



A K-Means Based Approach for Optimal Gateway Deployment in LoRaWAN-SIM

Thomas Djotio Ndie¹(✉), Antoine Junior Tsagmo Denkeng², Karl Jonas³,
and Roblex Nana Tchakouté^{1,2,3}

¹ Bonn-Rhein-Sieg University of Applied Sciences, Rheinbach, Germany
tdjotio@gmail.com, nanatchakouteroblex@gmail.com

² Faculty of Sciences, University of Yaounde 1, Yaounde, Cameroon
juniorttsagmo@gmail.com

³ National Advanced school of Engineering of Yaounde 1, Yaounde, Cameroon
karl.jonas@h-brs.de

Abstract. LoRa is a technology that enables low-energy wireless communications over very long distances. Under these conditions, the simulation of LoRa projects becomes essential. Indeed, it allows on the one hand to virtually model and evaluate the performance of projects in the field and, on the other hand, to anticipate the optimization of the often expensive costs during the implementation of these projects. In this paper, we revisit the issue of simulation for LoRaWAN networks. Our research has led us to conclude that a LoRaWAN simulator's relevance can be assimilated to the problem of better gateway placement with the best exploitation of their parameters. Our approach consists of first studying some of the existing simulators with the aim of proposing a generic architecture model for such a tool and then, proposing recommendations for the selection of relevant simulators. To validate our model, we used the K-means machine learning technique to solve the problem of bad gateway locations in the LoRaWAN-SIM simulator. From our prototyped **K-LoRaWAN-SIM** simulator, we observed a gain of 10 gateways compared to the LoRaWAN-SIM simulator considered as our reference prototype. In fact, the results obtained with 25 gateways are close to those obtained with 35 gateways in LoRaWAN-SIM.

Keywords: Internet of Things · LPWAN · LoRa Technology · Simulation

1 Introduction

Considering the growth of the world's population, the main future challenges are intrinsically linked to energy management, natural resource monitoring, pollution control, the efficiency of environmental protection infrastructures, disaster prevention, medical monitoring and smart agriculture; to name but a few [1]. At the heart of these issues is the Internet of Things (IoT) and, LoRa technology is part of it. In such a context, the simulation of LoRaWAN networks becomes very important, as it can be used to observe the virtual behavior of the network

with the aim of getting as close as possible to the real world for decisions and actions to be applied. LoRaWAN is considered very promising due to its open accessibility, simplicity and flexibility both technologically and commercially [2]. A good understanding of the constraints inherent to this technology is necessary for the modeling and realization of a simulator in the field. An LPWAN simulator is a software that allows giving an overview of the real operation of an LPWAN network under specific conditions.

LoRa devices operate in ISM (Industry, Science, Medicine) bands that are subject to interferences [3]. Uplink messages are sent by nodes to the network server and are relayed by one or more gateways. Downlink messages are sent by the network server to a single terminal and are relayed by a single gateway [3]. For some sensitivity thresholds, the gateway decodes the signal more easily than for others. Any packet with a sensitivity below the minimum sensitivity will not be detected by the gateway and will be considered lost.

LoRaWAN supports spread factors (SF) between 7 and 12, which allow for a trade-off between range and data rate. SF7 has the shortest range and highest data rate; SF12 is the exact opposite. The SF value is assigned when a new node connects to the base station. Since the SF symbols are orthogonal, the same gateway can receive different transmissions with different SFs on the same channel simultaneously, although in real-life scenarios this is not always the case [4]. The base station sends an acknowledgment frame (ACK) to the node when it receives a successful transmission. However, if a node does not receive an ACK frame, it retransmits the previous data until a successful ACK occurs [5].

However, in this panorama of upcoming challenges, to our humble knowledge, the academic community and individuals do not have clear information and guidance on the appropriate simulator(s) for the work they have in mind. Indeed, a number of open-source network simulators have been developed in recent years; but, most of them were developed for specific research purposes (first generation). Therefore, they tend to focus on a specific aspect. They are often limited by a number of assumptions. They also cannot properly capture the operation of the entire LoRaWAN protocol stack and therefore cannot properly simulate a real-world network. For this reason, our attention is mainly focused on the so-called second generation simulators. Our research has also led us to conclude that the relevance of a LoRaWAN simulator resides in the problem of placing gateways with the best exploitation of their characteristics in the simulator. The contributions made in this paper are to (1) - Propose a generic model of a LoRaWAN simulator architecture and suggest recommendations for relevant selection of a simulator. (2) - Propose an optimal solution for the gateway deployment problem in a LoRaWAN network, applied to the LoRaWAN-SIM simulator case. To validate our model, we used the K-means machine learning technique to solve the gateway placement problem in the LoRaWAN-SIM simulator. From our prototyped K-LoRaWAN-SIM simulator, we observed a gain of 10 gateways compared to the LoRaWAN-SIM simulator considered as our reference prototype. Indeed, the results obtained with 25 gateways in K-LoRaWAN-SIM are close to those obtained with 35 gateways in LoRaWAN-SIM.

