Comparison of LoRa Simulation Environments

Christos Bouras[1,2], Apostolos Gkamas[3], Spyridon Aniceto Katsampiris Salgado[2], Vasileios Kokkinos[2]

[1]Computer Technology Institute & Press “Diophantus”, Patras, Greece
[2] Computer Engineering and Informatics Dept. Univ. of Patras, Greece
[3] University Ecclesiastical Academy of Vella, Greece

Abstract. Internet of Things (IoT) is used more and more in our everyday life, connecting different wireless devices, and their distance can vary from some centimeters to many kilometers. New IoT technologies such as Long Range (LoRa) are emerging enabling power efficient wireless communication over very long distances. Simulation of LoRa networks is quite important, because can be used for the design and the evaluation of LoRa based application without the need of costly implementations or before to proceed to the actual implementation of the system. Choosing the right parameters of the systems like spreading factor can improve the energy consumption of the wireless devices. This paper presents the most important LoRa simulation environments available in the literature and after that, we present a comparative evaluation of LoRa simulation environments. The benefits, the disadvantages and the highlights of each LoRa simulation environment is presented.

1 Introduction

Nowadays, Internet of Things (IoT) has been established in our everyday life, as it offers a number of capabilities. So, more and more devices and systems are being created that aim to offer solutions that need technologies that can interconnect wireless devices over long distances. Two candidates that try to solve this are LoRa technology [1], proposed by the LoRa Alliance, offering a low power, wide area network protocol for IoT devices and Narrow Band Internet of Things (NB-IoT) technology as part of 5G networks [2]. So, it is necessary to study this kind of technologies, because the IoT market is gaining exponential popularity introducing more solutions, and our lives can be improved in various ways.

LoRa and NB-IoT have been evaluated and compared in a high extent, using metrics such as latency performance Quality of Service (QoS), range, coverage, battery life, cost efficiency [3], as these two technologies are very promising LPWAN (Low Power Wide Area Network) technologies. LoRa could be a good choice to integrate into IoT applications. Choosing the right combination of the spreading factors and bandwidth can reduce the transmission time, increase the data rate, thus improving the energy consumption of the IoT devices [3][4]. Despite this, LoRa is an asynchronous protocol, so the engineers and scientists are able to choose the periods of sleep mode that the devices can enter, in contrast to the NB-IoT that needs synchronization which leads to additional power consumption. In conclusion, LoRa should be used to send small data packets over large distances.

LoRa [17] is an emerging new technology which enable power efficient wireless communication over very long distances. LoRa is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. LoRa Technology enables smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, disaster prevention, and more. LoRa Technology has amassed over 600 known uses cases for smart cities, smart homes and buildings, smart agriculture, smart metering, smart supply chain and logistics, and more. With 97 million devices connected to networks in 100 countries and growing, LoRa Technology seems to be a very important technology for implementation of IoT, Networks. Devices typically communicate directly to a sink node which removes the need of constructing and maintaining a complex multi-hop network. LoRa provides a range of communication options (center frequency, spreading factor, bandwidth, coding rates) from which a transmitter can choose. Simulation of LoRa networks is very important, because can be used for the initial design and evaluation of LoRa based applications without the need of costly implementations.

All these LoRa simulators have been used for testing various scenarios, but there is not, to the best of our knowledge, any a comparison between them. For example, PhySimulator has been used for the evaluation of link-level of LoRa, showing that despite the theoretical point of view that spreading factors can be considered as orthogonal, in reality the inter-spreading factor collisions are problem to LoRa [5]. FloRa simulator has been used for the evaluation of the performance of LoRa using Adaptive Data Rate (ADR) mechanism, showing that ADR is an effective way to increase the delivery ratio in an energy efficient way [9]. LoRaSim has been used to test the scalability of the LoRa Low-Power Wide-Area networks [6]. Last but not least, the LoraWAN Module for ns-3, was created in order to provide to the community a powerful tool for real networks, instead of using a simplified MAC protocol. So, it implements in addition to the LoraSim the acknowledgments, so as to give the ability to the users to test a network where the spreading factor can change according to given feedback from the gateway [11].

