Indeed, unlike traditional cellular networks where the allocation of resources is based on the assignment of resource blocks, which facilitates the operation of resources' sharing between different services, the access to LoRaWAN networks is completely random (i.e., Aloha-like), which makes it very difficult to virtualize [3].

Very few works in the literature address the problem of LoRaWAN networks' slicing [**?**], [4]–[7]. These works rely on the assignment of the available frequencies to the IoT devices, which allows improving their distribution. However, allocating an entire frequency to a slice can represent a significant waste of resources in a network where resources are very scarce. Moreover, the number of frequencies being very limited, this restricts the applicability of the approaches. Finally, the fact that no admission control is available means that no qualitative guarantees can be met in such networks.

In LoRaWAN networks, there is obviously no support to enforce service differentiation, except the allocation of frequency to the devices, which suffers from the limitations introduced earlier. There is, also, no straightforward means to control the access attempts of the IoT devices, unlike NB-IoT networks with the Access Class Barring (ACB) mechanism [8]. In this respect, we propose in this paper a probabilistic approach for the distribution of the devices between the different available channels, in a way to guarantee the quality of the supported services (i.e., slices), while maximizing the use of the resources compared to a classical solution. In addition, our proposed frequency-based approach incorporates an admission control, which constitutes a precondition for achieving the targeted quality criteria when dealing with massive number of IoT devices. The proposed probabilistic approach is based on the transmission by the LoRaWAN Network Server (LNS) of probabilities of access to frequencies (i.e., channels) as well as a probability of being blocked, which concerns only non-priority devices. The problem of assigning the different probabilities being very complex (i.e., infinite action space), we propose in this work an evolutionary algorithm to solve this problem more efficiently.

The remainder of this paper is organized as follows: Section II provides the background on NB-IoT Access class Barrier strategy, and gives an overview of the literature work on slicing on LoRaWAN. Section IV describes the system model for LoRaWAN, and provides a mathematical formulation of

INTELLIGENTSIA is a collaborative research project funded by the French Agence National pour la Recherche (ANR) public funding agency.

the problem. Section V presents the details of the proposed solution, based on the Genetic algorithm. Section VI presents the simulation environment of the proposed approach and shows its efficiency compared to the existing one. Finally, the paper concludes with a summary of the main advantages and achievements of the proposed system in Section VII.

II. Background

LoRaWAN is a Long Range communication protocol [9] often used to create Low Power Wide Area Networks (LPWANs) with an operating range that goes from 300 meters up to 10 kilometers. Among the main LPWANs protocol, LoRaWAN is one of the most known and used, given its open architecture.

LoRa modems modulate symbols into increasing and decreasing frequencies chirps, respectively called up-chirps and down-chirps. LoRa modulation has many parameters, which can partially be modified, depending also on the operating region of the system. These parameters are: carrier frequency, signal bandwidth, coding rate, spreading factor, chirp polarity, and sync word.

More precisely, the spreading factor (SF) defines how many chirps are sent per second. It ranges among SF7 and SF12. In detail, a large SF increases the symbol airtime and the energy consumption, thus improving the communication range, but reducing the available data rate and messages' payload size. In this paper, we consider that all the devices use the Adaptive Data Rate (ADR) mechanism for SF selection [10]. The LoRaWAN ADR is a scheme that allows to optimize the network's data rates, time-on-air (ToA) and energy consumption by controlling the data rate and the transmission power for all the end nodes independently.

As for the LoRaWAN end devices classes, they can be classified into three categories: Class A, B, and C. A Class A device has the lowest energy consumption mode, it can send uplink messages at any time, however downlink messages can be received only during two specific windows of time after an uplink transmission. Class B devices can send uplink messages any time, and have a time-synchronised receiving window for the downlink. Class C devices has the highest power consumption, they keep the receiving window open unless they are transmitting again which is the case of LoRaWAN gateways.

III. Related work

The need to support personalized services, with their own constraints, has led to the emergence of the concept of network slicing, which has taken shape with 5G cellular networks, and will be generalized in the next 6G to include not only the mobile core network but any other service [11].

IoT networks are very low cost and best effort networks by definition, but in which there is more and more the need to differentiate services [12]. Very few works in the literature tackle this subject, the main attempts concern NB-IoT networks [13].

Integrating slicing in IoT network requires the virtualization of two parts: the core network virtualization and the access network virtualization. With the convergence of the cloud and networking, especially with service virtualization and orchestration technologies, core virtualization has not only become possible, but several solutions already exist. The virtualization of the network access is, however, more challenging.

