



HAL
open science

LoRa-MAB: A Flexible Simulator for Decentralized Learning Resource Allocation in IoT Networks

Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cédric Adjih, Steven Martin

► **To cite this version:**

Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cédric Adjih, Steven Martin. LoRa-MAB: A Flexible Simulator for Decentralized Learning Resource Allocation in IoT Networks. WMNC 2019 - 12th IFIP Wireless and Mobile Networking Conference, Sep 2019, Paris, France. pp.55-62, 10.23919/WMNC.2019.8881393 . hal-02431653

HAL Id: hal-02431653

<https://inria.hal.science/hal-02431653v1>

Submitted on 8 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LoRa-MAB: A Flexible Simulator for Decentralized Learning Resource Allocation in IoT Networks

Duc-Tuyen Ta^{1,4}, Kinda Khawam², Samer Lahoud³, Cédric Adjih⁴, and Steven Martin¹

¹ROCS, LRI, Université Paris Saclay, 91190 Gif-sur-Yvette, France

² Université de Versailles Saint-Quentin-en-Yvelines, 78000 Versailles, France

³ Faculté d'ingénierie ESIB, Université Saint-Joseph de Beyrouth, Lebanon

⁴ Inria, CRI Saclay-Île-de-France, 91120 Palaiseau, France

Abstract—LoRaWAN is a media access control (MAC) protocol for wide area networks. It is designed to allow low-powered devices to communicate with Internet-connected applications over long-range wireless connections. The targeted dense deployment will inevitably cause a shortage in radio resources. Hence, autonomous and lightweight radio resource management is crucial to offer ultra-long battery lifetime for LoRa devices. One of the most promising solutions to such a challenge is the use of artificial intelligence. This will enable LoRa devices to use innovative and inherently distributed learning techniques, thus freeing them from draining their limited energy by constantly communicating with a centralized controller.

Before proceeding with the deployment of self-managing solutions on top of a LoRaWAN application, it is sensible to conduct simulation-based studies to optimize the design of learning-based algorithms as well as the application under consideration. Unfortunately, a network simulator for such a context is not fully considered or lacks real deployment parameters. In order to address this shortcoming, we have developed a LoRaWAN simulator which aims for resources allocation problem in LoRaWAN network. The Multi-Armed Bandit and its reinforcement learning based algorithm are used to formulate and finding a resource allocation solution. To demonstrate the usefulness of our simulator, extensive simulations were run in a realistic environment taking into account physical phenomenon in LoRaWAN such as the capture effect and inter-spreading factor interference. The simulation results show that the proposed simulator provides a flexible and efficient environment to evaluate various network design parameters and self-management solutions as well as verify the effectiveness of the distributed learning algorithms for resource allocation problems in LoRaWAN.

Index Terms—LoRAWAN, network simulator, reinforcement learning, discrete-event simulator.

I. INTRODUCTION

The deploying of the Internet of Things promises to provide a low-cost, large-scale and ultra-durable connectivity for everything which can benefit from being connected. LoRa and its network architecture LoRaWAN [1]–[3] is currently the most promising IoT solution over the unlicensed band with a simplified connectivity procedure over long-range wireless connections. LoRaWAN has unique particularities, including chirp spread-spectrum modulation technique, regulatory limitations on radio duty cycle, and the use of ALOHA protocol. The start network topology is used for transmitting data, in which a gateway relays messages between a network server (NS) and end-devices. The transmission between end-devices and the gateway is possible on one of the available sub-channels

and with the one of the 6 spreading factors. A collision occurs when two or more LoRa transmissions overlap at the receiver. A collision is provoked by the selection of the same channel and spreading factor (SF) by different devices, with timing overlap [4]. In addition, even with different SFs, a collision between signals on the same sub-channel will occur due to the imperfect orthogonality of SFs, called inter-SF collision [5], [6]. Conversely, if there are concurrent transmissions on the same radio resources (the same SF and sub-channel), the gateway is able to successfully receive one of them if its Signal-to-Interference-and-Noise-Ratio (SINR) is higher than a threshold value, *e.g.* 6 dB, for any SF. The latter is deemed capture effect (CE) [3], [7] and will be also accounted for.

Before proceeding with the deployment of the IoT application on top of LoRaWAN, it is sensible to conduct simulation-based studies to evaluate the network design as well as protocol parameters for the considered application. Since selecting the same SF and sub-channel in LoRaWAN is inevitable due to the shortage in radio resources, it is necessary to evaluate the network design, under the impact of capture effect and inter-SF collision, to minimize the number of collisions. Therefore, a LoRaWAN network simulator must be deployed to provide early insight into different physical aspects that can affect network performances.

In LoRaWAN, each device must be able to operate autonomously by using innovative and inherently distributed learning techniques to select radio resources, as it is not reasonable to assume that they can communicate frequently with the NS, given their strict energy and capacity constraints. Moreover, in a massive deployment, the NS will not be able to manage this access by respecting strict deadlines. Therefore, completely self-managed solutions to select radio resources with real deployment parameters must be assessed through an adequate network simulator.