The aim of this paper is to study and examine the most common LoRa simulators that are available in the literature, provide a detailed comparison featuring their main advantages, disadvantages and showcasing the various highlights. The above comparison provides to the academic community the required data to help choose the necessary and more suitable simulator depending on the needs, the condition, the knowledge and programming experience of the user. Such features will be the operating systems that are supported, the programming language that is used, the existence and quality of the graphical user interface, the community support etc.

The rest of this work is organized as follows: Next section describes the basics of LoRa technology. We briefly discuss the basic network simulation process in section 3. The section 4 presents the features of the LoRa simulation environments available in the literature. After that, the section 5 presents a comparative evaluation of these LoRa simulation environments. Finally, section 6 presents the conclusions and section 7 discusses the future work of our paper.

2 LoRa Technology

LoRa technology can be considered as two parts, the first one is the LoRa and the second one is the LoRaWAN. LoRa is a proprietary wireless technology owned by Semtech, derived from the chirp spread spectrum modulation (CSS), while LoRaWAN is an open standard network protocol [1].

The physical layer of LoRa aims to provide the ability to communicate in an energy efficient way for energy constrained devices over long distances, supposedly coverage capability over 15 km Line of Site (LoS). The fact that is similar to chirp spread spectrum modulation allows to provide to the network a trade-off between the data rate for sensitivity within a channel bandwidth. Moreover, this fact helps in maintaining the low power features of the frequency shifting keying, while increasing the coverage, with low cost. But the fact that physical layer of the LoRa is proprietary means that we don't have a lot of information available and the documentation it is not freely available to the scientific community [1][4].

In Fig.1. is shown the LoRa stack, depicting the layers of LoRa technologies.

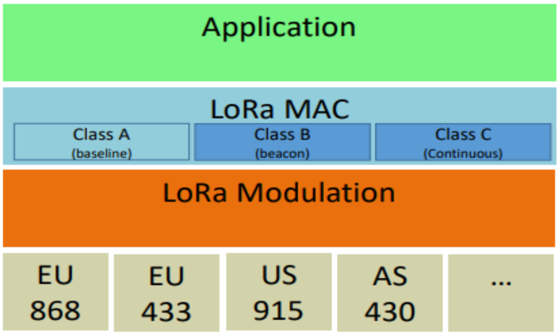

Fig. 1. The LoRa stack [1].

In contrast to the physical layer of LoRa, LoRAWAN is promoted by the LoRa Alliance that is consisted by various companies such as Semtech, IBM, etc. It defines the communication protocol of the network and is the MAC (Medium Access Control) layer protocol. Its primary goal is to determine in the best way possible the battery lifetime of the device the network capacity, the QoS (Quality of Service) etc. The LoRaWAN protocol categorizes the devices-nodes into 3 district classes A, B and C. All devices must support at least the Class A functions. The other two classes are optional and depending on the hardware can be certified for the other classes. So, class A enables the communication of the end devices with the server, where each end device sends uplink transmission and follows two small downlink windows. It is the most power efficient class. Class B adds to class A the ability to the end devices to open extra downlink transmissions with fixed latency. Last but not least, Class C considers that the end devices have continuously open receive windows. From the three classes, class A is the most power efficient and Class C needs the biggest amount of energy in order to operate.

3 Network Simulation

Network design with specific parameters is a challenging task, because the choice of the different parameters is not unique but should be suitable for the specific condition or application. For this reason, it is important to use a simulator.

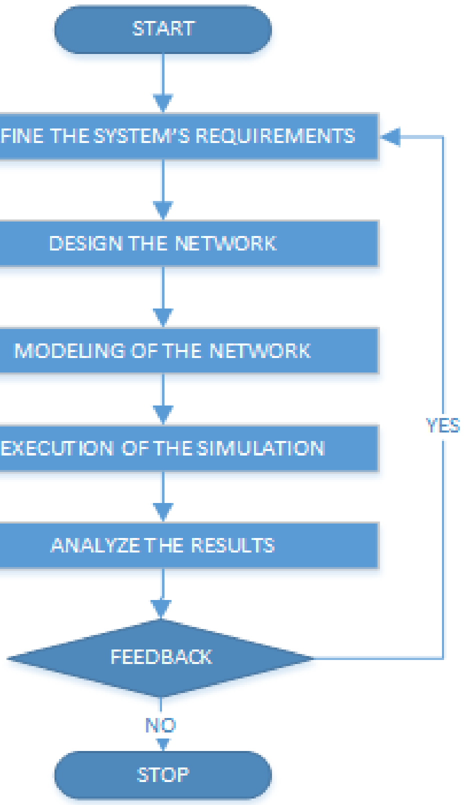

Fig. 2. Flowchart describing the process of the simulation.

