

Performance Enhancements of LoRaWAN Using Machine Learning on the Edge

by

Anuar Oтынshin

B.S., Nazarbayev University (2014)

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at the

NAZARBAYEV UNIVERSITY

Apr 2021

© Nazarbayev University 2021. All rights reserved.

Author
Department of Computer Science
Apr 11, 2021

Certified by.....
Dimitrios Zorbas
Assistant Professor
Thesis Supervisor

Accepted by
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

Performance Enhancements of LoRaWAN Using Machine Learning on the Edge

by

Anuar Oтынshin

Submitted to the Department of Computer Science
on Apr 11, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

LoRaWAN is becoming the dominant long range protocol for Internet of Things (IoT) devices. However, LoRaWAN's performance suffers from a high number of collisions in saturated LoRa networks. To mitigate the number of collisions that happen due to the time overlap of transmissions on the same channel, we use an edge machine learning approach. To do so, Reinforcement learning (RL) is leveraged. RL is a field of machine learning that aims at maximizing the reward by interacting with the environment. SARSA is an on-policy RL algorithm that uses previous actions to update the Q-value. This study aims to improve the performance of congested LoRa networks by allowing RL-based applications on individual nodes. Specifically, it explores whether periodic applications driven by SARSA can improve the performance of the network and adapt the period transmissions of the nodes. In this thesis, two versions of SARSA have been developed, evaluated, and compared to the baseline of LoRaWAN. To achieve that, several simulations with different configurations are performed. The simulations include networks with hundreds of nodes and different number of maximum retransmissions. The results of the simulations have shown that networks where SARSA algorithms are used present a better performance compared to the typical LoRaWAN periodic application in certain examined scenarios. The results demonstrate that RL-based algorithms can significantly improve the performance of networks with high load. Nevertheless, there is still room for further improvement and better understanding of the internal mechanisms of the proposed RL approaches.

Thesis Supervisor: Dimitrios Zorbas

Title: Assistant Professor

Acknowledgments

I have received a great deal of support and assistance while writing the dissertation. First, I would like to express my gratitude for my thesis supervisor, Professor Dimitrios Zorbas, who supported me throughout the writing process and contributed his ideas that formed the basis of this study. Your proficient knowledge and perceptive feedback allowed to develop my work to higher degree. Second, I would like to thank Professor Mona Rizvi, who directed me at the early stages of developing my dissertation and helped with selecting my supervisor. Finally, I would like to acknowledge my parents for their constant emotional support and listening for my ideas and thoughts.

Contents

1	Introduction	13
2	Research Methodology	17
2.1	Literature Review	17
2.2	Algorithmic Design	18
2.3	Implementation	19
2.4	Evaluation	19
2.5	Conclusions	20
3	Literature Review	21
3.1	Classification	21
3.1.1	Research Focus	21
3.1.2	Evaluation Methods	25
4	System Architecture	27
4.1	System Model	27
4.1.1	Baseline	27
4.1.2	The SARSA-based RL algorithms	29
5	Results & Discussion	33
5.1	Results	33
5.2	Discussion	36
6	Conclusion	37

List of Figures

1-1	LoRa modulation	14
4-1	Nodes Distribution (a. 100 nodes, b. 200 nodes, c. 500 nodes)	28
5-1	PDR throughout the simulation (100 nodes, $r = 8$)	35
5-2	PDR throughout the simulation (500 nodes, $r = 4$)	35
5-3	PDR throughout the simulation (500 nodes, $r = 8$)	36

List of Tables

5.1	Average PDR (100 nodes)	34
5.2	Average PDR (200 nodes)	34
5.3	Average PDR (500 nodes)	34

Chapter 1

Introduction

Internet of Things constitutes to the connection of a large number of devices to the Internet. New developments, protocols, and devices has been recently proposed to make the concept of IoT a reality but also to facilitate the rapid growth of the number of devices. IoT finds applications in a large number of every day domains such as the healthcare, the agriculture, and the smart cities. The main medium used by the IoT devices to communicate is the air through wireless communications and networks. Low Power Wide Area Networks (LPWAN) is the type of networks that allow long range and power efficient communications for IoT devices.

One of the most prominent LPWAN technologies is the LoRa radio technology. LoRa (Long Range) is a proprietary modulation based on the Chirp Spread Spectrum technique. According to this technique all the available channel bandwidth is used while the chirps are spread diagonally during a transmission as it is depicted in Figure 1-1. The amount of spread to use is decided by a parameter called Spreading Factor (SF). The higher the SF, the longer the range but the longer the transmission time and, thus, the energy consumption. LoRa is known for its high customizability and low power consumption compared to other LPWAN technologies.

The LoRaWAN protocol [1] sits on top of LoRa at the MAC and link layers and it also provides some other functionalities such as end-to-end encryption, back-end connectivity, and adaptive LoRa settings management. In contrast with LoRa physical layer, LoRaWAN is an open source networking protocol. LoRaWAN supports

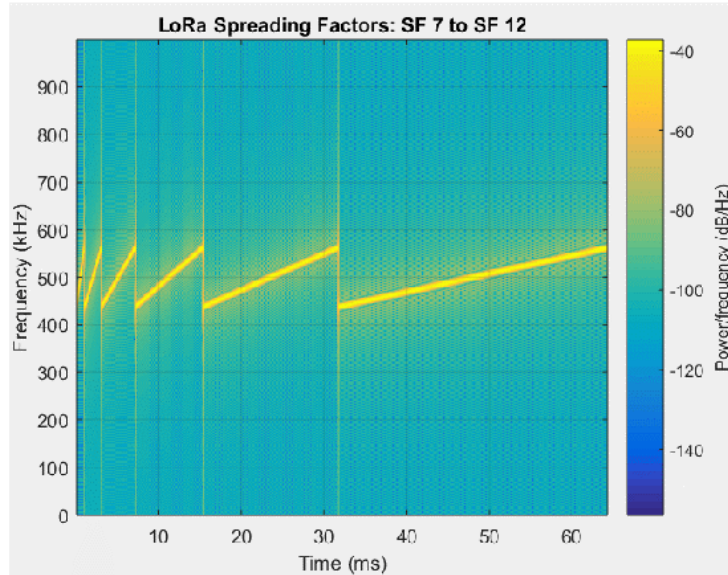


Figure 1-1: LoRa modulation

three classes of network devices: Class A, Class B, and Class C. Each class has its own features and use cases, but class A is the dominant one. All LoRaWAN devices support bi-directional communications, however, the main differences between classes are the downlink communication time window and energy consumption. In Class A, uplink communications can be initiated at any time by the end-device, followed by two receive windows, during which a downlink communication is allowed for acknowledgements and commands. Class B devices open downlink communication periodically and synchronized, while Class C devices open the downlink communication channel continuously. The Class A devices are the most power efficient, while Class C devices usually require continuous power supply. LoRaWAN, since it is at its early stage of development, there is room for improvements and modifications.