In M2M networks, which are at the origin of NB-IoT networks, massive access of IoT terminals has been considered very early and several efficient processes have been integrated to address this issue. The Access Class Barring (ACB) strategy is certainly one of the most prominent, as it allows blocking devices before they even have access to the network. This technique consists in assigning an access probability to the various devices, which may enable the support of slicing in this type of access network [14]. In fact, it is simply a matter of finding different access probabilities for each class of service or slice, which allows for a more refined sharing of the network [15].

In the case of LoRaWAN networks, it's a completely different situation. In fact, very few works in the literature address the problem of slicing in LoRaWAN networks [4]. This can be explained by the fact that LoRaWAN networks are unlicensed open networks where there is no technical means to limit congestion [16]. Indeed, the only options offered by LoRaWAN are to: limit the transmission of devices through the "duty cycle" mechanism, which indicates the fraction of time a resource is busy, or to assign a terminal to a particular frequency when joining the network. These options are very insufficient to mitigate congestion, and even less to have any guarantee.

The existing works are almost all from the same author [**?**], [4]–[7]. There main aspect is that the end devices are grouped based on their requirements by a 4 steps strategy: 1) a *clustering* step to determine the number of slices, 2) a *throughput estimation* step based on the average throughput of each device, 3) a *resource allocation* step by dynamically allocating the channels to gateways and 4) a final step for *devices parameters' optimization* by choosing either the spreading factor using a utility function [4], [5] or both the spreading factor and the transmission power using TOPSIS [**?**], [6]. The main drawbacks of these works is that it is based on assigning frequencies to the gateways, which is far from reality. Moreover, given the limited number of channels, assigning a whole channel to a slice does not seem very realistic. Finally, there is no guarantee to be provided in the constrained scenarios since there is no admission control.

In this work, we propose a far more efficient approach, avoiding the wastage of the radio resource. Beyond the algorithmic aspects, we propose here an approach that is simple to implement and compatible with current LoRaWAN networks, unlike existing approaches.

IV. System Model

As shown in Figure 1, LoRaWAN deploys a simple star-of-stars network topology and is composed of: end devices, centralized gateways, and a remote network server, more commonly known as the LNS which stands for LoRa Network Server. In particular, LoRaWAN end devices are energy-efficient sensors and actuators featuring low data rates ranging from 0.3 kbps to 50 kbps [9] with long-life batteries lasting up to 10 years [17].

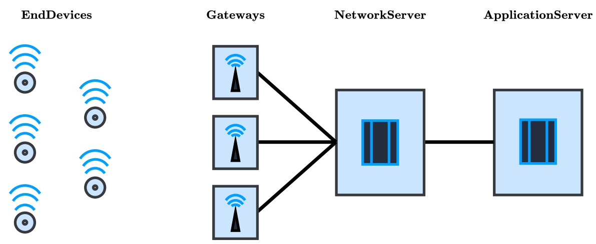


Fig. 1: LoRaWAN architecture.

Although our approach can be easily generalized to multiple gateways, for the sake of simplicity, we assume in the following a network with a single gateway. Besides, the same assumptions as in [18] are adopted: (1) a uniform distribution of the devices, (2) devices transmitting packets of the same duration T_f for a spreading factor (SF) f, (3) packets transmission following a Poisson process, and (4) no inter-SF collisions. In such a situation, the probability of a successful delivery called Packet Data Rate (PDR), can be given by [18]:

$$P_f(x) = \exp^{-2T_f \theta N_f \frac{(\min(xR,d))^2}{d^2}}, \tag{1}$$

where x is the distance of the end device to the gateway, θ is the intensity of packets transmission in packets/sec, N_f is the number of nodes with SF $f \in \{1, \cdots, N_{SF}\}$, and d is the range of deployment for the given SF. xR represents the range of devices that may interfere with the current communication, where $R = 10^{\frac{P_{\text{thld}}}{10\gamma}} > 1$, with γ the path loss exponent and P_{thld} the minimum threshold of the received energy. d being the maximum distance of a device using a given SF when applying ADR technique for SFs selection. Thus, the smaller the distance x to the gateway, the higher the probability $P_f(x)$ of successful delivery due to the reduced number of interferers.

A. PDR evaluation

We consider, in the following, the case of two slices/classes of services, a priority class and a best effort (BE) class, with a probability p for a device to belong to the BE class. We also consider the general case of K frequencies, which is for example equal to three (i.e., 383.1, 383.3, and 383.5 MHz) in the European standards.