Generally, a simulation software should provide the user the ability to define the network topology, to specify the characteristics and features of each node, the link between them and the traffic model and the packet routing algorithms that can be used. Moreover, the user should get the performance metrics for the simulated network and if it is available to the simulation software to get some visualization of these metrics. These metrics for a LoRa simulation could be Data Extraction Rate

(DER), Network Energy Consumption (NEC) [6] and bit error rate etc. There are simulators that provide a good graphical interface, and others that can provide only some graphical representations through plots, or even command line like outputs. In general, simulators can be free open source, or commercial, but here we will examine only free simulators for academic purposes or open-source, and until now there are not priced LoRa simulators. Another important aspect of a simulator is its performance.

First of all, as Fig. 2. shows, the users should define clearly the problem that is to be solved, in order to define correctly the system's requirements and the environment. Then they can proceed to the modeling of this network, so to define the details in a greater extent and set the necessary parameters. The next step is to execute this simulation, so as to get the results. Finally, after the analysis of these information can repeat again all these steps with the given feedback to the simulation software or stop the process and analyze the results.

4 LoRa Simulation Environments

In this section we present the following LoRa simulators:

- PhySimulator [7]
- FLoRA [8]
- Ns-3 module [11]
- LoRaSim [12]

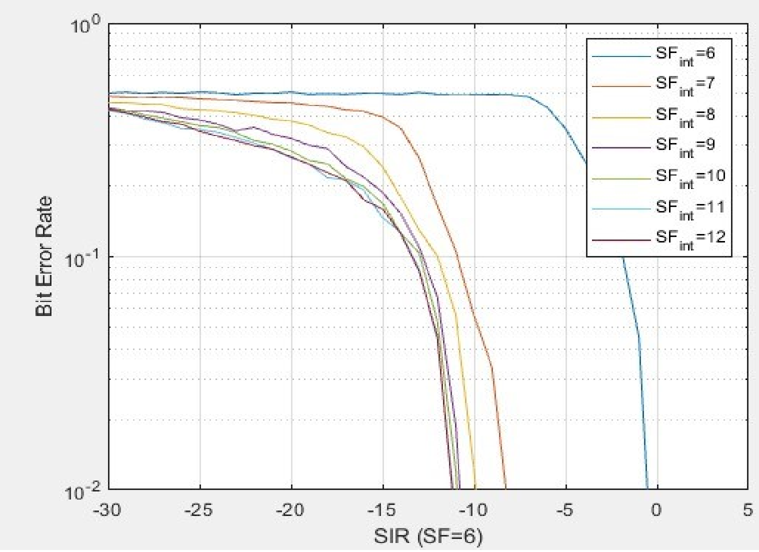

Fig. 3. An example of plot the BER-SIR.

4.1 Physimulator

The first simulator is called PHY Simulator and aims to implement the link level of LoRa. PhySimulator is written in MATLAB. The aim of this simulator is to test the reception of two overlapping-interfering LoRa transmissions that have been modulated with different spreading factors.

After each run of the program, 8 figures are generated, showing the packet, symbol and bit error rate. Specifically, the output is the packet, symbol and bit error rate for each spreading factor, interfered with any other spreading factor.

This simulator gives the ability to the user to edit various parameters (changing the values of the variables on the code). For example, you can change the bandwidth, the payload bits and the max trials per step etc. All these parameters cannot change through a graphical interface, but the user has to edit them changing directly the MATLAB code. Next it is presented a plot showing the bit error rate and the SIR using the physimulator. In Fig. 3. we present an example of plot using physimulator

4.2 FLoRa

The FLoRa simulator, is a simulation framework, using the well-known OMNeT++ discrete event simulation library that is distributed under the Academic Public License, so it is free for non-profit or academic use. In spite of OMNeT++ framework, FLoRa, is based also on the INET Framework, which is an open-source library for OMNeT++ and its purpose is to help the process of experimentation for different network protocols. FloRa is written in C++.