One of the main research challenges of LoRaWAN is the packet collision in highly populated networks. In the case when there are hundreds or thousands of network nodes, data may be lost due to collisions. This mostly happens due to the Aloha-based MAC that has been adopted. Moreover, a LoRaWAN end-node may send messages that require an ACK response from a gateway. These messages are called confirmed messages. Therefore, a network that has several nodes that communicate with a

gateway using confirmed messages is a network with confirmed traffic. According to the protocol, in confirmed traffic, a node is allowed to perform a maximum number of retransmissions (r). Basically, a node in confirmed traffic transmits the message and waits for an ACK packet from a gateway. Then, if no gateway responds, the message is considered a failure and the node transmits the message again, until the number of transmissions reaches r or the node receives an ACK from the gateway.

The main metric that is used to evaluate the performance of a network is the packet delivery rate (PDR). This is a ratio between the number of messages received by a gateway and the total number of messages sent by the nodes.

$$\text{packet delivery rate (PDR)} = \frac{\text{packets received at the gateway}}{\text{total number of packets sent}}$$

Often, devices in the IoT are responsible for sending small amount of data periodically. Thus, they usually support a periodic application that requires data reception of every few minutes to capture its behavior. In saturated networks, this might become a problem, because the devices do not perform any kind of medium sensing and, thus, they do not adjust their transmission policy accordingly.

In this study, we investigate the case where the nodes are allowed to individually select the transmission time depending on the status of previous transmissions, hoping that this technique will improve the PDR in periodic data transmissions. Using these past pieces of information, the nodes can learn the behavior of the network (given the ACKs that have been received or not) and choose an eventually better transmission timing for their next transmission. To do so, we leverage machine learning (ML). A machine learning approach is used to adjust overlapped transmissions and choose better timings for future transmissions. Nevertheless, a node only knows whether the last transmission was successful or not and, moreover, due to their low computational and memory storage capabilities, they cannot support too complex approaches. So, data-driven or pre-trained machine learning algorithms cannot be used. Reinforcement learning (RL) is a field in machine learning where an agent requires no predetermined information about the environment to increase the reward. It allows

agents to learn the environment by interacting with it. Despite the popularity of LoRaWAN, there is not much research done on how the protocol could benefit from RL algorithms.

The main contributions of the study are the description and classification of recent studies on LoRaWAN, the development of two application-layer RL algorithms (SARSA-1 and SARSA-2), and the evaluation of performance of the algorithms against the native LoRaWAN transmission policy for a periodic application.

In the next chapter, the research methodology followed throughout this study is presented. Chapter 3 surveys the literature, and Chapter 4 describes the system architecture and system model. In Chapter 5, the results of simulations are provided and the importance of the results is discussed. Finally, in Chapter 6, some future directions are considered and the study is concluded with some final remarks.

Chapter 2

Research Methodology

The main purpose of the study is to analyze the performance of the LoRa network, where nodes individually adjust their transmission time according to RL algorithm. To obtain this goal, existing data from similar studies was explored and evaluated, algorithms were developed conceptually and practically, the performance data from the algorithms was collected and analyzed, and several conclusions from these results were drawn.

2.1 Literature Review

To identify the most relevant and important studies, various research papers from different databases were collected, including the IEEE Xplore, the ACM Digital Library, and the NU Library. The papers were searched using several keywords related to the topics above. However, the number of papers found was too high to process and analyze, moreover, some studies were not relevant to the purposes or were out of scope of this study, so they were needed to be further filtered.

The filtering was done by using several selective criteria, which can be additionally split into objective and subjective criteria. The objective criteria were used to narrow down the studies based on quantitative metrics, such as duplicate papers, publication date, and availability. Hence, the duplicate papers were excluded from the review. Papers that are not publicly available or papers written in non-English language were

also removed from the analysis, the same for studies written before 2015, because the first version of the LoRaWAN protocol was developed in 2015, so any earlier studies would not be related to our study.

The subjective criteria are based on more qualitative features of the paper, such as the relevance to the current research. This type of filtering is based on the title, the abstract, and the contents of the paper.

Finally, after the final literature selection is identified, a comparative analysis of the studies was done and missing points of the research were determined. The detailed analysis and review is presented in the following chapter.

2.2 Algorithmic Design

The literature analysis helped to establish that most of research is focused on optimizing network performance by either adjusting network parameters or implementing the algorithms that operate on gateways and manage network processes by sending messages to end devices. It was recognized that there is little work done on exploring the influence of application layer parameters of individual nodes on network performance.

This led to the development of a concept that each node individually might learn the behavioral patterns of network and positively affect network performance. Data-driven algorithms are not useful for individual nodes as network system is a changing environment. Machine learning algorithms are particularly useful at recognizing patterns in systems. Usually, end devices in IoT environment are sending their data periodically. Therefore, theoretically, behavioral patterns of a network can be identified by individual nodes to optimize data transmission. RL is a machine learning paradigm that aims at increasing the cumulative reward in a certain environment. RL is usually applied in gaming environments, where an agent has to make different decisions from action space in order to maximize the reward (points, in-game currency, etc.). It is also effective in changing environments as the agents learn while making decisions. The key idea of RL algorithms is to make decisions based on prior knowledge. The RL algorithm that has gained much popularity in the last decade is

Q-learning. It is a model-free algorithm that does not require prior knowledge about an environment to operate.

2.3 Implementation

Once the ML algorithm design was identified, it was needed to implement the code for the algorithm and test it. Before implementing the algorithm, it was needed to determine the testing environment that was going to be used to test the approach. There are several ways to test the network system: developing a mathematical model, perform simulations, and conduct real-world experiments. Though the last method is the most effective, it is not feasible to implement, because it is needed to test a densely populated network with hundreds of devices which is not possible due to budget and time limitations of the study. The mathematical model might effectively predict the outcomes and results of the algorithm, however, simulations construct a closer to real-world environment, which can be easily controlled via network variables.