We define the probability for a best effort device, respectively priority device, to select a frequency i on which it transmits as p^i_{BE}, respectively p^i_{Prio}, with $i \in \{1, \cdots, K\}$. Thus, the number of end devices using the SF f at the frequency i is:

$$N^i_f = N_f((1-p)\, p^i_{\text{Prio}} + p\, p^i_{\text{BE}}) \tag{2}$$

We assume that the devices are choosing the SF based on their geographical location or using the Adaptive Data Rate (ADR) mechanism. For example, the devices that are located at a distance where the SF 7 is used, will be using the SF 7 regardless of their service class. In order to evaluate the PDR for both priority and BE classes, we start by evaluating for each device the eventual PDR when the IoT device m connects using a frequency i. For this aim, we define the PDR per device/per frequency as follows:

$$PDR^i_m = \exp^{-2T_f \theta N^i_f \frac{(\min(xR,d))^2}{d^2}} \tag{3}$$

Then, the average PDR per frequency, PDR^i is the same for priority and BE classes, and is calculated as the average of the PDR of all devices connected to this frequency:

$$PDR^i = \frac{1}{N^i_f} \sum_{m=1}^{N^i_f} PDR^i_m \tag{4}$$

Finally, the PDR for priority and BE end devices are evaluated as follows:

$$PDR_{\text{Prio}} = \sum_{i=1}^{K} p^i_{\text{Prio}} PDR^i \tag{5}$$

$$PDR_{\text{BE}} = \sum_{i=1}^{K} p^i_{\text{BE}} PDR^i \tag{6}$$

B. Model with admission control

In this section, we introduce the admission control probability for BE end devices. For this reason, we redefine the number of devices per SF, per frequency as follows:

$$N^i_f = N_f((1-p)\, p^f_{\text{Prio}} + p\, p^i_{\text{BE}}\, p^i_{\text{ac}}) \tag{7}$$

with p^i_{ac} being the probability of blocking BE devices at the frequency i.

Thus, the PDR for BE end devices is evaluated as follows:

$$PDR_{\text{BE}} = \sum_{i=1}^{K} p^i_{\text{BE}}\, p^i_{\text{ac}}\, PDR^i \tag{8}$$

In practice, this admission control parameter will be applied at the end device side before transmitting a packet. The devices select a random number between 0 and 1, and only transmits if the number is higher than the admission control probability for the frequency to be used.

C. Parameters' communication

Our proposal consists of a centralized estimation of the frequency selection probabilities. The network server, will learn, based on the PDR achieved by different end devices from different classes of priority, the optimal values that will maximize the target PDR.

The optimal values need to be actively requested by the end device. This request/response process involves a two-messages protocol, described below, between the end device and the network server.

1) Access parameters' update request: Initially, an end device sets up an over the air activation (OTAA) before transmitting, with assigning equal probabilities for frequency selection, and admission control probabilities equal to 1. This means that all the devices have equal chances to transmit on any frequency, and all low priority devices will be allowed to transmit their packets without any constraints on any selected frequency.

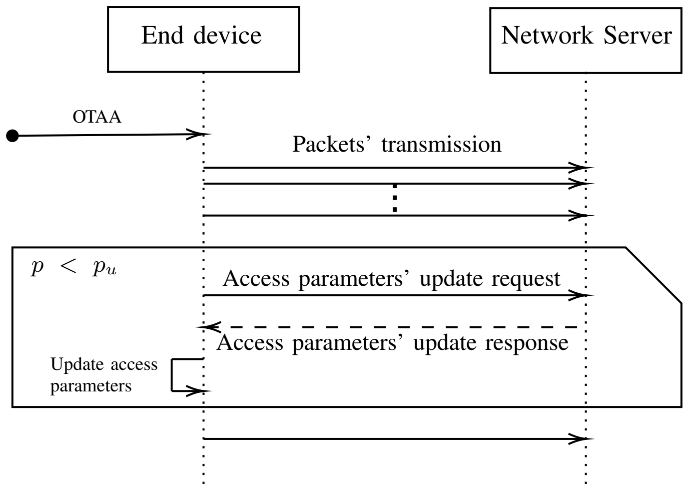


Fig. 2: Update request/response communication between the end device and the network server.

Payload	Size (Bytes)
CID (0xAA)	1
UReq	1

TABLE I: MAC Command: Update Request

The *update request* mechanism is randomly triggered by an end device with a probability p_u, which is equal to 0.1 in our solution. Thus, in average, an update request will be triggered every 10 packets. The network server will gather information about other end devices' performance, and optimize the value of the frequency selection and admission control probabilities as a function of the service level agreements of the supported services/classes.

The communication between an end device and the network server is illustrated in Fig. 2.

2) Access parameters' update response: The implementation of the *update request/response* is defined by two MAC commands, following the LoRaWAN specifications. The proposed commands are smaller than 15 Bytes in order to be piggybacked with the application data. The transmission of the update request is done by the end device by taking 2 bytes. The update response is transmitted by the network server using $7Bytes$, the frequency probabilities of the end device class and their admission control probabilities.

This proposition is easily generalized, since each device will receive the information related to the priority class to which it belongs, meaning that it is possible to have a large number of classes and management strategies to communicate without having to increase the size of the update response.