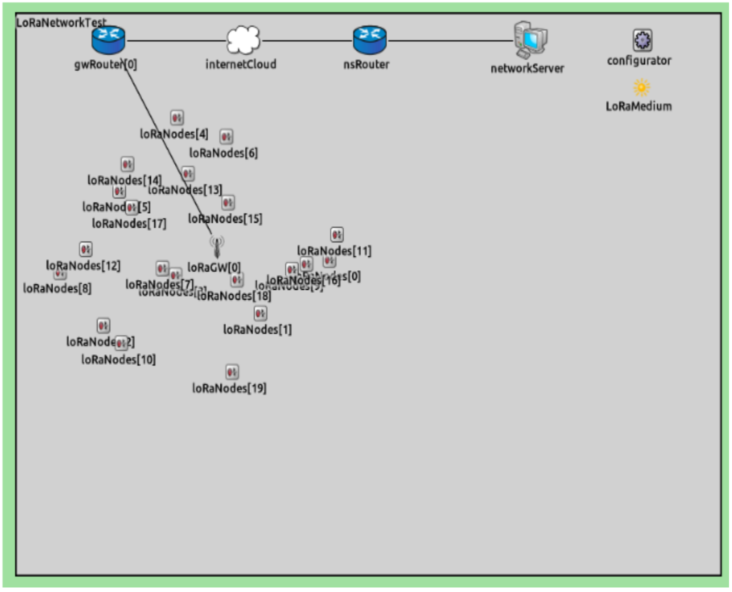

Fig. 4. An example of FLoRa [8].

In Fig. 4. we present an example of the GUI result of FLoRa simulator.

It allows the creation of LoRa nodes, gateways and a network server [8]. Furthermore, its modules aim to simulate the physical layer, and the LoraWan MAC protocol [9]. In contrast to the other simulation software, this provides a very good graphical interface, because it is based on the OMNeT ++ and a graphical representation of the network. It offers an accurate model for the LoRa physical layer. It emphasizes to giving statistics for the energy consumption of the network as well.

The features of FloRa simulator according to [8] include the following:

- Accurate model of LoRa physical layer (including collisions and capture effect)
- Simulations with one (or more) gateways in the network
- End-to-end simulations, including accurate modeling of the backhaul network
- Statistics of energy consumption in network

After each run, a number of files containing the statistical data are created. Also, the FLoRa provides some sample scenarios.

4.3. NS-3 Module

This simulation software is not actually a distinct simulator, but a module that plugs in the ns-3, a free discrete event network simulator, that is designed for academic and research purposes. It supports a big range of protocols and networks including IP and non-IP networks, and wireless simulations. Ns-3 is designed with modularity in mind and provides the ability to work in both graphical interface and command line. It is written in C++ and python. This particular module it is compliant with class A of the LoRaWAN 1.0 specifications. This means that it simulates the case where the devices send only uplink transmission and the server sends only downlink transmission. This class is the most energy efficient end-service system, in contrast to the other two available classes of LoRa. The physical layer, the MAC layer and the Transport and Application has been created, trying to give an agile and highly configurable solution. It gives the ability to integrate new algorithms on the server side to this ns-3 module [11]. Moreover, this proposed module contains an extra class in order to track the power consumption named LoRaRadioEnergyModel, tracking the energy of the states of the physical layer of the LoRa protocol. Furthermore, it was tested and evaluated by its creators using three scenarios.

- The first one considers a circular topology with one gateway that is located in the center, without the need of acknowledgments
- The second one uses more gateways, assuming circular topology too.
- The last scenario considers a network like the first scenario with the difference that the messages now need confirmation. This scenario according to the authors of [11] shows that is impossible to acknowledge all the packets.

