Experimental Analysis of the Over-The-Air Activation procedure in LoRaWAN

Chékra El Fehri*, Nouha Baccour†, Pascal Berthou‡ and Inès Kammoun*
*LETI Laboratory, National School of Engineers of Sfax (ENIS), Sfax, Tunisia.
†REDCAD Laboratory, National School of Engineers of Sfax (ENIS), Sfax, Tunisia
‡LAAS-CNRS, University of Toulouse, Toulouse, France
Email: chekra.elfehri@stud.enis.tn, nouha.baccour@enis.tn, berthou@laas.fr, ines.kammoun@ieee.org

***Abstract*—Long Range Wide Area Network (LoRaWAN) technology is knowing an impressive growth. It is considered by many researchers as the new era of Internet of Things (IoT) communication. Several studies focus on evaluating the LoRaWAN performance according to many features such as coverage, scalability, and communication reliability. However, these studies assume that LoRaWAN end-devices are already activated by the network server. Thus, the performance of LoRaWAN activation procedure, referred as Over-The-Air Activation (OTAA), is not widely treated.**

In this paper, we elaborate an experimental analysis of the OTAA procedure performance using a real field LoRaWAN deployment. Our objective is to analyse the end-device activation delay and power consumption at large scale LoRaWAN. To achieve this goal, we design an experimental scenario of 30 end-devices competing for being activated by sending network join-requests. Upon its activation, each end-device transmits unconfirmed data at high rate, which simulates a large-scale LoRaWAN where hundreds of end-devices send their join-requests concurrently. Results show that OTAA procedure incurs high activation delays and power consumption, especially in large scale where the network traffic is high. This is due to three main factors: collisions, the back-off retransmission mechanism and join-request duty-cycle.

***Index Terms*—LoRaWAN, Over-The-Air Activation, delay, power-consumption**

I. Introduction

Low power wide area networks (LPWANs) are invading the Internet of Things market as they provide long-range, low-cost and low-power connectivity. LPWAN communication technologies are competing for providing large-scale connectivity. The Long Range Wide Area Network (LoRaWAN) is one of the leading technologies. Since its appearance, researchers and experts didn't stop exploring the LoRaWAN features promoted by its specification. In brief and as described in [1], LoRaWAN consists of a set of LoRa end-devices connected to a LoRa gateway via a star-of-star topology. The LoRa gateway acts as a bridge to relay bi-directional communication between end-devices and the network server which represents the intelligent entity of the network. The end-devices operate over the ISM bands and implement three mandatory channels specified in the LoRaWAN specification [2].

Based on the application requirements in terms of downlink latency and power consumption, a LoRaWAN end-device can operate according to one of three classes A, B or C, where A is the default class. Regardless of the end-device class, the uplink communication is based on ALOHA-like channel access technique. However, the downlink communication differs from one class to another. In class A, after an uplink packet transmission, the end-device opens two receive windows to receive downlink traffic. In class B, end-devices are synchronized via a Beacon message broadcasted periodically by the gateway and downlink packets are received by the end-devices during defined time slots. As for class C, end-devices are in a continuous listening for downlink traffic if they are not sending. Before being able to operate over the network and exchange any data, the end-device in LoRaWAN should first join the network as a class A end-device using one of the two activation methods. The first is Activation By Personalization (ABP) which consists in embedding all needed parameters directly in the end-device, which allows it to join the network without the need to request joining from the network server. The second is the Over-The-Air Activation procedure (OTAA) where the needed parameters are provided to the end-device by exchanging join-request and join-accept packets with the network server. For instance, several session keys for secure data exchange are generated during the OTAA procedure.

Many research works were devoted to the performance evaluation of LoRaWAN such as analyzing the LoRaWAN capacity using either methematical models [3], simulations [4] or real world deployments [5]. The interest to LoRaWAN was not limited to the network capacity evaluation, several interesting contributions were introduced to improve the LoRaWAN network performance [6], scalability and reliability [7], and energy efficiency [8]. These research works assume that LoRaWAN end-devices are already activated, i.e., connected to the network server.