For the purposes of this study, it was decided to use the NS-3 simulator. NS-3 is an open-source simulator designed for building various network environments. It also has a LoRaWAN module [22] that implements different classes and functions to thoroughly set up LoRa network environment. The algorithm was developed as an application layer and the results were recorded by modifying some code parts of the module and simulator. The detailed algorithm description and implementation are given in Chapter 4.

2.4 Evaluation

After the algorithm was implemented, it was needed to run a series of experiments (simulations) under different conditions using the NS-3 simulator to evaluate the performance. According to the hypothesis, the algorithm should improve the overall performance of the network (i.e., the PDR) in high node populations. Hence, before running the simulations, there were a few things that were needed to be determined.

Firstly, it was needed to identify a baseline approach which would be used as a comparison with our approach, and thus, to prove or reject the hypothesis and address the research question. Secondly, it was needed to establish the main metric, which will be used to compare the approaches.

For the baseline approach, a simple periodic application set up on the nodes that periodically sends a fixed-size packet throughout the simulation was selected. This is a quite basic application that is expected to be used on a typical IoT device. To evaluate the results of the approaches, the packet delivery rate (PDR) was selected. It is one of the most popular metrics to evaluate network performance because it gives a clear picture of how many packets are delivered or not in the network. The results of the algorithmic performance and their description are presented in detail in Chapter 5.

2.5 Conclusions

Finally, once the results are analyzed, their implications and the applications of the algorithm are discussed. The general remarks are made based on the outcome and future work is also pinpointed.

Chapter 3

Literature Review

Despite the novelty of the LoRaWAN protocol, version 1.0.0 was published in early 2015, its promising potential facilitated many researchers to study the applications of LoRaWAN as well as to examine and improve the protocol. To develop a sufficient knowledge of existing trends in the field and to find out useful research methodologies, the most recent studies related to the topic were selected and described.

3.1 Classification

The classification of literature serves several purposes. Firstly, it helps to understand the main trends in the field and find out what areas might be improved with the study. Secondly, it is a good source for other researchers' methodologies that can be used in this research work. Thus, it was decided to classify the literature by research focus and by methodology.

3.1.1 Research Focus

In the past decade, the increased interest in the IoT led to exploration of new protocols to efficiently transfer data between devices over long distances. Due to its promising features, LoRaWAN became one of the most popular protocols in the area of LPWAN. The studies can be classified as practical and theoretical research. The

first type of research focuses on different applications and deployments of LoRaWAN in various domains, while the theoretical research studies the limitations and optimization techniques of the protocol in a more mathematical way.

Practical research

LoRaWAN is highly customized protocol that can be applied in different fields, such as smart cities ([10], [18], [4]), data monitoring ([20], [2], [10]), and search and rescue operations ([9]). In [10], the author used a sensor that collects temperature data from the environment and transmits it to the gateway over a long distance. The author concludes that the LoRaWAN is suitable for city-like environments, however, the study is quite limited in scope as it only considers one ED that communicates with the gateway. A more extensive study was done by [18] and [4], where the authors deployed networks as part of a smart city application in Hamburg and Bristol. The authors came to similar conclusions that the LoRaWAN is suitable for smart cities due to its long range and resilience against interference.

One of the key advantages of LoRa is the ability to transmit data packets over long distances. This feature is perfect for data monitoring applications in rural areas. This is because the radio utilizes only a small portion of battery, which is important if the sensors are placed in hard-to-reach locations. This benefit is explored in [20], where the authors installed LoRa end-devices on weather stations that inform about the weather conditions. Usually these weather stations are placed in remote areas to inform about upcoming disasters. Thus, long range networks are extremely useful in such situations.

Theoretical

LoRaWAN is a novel protocol but it has many limitations, hence, many researchers aim to improve different aspects of its functionality. For example, there are several studies ([14], [25], [13], [27]) that focus on the security procedures of LoRaWAN. Despite that each transaction is encrypted with AES encryption, there might still be potential to crack the network. In [14], the authors consider two types of attacks on

the network and determine the possible methods to prevent such types of attacks. Probably, the most popular type of attack on the networks are the Denial-of-Service (DoS) attacks. The authors of [25] investigate the protection against DoS attacks in LoRaWAN. However, the authors do not provide any solutions for preventing DoS attacks.

Another field of LoRa networks optimization is energy efficiency optimization. The key feature of LoRa is that it consumes a small amount of energy, thus, it extensively extends the end devices' battery lifetime compared to other long range solutions. However, there are still ways to improve energy efficiency of LoRaWAN devices. In [6], the authors implemented a model of capacitor that may store accumulated energy and supply it as needed for end devices. In [5], the authors investigated how to improve energy consumption using a slotted ALOHA approach in a highly populated LoRaWAN network.

Finally, there are many studies that focus on the improvement of the LoRaWAN's throughput. Many sources point out the decreasing performance of LoRaWANs in highly-populated environments ([16], [7], [3]). Thus, there is a need for optimization of networks with many nodes in a small area. The problem manifests itself notably in congested network with confirmed traffic. Authors of [7] suggest that incorrect use of confirmed traffic might severely degrade the performance in a highly congested network. The authors of [17] draw similar conclusions by conducting a survey on recent work to optimize confirmed traffic in LoRaWAN. In the study, authors investigate common factors that affect the performance of confirmed traffic, such as, the spreading factor, the number of re-transmissions, and the ACK timeout time. They found out that in its current state LoRaWAN support confirmed traffic well in small networks, however, as the number of communication links increases and as the data is sent more frequently, the performance drops significantly.

To accommodate the demand for optimization techniques in dense networks, researchers examined the problem and tried to implement optimization techniques. Many studies mention the significance of selecting the SF ([29], [16], [8], [17], [23]) and its influence on the average performance of the network. They attempted to

optimize it for different use cases and demonstrate that a better selection of SF parameter positively affects the PDR and allows to outperform the traditional ADR used in LoRaWAN. For example, in [8], the authors were able to identify the traffic type (more or less aggressive) and adapted the SF value individually for each device, which resulted in an improved performance of the network. This is similar to [23], where the authors utilized machine learning to select an optimal SF depending on the number of nodes in the network.