4.4 LoRaSim

Finally, the last tool examined is the LoRaSim, is a discrete event simulator which aim is to analyze the ability of scalability of a LoRa network and the collisions. It allows us to place the LoRa nodes in a 2-dimensional grid. The LoRaSim is written in python ver.2.7, something that is considered as a drawback because only version 3 of python is updated. Also, it uses NumPy, matplotlib and SimPy python libraries. This tool has four different simulations, each one for different properties of the network and the nodes. More specifically, there are simulations for single base station and one that can simulate up to 4 base stations. Another categorization of the simulations is the type of the antennae, as there are files that simulate nodes with directional antennae.

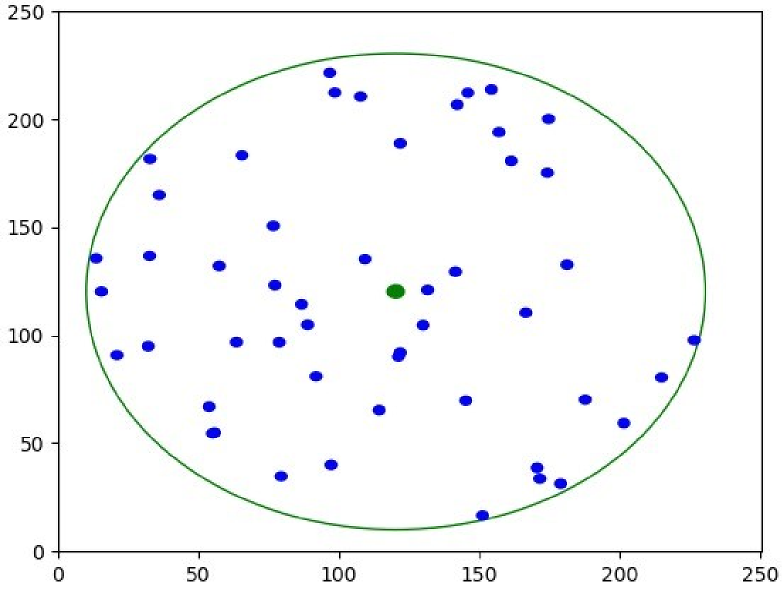

Fig. 5. An example of LoRaSim plot, using one sink, 50 nodes (blue: nodes, green sink).

In this simulator, you can change a number of parameters, such as the number of the nodes, the number of the base stations to simulate. Furthermore, it is possible to choose between full and simplified check of the collisions, the number of the LoRa networks, the time that the simulation will run and the distance between two base stations.

LoraSim, when executed provides the user some plots if the variable graphics is set to 1, but the majority of the information is presented through the command line and it exports these details to a file called expX.dat. Apart from the visualizations through the plots there is not a graphical interface, and the user has to work through the command line. Below in Fig. 5. there is an example of the plot that is created, in the case of one sink and fifty nodes.

5 Comparative Comparison

In this section, we present an overall comparison of the features of the examined simulators. Some of the features we have evaluated are the operating system support, the license type, the GUI, the availability of the statistics of energy consumption and the available documentation.

Table 1. Comparison Of The Simulators

Features	PhySimulator	FLoRa	Ns-3 module	LoRaSim
Event	discrete	discrete	discrete	discrete
License Type	free	Open source (study and research)	Open source	Creative Common Attributes 4.0
Language	Matlab	C++	C++, python	python
Operating System	Windows, Linux, MacOS	Windows, Linux, MacOS	Linux, MacOS, Windows through virtualization	Windows, Linux, MacOs
GUI	Only plots	yes	yes	Only plots
Energy Consumption statistics	no	yes	yes	no
Documentation	Ok	ok	average	ok
Number of published papers	2	1	1	2
Website	yes	yes	no	yes
Community Support	Good	Limited	Very Good	Limited

The reasons we chose to evaluate all these features are the following: it is very important for the user to know in what platform should work, what programming languages should know for the specific problem that has to be solved and especially for the scientific community and the researchers to know if it is open source. This is necessary in order to expand the exissting tools taking into consideration additional scientific aspects, novelties and as happened with the ns-3 module to use the existing protocols and “insfrastracture” for other protocols such as LoRa. In this framework we examine the existance of related publications, website and the community that support each software. In addition to this, in order to understand and evaluate the results of the simulation so as to check your system model, it is also necessary to have some visualized data and statistics, for this reason this feature is examined too.