In this paper, we develop a lightweight and flexible network simulator, called LoRa-MAB, in order to investigate the performance of resource allocation in LoRaWAN through simulation. Furthermore, we resort to the popular EXP3 (Exponential Weights for Exploration and Exploitation) algorithm to steer autonomously the decision of LoRa end-devices towards the most suitable resources (*e.g.* spreading factors, sub-channels).

The contributions of the paper are the following:

- *LoRaWAN Simulator*: we develop a flexible, open-source

simulation framework, called LoRa-MAB, in Python with Simpy library [8]. This framework captures specific LoRa link behavior for multiple network settings with the impact of the capture effect and inter-SF collision. The simulation framework can be found at <https://github.com/tuyenta/IoT-MAB>.

- *Distributed Learning Algorithm*: we deploy a distributed learning algorithm for radio resources selection in LoRaWAN. The goal is to steer autonomously the decision of each end-device towards the most adequate radio resources ensuring reactivity to the possible changes that can occur in the common resource usage.
- *LoRaWAN Evaluation*: we carry out a thorough evaluation of network performances by using the proposed simulator. We show that the distributed learning approach is much more efficient in minimizing the number of collisions in comparison with a uniform distribution or a trivial random distribution over the set of resources.

The rest of the paper is organized as follows. The related work is sketched in Section II. The simulation framework, followed by the implementation of distributed learning solutions, are presented in Section III and Section IV. Performance of LoRaWAN network with distributed learning solutions is evaluated in Section V by using the proposed simulator. The conclusion and future work are presented in Section VI.

II. RELATED WORKS

The most popular simulator for LoRaWAN is LoRaSim [4], which is a discrete-event simulator based on Python with SimPy simulation library [8]. This simulator implements a radio propagation model based on the well-known log-distance path loss model. The collision between transmission packets in LoRaWAN is then evaluated through some experiments with given radio settings such as spreading factor (SF), bandwidth (BW) and code rate (CR). At the end of a simulation, the simulator reports the corresponding packet delivery ratio and total energy consumption of the network. The LoRaSim-based simulators extend the simulator for multiple IoT applications by adding the application data generation model along with data packet size [9] or for the specifications of LoRaWAN in the United States [10]. However, such simulators are limited due to some fixed radio settings while realistic physical setting (such as inter-SF interference) is not considered. More importantly, no self-managing solution is implemented.

Similar simulators for LoRaWAN are based on the NS-3 network simulator [11] or the OMNeT++ network simulator [12]. However, distributed learning is not considered.

Recent work on distributed learning for radio resources allocation in LoRaWAN had recourse to Multi-Armed Bandit (MAB) problem [13], [14]. Each end-device is considered as an intelligent agent that chooses a given SF and/or channel to improve the success transmission rate [13] or the reliability and energy-efficiency tradeoff [14], through an adequate reward process. In [13], the authors assumed that all end-devices use the same SF and adopted the stochastic MAB algorithm

to determine the frequency selection. However, such an assumption is impractical in reality due to the mutual coupling between multiple intelligent end-devices. The work in [14] has presented stochastic and adversarial based distributed learning algorithms for resource allocation in an IoT network. However, the capture effect and inter-SF interference are overlooked. In addition, the committed simulators, which are based on Matlab, are not truly network simulator and maybe not able to completely simulate the real operations of LoRaWAN.

III. THE SIMULATION FRAMEWORK

LoRaWAN [1] is the network architecture designed for wireless connections operated on LoRa technology. It specifies different types of devices, resources, transmission protocols, and encryption methods to build a secure wireless network in long-range with low-power consumption. Depending on the geographical region, communications in LoRaWAN occur in one of the sub-channels in the ISM band, with a bandwidth of 125 kHz, 250 kHz or 500 kHz. Towards this end, the Chirp Spread Spectrum (CSS) modulation has been used, which enables signals with different spreading factors to be distinguished and received simultaneously, even if they are transmitted at the same time and on the same channel. Besides selecting a spreading factor and sub-channel for packet transmission, each end-device selects a transmission power level up to 20 dBm. Further, a duty cycle restrictions of 1% or 0.1% are imposed for preventing any end-device from sending data too often in order to make room for the other end-devices.

A. Simulation Model

We consider a LoRaWAN-like network composed of one or more gateways, each located at the center of a disc-shaped network of radius R , with N end-devices uniformly distributed. For simplicity, we assume the functions of NS is embedded in the gateway. Thus, there are 3 components: end-devices, propagation model and gateways. Depending on the radio resources selection method, we separate the set of end-devices into 2 kinds: i, the normal end-device which selects the resources by following a uniform or a trivial distribution, and ii), the smart end-device which selects the resources by resorting to a learning algorithm. The model of LoRaWAN-MAB for simulating the operation of LoRaWAN is then presented in Figure 1 with the input parameters and the structure of the simulator.

B. System Parameters

1) Radio Resources:

- **Spreading Factors**: the spreading factor (SF) can be selected from 7 to 12. The higher the SF, the higher the signal-to-noise ratio, the sensitivity and range, but also the time on air (channel occupation time).
- **Channels**: depend on world region, LoRa communication can operate in the 433MHz and 868Mhz ISM frequency bands or the 900MHz band.
- **Transmission Power Level**: in LoRa communication, each end-device can adjust its transmission power from -4dBm to 20dBm , in 1dBm step.