The rest of this paper is organized as follows: Sect. 2 is devoted to works related to LoRa simulators and suggests an argued recommendation. In Sect. 3, we propose a generic architecture model for a LoRaWAN simulator. In Sect. 4, we also propose a solution to the gateway deployment issue in LoRaWAN-SIM based on the K-Means algorithm. Before concluding, we present our experiments and interpretations of the results in Sect. 5.

2 Related Research and Suggestions

2.1 State of Research on LoRaWAN Simulators

The research question on LoRa technology simulators has already been addressed in the literature by Silva and al. [6] and Christos Bouras and al. [1]. They produced surveys presenting respectively LoRaWAN simulations of the NS3 Module only and, the LoRa simulators (too specific), mostly of the first generation. However, these works are limited today because they do not take into account more complete or complex simulations of LoRa networks. This hampers users in their simulation projects. Contrary to these surveys, we focus in this paper on more complete simulators so-called second generation. Indeed, their varied functions allow us to better simulate real-world constraints, which makes them more relevant.

Indeed, we have gone through some works that have been interested in different LoRa network simulators, capable of performing useful evaluations. We can quote without being exhaustive: *NS3* [6], [7], *LoRaSim* [8], *LoRaFREE* [9], *LoRaEnergySim* [10], *LoRaWAN_Simulator* [11], *IoT-MAB* [12], *LoRaWAN-SIM* [13] and *FLoRa* [14]. However, many of these simulators are deployed for specific research purposes. They are generally oversimplified and contain a number of assumptions that affect the accuracy of their results. Here, we will only present simulators that are in line with a logic of completeness of functionality.

2.2 Recommended LoRa Simulators

The LoRaWAN network simulation model illustrated in Fig. 1, consists of the set of steps that any simulator should necessarily take into account for its operation. Indeed, a user must be able to model his network, specify the traffic model and, the routing algorithms that can be used. He must be able to obtain performance measurements of the simulated network. [1]. This model suggests an abstract or generic architecture for LoRaWAN simulators. From this model could be derived different architectures that could lead to the implementation of simulation tools in the field.

Ideally, the expected recommendation would be to propose simulators to users according to their sector of activity (e.g. medical monitoring, intelligent agriculture, environmental monitoring). Unfortunately, current simulators are not yet at this level of development. Nevertheless, there are some that are trying to address these issues. Figure 2 presents the general architecture of a LoRaWAN

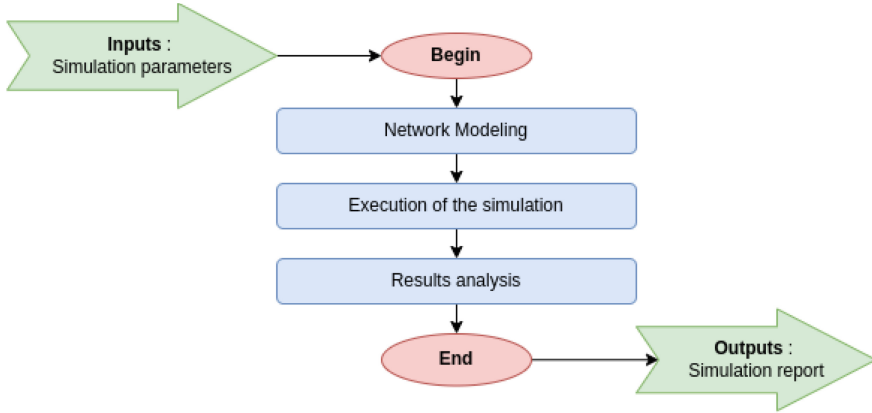


Fig. 1. LoRaWAN simulator model

network [11]. The gateway is used to transfer messages from the end nodes to the network server. In a LoRaWAN network, end nodes are not tied to a specific gateway. The data transmitted by an end node is received by multiple gateways. Each gateway forwards the packet received by the end node to the network server via a link (cellular, Ethernet, satellite or Wi-Fi).