The *Update Request* message command syntax can be seen in Table I. The command will trigger the parameters' optimization at the network server side.

The update Response message command syntax is presented in Table II.

The probability values are coded on 1 bytes, i.e. 8 bits, using a Mini-float representation technique, no sign bit required, and with a bias of -7 which will allow to have a wide range of numbers between 0 and 1. An exponent of 3 bits is considered,

Payload	Size (Bytes)
CID (0xAA)	1
P_{f1}	1
P_{f2}	1
P_{f3}	1
P_{ac1}	1
P_{ac2}	1
P_{ac3}	1

TABLE II: MAC Command: Update Response (case of three frequencies)

and a significand of 5 bits.

V. ML-based probabilities calculation

The Genetic algorithm (GA) [19] is a population based search algorithm. GA is a well-known meta-heuristic algorithm inspired from biological evolution process, which utilizes the concept of survival of the fittest. It consists on using genetic operators on individuals present in the population in order to produce new populations. These elements/operators are:

- Chromosome representation
- Chromosome selection
- Chromosome crossover
- Chromosome mutation
- Fitness function

In this paper, we consider that the population consists of N chromosomes, and is defined as $\mathcal{C} = \{c_1, c_2, \cdots, c_N\}$ with a chromosome $c_n = [p^1_{Prio}, \cdots, p^K_{Prio}, p^1_{BE}, \cdots, p^K_{BE}, p^1_{ac}, \cdots, p^K_{ac}]$. We consider that $p^k_{Prio} \in [0,1]$, $p^k_{BE} \in [0,1]$, and $p^k_{ac} \in [0,1]$ in order to obtain optimal solutions for the probability between 0 and 1.

The procedure of GA is described in Algorithm 1. First, a population $\mathcal{C}^0$ of size N chromosomes is randomly initialized. The fitness of each chromosome is calculated using the fitness function defined in Eq. 9. The first two chromosomes of highest fitness values are selected as the parents c^0_j and c^0_l for generating the next population. The genes of the parents are crossed-over using a single point crossover strategy. Using these parents a new generation is created by applying a mutation on the offspring calculated for the parents. The new population will replace the old population. This process is repeated until the best solution is found or the maximum number of generations is reached.

The fitness function *fitness* we use for this problem is defined as follows:

$$\textit{fitness} = \frac{1}{1 + |\textit{PDR}_{\text{Prio}} - \textit{PDR}^{\text{target}}_{\text{Prio}}|} \tag{9}$$

This fitness function ensures a good fitness if the priority class devices reach their target PDR, and a low fitness if it is not reached (< 1). Moreover, this fitness function has also lower value if the PDR of priority class devices is higher than the target PDR. This ensures an optimal PDR for the priority class devices and avoids the PDR degradation of BE class devices once the target PDR of priority class is reached.

Algorithm 1: Genetic Algorithm (GA).

Data: Population $\mathcal{C}$ size, N;
Maximum number of iterations, MAX.
Result: Global best solution, c_{best}.
Begin:
Generate initial population of N chromosomes $\mathcal{C}^0 = \{c_1^0, c_2^0, \cdots, c_N^0\}$ $(i = 1, 2, \cdots, n)$;
Set iteration counter $it = 0$;
Compute the fitness value of each chromosomes;
while $it < MAX$ **do**
  Select a pair of chromosomes, c_j^{it} and c_l^{it}, from initial population $\mathcal{C}^{it}$ based on their fitness;
  Apply crossover operation on selected pair with crossover probability;
  Apply mutation offspring with mutation probability;
  Replace old population with newly generated population;
  Increment the current iteration it by 1;
end
Return the best solution c_{best};

The main advantage of using the genetic algorithm to solve this problem is that it allows to explore a very large set of solutions while providing a solution that is globally optimal.

VI. Evaluation

In this section, we evaluate the proposed slicing strategy in LoRaWAN using the genetic algorithm. The evaluation consists of comparing the PDR of both priority classes for different PDR targets for the high priority class.

A. Simulation setup

Our proposal is evaluated in a simulator written in Python. The model parameters are as follows:

- Total number of devices: $\{200, 400, 600, 800, 1000\}$
- The SFs maximum distance from a gateway following ADR strategy for SF selection: $d = \{2200, 2500, 3000, 3700, 4100, 6400\}$ [m] corresponding to SFs $\{7, 8, 9, 10, 11, 12\}$, respectively.
- Time on Air for each SF: $\{79.1, 147.97, 275.46, 509.95, 1101.82, 2039.81\}$ [ms] starting from SF7 and ending with SF12.
- Packets transmission intensity: $\theta = 1/1200$ [packets/sec].
- Path loss Exponent: $\gamma = 2.08$
- Power threshold: $p_{thld} = 6$
- Probability of BE devices: $p = 0.7$
- Number of frequencies: $K = 3$

As for the genetic algorithm, we consider 200 generations of 20 individuals/chromosomes per population. A chromosome is composed of 9 genes. Two parents are selected from each generation using a tournament parent selection, and a single point crossover. A random mutation is used with 20% for genes mutation. We use the PyGAD Python library for building the genetic algorithm and optimizing its results [20].