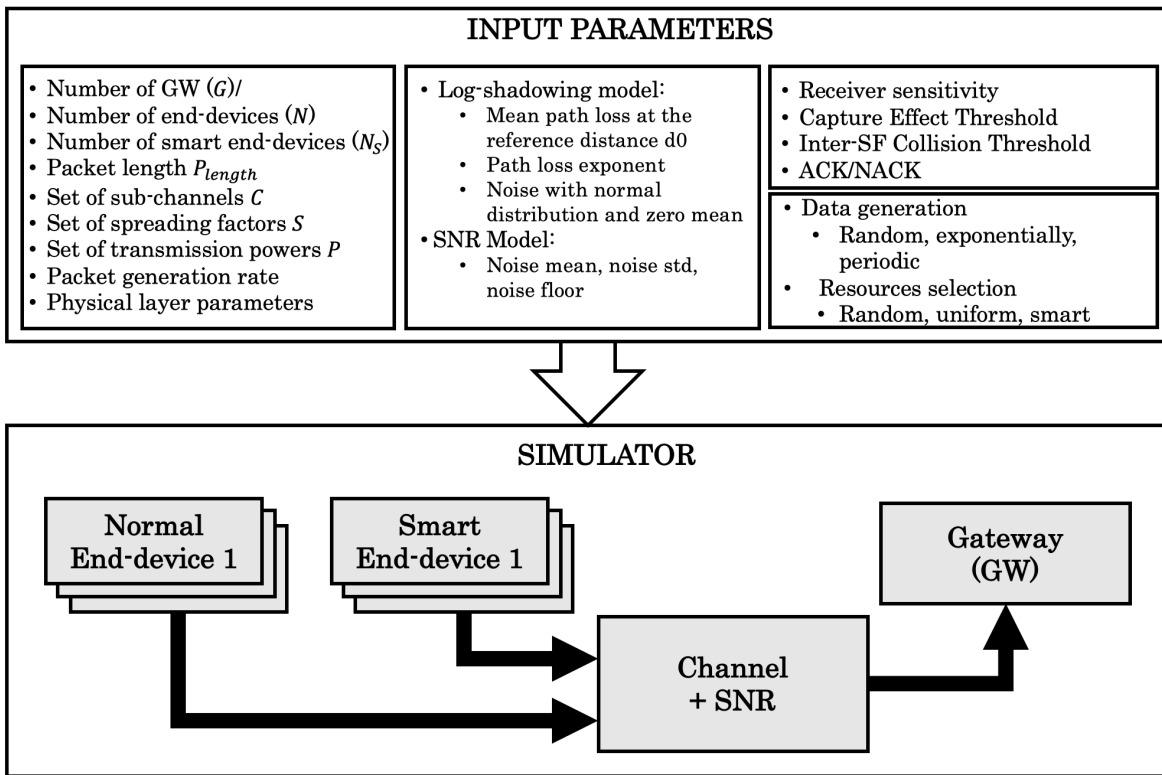


Fig. 1: Simulation model of LoRaWAN network.

- **Coding Rate (CR):** Forward Error Correction code is used to protect LoRa communication against interference. CR can be set either to 4/5, 4/6, 4/7 or 4/8. A higher CR offers more protection but increases time on air.

2) *Bandwidth:* LoRa communications can operate in the bandwidth (BW) of 125kHz, 250kHz or 500kHz. Higher BW gives a higher data rate (thus shorter time on air), but a lower sensitivity (because of integration of additional noise). Conversely, a lower BW gives a higher sensitivity, but a lower data rate.

3) *Packet Length:* The length of the transmitted packet for each end-device. We set the default packet length to 50 bytes.

4) *Duty Cycle Restriction and Packet Generation Rate:* Let T_s be the time necessary to transmit a packet of l bytes on spreading factor s . Then, given a duty cycle limitation of δ , the packet generation rate for each end-device operating on SF s , denoted by λ_s , must verify $\lambda_s \cdot T_s \geq \delta$ [1].

5) *Propagation Model:* We implement a radio propagation model based on the well-known log-distance path loss model, which is presented in [4], [15]. Using this model, the path loss $L_{pl}(d)$ that depends on the communication distance d can be described as:

$$L_{pl}(d) = \bar{L}_{pl}(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + X_\sigma \quad [dB], \quad (1)$$

where $\bar{L}_{pl}(d_0)$ is the mean path-loss at the reference distance d_0 , γ is the path loss exponent, and $X_\sigma \sim \mathcal{N}(0, \sigma^2)$ is the

normal distribution that accounts for shadowing, with zero mean and σ^2 variance.

6) *Sensitivity and Link Behavior:* The sensitivity of a radio receiver at room temperature is computed by what follows:

$$S = -174 + 10 \log(BW) + NF + SNR \quad [dB], \quad (2)$$

where the first term describes thermal noise in 1Hz of bandwidth and can only be influenced by changing the temperature of the receiver. BW is the receiver bandwidth. NF is the receiver noise figure that is fixed for a given hardware implementation. SNR is the signal-to-noise ratio required by the underlying modulation scheme and is determined by the spreading factor SF. The higher the SF, the higher the SNR.