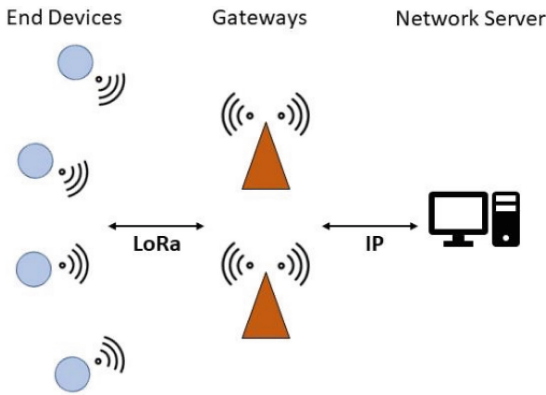


Fig. 2. Architecture of the LoRaWAN network

The NS3 Simulator. NS3 is a popular simulator for Internet systems, which supports LoRaWAN simulations. It is intended for teaching and research purposes with the goal of simulating a wide variety of networks. It includes over 700 different simulation scenarios. It supports a wide range of protocols and wireless network simulations as well. It is written in C++ and Python and, makes use

of class A of the LoRaWAN specifications for its terminals [7], [6]. This is the least energy-consuming class compared to classes B and C. It is free and open source, available under the GNU GPLv2 license, and maintained by a vibrant global community.

LoRaWAN_Simulator. LoRaWAN_Simulator [11] is a working environment implemented in MATLAB. It is developed to characterize the behavior of LoRaWAN networks, taking into account physical, media access control, and network aspects. The wide variety of parameters it handles allows it to stand out as one of the most complete simulators currently available in the field. The points that make it special are: the accuracy of its physical layer for the examined settings, support for multiple gateways, modeling of half and full duplex gateways, the ability to enable or disable receive windows, consideration of uplink/downlink interference, and the impact of different coding rates in interference and noise limited scenarios.

IoT-MAB. IoT-MAB [12] is a SimPy-based simulator developed in Python 3. It uses machine learning techniques to enable LoRa devices to use innovative learning techniques to simulate the intelligent allocation of distributed resources in a LoRa network. Its goal is to introduce artificial intelligence into LoRa networks for better management of network resources. However, its handling and use are not easy.

LoRaFREE. LoRaFREE [9] is a more comprehensive SimPy simulator than LoRaSim. Unlike LoRaSim, it takes into account a packet error model, imperfect orthogonality of spreading factors, fading impact, and duty cycle limitation at the device and gateway level. LoRaFREE supports two-way communication by adding downlink capability and retransmission strategy in case of confirmable uplink transmissions. LoRaFree also extends LoRaSim's energy consumption profile to consider the energy consumed at the time of reception. In addition, this simulation environment is capable of simulating the timing of synchronized transmissions.

LoRaWAN-SIM. The LoRaWAN-SIM simulator [13] is completely written in Perl. It is designed primarily to evaluate confirmed transmissions, unconfirmed transmissions, and, large device simulations that include multiple gateways. It consists of two major modules that run sequentially. Based on a time-slicing approach, it has also been shown to be interesting for evaluating the optimal data collection time in LoRa networks.

The important remark is that the vast majority of LoRa simulators currently available are only used to evaluate the metrics of a LoRaWAN network. To our humble knowledge, (1) many of them are not yet specialized for specific areas of our daily life (for example simulators dedicated to medicine, precision agriculture, environmental monitoring and others); (2) only 03 of these simulators (NS3

Module, LoRaWAN_Simulator, and LoRaWAN-SIM) would be fairly suitable for use in complex applications such as the smart city. Although LoRaWAN-SIM has caught our attention, in its version at the time of writing this article, it does not yet allow an optimal deployment of gateways.

3 Proposal of an Ideal Simulator Model

We define an ideal or generic simulator as one that would accept a high number of (1) simulation settings as input and, (2) simulation results or data as output. Such a simulator is characteristic of a high level of network manageability on the one hand and, on the other hand, of a high level of understanding of the said network with as much detail as possible.

3.1 Properties of an Ideal Simulator

These are essential metrics for the proper functioning of the network in a massive IoT:

- A large number of nodes (High Nber of ED): greater than or equal to 1000
- A large number of gateways (High Nber of GW): greater than or equal to 10
- Rate of packets delivered (Delivery ratio)
- Rate of packets received (Reception ratio)
- Downlink and ACK (Downlink & ACK)
- Evaluation of energy consumption (Energy evaluation)
- Collisions evaluation
- Imperfect orthogonality of SF (SF Orthogonality)
- Adaptive data rate implementation (ADR)
- A large number of results returned (Nber of Returns): greater than or equal to 10.

3.2 Mathematical Model of an Ideal Simulator

Let P_R = our set of user parameters and the results that a simulator can offer. Let $SimI$ denote the ideal simulator, $S = \{s_1, s_2, \dots, s_n\}$ a set of simulators, and $Sim(s_i)$ a function that takes a simulator s_i as parameter and returns the total number of parameters and results in P_R . $n = total\ number\ of\ simulators\ in\ our\ study$. We model this ideal simulator by :

$$SimI = argmax(Sim(s_i)); 1 < i < n$$

$$Sim(s_i) = \sum_{j=1}^m P_j; \text{ With } P_j = \begin{cases} 1 \text{ if } s_i \text{ has the } P_R_j \\ 0 \text{ if not (The } P_j \text{ are equal)} \end{cases}$$

Let us recall here that we are not trying to evaluate the relevance of a parameter in relation to another, but rather the presence or not of a parameter in the simulator.