Speaking of machine learning, it opens great possibilities in a variety of fields in computer science. Computer networks is definitely one of the fields that benefit from ML algorithms. Possible applications of ML are shown in [24], where the authors compiled a survey on different ML approaches to improve the performance of wireless networks. In this study, a variety of applications were investigated, including 5G, WiFi and Bluetooth, which are key for the development of IoT. However, the survey does not contain any information about LPWA Networks. Nevertheless, it is important to understand how ML algorithms could be implemented in different layers of wireless networking. The paper contain thorough analysis of various ML algorithms such as Q-learning, Collaborative Filtering, Deep Learning, which are also described in [15] and [11], K-Means Clustering, and applications in resource management and network performance.

In the last few years, some attempts to apply ML algorithms in LoRaWAN appeared in the literature ([23] [19], [11]). The authors of [19] explore the specific use case of LPWAN, where network nodes are located indoor and the signal might be blocked by walls. The authors compare the LoRaWAN performance with existing indoor wireless solutions such as the WiFi, the Bluetooth, and the Zig-Bee, and they came to the conclusion that LPWANs, especially LoRaWAN, supersede existing solutions in many IoT applications. Also, they exploited deep learning approach in LoRaWAN to predict indoor location with 98% accuracy. However, the data collected from the study is quite small to make any conclusive results. An interesting approach was taken by the [11], where the authors applied deep reinforcement learning algorithm installed on the gateway to optimize the network. The results show

that the performance increases to up to 500% in some cases.

Nonetheless, most of research is focused on improving the network as a whole, from the perspective of the gateway or network servers. Studies that attempted to increase network performance by tweaking application level parameters were not identified. The work in [26] focuses on application layer parameters, however, the study is quite limited in scope and does not provide sufficient knowledge on the influence of application-layer parameters on the network performance.

3.1.2 Evaluation Methods

Every study has different approaches to apply and test LoRaWAN. The cases vary from employing the protocol in smart city environment, such as in [4], or testing in a localized environment like in [19]. In general, the studies can be classified by evaluation approach into three types: mathematical models, simulations and real-world testing. For example, in [3], authors construct a mathematical model that analyzes the relation between the network population and the packet error rate. Another example is the attempt to optimize LoRa parameters using ML techniques in [23]. The authors of [12] develop an analytical model that computes the energy consumption and delay for uplink transmissions. They also suggest that there is a need for developing mathematical models, because it is more time effective compared to other types of evaluation techniques. However, this type of evaluation requires sufficient mathematical knowledge and might miss various nuances of the real-world environment, such as buildings, mobility of end-devices and different obstacles. Thus, it is sometimes difficult to construct a sound mathematical model for specific environments.

Another way of evaluating LoRa networks is the real-world tests ([2], [4], [10], [18], [19], [20], [26]). This approach gives a more accurate representation of the environment and does not omit the factors described above. For example, LoRaWAN was utilized in a Things network in Bristol ([4]) and Hamburg ([18]). In [19], authors placed a temperature sensor in an indoor location to compare the performance with traditional wireless networks (5G, WiFi, Bluetooth). In application-focused practical studies ([2], [10], [20]), the authors used real-devices to communicate with the

gateway. Nevertheless, constructing a real-world environment might be too cost ineffective and it is difficult to simulate a highly congested network, because there are not many suitable environments with many end devices. Also, it might be challenging to perform extensive simulations with altering network parameters in such environments as it will take too much time.

The third evaluation method is performing simulations. Understandably, this method is the most popular among researchers ([5], [7], [6], [8], [9], [25], [16]) because it is easily scalable, flexible and time efficient. Moreover, most of the modern simulators are advanced enough to provide relatively accurate representation of the environment. There are many network simulators that are used by the researchers and each simulator serves the purpose of the research. For example, the work by [5] is focused on improving energy efficiency of LoRa devices. Therefore, the authors use Python-based LoRaEnergySim simulator to assess the results of their approach. In [25], the authors used CPNTools to develop a model of LoRaWAN using Colored Petri Nets (CPNs) and analyze the vulnerabilities to DoS attacks. However, these tools are quite narrow in scope and are focused for a specific set of objectives. So, more general network simulators that might help in assessing network performance are needed to be considered. In the selected literature, authors often rely on the following tools: MATLAB ([9], [16], [21]), SimPy ([8], [28]), LoRaFREE ([29]) and NS-3 ([6], [7]). Each tool comes with a special module for developing and evaluating LoRaWAN.

To sum up, existing literature on network optimization was reviewed. Specifically, attempts at improving LoRa networks were investigated. Also, the techniques used to evaluate the results of research were analyzed. As a result of the analysis, it was identified that there is little research conducted on the influence of application-layer parameters on network performance. In addition, it was concluded that using simulations to assess the results is the most practical and convenient to evaluate the network performance in congested environments.

Chapter 4

System Architecture

4.1 System Model

In simulations, a highly congested network with a single gateway located in the middle of the terrain and N nodes spread uniformly in radius $R = 7500m$, where N varies from 100 to 500 nodes, is considered. Examples of the nodes distribution for 100, 200, and 500 nodes can be seen in Figure 4-1. All nodes are static devices and periodically transmit confirmed messages over time $T = 20$ days within a period $p = 600s$ within a time window defined by the algorithm (hereafter defined as the *delay*). Also, LoRaWAN allows to set the maximum number of retransmissions. If a packet is not successfully acknowledged after this predefined number of retransmissions, it is dropped. The results are evaluated according to the achieved packet delivery rate (PDR).