A transmitted LoRa packet will be received if its power at the receiver is higher than or equal to the receiver sensitivity, which is given in Table I.

TABLE I: LoRa characteristics at BW = 125 kHz.

SF	Bit-rate [kbps]	Receiver Sensitivity [dBm] [2]	Reception Thresh. [dB]	Inter-SF collision Thresh. [dB] [5]
7	5.47	-123	-6	-7.5
8	3.13	-126	-9	-9
9	1.76	-129	-12	-13.5
10	0.98	-132	-15	-15
11	0.54	-134.5	-17.5	-18
12	0.29	-137	-20	-22.5

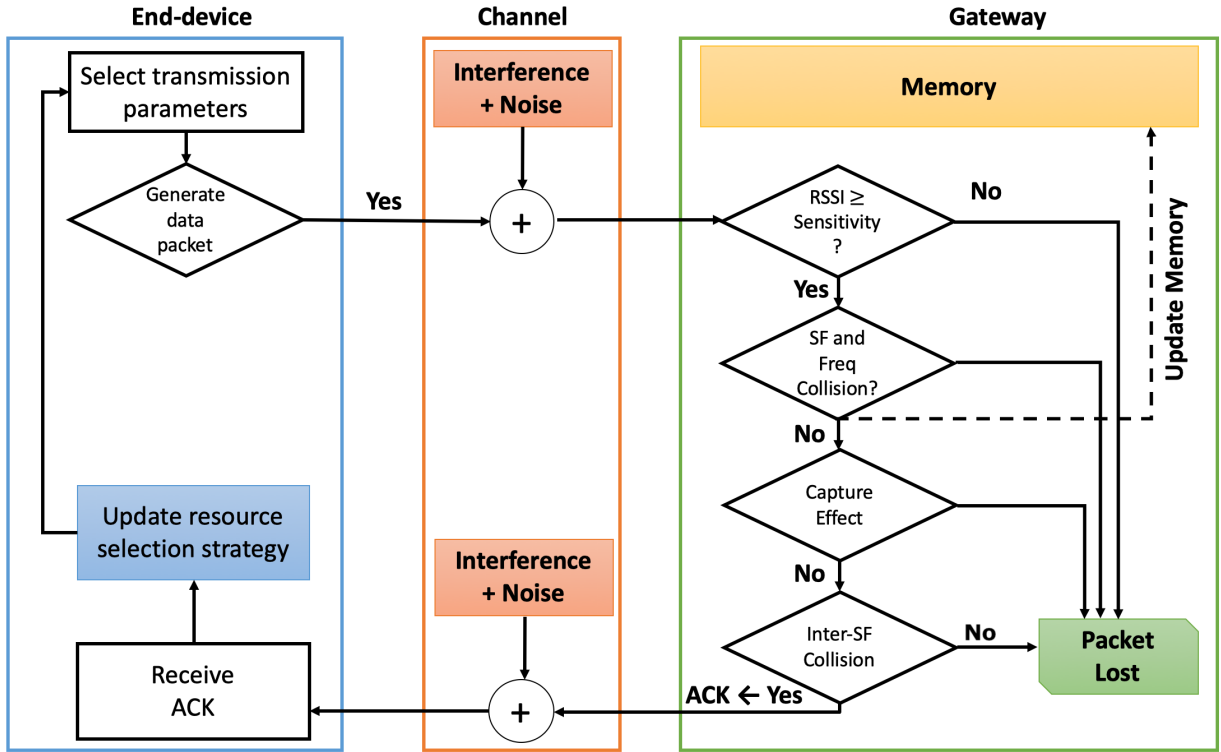


Fig. 2: Link behavior and simulation process

C. Collision Behavior

Due to resource limitations, two or more LoRa transmissions can overlap at the gateway. In such a case, there are some conditions that determine whether both packets can be successfully decoded, just one, or neither. These conditions rely on spreading factor, channel, power, and timing [7]. First, a collision occurs when two or more devices transmit at the same time in the same SF and channel. Next, a timing collision will happen if these transmissions overlap in their critical section (i.e., the section that starts at the last 5 preamble symbols of the packet). In addition, if there are several signals transmitting with the same SF and on the same sub-channel simultaneously, the NS is still able to successfully receive the strongest signal if its SINR is higher than a threshold of 6 dB. This phenomenon is called *capture effect* [3], [7].

Nevertheless, perfect orthogonality is not guaranteed, and interference among communications using different SF must also be accounted for [5]. In fact, the NS can successfully receive a signal using SF s if its power is higher by a given threshold (in Table I) than the multiplication of total interference from signals using SF $s' \neq s$. This phenomenon is deemed *inter-SF collision*.

D. Simulation Process

The simulation process of the proposed LoRaWAN simulator is shown in Figure 2. First, each end-device generates a packet by following a random distribution, such as the exponential distribution. The packet generation rate is chosen to satisfy the duty cycle restriction. The generated packet

then is transmitted to the gateway by using the selected radio resources.