4 Optimal Gateway Deployment

4.1 Issue Focus: Poor Deployment of Gateways

From our previous study, we found that the simulation question essentially raises 2 important concerns: the optimal placement and deployment of gateway nodes, so that the results obtained during simulation minimize errors during actual network deployment. In other words, the goal is to provide a means to determine the optimal positions of the gateway nodes for efficient network operation. Indeed, a good placement of gateway nodes inevitably leads to an optimal coverage of the area to be monitored, and consequently, to an economy of the global number of these gateway nodes. We have chosen the LoRaWAN-SIM simulator as the basic framework for experimenting with this problem of optimal positioning of gateway nodes for deployment that would ensure good network performance. As presented in the previous section, LoRaWAN-SIM is described as a package formed of two modules: one for selecting the parameters and the other for launching the actual simulation.

When using LoRaWAN-SIM, we found that the gateway and sensor nodes are completely generated randomly in a plot. The problem with this is that the network implements a haphazard arrangement of gateways. As a result, this non-optimal arrangement of the gateways has consequences on the network's overall performance. Indeed, their bad placement does not allow optimal coverage and could affect the quality of service (QoS) of the network. This is what motivates us to look for a technique to propose an approach that will allow a better disposition of the gateways that will optimize the network performances.

4.2 Motivations

We wish to offer the possibility to generate during the simulation, a network whose distribution of sensor and gateway nodes will be closer to reality. The node placement model should be economical in the number of gateway nodes needed to provide better QoS performance during network operation.

4.3 Resolution Approach

To solve this problem of finding the best gateway layout, we propose a solution based on machine learning, particularly clustering algorithms. These clustering algorithms allow to partition of the data into subgroups, or clusters, in an unsupervised way [15].

Among these unsupervised learning methods, we can distinguish hierarchical approaches; centroid methods (K-means and K-medoids), and, density approaches (DBSCAN). We have chosen the K-means algorithm because our data (points in the plane) are numerical and unlabeled. Our proposed approach allows for a more intelligent grouping of gateways in the network into node groups.

The K-Means Algorithm: K-means is an iterative algorithm that partitions the data set into K distinct subgroups (clusters). It tries to make the intra-cluster data points as similar as possible while keeping clusters as distant as possible [16]. It assigns data points to a cluster such that the sum of the squared distance between these points and the centroid of the cluster is at a minimum. The manual approach shows limitations in determining the optimal positioning of gateway nodes in the simulators studied above, especially in the context of a massive IoT network.

$$\arg \min \sum_{k=1}^K \sum_{\vec{x}_i \in C_k} d(\vec{x}_i, \mu_k)^2$$

However, it is difficult to find the exact cluster centers, so we use a heuristic.

Algorithm 1. K-means clustering algorithm

Require: dataset X of nodes coordinates, number of groups $K \in N$

Ensure: k groups centers coordinates for gateways

- 1: $I \leftarrow \infty$
 - 2: take K arbitrary centres $c_k \in D$
 - 3: **repeat**
 - 4: Initialize all k groups to empty $G_k \leftarrow \emptyset$
 - 5: **for** $i \in 1, \dots, N$ **do**
 - 6: $k^* \leftarrow \operatorname{argmin}_{k \in 1, \dots, K} d(x_i; c_k) G_{k^*} \leftarrow G_{k^*} \cup x_i$
 - 7: **end for**
 - 8: Re-compute the centers: $c_k \leftarrow$ centre of gravity of G_k
 - 9: $I \leftarrow I_W$
 - 10: Calculate I_W
 - 11: **until** $I - I_W < \text{threshold}$
-

4.4 Architecture of K-LoRaWAN-SIM

Here, K-means is included in the first module to solve the target problem. Figure 3 depicts the workflow that models the proposed enhanced simulator architecture named **K-LoRaWAN-SIM**.