4.1.1 Baseline

The study aims to develop an application layer algorithm that optimizes saturated networks, and the results will be compared to the results of the baseline approach, which is the typical LoRaWAN periodic application that transmits a packet every p amount of time, where p is constant. For example, if the the first packet was transmitted at time d_1 , then the next packet will be sent at time $d_1 + p$. In the simulations,

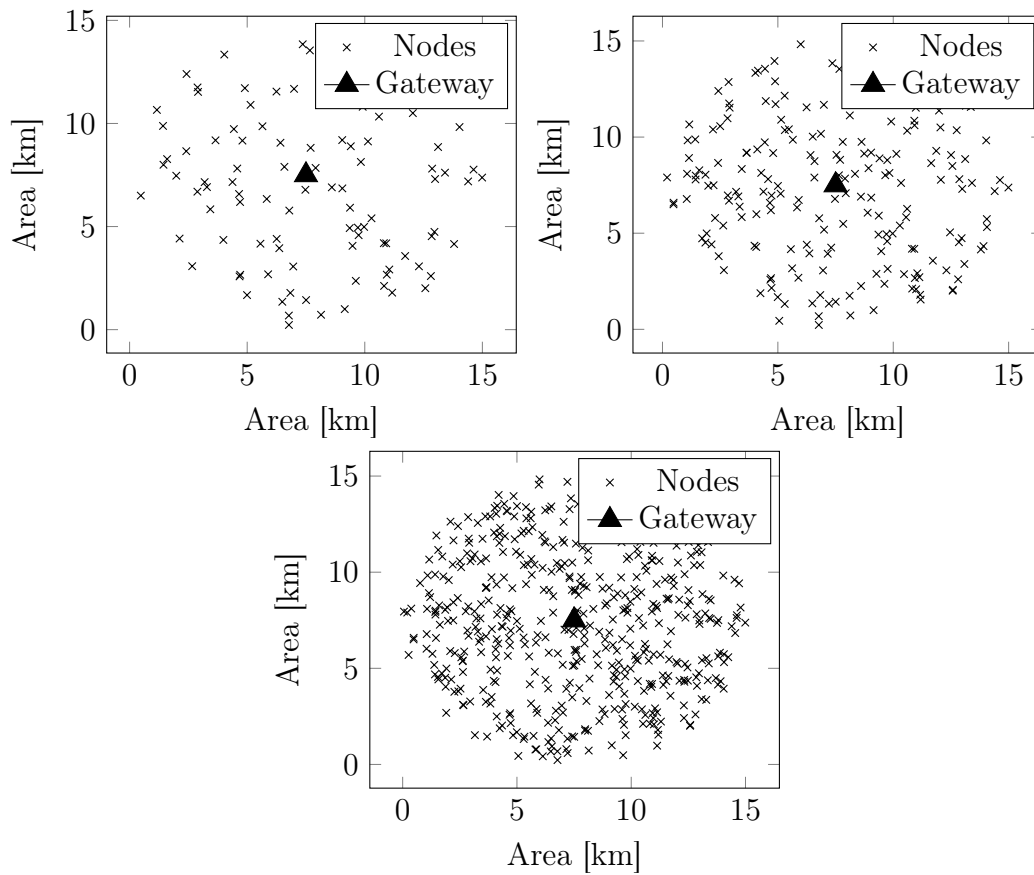


Figure 4-1: Nodes Distribution (a. 100 nodes, b. 200 nodes, c. 500 nodes)

the transmission delay is defined randomly in range $[0, p)$ for each individual node.

4.1.2 The SARSA-based RL algorithms

The main approach will be based on the State-Action-Reward-State-Action (SARSA) algorithm, which is a RL algorithm. The algorithm adjusts d to select the transmission time of the next packet. The idea is that at the beginning of each period, the node will select the transmission delay based on the status of the previous transmissions. SARSA is a Q-learning algorithm, however, while the Q-learning algorithm's policy is based on the maximum reward, the SARSA is an on-policy algorithm that may define its own policy. Nevertheless, both use a so-called Q-table or a state-action table based on which the next action is selected. The reward formula is given as follows:

$$Q(s_i, a_i) = Q(s_i, a_i) + \alpha(r_i + \gamma Q(s_{i+1}, a_{i+1}) - Q(s_i, a_i))$$

Here, s_i and a_i are the previous state and the action taken on that state, respectively; r_i is the reward received after taking the action a_i ; the s_{i+1} is the state, which an agent (or a node) enters after selecting the action a_i ; and the a_{i+1} is the next action that the agent takes on the state s_{i+1} . The s_{i+1} and the a_{i+1} values will be used in the next iteration of the algorithm as the s_i and the a_i . The α and γ parameters are fixed throughout simulation and are responsible for the algorithm's learning rate and the discount factor, respectively. The learning rate defines the importance of the most recent rewards compared to the old information. The discount factor defines the significance of the future rewards.

To properly set up the SARSA algorithm, it is needed to define the state space (S), the action space (A) and the reward (r) in the current environment. Two versions of the algorithm were developed, which differ in the number of actions available. As we mentioned, the reward r is based on the status of the previous transmission. Since, the nodes transmit confirmed messages, each node awaits for an ACK packet from the gateway. If a node successfully receives an ACK, r is set to 1, otherwise is set to 0.

Both versions use the same state space S , which divides the available time range (p) into slots of 10 seconds each. Also, unlike the typical LoRaWAN periodic application, the SARSA algorithm may select any time to transmit the packet. Since the next selected transmission may happen while the previous packet is still on air, some restrictions have been applied. Due to duty cycle restrictions and the reward-based nature of the algorithm, it is not allowed to transmit several packets within a very short amount of time. For example, 20 seconds will be deducted from a total period of 600 seconds ($p = 600s$) before and after the transmission, which will leave 560 second window to transmit a packet. This window is further split into 56 slots, which constitutes the state space S . So, the state space S can be defined as:

$$S \in [0, \frac{p - 2safeTime}{10}).$$

As it was mentioned before, two versions of the algorithm were developed. The two versions differ in the definition of the action space. The first algorithm, or SARSA-1, has the following action space:

$$A \in \{\text{don't change delay} = 0, \text{increase delay} = 1, \text{decrease delay} = 2\}$$

This definition of the action space is inspired by other applications of Q-learning algorithms in gaming environments that uses maps to define the state space and the {left, up, right, down} movements to define the action space. In our environment, S is represented as a single-dimensional map and the actions are to move to the previous slot, to the next slot or not to move at all.

The second version of the algorithm, SARSA-2, makes use of the whole state space S and instead of moving across the map slot-by-slot, as in the SARSA-1, it selects the action from the state space. Thus, the action space is identical to the state space:

$$A = S = [0, \frac{p - 2safeTime}{10})$$

The pseudo-code of the algorithms is presented below. They contain several high-

level descriptions of other methods used by the algorithm. However, their implementation is not relevant to the study. Generate-Packet() method creates a random-size packet to send to the gateway and Send-Packet() is responsible for transferring the packet through physical layer.