After sending a packet, the end-device waits for an acknowledgment (ACK) sent by the NS through the gateway [2]. We assume that there is no collision between the ACK and uplink packets. In fact, the ACK can be delivered on a separate channel with a higher duty cycle. Hence, if an end-device receives an ACK for its transmitted packet, then either there was no collision, or the capture effect has occurred. Conversely, when ACK is not received, either the packet was lost due to collision with another packet transmitted with the same SF, or due to the inter-SF collision.

Depending on the received ACK or NACK, the end-device will update its strategy to reselect appropriate radio resources to minimize the number of collisions.

IV. DISTRIBUTED LEARNING FOR RADIO RESOURCES ALLOCATION IN LORAWAN

To implement the distributed learning for resource allocation in LoRaWAN, each end-device is considered as an intelligent agent which aims at minimizing the collision rate in a distributed manner by choosing the best radio resources $A_i = \{s_i, c_i, p_i\} \in \mathcal{A}$, where $c_i \in \mathcal{C}$, $s_i \in \mathcal{S}$ and $p_i \in \mathcal{P}$ are the selected channel, spreading factor and transmission power. After choosing the action $a_i(t)$ at time t , device i receives the corresponding reward, denoted by $r(t) \in \{0, 1\}$, where 1 stands for a successful transmission (ACK) and 0 represents a lost packet (NACK). Such kind of learning can be applied though Multi-Armed Bandit (MAB) problem [16] that only makes

use of local information available at the LoRaWAN end-device level. The result of the learning algorithm, for each device, will be a set of sub-channels, spreading factors and transmission power levels that suffer the least from collisions. To reduce the resource occupation of the neighboring devices, each device follows a set of rules that steers its decision and allows it to make a balance between (i) Exploiting the cumulated knowledge by choosing the most appropriate resources (s, c, p) and transmitting on them, and (ii) Exploring other resources that could be interesting to exploit.

A. The EXP3 algorithm

We resort to the popular EXP3 algorithm [17], [18] to address the distributed resource allocation problem. The goal is to steer the decision of each device i to choose autonomously the most appropriate (c_i, s_i, p_i) with minimum collision rate while ensuring reactivity to the possible changes that can occur in the common resource usage.

Let $a(t)$ be the chosen action at time t , the latter gives a reward $r_a(t)$. At each iteration t (at packet arrival), any device i selects a strategy with distribution $p_a^i(t)$ over \mathcal{A} . The goal of any device i is to update $p_a^i(t)$ in order to get the largest reward at horizon T in comparison with the best fixed strategy. We initialize the algorithm with the uniform distribution $p_s^i(0) = \frac{1}{A}$, where A is the cardinal of strategy \mathcal{A} .

The EXP3 algorithm for spreading factor selection in LoRaWAN is then presented in Algorithm 1. Note that in case of packet loss, $r_s(t) = 0$ and no update will be operated on the distribution strategy, and hence no learning either.

V. EXPERIMENTAL EVALUATION

We use LoRa-MAB simulator to investigate the performance of LoRaWAN deployment and the distributed learning solution for resource selection. To analyze the network performance, we use two evaluation metrics: i, the successful transmission rate, i.e., the rate of successfully received packets, and ii, the average energy consumption per successfully transmitted packet per end-device.

We evaluate the performance of LoRaWAN by using a simple setup where $N = 100$ end-devices transmit packets to one gateway. The configuration of the network is shown in Figure 3.

We consider three experiments:

- **Experiment 1:** each end-device can select one of 6 possible SFs from 7 to 12 with one sub-channel and the transmission power of 14 dBm.
- **Experiment 2:** each end-device can select one of 6 possible SFs from 7 to 12 with one of three sub-channels and the transmission power of 14 dBm.
- **Experiment 3:** each end-device can select one of 6 possible SFs from 7 to 12 with one sub-channel and the transmission power of $\{8, 11, 14\}$ dBm.

For each experiment, the time horizon for simulation is $T = 10^7$. The 1% LoRaWAN duty cycle limitation [1] is respected by setting the packet generation rate of each end-device to $\lambda = 15$ packet/hour and the packet length $l = 50$ bytes. The

Initialization:

- Let $a = A_j \in \mathcal{A}$ be the strategy chosen by device j .
- Set the initial weights $\omega_a^j(0) = 1, \forall a \in \mathcal{A}, \forall j \in \mathcal{N}$ and the uniform distribution of strategies per device.
- Set the learning rate $\gamma = \min \left\{ 1, \sqrt{\frac{K \log(K)}{(e-1)T}} \right\}$ where $e \approx 2.71828 \dots$ is the base of the natural logarithm.

for $t = 1$ to T do

initialization ;

foreach end-device j do

At time t , draw strategy $a \in \mathcal{A}$ according to the distribution $p_a^j(t)$;

if Transmit then

Receive reward

$$r_a^j(t) = \begin{cases} 1 & \text{if ACK is received,} \\ 0 & \text{otherwise.} \end{cases}$$