The performance of LoRaWAN activation procedure, the so called OTAA procedure, was not broadly handled, except in [9]. In this study, the authors evaluate the performance of OTAA in terms of activation delay and energy consumption using a mathematical model. However, the study involves a set of unrealistic assumptions such as ignoring the capture effect despite the fact that LoRa modulation scheme exhibits the capture effect as an important metric to conquer collisions. Further, most related literature on LoRaWAN OTAA is about security aspects [10] [11] [12] [13] and ignore the network performance in terms of latency and energy efficiency. In

Fig. 1. The join-procedure timing.

LoRaWAN, low powered end-devices use an ALOHA-like channel access technique when sending join-request packets. Therefore, the performance of OTAA procedure may be affected, especially at large scale, when thousands of nodes compete for sending their join-request packets to the network server via the gateway leading to collisions and join-request retransmissions. As a result, the end-device activation would be delayed in addition to the increase of power-consumption.

In this paper, we elaborate the experimental evaluation of LoRaWAN OTAA procedure, in terms of activation delay and power consumption in large-scale scenario. LoRaWAN provides long range communication and thus thousands of devices can reach the gateway. The ability of LoRaWAN to manage a high number of end-devices, the so called scalability, is of paramount importance. This what explain the several studies on LoRaWAN scalability, but this topic remains not well explored as most studies are based on simulations or mathematical models. To analyze the real-life performance of OTAA in large-scale, we designed an experimental scenario intended to explore the performance of OTAA under high network traffic and a small number of devices (30), which approaches to a large-scale scenario having thousands of devices transmitting with low data rates, typically one packet each 30 minutes to 24 hours [14].

The remainder of the paper is as follow: First, we give an overview of LoRaWAN OTAA procedure and present the handled problem in section II. Then, we give the hardware architecture and experimental setup in section III. We present the experimental results and discuss them in section IV. Finally we conclude the paper in section V.

II. The LoRaWAN OTAA: Overview and Problem Statement

A. OTAA- Overview

In LoRaWAN, to be able to start operation and initiate packets exchange, end-devices should be activated to join the network. The outcome of this activation is the setting of end-device parameters (e.g., the device address and session keys). Activation can be established by two methods: ABP method which is convenient when we have to deploy few number of end-devices, as activation parameters are defined manually and stored into the end-device without any negotiation with the network server. On the other hand, OTAA activation procedure is recommended at large scale network deployment as activation parameters are set automatically, following an exchange between the end-device and the network server (join-request/join-accept). As shown in Fig. 1, after a JOIN_ACCEPT_DELAY1 from sending the join-request packet, the end-device opens a first receive window RX1 during which it waits for a join-accept packet from the network server. If no join-accept is received by the end-device, it will open a second receive window RX2 after a JOIN_ACCEPT_DELAY2 from sending the join-request packet. The network server in turn generates the needed parameters and sends them to the end-device in the join-accept packet either in RX1 or RX2. The JOIN_ACCEPT_DELAY1 and JOIN_ACCEPT_DELAY2 are set to $5s$ and $6s$ respectively. If no join-accept packet is received in RX2, the end-device should wait a certain back-off time before the retransmission of join-request packet. In addition to ISM bands duty cycle restrictions imposed by ETSI regulations [15], the LoRaWAN specification introduces more constraining restrictions in join-request retransmissions [2]: During the first hour from its powering-up, the end-device should not exceed $36s$ as a total time-on-air, i.e., for all transmitted join-request packets, which corresponds to 1% of duty cycle. During the next $10hours$ following the first hour, the end-device is not allowed to transmit join-request packets more than $36s$ as a total time-on-air. After having spent $11hours$ trying to join the network without any reply from the network server, only $8.7s$ of total time-on-air is allowed for join-request transmissions every $24hours$. There is a set of parameters that should be stored into the end-devices as a prerequisite to initiate OTAA activation procedure. As illustrated in Fig. 2, (1) these parameters which includes the network and the device identifiers, as well as the device root keys will be encapsulated in a join-request packet and sent to the join-server . Once received, (2) the join-server will reply with a join-accept packet containing the end-device and network information needed by the end-device to operate over the network. (3) Then, both end-device and join-server will generate specific session keys. Finally (4), specific generated keys will be sent by the join-serevr to the application server and the network server for the encryption/decryption use of specific data.