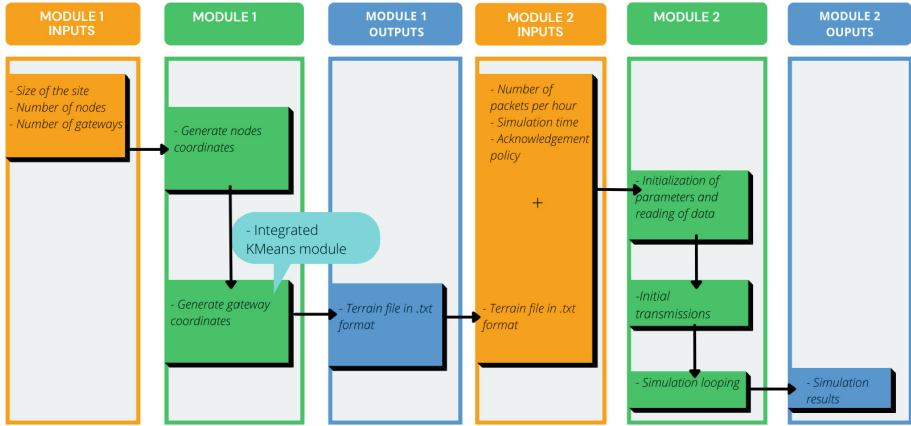


Fig. 3. Workflow of K-LoRaWAN-SIM

5 Experiments and Results (Case of LoRaWAN-SIM)

5.1 Experimental Protocol

Working Environment

- **Working machine:** a PC running Ubuntu 20.04.3 LTS 64-bit OS, with an Intel processor CoreTMi5-9300HF CPU @ 2.40GHz × 8 and 16 GB of RAM.
- **Programming language:** we implemented our K-Means learning module with the scikit-learn library in Python3, which we then coupled with LoRaWAN-SIM initially developed within Perl 5.30.

Simulation Settings: The different parameters used for our experiments are recorded in Table 1. These settings are independently used first in the LoRaWAN-SIM simulator version we took as a reference, and then in our developed prototype K-LoRaWAN-SIM simulator.

Our example use case consists in simulating a typical scenario of environmental monitoring applications such as air quality monitoring in a target area by a massive IoT network. To illustrate this, we have chosen an experimental framework with an area of nearly 9Km² (assimilated to a community) in which we have dispersed nearly 3000 nodes, interconnected by a variable number of gateways.

Methodology for Results Validation: The choice of the minimum number of gateways (35 and 9) used to launch the network was made each time following numerous experiments in the **LoRaWAN-SIM** and **K-LoRaWAN-SIM** simulators.

The optimal choice of gateways: We first performed about ten runs and each time we observed the state of the network connectivity, i.e., if all the nodes

Table 1. Simulation parameters: In order to make our results reproducible.

Parameters	LoRaWAN-SIM	K-LoRaWAN-SIM	
	35 Gateways	35 Gateways	9 Gateways
Site size (meters)	3000 m x 3000 m	3000 m x 3000 m	3000 m x 3000 m
Number of nodes	3000	3000	3000
Number of gateways	35	35	9
Number of packets sent per hour for each ED	12 : 1pkt/5min	12 : 1pkt/5min	12 : 1pkt/5min
Simulation time(sec)	10000 sec	10000 sec	10000 sec
ACK Policy	2	2	2
Coding rate	4/5	4/5	4/5
Bandwidth (BW)	125 KHz	125 KHz	125 KHz
Path Loss Model	$L_{pl}(d_0) = 95dB$, $d_0 = 40m$, $\gamma = 2.08$, $\sigma = 3.57$	$L_{pl}(d_0) = 95dB$, $d_0 = 40m$, $\gamma = 2.08$, $\sigma = 3.57$	$L_{pl}(d_0) = 95dB$, $d_0 = 40m$, $\gamma = 2.08$, $\sigma = 3.57$
Receiver sensitivity (per SF for BW125)	$[-124, -127, -130, -133, -135, -137]dBm$	$[-124, -127, -130, -133, -135, -137]dBm$	$[-124, -127, -130, -133, -135, -137]dBm$
Power & consumption T_x	2dBm, 0.0396 mW; 7dBm, 0.099 mW; 14dBm, 0.2508 mW	2dBm, 0.0396 mW; 7dBm, 0.099 mW; 14dBm, 0.2508 mW	2dBm, 0.0396 mW; 7dBm, 0.099 mW; 14dBm, 0.2508 mW
Consumption R_x	0.15 mW	0.15 mW	0.15 mW

of the network were well connected by all the gateways. We then gradually reduced the number of gateways in the LoRaWAN-SIM use case to 35 gateways. We concluded that this was the minimum number needed to cover all nodes in the network. Finally, to highlight the effectiveness of our K-means based optimization algorithm in terms of minimizing the number of gateways (due to a good layout from now on), we proceeded to progressively decrease the number of gateways in the K-LoRaWAN-SIM use case down to 9 gateways. We managed to achieve the same results in terms of coverage and network connectivity.

Validation of simulation results: We generated 10 instances per simulation for each set of parameters and took the average of these results to populate tables. For example, for the values of *the average energy consumed by a node*, we took the sum of the average energies of each run and divided it by the number of times we performed these runs. In order to confirm the choice of the average, we have evaluated the standard deviation (σ) on the obtained results in order to corroborate that there is not too much variability in the obtained results. e.g.:

- For the energy, we have about one standard deviation of $\sigma = 2mJ$
- For the ratio, we have about a standard deviation of $\sigma = 0.2\%$

5.2 Experimental Results, Comparisons and Discussions

Analysis and Interpretation of Results: Figures 4, 5 and 6 illustrate the graphs of the different configurations relative to the Table 1. It can be seen that the last two propose a better arrangement of gateways than the first one.