Update weights and distribution of available strategies:

$$\omega_a^j(t+1) = \omega_a^j(t) \exp\left(\frac{\gamma r_a^j(t)}{K \cdot p_a^j(t)}\right)$$

$$p_a^j(t+1) = (1 - \gamma) \frac{\omega_a^j(t+1)}{\sum_{a=1}^K \omega_a^j(t+1)} + \frac{\gamma}{K}$$

end

end

end

Algorithm 1: EXP3 algorithm for fully distributed SF allocation in LoRa network

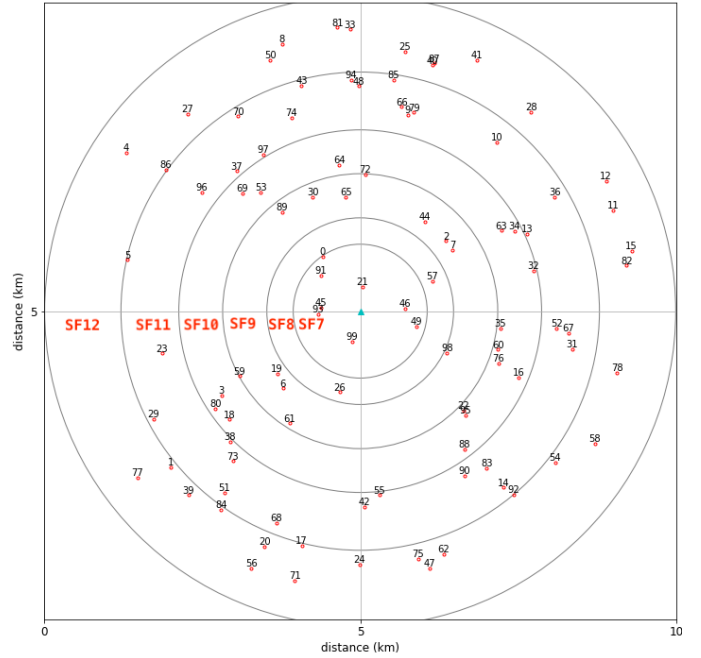


Fig. 3: Network configuration

packet is generated through an exponential distribution. End-

TABLE II: Parameters for performance analysis

Parameters	Values
Aera	Disc of radius 4.5 km
Packet length	50 bytes
Bandwidth (BW)	125 kHz
Code rate	4/5
Frequency set	868100 Hz
Capture Effect Threshold	6 dB
Inter-SF Collision Threshold	Table I
Transmission Power	14 dB

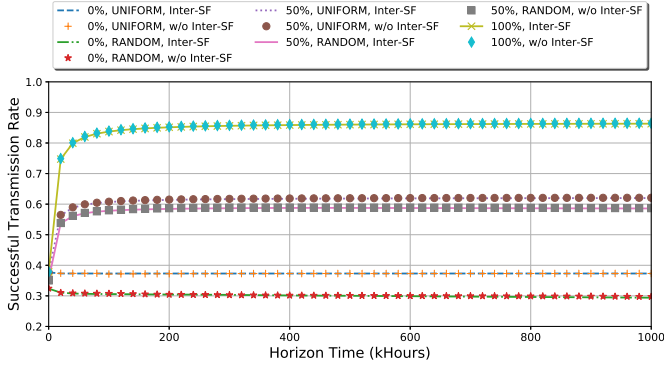


Fig. 4: Experiment 1: Successful transmission rate

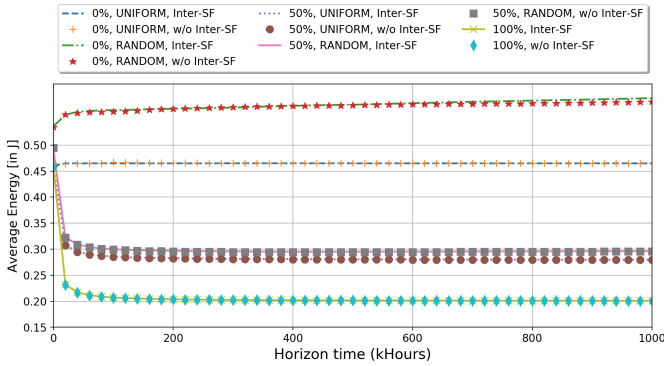


Fig. 5: Experiment 1: Average energy consumption per a successful transmit packet per device

devices are located in a circle with radius $r = 4.5\text{km}$ through a uniform distribution. In our simulations, we consider the log-distance path loss model with flat fading, where the reference distance is $d_0 = 40\text{m}$, the path loss at the reference distance is $PL_0 = 107.41\text{ dB}$, and the path loss exponent is $\gamma = 2.08$. The other simulation parameters are presented in Table II.

We will evaluate the EXP3 performances in a real setting that accounts for both the capture effect and the inter-SF collision. Further, to fully assess the reinforcement learning based algorithm, we will compare the EXP3 against simple mechanisms where each device selects its action over the set of actions according to i, the uniform distribution and ii, a Gaussian distribution.