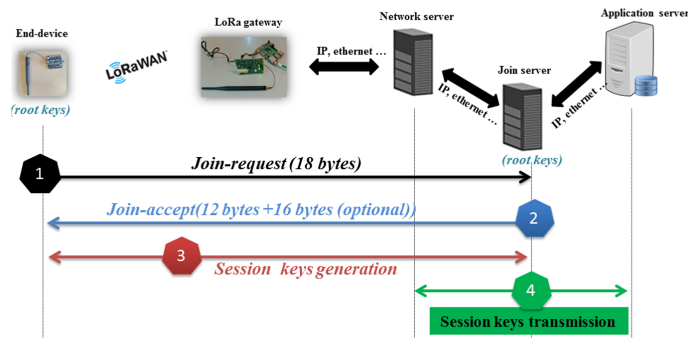

Fig. 2. The OTAA procedure

B. Problem statement

As shown in the above description of OTAA procedure, several important parameters such as session keys are exchanged between the end-device and the join-server during the activation. These parameters should be confidential for secure

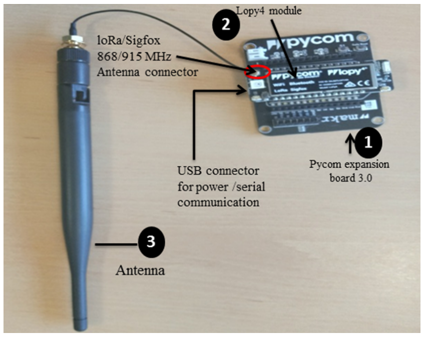

Fig. 3. The LoRa pycom/lopy4 end-device

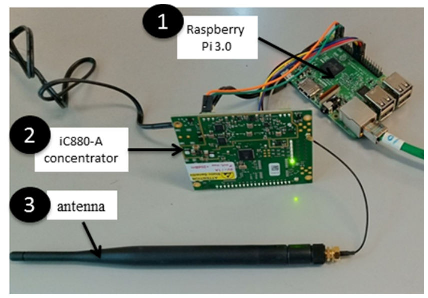

Fig. 4. The LoRa gateway hardware

TABLE I
THE EXPERIMENTAL SETTINGS

Channels Frequencies	Spreading factor (SF)	Coding Rate (CR)	Band width (BW)	TX_Power
868.1 MHz	7	4/5	125 kHz	14 dBm

data exchange between network entities. In fact, the security of OTAA procedure was widely analyzed in the literature but its network performance in terms of activation delay or power consumption is not explored, except in [9]. In this work, the authors derive the end-device activation delay and energy consumption using a markov chain model. However, they consider a network of only 30 end-devices and only joining traffic is generated over the network. Further, a set of unrealistic assumptions are considered such as ignoring the capture effect and assuming that the gateway is not in a transmission state when receiving a join-request. Moreover, the study is based on an earlier LoRaWAN spcification [16] that does not include retransmission time back-off mechanism and the duty-cycle restrictions in join-request transmissions.

III. HARDWARE ARCHITECTURE AND EXPERIMENTAL SETUP

In large-scale networks or in applications generating high data traffic together with OTAA joining traffic, collisions and join-request retransmissions may happen frequently due to the ALOHA based channel access technique used in LoRaWAN. As a result, the end-device activation would be delayed in addition to the increase in power-consumption resulting from excessive retransmissions. Thus, many questions have to be answered; (1) In large-scale LoRaWAN, what is the activation delay for an end-device ? (2) If the join-request is not acknowledged by a join-accept, how many join attempts (join-request retransmissions) are required so the end-device can successfully join the network? (3) To what exent join-request retransmissions can impact the power consumption of the end-device during the OTAA procedure? (4) What are the main factors that impact the performance of OTAA? To answer these questions, we elaborate an experimental evaluation of OTAA procedure using a real LoRaWAN enviroment.