The comparison of the features of each simulation software is shown in table 1. As we see in the table where there are all the features of the examining simulators side by side, and we can say that all available simulators are quite good.

First of all, all four simulators are discrete event. This means that they are modelling the system as a sequence of discrete events in the time domain. This allows the simulators to move to the next event, assuming that there is no change in the system between two consecutive events, thus there is no need to track the system continuously.

Concerning the programming languages which have been used in the implementation of the simulators, all the simulators are based on well know programming environments with a great support community. The above is very important because a researcher can easily expand the simulator capabilities by implementing new modules (for example support new network protocols or integrate additional tools on the existing available environment etc). In summary, PhySimulator is implemented in Matlab, FLoRa is implemented in C++, Ns-3 module is implemented in C++ and python and LoRaSim is implemented in python.

The FLoRa through OMNeT++ and Ns-3 through NetAnim have a more extended graphical interface compared to the other simulators. PhySimulator and LoraSim give only some plots. All the examined simulators have been published to the scientific community and FloRa, ns-3 module have 1 related publication and PhySimulator and LoRaSim have 2, and all simulators have their own website except the Ns-3 module. Nevertheless, Ns-3 is an open source project that has a great and big community that supports it [13]. FloRa, Physimulator, LoRasim have both related publications and a related website in contrast to the Ns-3 module that has only related publication. For this reason, the three simulators have more detailed information about the process of installation and how to use the tools. The Ns-3 module is open source and the code is available on Github.

LoRA is a technology for the implementation of IoT applications where the devices are powered usually by a battery, so the optimization of energy consumption is a key factor for the deployment of IoT applications. As result, the investigation of energy consumption through Energy Consumption statistics is a very important feature of a LoRA simulator. From the simulators which we study in this paper only FLoRa and Ns-3 module supports Energy Consumption statistics.

Another feature that was examined is the license of each simulator. FloRa is free for academic and study use. LoRaSim on the other hand is provided under the license Creative Commons Attribution 4.0. This means that is free to use, share and change the code, giving the appropriate credit to the authors and provide link with the license.

All simulators can be run in all operating systems in one way or other. PhySimulator can be executed wherever Matlab can be executed, so the supported operating systems include Linux MacOs and Windows. In the same way, LoraSim can be used in whatever operating system can support python. So LoraSim is used in Windows, Linux and MacOs. The ns-3 module, because it is based on ns 3, it can be executed on Windows only by using a virtual machine. It supports Linux and MacOS natively. As we have mentioned before, one very important aspect is the energy consumption. The FloRa simulator and the Ns-3 module focus on providing the user this kind of information.

One other important factor which plays an important role in the usability of the simulator, is the detailed documentation of the simulator operation. Except of Ns-3 module (which offers an average documentation), all the other simulators provide a detailed documentation something which is very important. Also, there is a number of tutorials for the ns-3 and OMNeT++ simulators that can be used in order to understand how each simulator works. If we discuss for the community support, FLoRa and Ns-3 module seems to have the better community support, because both the underline network simulators, Ns-3 and OMNet++ respectively, have a strong user community. Ns-3 being open source means that there are less maintainers to respond to questions or fix reported bugs and abnormalities. However, it is extremely widespread and used by many students, scientists and academics that the online community can help and offer great support.

More specifically, according to [14] a Google Scholar search of the ‘ns-3 simulator’ results since 2017 (excluding patents and citations) yields over 2000 links (with some false positives). In addition, the IEEE digital library lists 145 ns-3 publications for 2017, and the ACM digital library lists 2579 publications matching the search term ‘ns-3’ in 2017. In addition, there are organized Workshops on Ns-3 and the related proceedings are published in the ACM digital library [15]. The above facts ensure the important acceptance of ns-3 simulator as network research tool. In addition to Ns-3, also OMNeT++ has an active community which have organized 5 OMNeT++ Community Summits until 2018 [16]. As result, if we compare the above simulators in terms of research community support seems that Ns-3 and OMNeT++ have the most active research community which organize relative workshops about the evolution of the simulation software. This seems reasonable based on the fact that both Ns-3 and OMNet++ can be obtained at no cost.