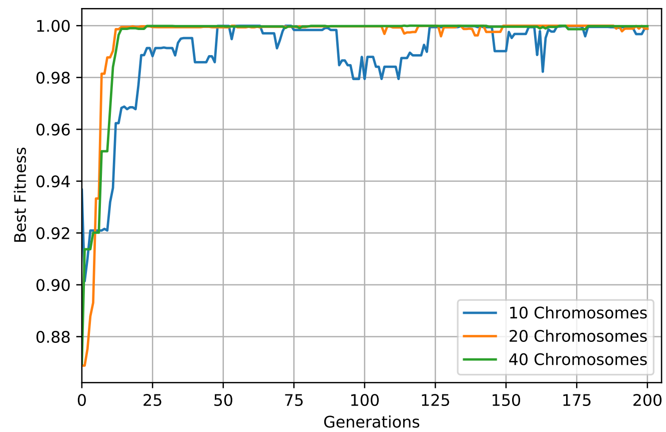


Fig. 3: Fitness function variation as a function of the generation number.

B. General performance of the genetic algorithm

In this section, we present the simulation performance of the genetic algorithm for a target PDR of 0.7 for the priority class of end devices. We compare the best fitness value as a function of the generations for a population of 10, 20, and 40 chromosomes. Fig. 3 shows that it is possible to reach an optimal solution for 20 chromosomes with a comparable performance with 40 chromosomes. The GA starts to find an optimal solution almost around the 15^{th} generation.

C. Comparison of different desired PDR output

In Fig. 4, we evaluate the PDR of BE and priority devices as a function of the number of devices in the system, for different targets of PDR (70%, 80%, 90%).

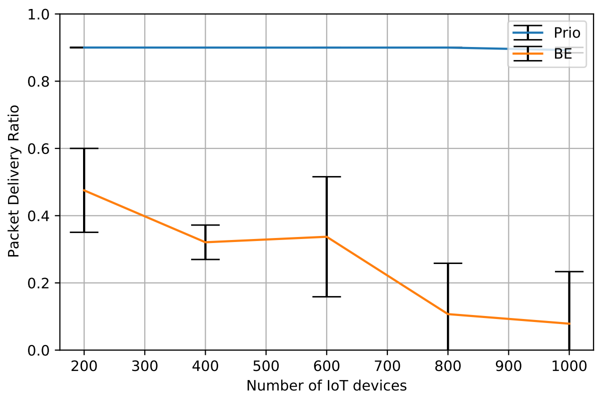
We observe in Fig. 4a that the priority class achieves the target PDR of 90%. The BE class presents a very low PDR, starts with 45% for 200 IoT devices and decreases to 9% for 1000IoT devices. This is explained by an increase in the number of BE blocked by the admission controller, see Fig. 4b.

We show in Figs. 4c and 4d the PDR and the number of blocked BE devices for a priority devices target PDR of 80%. We observe that the proposed genetic algorithm ensures that the priority devices achieve their target PDR, while trying to maximize the BE class PDR. The number of blocked devices has been reduced from 630/700 to 400/700, where 700 is the total number of BE devices (70% of total devices).

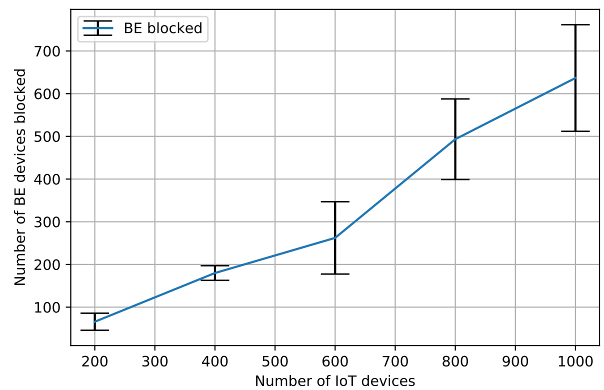
Finally, for a target priority class PDR of 70%, the performance of BE devices improves for a number of devices lower than 600, see Figs. 4e and 4f. As for higher than 600 devices, the obtained performance of BE devices is similar to that obtained with a target of 80%. The reason behind this performance could be the blockage probability for BE devices.