Algorithm 1: SARSA-1

```

Input:  $Q$ , packet  $p_0$ ,  $a_0$ ,  $s_0$ ,  $\alpha$ ,  $\gamma$ ,  $\epsilon$ 
 $p_1 = \text{Generate-Packet}()$ ;
if  $p_0.success$  then
  | reward = 1
else
  | reward = 0
switch  $a_0$  do
  | case 0 do
  |   |  $s_1 = s_0$ 
  | case 1 do
  |   | if  $s_0 < 55$  then
  |   |   |  $s_1 = s_0 + 1$ 
  |   |   else
  |   |     |  $s_1 = s_0$ 
  | case 2 do
  |   | if  $s_0 > 0$  then
  |   |   |  $s_1 = s_0 - 1$ 
  |   |   else
  |   |     |  $s_1 = s_0$ 
 $d = \text{random number in range } [10s_1; 10s_1 + 10]$ ;
Send-Packet( $d$ );
 $a_1 = \text{Select-Action}(Q, s_1, \epsilon)$ ;
 $Q(s_0, a_0) = Q(s_0, a_0) + \alpha(\text{reward} + \gamma Q(s_1, a_1) - Q(s_0, a_0))$ ;
 $s_0 = s_1$ ;
 $a_0 = a_1$ ;
 $p_0 = p_1$ ;

```

Finally, as the SARSA algorithm does not use a greedy policy as in the Q-learning method, it is important to explain the policy used by the implementation of the algorithm in this study. A so-called ϵ -greedy policy was used, which is defined below in Select-Action() algorithm.

Here, the ϵ is constant and represents how much the algorithm will rely on the previous rewards. If $\epsilon = 1$, then each next action will be defined randomly, whereas if $\epsilon = 0$, the actions will always be defined by the previous rewards.

Algorithm 2: SARSA-2

Input: Q , packet p_0 , a_0 , s_0 , α , γ , ϵ
 $p_1 = \text{Generate-Packet}()$;
if $p_0.success$ **then**
 | reward = 1
else
 | reward = 0
 $s_1 = a_0$ $d = \text{random number in range } [10s_1; 10s_1 + 10)$;
Send-Packet(d);
 $a_1 = \text{Select-Action}(Q, s_1, \epsilon)$;
 $Q(s_0, a_0) = Q(s_0, a_0) + \alpha(\text{reward} + \gamma Q(s_1, a_1) - Q(s_0, a_0))$;
 $s_0 = s_1$;
 $a_0 = a_1$;
 $p_0 = p_1$;

Algorithm 3: Select-Action

Input: Q , s , ϵ
Output: action a
 $p = \text{random variable from } 0 \text{ to } 1$;
 $a = -1$;
if $p < \epsilon$ **then**
 | $a = \text{random action from } Q(s, :)$;
else
 | $a = \max(Q(s, :))$;
return a ;

Chapter 5

Results & Discussion

5.1 Results

In this chapter, the simulation results of the developed RL algorithms are presented. The simulations provide the performance data of a LoRa network for 20 days under various conditions. First, a typical LoRaWAN periodic application was set up on each node and the performance was recorded. These results will serve as the baseline approach and include performance metrics for network simulations with 100, 200, and 500 nodes. The average PDR for a period of every 2 hours was recorded throughout the simulation. The scenario was repeated using the SARSA approaches. Finally, the results of the baseline approach were compared to the developed RL algorithms. Also, due to the learning mechanism, the RL algorithms are expected to improve the performance over some time. Thus, the average PDR for the second half of the simulation was also included.

Tables 5.1, 5.2, and 5.3 present the simulation results for 100, 200, and 500 nodes, respectively. The results reveal that SARSA-1 does not significantly improve the PDR when the number of retransmissions is set to 1, 2, or 4. When this number is equal to 8, the improvement is high for 100 nodes. SARSA-2 performs better when $r = 8$ but struggles to adapt in scenarios with a lower number of retransmissions for all node populations. What we can see from the results is that when the network is heavily saturated ($r = 8$), the search of the entire state space of SARSA-2 performs

better. However, it fails to adapt for simpler traffic scenarios. In those scenarios, SARSA-1 performs better.

	# of retransmissions $[r]$			
	1	2	4	8
Baseline	81.62%	94.65%	94.52%	73.17%
SARSA-1	83.53%	94.45%	95.43%	94.50%
SARSA-2	81.39%	90.30%	94.77%	93.06%

Table 5.1: Average PDR (100 nodes)

	# of retransmissions $[r]$			
	1	2	4	8
Baseline	73.31%	90.53%	89.05%	64.49%
SARSA-1	77.65%	90.54%	91.27%	58.95%
SARSA-2	69.98%	83.55%	88.79%	77.96%

Table 5.2: Average PDR (200 nodes)

	# of retransmissions $[r]$			
	1	2	4	8
Baseline	72.51%	83.64%	70.38%	64.27%
SARSA-1	68.15%	80.57%	78.62%	64.21%
SARSA-2	50.88%	70.18%	74.99%	71.74%

Table 5.3: Average PDR (500 nodes)

A more detailed view is given by Figure 5-1, where the PDR trend over time is displayed for a scenario with 100 nodes and 8 retransmissions. It is clear that both SARSA approaches outperform the baseline in this case. Two other cases are also presented. In Figure 5-2, it can be observed that the baseline results gradually fall over time, whereas the RL algorithms maintain the performance during the simulation time. Similarly, in Figure 5-3, it can be seen that the performance decreases over time for all the applications. Nonetheless, the SARSA-2 experiences the least amount of loss in PDR, whereas the SARSA-1 and the baseline perform similarly.