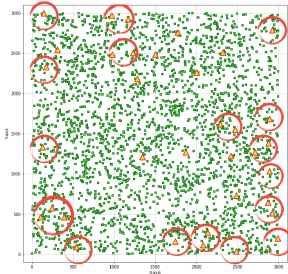


Fig. 4. $9km^2$, 3000 ED, 35 GW without K-means

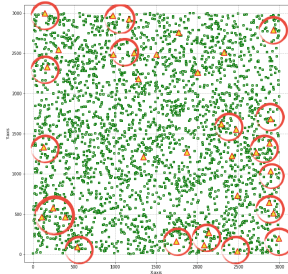


Fig. 5. $9km^2$, 3000 ED, 35 GW with K-means

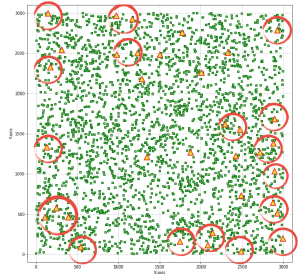


Fig. 6. $9km^2$, 3000 ED, 9 GW with K-means

The energy consumption is represented by the diagrams in Fig. 7. This low value obtained is consistent with the expected good energy performance of the network. The packet delivery and reception rates are represented by the diagrams in Fig. 8. The high value obtained is a good indicator of good performance of the ratio of sending and receiving packets in our network. These two figures are only the representation of the results of the output of simulations quantified in Table 2.

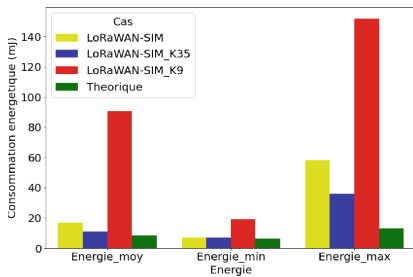


Fig. 7. Energy consumption

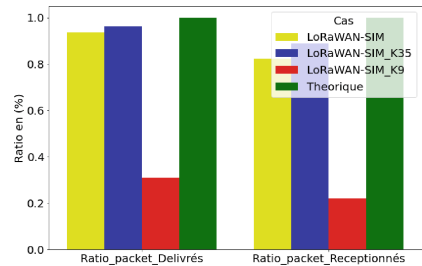


Fig. 8. Ratio of delivered/received packets

However, we observe that there is a relationship between the poor performance of the 3rd experiment (4th column of Table 2) and the reduced number of gateways. Similarly, there is a relationship between the very high energy consumption and the poor packet delivery and reception ratio. We say that the numbers in the 4th column of Table 2 justify the energy consumption for this set of parameters. **The massive data traffic to the only available gateways affects the operation of the entire network by decreasing the quality of service (QoS) including minimizing the network throughput.** This large amount of retransmission is the cause of the increase in energy consumption resulting in a rapid depletion of the batteries. The very high number of lost packets cannot guarantee a good QoS either.

Table 2. Summary of observations

Parameters	LoRaWAN-SIM	K-LoRaWAN-SIM		
	35 Gateways	35 Gateways	9 Gateways	Theorics Val
Average energy consumed by the node (mJ)	16,72881	11,06441	90,46371	8,2764
Minimum energy consumed by the node (mJ)	6,86721	6,95059	19,15122	6,3162
Maximum energy consumed by the node (mJ)	58,33216	35,90506	151,81651	13,2858
Total number of single transmission	102395	102170	156541	99000
Total number of retransmissions	115860	44620	1122483	0
Total number of transmissions	214747	146163	1173782	99000
Total number of packets acquired	95887	98542	48299	99000
Total number of packets delivered	176177	129769	258842	99000
Total number of lost packets	3508	627	105241	0
Ratio of packets delivered	0,93650	0,96449	0,30854	1
Ratio of packets received	0,82353	0,88784	0,22052	1

Consequently, we are looking for a compromise to reduce the number of gateways while maintaining good performance.

Trade-Off Between Gateway Reduction and Performance: We have found the compromise of the number of gateways per experiment: We performed many executions and each time we observed the packet delivery and reception ratios because this information is a function of the network energy consumption. It also allows us to characterize the network from the number of retransmissions recorded. As a standard, the desired packet delivery and reception ratios are respectively at least 90% and 80%. To this end, we have gradually increased the number of gateways in the K-LoRaWAN-SIM use case to 25, the average number representing the best compromise sought at the outset. This represents a reduction of 10 gateways compared to the LoRaWAN-SIM reference tool and, in addition, provides good performance in terms of adequate QoS in the network. We make the following observation: the new set of settings passed to the K-LoRaWAN-SIM simulator provides quite good results in output. We have indeed found the desired compromise that satisfies us both on the number of gateways used and on the network performance (see Table 3, Figs. 10 and 11). We can then say that compared to LoRaWAN-SIM, K-LoRaWAN-SIM has even more advantages, especially the reduction of the number of gateways needed and the good performance for the network (Fig. 9).