D. Comparison of performance under different problem settings

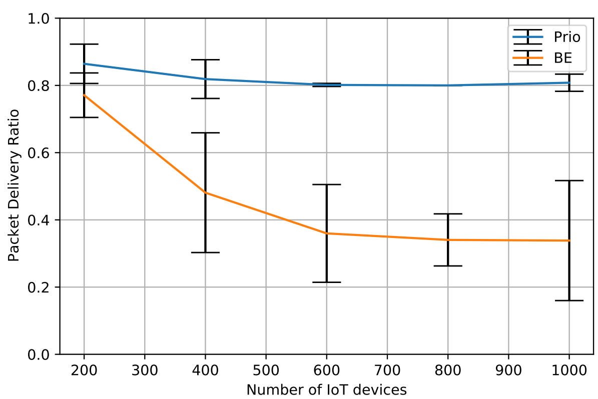
In this section, we perform a comparison of the packet delivery rate as well as the number of blocked devices under



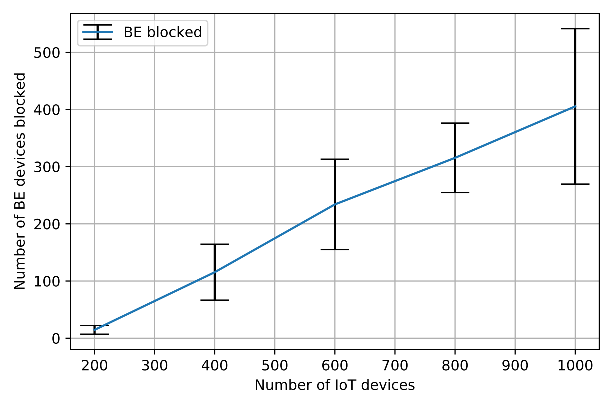
(a) PDR per slice for a target PDR=90%



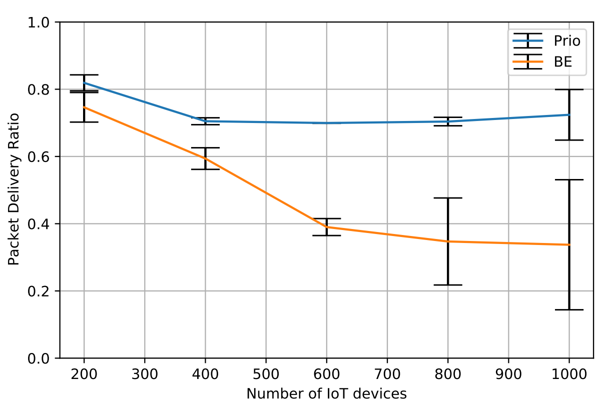
(b) Number of blocked IoT devices from the low priority slice for a target PDR=90%



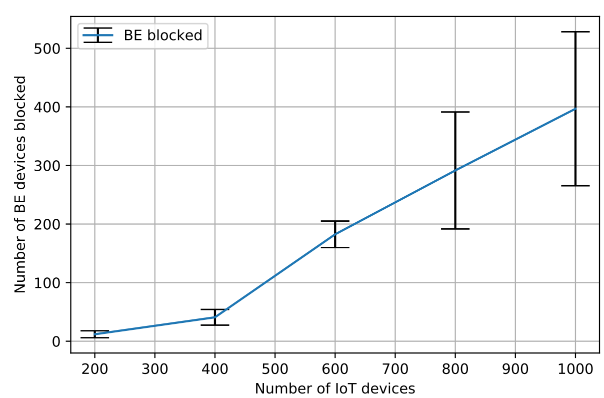
(c) PDR per slice for a target PDR=80%



(d) Number of blocked IoT devices from the low priority slice for a target PDR=80%

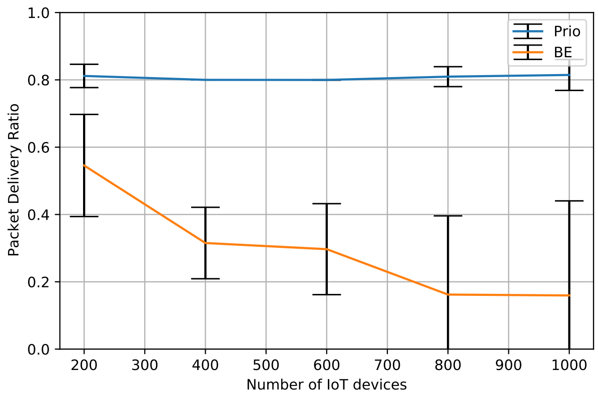


(e) PDR per slice for a target PDR=70%

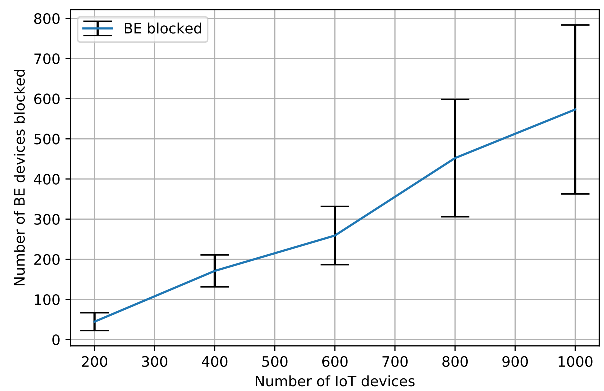


(f) Number of blocked IoT devices from the low priority slice for a target PDR=70%