1) *Experiment 1*: Figure 4 shows the successful transmission rate of LoRaWAN. We can see clearly that the successful

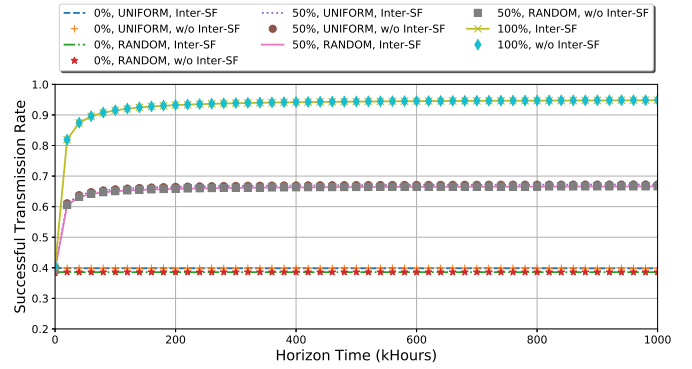


Fig. 6: Experiment 2: Successful transmission rate

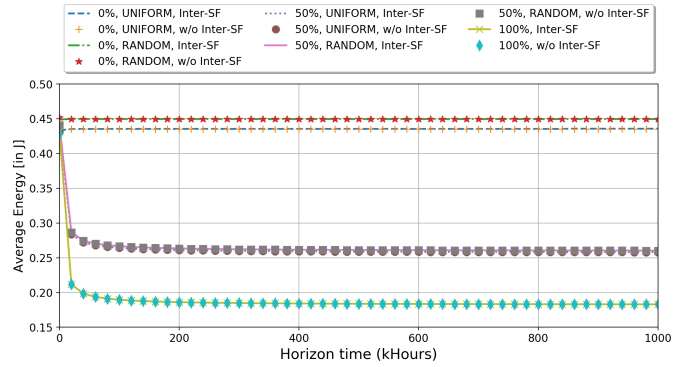


Fig. 7: Experiment 2: Average energy consumption per successful transmit packet per device

transmission rate of the system with distributed learning is significantly higher than that of the uniform SF selection and random SF selection. In addition, the larger the number of intelligent end-devices using distributed learning, the higher the packet reception rate.

Figure 5 shows the network average energy consumption. Interestingly, the larger the number of intelligent end-devices using distributed learning, the lower the energy consumption. Hence, intelligent resource allocation improves both network performances and energy efficiency.

2) *Experiment 2*: Similar to Experiment 1, in Experiment 2, the successful transmission rate of the network with distributed learning is significantly increased compared to the uniform or random SF selections (Figure 6). Consequently, the average network energy consumption is significantly reduced compared to the uniform and random SF selections. Moreover, Figure 7 shows the average energy consumption in the case where 100% of end-devices use the EXP3 algorithm in Experiment 2 is smaller than that in Experiment 1 since the successful transmission rate in Experiment 2 is higher (0.96 v.s. 0.845).

3) *Experiment 3*: Figure 8 and 9 show the successful transmission rate and energy consumption in LoRaWAN for Experiment 3 with 3 power levels. Similar to the previous experiments, the successful transmission rate with distributed

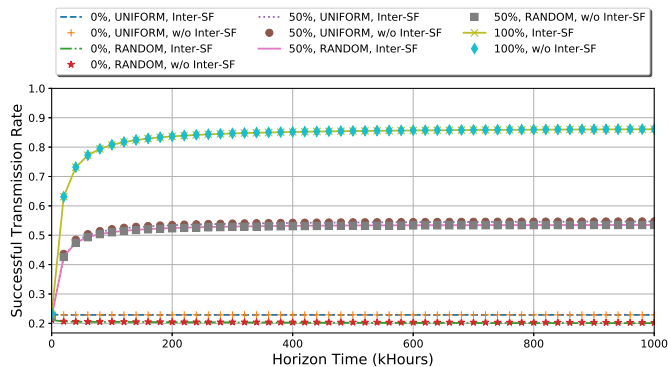


Fig. 8: Experiment 3: Successful transmission rate.

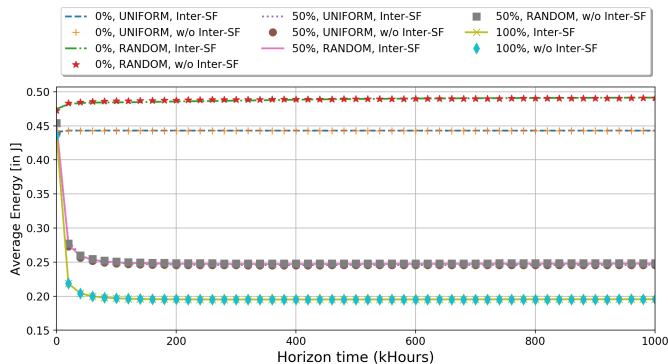


Fig. 9: Experiment 3: Average energy consumption per a successful transmit packet per device.

learning is significantly increased while the average power consumption is significantly decreased compared to the uniform or random SF selections.

Figure 10 then compares the successful transmission rate and the corresponding average power consumption of Experiment 1 and Experiment 3. We observe that the successful transmission rates in both experiments are close together. Conversely, the average energy consumption in Experiment 3 was lower than 0.006 J per successfully transmitted packet per end-device compared with that realized in Experiment 1. Such a value is small for one packet per end-device, however, it is a huge value for the transmission process of a dense IoT network. It means that by using distributed learning algorithm for the radio resource selection with multiple power levels, we can reduce the total energy consumption of the network, hence increase sensibly the lifetime of LoRaWAN.