Discussions. Compared to **LoRaWAN-SIM**, the proposed tool **K-LoRaWAN-SIM** takes into account the gateway centralization parameter with the aim of providing better performances.

The optimization brought by the **K-means** to this simulator has a global impact on the network settings, such as: the global energy consumption of the

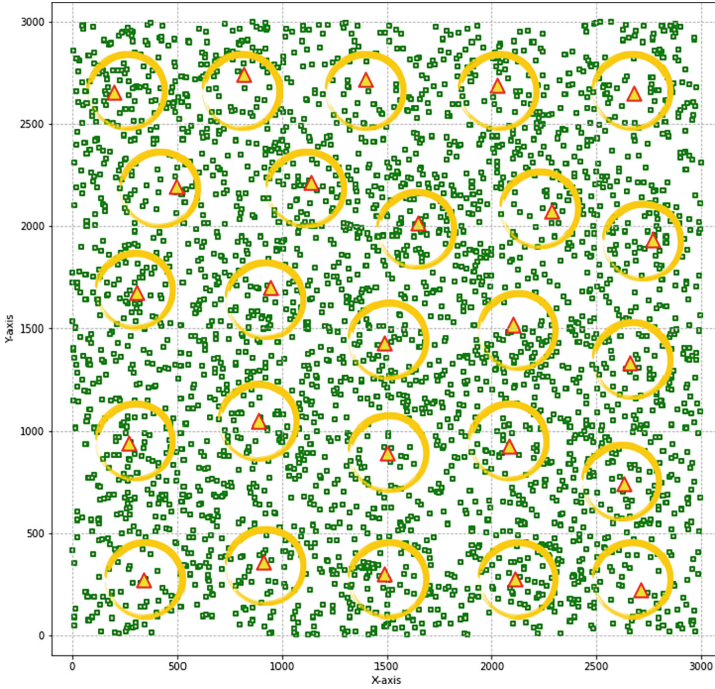


Fig. 9. $9km^2$, 3000 ED, 25 GW with K-means

Table 3. Summary of observations

Parameters	LoRaWAN-SIM	K-LoRaWAN-SIM			
	35 Gateways	35 Gateways	9 Gateways	25 Gateways	Theorics Val
Average energy consumed by the node (mJ)	16,72881	11,06441	90,46371	16,74796	8.2764
Min energy consumed by the node (mJ)	6,86721	6,95059	19,15122	8,94078	6.3162
Max energy consumed by the node (mJ)	58,33216	35,90506	151,81651	37,18431	13.2858
Total number of single transmission	102395	102170	156541	102120	99000
Total number of retransmissions	115860	44620	1122483	128367	0
Total number of transmissions	214747	146163	1173782	227080	99000
Total Number packet Acquity	95887	98542	48299	95712	99000
Total number of packets delivered	176177	129769	258842	185467	99000
Total Number of losts packets	3508	627	105241	3409	0
Ratio of packets delivered	0,93650	0,96449	0,30854	0,93724	1
Ratio of received packets	0,82353	0,88784	0,22052	0,81675	1

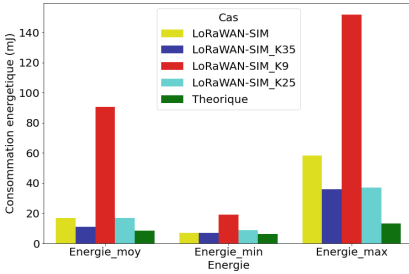


Fig. 10. Energy consumption

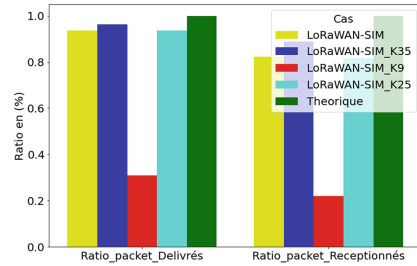


Fig. 11. Ratio of delivered/received packets

network, the data collection time, the transmission reliability, the reduced number of gateways in the network, the transmission speed (packet delivery rate). Indeed, a better positioning of gateways allows us to use good spreading factors, which are important settings to maintain an acceptable level of power consumption of the sensor devices, to have a larger coverage, a good quality of service, just to mention a few. On the other hand, our prototype takes just a little longer to generate the simulation results (5 s longer on average). **K-LoRaWAN-SIM** is accessible at the URL <https://github.com/antoinetsagmo/K-LoRaWAN-SIM>.