The LoRaWAN experiments were conducted using three kind of hardware: 30 end-devices, one gateway and one network server (netServer). End-devices are built by putting together: (1) An expansion board 3.0 from pycom [17], a kind of printed circuit offering useful features such as a micro USB connector allowing serial communication with our laptop and embedded GPIO connectors; (2) Lopy4 from pycom, a Micropython-programmable micro-controller integrated in the expansion board via GPIO connector allowing four different network connections: WiFi, BLE, Sigfox, and LoRa/LoRaWaN; and (3) An external antenna plugged into the Lopy4 Sigfox/LoRa antenna connector that operates in EU868-870 MHz band (Fig. 3). LoRaWAN firmware is thus integrated in the Lopy4 and can be updated via the expansion board using our laptop and serial communication. Firmware update includes the set up of the regional sub-band, the bandwidth, the coding rate, etc. Note that Lopy4 supports class A and class C and implements the LoRaWAN specification v1.0.2 [2]. The gateway [18] includes (1) a raspberry Pi 3.0 which is a fully featured micro-computer running the gateway software, (2) a LoRaWAN concentrator iC880-A which is a transmitter/receiver module allowing to receive up to 8 LoRaWAN uplink packets simultaneously using different spreading factors, and (3) an external LoRa antenna (Fig. 4). The netServer is a computer machine that runs ChirpStack software [19]. It is an open-source LoRaWAN network server allowing to orchestrate all the network exchanges.

As depicted in Fig. 5, our experiments are held in the LAAS-CNRS building in Toulouse (France), where 30 end-devices were deployed together with a LoRa gateway located about 45 meters far from the end-devices and a ChirpStack netServer. Table I summarizes the main settings of our experimental scenario. It is important to note that the Lopy4 LoRaWAN firmware is not open source in the hardware. However, it is possible to communicate with the firmware using a rich set of predefined methods such as join() to launch OTAA procedure, has_joined() to check if the end-device has joined the network, add_channel() etc. Hence, we developed a pymakr/Atom [20] project that we load into the end-device Lopy4 to perform the following tasks: (1) configure LoRaWAN firmware with a set of network settings (Table I), (2) define the experimental scenario, and (3) report performance metrics: generated traffic, the activation delay and number of attempts during OTAA. The experimental scenario consists first in executing OTAA procedure. Each end-device sends a join-request packet every 15s until the reception of

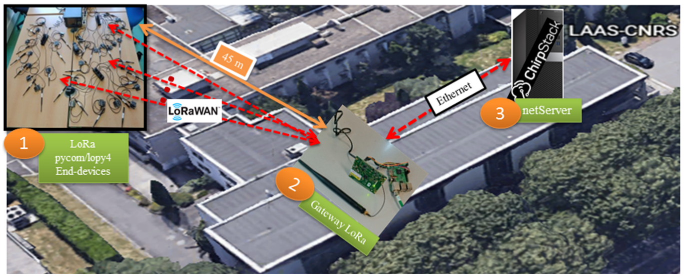

Fig. 5. The experimental setup.

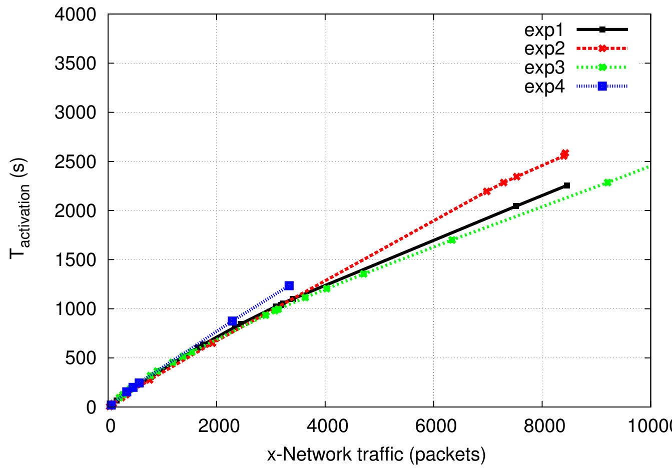

Fig. 6. The Eds activation delay.