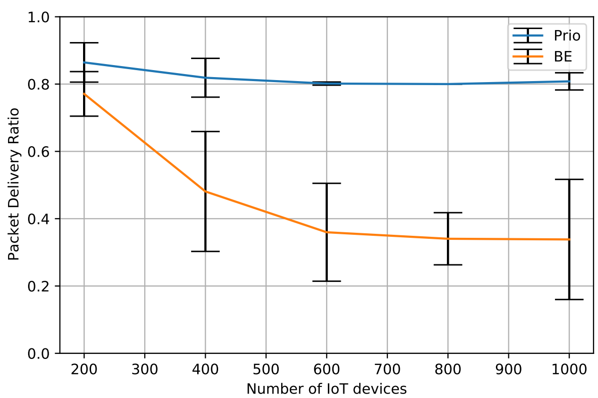
Fig. 4: Variation of PDR and Number of blocked devices as a function of the target PDR for priority slice.



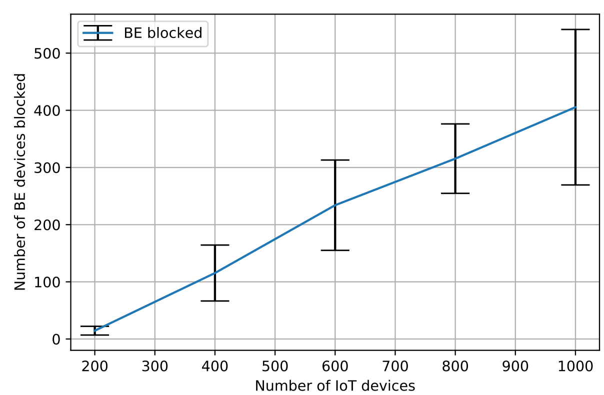
(a) PDR per slice, $\theta = 1/600$ [packets/sec]



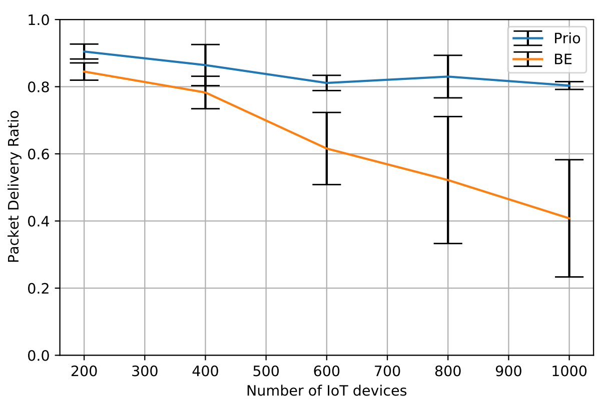
(b) Number of blocked IoT devices from the low priority slice, $\theta = 1/600$ [packets/sec]



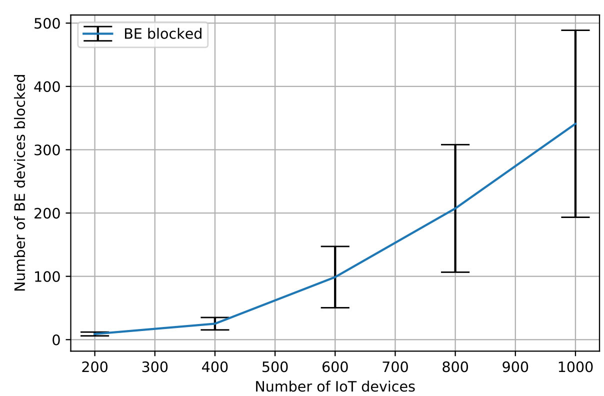
(c) PDR per slice, $\theta = 1/1200$ [packets/sec]



(d) Number of blocked IoT devices from the low priority slice, $\theta = 1/1200$ [packets/sec]



(e) PDR per slice, $\theta = 1/2400$ [packets/sec]



(f) Number of blocked IoT devices from the low priority slice, $\theta = 1/2400$ [packets/sec]

Fig. 5: Variation of PDR and Number of blocked devices as a function of the packets intensity, for a target PDR of 80% for priority slices.

different problem settings: the case of low charged, medium charged and highly charged systems. For this aim, we modify in this case the intensity of packets transmission θ as follows:

- Low charged: $\theta = 1/1800$ [packets/sec]
- Medium charged: $\theta = 1/1200$ [packets/sec]
- Highly charged: $\theta = 1/600$ [packets/sec]

For the following, we consider a target PDR for priority class of 80%.