6 Conclusion and Future Work

LPWAN technologies are gaining more importance in almost every area of life in this increasingly connected world. This paper presents the best LoRa simulators in the literature to our humble knowledge. However, **LoRaWAN-SIM**, an open-source simulator has caught our attention thanks to its ease of use and the numerous results it provides. Following the experiments, in particular by visualizing the layout of the gateways generated by the tool, we realized that the essential problem of simulation in a LoRaWAN network can be assimilated to that of bad placement of gateways. We provided a solution based on the K-means machine learning technique.

K-LoRaWAN-SIM is available online at <https://github.com/antoinetsagmo/K-LoRaWAN-SIM>. It allowed us to conduct experiments and compare the results with those of the LoRaWAN-SIM reference tool available at <https://github.com/deltazita/LoRaWAN-SIM>. This new LoRa network simulator offers the possibility to realize a better placement of gateway nodes in the network. It also allows realizing an interconnection guaranteeing a better coverage with a minimal number of gateway nodes while keeping a correct level of performance and quality of service. We have seen that K-LoRaWAN-SIM provides almost identical results to the basic LoRaWAN-SIM tool while saving about ten gateways on the following test parameters: energy consumption, total number of unique transmissions, the total number of lost packets, the ratio of delivered and received packets, etc.

In the future, we are interested in scaling up. Indeed, for execution time constraints, we did not simulate large networks corresponding to real-life scenarios for applications that can cover a geographical area on the scale of a city for example. This type of scenario has the characteristic of generally requiring a very large number of sensor nodes.

Acknowledgements. This work was carried out in the WiCoNet laboratory of the Ecole Nationale Supérieure Polytechnique and UMMISCO of Computer Science Department of Yaoundé 1 University.

References

1. Bouras, C., Gkamas, A., Salgado, S.A.K., Kokkinos, V.: Comparison of LoRa simulation environments. In: Barolli, L., Hellinckx, P., Enokido, T. (eds.) *Advances on Broad-Band Wireless Computing, Communication and Applications*, pp. 374–385, Cham (2020). Springer International Publishing
2. Sinha, R.S., Wei, Y., Hwang, S.-H.: A survey on LPWA technology: LoRa and NB-IoT. *ICT Express* **3**(1), 14–21 (2017)
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 Commun. Surv. Tutorials* **21**(2), 1561–1581 (2019)
4. Croce, D., Gucciardo, M., Tinnirello, I., Garlisi, D., Mangione, S.: Impact of spreading factor imperfect orthogonality in LoRa communications. In: Piva, A., Tinnirello, I., Morosi, S. (eds.) *Digital Communication. Towards a Smart and Secure Future Internet*, pp. 165–179, Cham (2017). Springer International Publishing
5. Tiago Afonso Antunes de Figueiredo. *Lorawan performance evaluation* (2019)
6. Silva, J., Flor, D., de Junior, V.A., Bezerra, N., Medeiros, A.: A survey of LoRaWAN simulation tools in ns-3. *J. Commun. Inf. Syst.* **36**(1), 17–30 (2021)
7. Reynders, B., Wang, Q., Pollin, S.: A LoRaWAN module for ns-3: Implementation and evaluation. In: *Proceedings of the 10th Workshop on Ns-3, WNS3 '18*, pp. 61–68, New York, NY, USA (2018). Association for Computing Machinery
8. Bor, M.C., Roedig, U., Voigt, T., Alonso, J.M.: 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, MSWiM '16*, pp. 59–67, New York, NY, USA (2016). Association for Computing Machinery
9. Abdelfadeel, K.Q., Zorbas, D., Cionca, V., Pesch, D.: Fine-grained scheduling for reliable and energy-efficient data collection in LoRaWAN. *IEEE Internet Things J.* **7**(1), 669–683 (2020)
10. Callebaut, G., Ottoy, G., van der Perre, L.: Cross-layer framework and optimization for efficient use of the energy budget of IoT nodes. In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6 (2019)
11. Marini, R., Mikhaylov, K., Pasolini, G., Buratti, C.: LoRaWANSim: a flexible simulator for LoRaWAN networks. *Sensors* **21**(3) (2021)
12. Ta, D.-T., Khawam, K., Lahoud, S., Adjih, C., Martin, S.: LoRa-MAB: a flexible simulator for decentralized learning resource allocation in IoT networks. In: *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 55–62 (2019)
13. Zorbas, D., Caillaouet, C., Hassan, K.A., Pesch, D.: Optimal data collection time in LoRa networks—a time-slotted approach. *Sensors* **21**(4) (2021)

14. Slabicki, M., Premsankar, G., Di Francesco, M.: Adaptive configuration of LoRa networks for dense IoT deployments. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9 (2018)
15. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.* **31**(8), 651–666 (2010). Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)
16. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, number 14 in 1, pp. 281–297. Oakland, CA, USA, (1967)