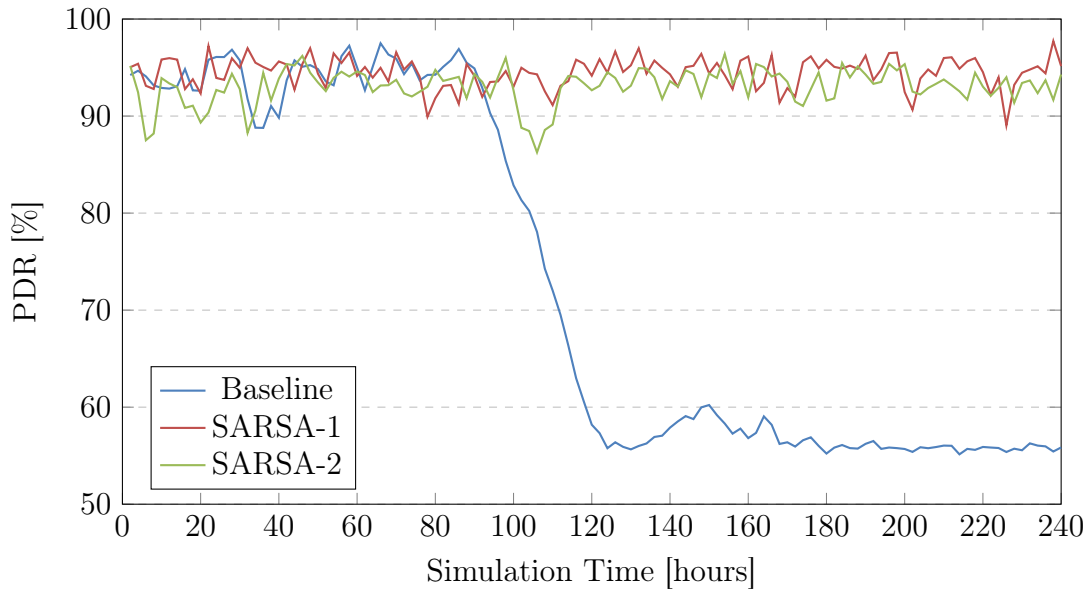


Figure 5-1: PDR throughout the simulation (100 nodes, $r = 8$)

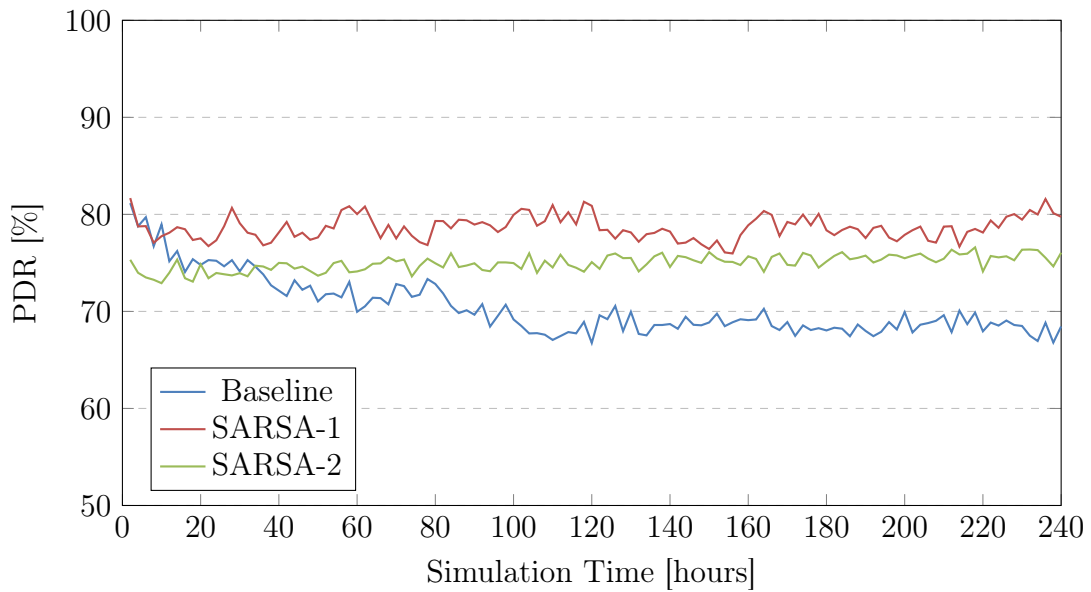


Figure 5-2: PDR throughout the simulation (500 nodes, $r = 4$)

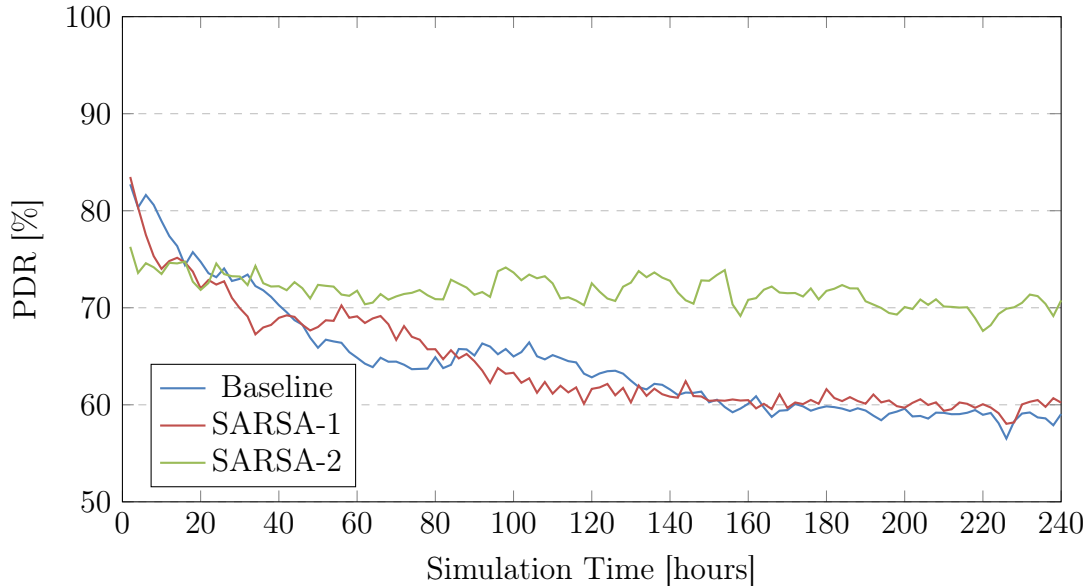


Figure 5-3: PDR throughout the simulation (500 nodes, $r = 8$)

5.2 Discussion

The results presented in the previous chapter suggest that LoRa networks could benefit from the RL driven applications installed on the nodes, however, the performance varies in different conditions.