We first start with the results of a highly charged system for a packets intensity of $1/600$ [packets/sec]. In Fig. 5a, we observe that the priority class devices achieve their target PDR, while the BE class devices get their PDR reduced as the number of devices increase, with a percentage of blocked BE devices over the total number of BE devices varying $10/140$ to $570/700$ (Fig. 5b).

As the packets intensity decreases, we observe that the PDR of BE increases. The system is less saturated and allows for a better service for BE devices while maintaining the target PDR for priority devices. The number of blocked BE devices also drops from a maximum of $570/700$ for $\theta = 1/600$, to $400/700$ for $\theta = 1/1200$, to $340/700$ for $\theta = 1/2400$.

VII. Conclusion

In this paper, we consider the problem of slicing in LoRaWAN for 5G/6G networks. We introduce an empirical formulation for the problem of network slicing, which ensures the access slicing by using a probabilistic model for the frequencies selection. An admission control probability is also introduced in order to control the back-off time for the lowest priority class. An evolutionary algorithm was used to find the optimal values for the frequency selection probabilities as well as the admission control probabilities. We show that it is possible, using the optimized probabilities for frequency selection and admission control, to achieve a target PDR for a given class of devices. It is possible to generalize this problem to the case of a higher number of classes of priority with different admission control strategies for different classes.

References

[1] A. Kaloxylos, "A survey and an analysis of network slicing in 5g networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, 2018.

[2] S. Zhang, "An overview of network slicing for 5g," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.

[3] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "Lora-mab: Toward an intelligent resource allocation approach for lorawan," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[4] S. Dawaliby, A. Bradai, Y. Pousset, and R. Riggio, "Dynamic network slicing for lorawan," in *2018 14th International Conference on Network and Service Management (CNSM)*, 2018.

[5] S. Dawaliby, A. Bradai, and Y. Pousset, "Adaptive dynamic network slicing in lora networks," *Future generation computer systems*, vol. 98, pp. 697–707, 2019.

[6] ——, "Network slicing optimization in large scale lora wide area networks," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019.

[7] ——, "Distributed network slicing in large scale iot based on coalitional multi-game theory," *IEEE Transactions on Network and Service Management*, 2019.

[8] M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar, and G. Rubino, "Estimating the number of contending iot devices in 5g networks: Revealing the invisible," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, p. e3513, 2019, e3513 ett.3513. [Online]. Available: https://onlinelibrary-wiley-com.passerelle.univ-rennes1.fr/doi/abs/10.1002/ett.3513

[9] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[10] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, "Adaptive data rate control in low power wide area networks for long range iot services," *Journal of Computational Science*, vol. 22, pp. 171–178, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877750317304428

[11] W. Guan, H. Zhang, and V. C. M. Leung, "Customized slicing for 6g: Enforcing artificial intelligence on resource management," *IEEE Network*, vol. 35, no. 5, pp. 264–271, 2021.

[12] S. Wijethilaka and M. Liyanage, "Survey on network slicing for internet of things realization in 5g networks," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, pp. 957–994, 2021.

[13] M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar, G. Rubino, and S. Tabbane, "Multiple access class barring factors algorithm for m2m communications in lte-advanced networks," in *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 195–199. [Online]. Available: https://doi.org/10.1145/2811587.2811624

[14] Y. H. Aoul and S. A. Chellouche, "Access control in nb-iot networks: A deep reinforcement learning strategy," *Inf.*, vol. 11, no. 11, p. 541, 2020. [Online]. Available: https://doi.org/10.3390/info11110541

[15] M. Bouzouita, Y. Hadjadj-Aoul, N. Zangar Tunisia, G. Rubino, and S. Tabbane, "Dynamic adaptive access barring scheme for heavily congested m2m networks," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 287–291. [Online]. Available: https://doi.org/10.1145/2988287.2989174

[16] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of lorawan channel access," in *2016 International Conference on Engineering and Telecommunication (EnT)*, 2016, pp. 10–14.

[17] L. Alliance, "Lorawan what is it?-a technical overview of lora and lorawan lora alliance," 2015.

[18] D. Zorbas, C. Caillouet, K. Abdelfadeel Hassan, and D. Pesch, "Optimal data collection time in lora networks—a time-slotted approach," *Sensors*, vol. 21, no. 4, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1193

[19] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021.

[20] A. F. Gad, "Pygad: An intuitive genetic algorithm python library," 2021.