a join-accept packet, i.e., its activation or the end of the experiment. Once activated, the end-device sends a 3-byte unconfirmed packet every 3s until the end of the experiment fixed to 2 hours. The 3s is the time-off duration during which the sub-band is not available for the end-device to send an unconfirmed packet according to the duty-cycle restrictions imposed by ETSI. Hence, end-devices generate unconfirmed data with maximum allowed rate (0.333 pkt/s). The experiment is repeated four times. At the end of a given experiment, only a subset of end-devices (e.g., 15 from 30) succeed to join the network before the end of the experiment. For all activated end-devices, we measure the activation delay, the number of join-request retransmissions (attempts), the total number of packets in the network (at the activation time), and the power consumption. It is important to note that the activation delay is the time duration between the transmission of the first join-request packet, which corresponds to the experiment starting time, and the time of join-accept packet reception.

To estimate the power-consumption of an end-device during its activation process, we use the Fluke norma 4000 power analyzer to measure the power $P1$ consumed by an end-device successfully activated after only one join-request attempt. Then, according to the number of attempts $nb\_attempts$ needed by each end-device to join the network, the end-device power consumption during the OTAA process is estimated according the formula: $Power\_consumption = P1 \times nb\_attempts$. We note that there is a slight difference between the power consumption when receiving a join-accept packet in RX1 and RX2.

IV. Results and Discussion

Fig. 6 shows the activation delay as a function of the total number of packets in the network. A point in the curve represents one of the activated end-devices in the given experiment (exp1, exp2, exp3 or exp4). From this curve, two main observations can be drawn:

Observation 1: The curves with respect to each experiment are almost overlapping during the first 20 minutes (1200 s), which corresponds to a total network traffic of 3500 packets. After 20 minutes, the curves are divergent as the network is more and more congested due to data traffic from activated end-devices and resistance to collisions differs from one experiment to another. If we take the example of one end-device ($node15$), the activation delay is about 37.5 min (2255s) in exp1, and 22.5 min (1355s) in exp3; however the node was unable to be activated in exp2 and exp4. Furthermore, $node18$ was activated after 42.5 min in exp3 and was unable to be activated for the three other experiences till the end of the experiment.

Observation 2: Only a set of end-devices from a total 30-nodes were able to join the network before the end of the experiment set to 2 hours: 15 , 12, 18 and 6 end-devices, with respect to exp1, exp2, exp3 and exp4. Moreover, all these end-devices were activated during the first hour of the experiment. More precisely, no end-device was able to be activated after 43 min, till the end of the experiment.

The analysis of these observations leads to define three main factors that highly impact the performance of the activation process in LoRaWAN: (1) Collisions, (2) retransmission back-off mechanism, and (3) join-request duty cycle.

1) **Collisions**
Recall that OTAA procedure uses ALOHA based channel access technique and it is well known that ALOHA leads to dramatic performance when the network is congested. However, during the first 20 minutes of the experiences duration, some collided join-request packets were correctly received by the gateway thanks to the channel capture effect. That means that some nodes were activated in the same time. This can be noticed in Fig. 6, exp3 (point 3 and 4 are overlapped). The more the number of activated end-devices increases, the more the unconfirmed traffic generated by activated end-device increases. The unconfirmed traffic (0.333 pkt/s) together with join-request trasmissions sent by the remaining non-activated nodes (0.06 pkt/s) lead to increasing network congestion (simulating large scale LoRAWAN scenario) and thus decreasing resistance to collisions and increasing activation delays. With different data traffic types competing to access the LoRa channels, the channel capturing effect is no more able to alleviate the collisions problem for join-request transmissions.