6 Conclusions

As we have said, LoRa is an ideal solution for long range communication of the IoT. It is known, that choosing the right variables such as spreading factor is very crucial, in order to maintain an acceptable level of power consumption of the device, to have greater coverage, good QoS etc. So, in order to find the best tradeoff between the above, it is helpful to use a simulator, as it can help the developers and scientists to choose the right parameters, executing the simulations with low risk and without the need of expensive implementations and investment. Moreover, it is possible to compare different protocols and technologies using simulators and evaluate the advantages and disadvantages of each technology. In this paper, we tried to give to the scientific community a general overview of the most common simulators available to the literature for LoRa technologies and a comparison between them.

7 Future work

Future work is the study and implementation of energy efficient mechanisms and systems for IoT devices using LoRa technology, with emphasis to search and rescue systems with various health data and medical monitoring sensors. So, the use of simulators is a key aspect of this process and a helpful tool, before we start the implementation of such systems, study the feasibility of such actions. In the context of our future work we plan to use LoRa simulators in order to optimize energy efficient algorithms in search and rescue systems with various health data and medical monitoring sensors.

Acknowledgments. The authors acknowledge that the starting point and the drive of this paper have been generated in the context of the project 'WeSAR (Wearable based Search And Rescue system)'. In this framework, we would like to thank all the project partners and employees of Econais AE, Yodiwo AE and CTI (Computer Technology Institute & Press "Diophantus") for their personal copy information and experience on this project. This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T1EDK-01520).

References

1. LoRa alliance official site, https://lora-alliance.org/ (accessed on 9/7/2019)
2. Mobile IoT in the 5G future, https://www.gsma.com/iot/wp-content/uploads/2018/05/GSMA-5G-Mobile-IoT.pdf (accessed on 9/7/2019)
3. Sinha, R. S., Wei, Y., Hwang, S. H., A survey on LPWA technology: LoRa and NB-IoT, ICT Expr.3(1), 14-21.
4. Bouras, C., Kokkinos, V., Papachristos N., Performance evaluation of LoraWan physical layer integration on IoT devices, in Global Information Infrastructure and Networking Symposium (GIIS 2018), Thessaloniki, Greece, 2018.
5. Croce, D., Gucciardo, M., Mangione, M., Santaromita, G., Tinnirello, I., Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance, in IEEE Communications Letters, vol. 22, no. 4, pp. 796-799, April 2018.doi: 10.1109/LCOMM.2018.2797057
6. Georgiou, O., Raza, U., Low Power Wide Area Network Analysis: Can LoRa Scale? in IEEE Wireless Communications Letters, vol. 6, no. 2, pp. 162-165, April 2017. doi: 10.1109/L. WC.2016.2647247
7. PhySimulator site, http://lora.tti.unipa.it/ (accessed on 9/7/2019)
8. FloRa official site, https://flora.aalto.fi/ (accessed on 9/7/2019)
9. Slabicki, M., Premsankar, G., Di Francesco, M., Adaptive configuration of lora networks for dense IoT deployments, NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, 2018, pp. 1-9.doi: 10.1109/NOMS.2018.8406255
10. NS3 simulator official site, https://www.nsnam.org/about/ (accessed on 9/7/2019)
11. 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). ACM, New York, NY, USA, 61-68. DOI: https://doi.org/10.1145/3199902.3199913
12. LoRaSim site, https://www.lancaster.ac.uk/scc/sites/lora/ (accessed on 9/7/2019)
13. Kabir, M., Syful I., Hossain, Md., Hossain, S., Detail Comparison of Network Simulators. 10.13140/RG.2.1.3040.9128.
14. NS-3 Statistics https://www.nsnam.org/about/statistics/ (accessed on 9/7/2019)
15. NS-3 Workshops https://www.nsnam.org/research/wns3/ (accessed on 9/7/2019)
16. OMNeT++ Community Summit 2018 https://summit.omnetpp.org/archive/2018/ (accessed on 9/7/2019)
17. Semtech LoRa Technology Overview | Semtech, https://www.semtech.com/lora (accessed on 9/7/2019)