VI. CONCLUSION

In this paper, we presented our devised discrete event simulator for performance evaluation of LoRaWAN in a realistic setting with the use of distributed learning for astute radio allocation. This simulator permits testing and evaluating different network design parameters and self-managing solutions for savvy radio resource allocation. The class-based simulator code is open source, making it easier for the scientific community to modify, adapt, and improve the simulator.

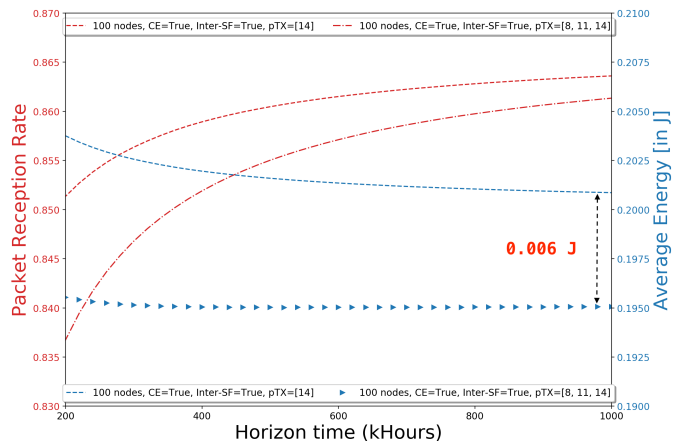


Fig. 10: Comparison of the performance of the network in experiment 1 and experiment 3.

We have shown that the distributed learning algorithm can significantly improve the performance of LoRaWAN in terms of successful packet ratio and energy consumption. However, convergence times are long, in the order of 200 kHours for the EXP3 algorithm. Therefore, in future work, we need to address the convergence time issue by using more computationally efficient distributed learning algorithms.

Furthermore, the location of each end-device can be used to select the most appropriate spreading factor which is distance related. Such information can constitute a warm start in EXP3 which can sensibly accelerate the learning process. Finally, the role of the gateway in the learning process will also be investigated. The gateway can act as an expert by advising adequately the end-devices to improve the resource selection process.

ACKNOWLEDGMENT

This research was partially supported by Labex DigiCosme (project ANR-11-LABEX-0045-DIGICOSME) operated by ANR as part of the program “Investissement d’Avenir” Idex Paris-Saclay (ANR-11-IDEX-0003-02).

REFERENCES

- [1] L. Alliance. [Online]. Available: <https://www.lora-alliance.org/>
- [2] O. Georgiou and U. Raza, “Low power wide area network analysis: Can lora scale?” *IEEE Wireless Comm. Letters*, vol. 6, pp. 162–165, 2017.
- [3] C. Goursaud and J. M. Gorce, “Dedicated networks for iot: Phy / mac state of the art and challenges,” *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, p. 150597, Oct 2015.
- [4] M. C. Bor and et al, “Do lora low-power wide-area networks scale?” 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: ACM, 2016, pp. 59–67.
- [5] D. Croce and et al, “Impact of spreading factor imperfect orthogonality in lora communications,” *Communications in Computer and Information Science*, pp. 165–179, 2017.
- [6] A. Waret, M. Kaneko, A. Guillon, and N. E. Rachkidy, “Lora throughput analysis with imperfect spreading factor orthogonality,” *IEEE Wireless Comm. Letters*, pp. 1–1, 2018.
- [7] T. Voigt, M. Bor, U. Roedig, and J. Alonso, “Mitigating inter-network interference in lora networks,” in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN. USA: Junction Publishing, 2017, pp. 323–328.

- [8] “Simpy – event discrete simulation for python.” 03 2018. [Online]. Available: <https://simpy.readthedocs.io>
- [9] M. O. Farooq and D. Pesch, “Poster: Extended lorasim to simulate multiple iot applications in a lorawan,” in *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*. USA: Junction Publishing, 2018, pp. 175–176.
- [10] A. Dongare. Lorasim extension. [Online]. Available: <https://github.com/adwaitnd/lorasim>
- [11] T. To and A. Duda, “Simulation of lora in ns-3: Improving lora performance with csma,” in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.
- [12] M. Slabicki, G. Premsankar, and M. D. Francesco, “Adaptive configuration of lora networks for dense iot deployments,” in *IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–9.
- [13] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, “Multi-armed bandit learning in iot networks: Learning helps even in non-stationary settings,” *Cognitive Radio Oriented Wireless Networks*, pp. 173–185, 2018.
- [14] A. Azari and C. Cavdar, “Self-organized low-power iot networks: A distributed learning approach,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.
- [15] J. Petajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pet-tissalo, “On the coverage of lpwans: range evaluation and channel attenuation model for lora technology,” in *2015 14th International Conference on ITS Telecommunications (ITST)*, 2015, pp. 55–59.
- [16] S. Bubeck and A. Slivkins, “The best of both worlds: Stochastic and adversarial bandits,” in *Proceedings of the 25th Annual Conference on Learning Theory*, vol. 23. PMLR, 25–27 Jun 2012, pp. 42.1–42.23.
- [17] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multi- armed bandit problem,” *Machine Learning*, vol. 47, no. 2/3, pp. 235–256, 2002.
- [18] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, “The nonstochastic multiarmed bandit problem,” *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.