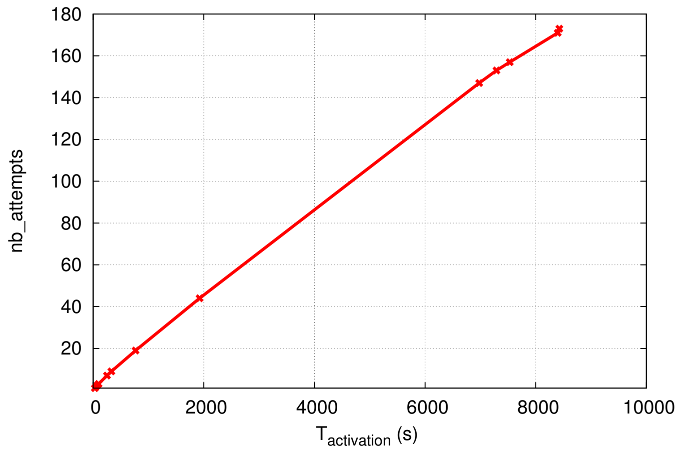

Fig. 7. The number of attempts (exp 2).

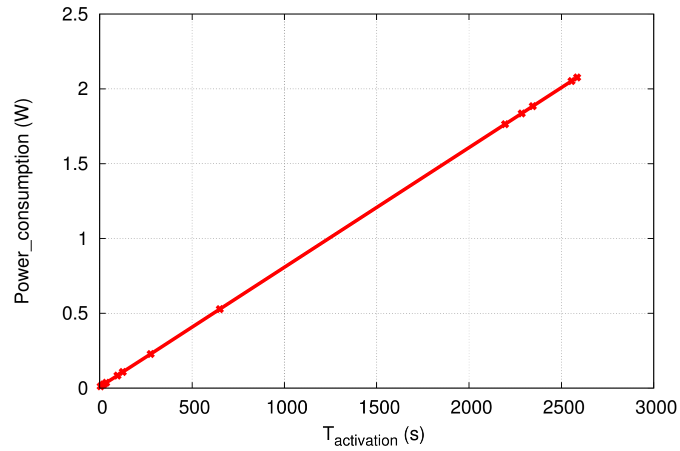

Fig. 8. The Eds power consumption during OTAA (exp 2).

2) **The retransmission back-off mechanism:**
Obviously, collisions lead to join-request packet retransmissions. LoRaWAN specification [2] defines a retransmission mechanism that aims to reduce the probability of simultaneous retransmissions (collision) and minimize the number of attempts that may be proceeded by low-powered end-devices to be successfully activated. It consists in defining the back-off delay, which corresponds to the time duration between the end of RX2 and the next join-request retransmission, as a random value selected in a given interval. The bounds of this interval are defined according to a sequence of time back-off and varies from one retransmission attempt to another. However, the specification does not give the values for this back-off sequence and keep it at the will of the LoRaWAN implementation. Indeed, in the pycom/Lopy4 hardware used in our experimentation, where LoRaWAN is implemented, the backoff delay is fixed to $15s$ for all retransmission attempts. This design choice contributes to excessive collisions in the ALOHA-based network. For illustrative purpose, we present in Fig. 7, the number of join attempts for each activated device in exp2. As it can be depicted from the figure, the maximum activation delay equal to 43 min, corresponds to a total of 173 attempts . This high number of join-request retransmissions leads to a power consumption of $2.076W$. Fig. 8 shows the power consumption with respect to each activated end-device in exp2. As it can be clearly noticed, the activation delay is proportional to the number of attempts and thus the power consumption. As a result, the retransmission back-off mechanism has a considerable impact on the performance of LoRaWAN activation procedure. It is worthy to explore new methods to define a time retransmission back-off sequence that can provide better LoRaWAN performance during OTAA.
3) **The join-request duty cycle:**
As stated above in observation **2)**, in our experiments only a set of end-devices were activated during the first hour of the experiment. The last end-device was activated after 43 minutes which correspond to a network traffic of 8427 packets. The remaining end-devices were unable to be activated till the end of the experiment, fixed to 2 hours. In other words, their activation delays would be greater than 2 hours. This observation is partially due to the join-request duty cycle that imposes a certain amount of total time-on-air for all transmitted join-request packets (refer to section II-A). Indeed, during the first hour, the join-request duty cycle (1%) does not constraint the retransmission of join-request packets since the back-off delay ($15s$) is greater than the wating time (time-off) imposed by the join-request duty-cycle ($5.049s$ given the time-on-air of a join-request packet is $0.051s$ and the total allowed time-on-air in one hour is 36s [21]). However, during the second hour till the 10th hour, the join-request duty cycle is dived by 10, i.e., 0.1%, which corresponds to $50.949s$ of time-off over the sub-band after each join-request packet transmission. Hence, the end-device can no longer use the retransmission back-off delay fixed to $15s$ (as in the first hour), which contributes to delayed activation.