Based on the results, a clear decrease in performance in regular periodic applications in highly saturated networks can be observed. The SARSA algorithms also experience similar losses, however it is notably less significant. It is suggested that the SARSA algorithms can greatly raise the performance in networks with high number of end devices. A significant improvement at high r values can be noticed, however, the performance at lower r values is either similar or sometimes worse than the baseline. Therefore, the typical LoRaWAN periodic transmission can still be used when the network is not too congested. Nevertheless, the performance in networks with more load can be improved considerably.

A more extensive simulation with more end-devices and different settings might provide a better understanding of the algorithms' behavior in congested networks. Also, the influence of α , γ and ϵ parameters is still to be determined. These two observations will be assessed in the near future as an extension of this research.

Chapter 6

Conclusion

This research aimed to demonstrate that a saturated network performance can be improved by modifying application layer parameters on individual nodes using reinforcement learning. While analyzing the literature, it was determined that the influence of application layer parameters in EDs is not thoroughly investigated. Therefore, this dissertation was aimed to address this issue. Based on the evaluation of the simulation results, it has been shown that the proposed SARSA-1 and SARSA-2 algorithms improve the performance of the network when compared to the typical LoRaWAN periodic application. In several cases, the RL algorithms were able to almost double the performance of the baseline algorithm.

Nevertheless, there is still room for improvement as it might be needed to optimize constant parameters for certain conditions. The results have shown that certain selection of the parameters might significantly affect the performance. In this study, static nodes that transmit only confirmed messages were investigated. In future works, the influence of mobile devices and unconfirmed traffic will also be investigated.

Bibliography

- [1] What is lorawan® specification, Mar 2021.
- [2] A. Aneiba, B. Nangle, J. Hayes, and M. Albaarini. Real-time iot urban road traffic data monitoring using lorawan. In *Proceedings of the 9th International Conference on the Internet of Things*, IoT 2019, New York, NY, USA, 2019. Association for Computing Machinery.
- [3] D. Bankov, E. Khorov, and A. Lyakhov. Mathematical model of lorawan channel access. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, 2017.
- [4] S. Battle and B. Gaster. Lorawan bristol. In *Proceedings of the 21st International Database Engineering & Applications Symposium*, IDEAS 2017, page 287–290, New York, NY, USA, 2017. Association for Computing Machinery.
- [5] L. Beltramelli, A. Mahmood, P. Osterberg, M. Gidlund, P. Ferrari, and E. Sisinni. Energy efficiency of slotted lorawan communication with out-of-band synchronization. *IEEE Transactions on Instrumentation and Measurement*, page 1–1, 2021.
- [6] M. Capuzzo, C. Delgado, J. Famaey, and A. Zanella. An ns-3 implementation of a battery-less node for energy-harvesting internet of things, 2021.
- [7] M. Capuzzo, D. Magrin, and A. Zanella. Confirmed traffic in lorawan: Pitfalls and countermeasures. In *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–7, 2018.
- [8] F. Cuomo, J. C. C. Gámez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani. Towards traffic-oriented spreading factor allocations in lorawan systems. In *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–8, 2018.
- [9] V. Delafontaine, F. Schiano, G. Cocco, A. Rusu, and D. Floreano. Drone-aided localization in lora iot networks, 2020.
- [10] J. Downes. Lorawan temperature sensors for local government asset management, 2020.

- [11] I. Ilahi, M. Usama, M. O. Farooq, M. U. Janjua, and J. Qadir. Intelligent resource allocation in dense lora networks using deep reinforcement learning, 2020.
- [12] F. H. Khan, R. Jurdak, and M. Portmann. A model for reliable uplink transmissions in lorawan. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 147–156, 2019.
- [13] J. Kim and J. Song. A simple and efficient replay attack prevention scheme for lorawan. In *Proceedings of the 2017 the 7th International Conference on Communication and Network Security, ICCNS 2017*, page 32–36, New York, NY, USA, 2017. Association for Computing Machinery.
- [14] L. S. Laufenberg. Impersonating lorawan gateways using semtech packet forwarder, 2019.
- [15] H. Li, K. Ota, and M. Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE Network*, 32(1):96–101, 2018.
- [16] A. Loubany, S. Lahoud, and R. El Chall. Adaptive algorithm for spreading factor selection in lorawan networks with multiple gateways. *Computer Networks*, 182:107491, 2020.
- [17] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke. A survey on the viability of confirmed traffic in a lorawan. *IEEE Access*, 8:9296–9311, 2020.
- [18] S. Meiling and T. C. Schmidt. Lora in the field: Insights from networking the smart city hamburg with riot, 2020.
- [19] M. A. Ouameur, M. Caza-Szoka, and D. Massicotte. Machine learning enabled tools and methods for indoor localization using low power wireless network. *Internet of Things*, 12:100300, 2020.
- [20] E. Pietrosevoli, M. Rainone, and M. Zennaro. On extending the wireless communications range of weather stations using lorawan, 2019.
- [21] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini. Slotted aloha on lorawan-design, analysis, and deployment. *Sensors*, 19(4):838, Feb 2019.
- [22] B. Reynders, Q. Wang, and S. Pollin. A lorawan module for ns-3: Implementation and evaluation. WNS3 '18, page 61–68, New York, NY, USA, 2018. Association for Computing Machinery.
- [23] R. M. Sandoval, A. J. Garcia-Sanchez, and J. Garcia-Haro. Optimizing and updating lora communication parameters: A machine learning approach. *IEEE Transactions on Network and Service Management*, 16(3):884–895, 2019.
- [24] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao. Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys Tutorials*, 21(4):3072–3108, 2019.

- [25] E. van Es, H. Vranken, and A. Hommersom. Denial-of-service attacks on lorawan. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, New York, NY, USA, 2018. Association for Computing Machinery.
- [26] S. Wang and T. Chen. Increasing lorawan application-layer message delivery success rates. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00148–00153, 2018.
- [27] A. Weinand, A. G. de la Fuente, C. Lipps, and M. Karrenbauer. Physical layer security based key management for lorawan, 2021.
- [28] A. Xanthopoulos, A. Valkanis, G. Beletsioti, G. I. Papadimitriou, and P. Nicopolitidis. On the use of backoff algorithms in slotted aloha lorawan networks. In *2020 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–4, 2020.
- [29] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch. Ts-lora: Time-slotted lorawan for the industrial internet of things. *Computer Communications*, 153:1–10, 2020.