To better understand the impact of the above factors in the OTAA performance, we analysed the distribution of activation delay between end-devices, as illustrated in Fig. 9. The figure is gathered based on exp2 but the same behaviour has been observed in the other experiments. Recall that in exp2, 12 out of 30 end-device were able to join the network during 2 hours. Fig. 9 clearly depicts three phases in the 2-hours experiment, which are analyzed next:
The first phase (1), involves 60% of activated end-devices and activation is performed relatively at short delays. In this phase, the network traffic is low and the channel capturing is in favor of join-request transmissions. However, the retransmission back-off mechanism is the main factor affecting OTAA performance, especially in terms of power consumption. For

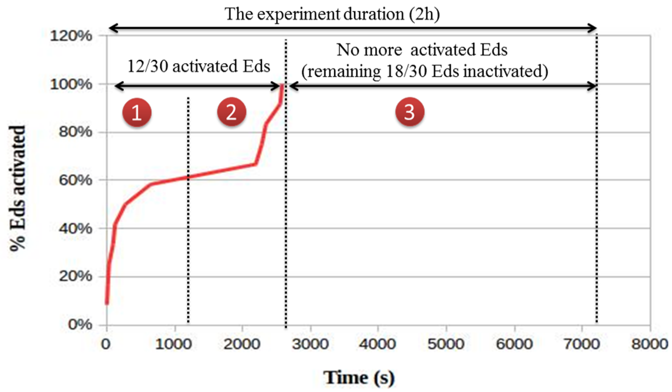

Fig. 9. The distribution of activation delay between end-devices (Eds)(exp 2).

instance, the last activated end-device in this phase makes 19 attempts to be activated, which corresponds to $0.228W$ of power consumption, Fig. 8.

In the second phase (2), end-devices join the network more slowly. The unconfirmed network traffic increases leading to increasing collisions. The channel capturing is no more able to resolve collisions. Moreover, the retransmission back-off mechanism worsen the situation. The second phase ends at $2585s$ with only 12 activated end-devices.

The third phase (3) consists in the remaining time of the experiment, where no end-device was able to join the network. It includes the second hour of the experiment duration where the join-request duty cycle is considered as a major factor of delaying the activation in addition to collisions.

V. Conclusion

To conclude, in this paper we evaluated the Over-The-Air Activation (OTAA) procedure in a real LoRaWAN indoor deployment. We deployed a LoRaWAN cell composed by 30 pycom/Lopy4 LoRa end-devices, one gateway and a Chirp-Stack netServer. Results show that at high network traffic, which simulates large scale LoRAWAN scenario, the OTAA procedure incurs large activation delays (e.g., 15 out of 30 end-devices require more than 2 hours to be activated), a high number of join-packet retransmission attempts (e.g., 173 attempts for an activation delay of 43 min) and consequently high power consumption (e.g., $2.076W$ for an activation delay of 43 min). Three main factors impact the performance of OTAA procedure in LoRaWAN, namely (i.) collisions and thus join-request retransmission attempts (ii.) the retransmission backoff mechanism which needs to be carefully improved taking into account network parameters, such as the network size and traffic rate in order to find a good balance between backoff-time and number of retransmissions, and (iii.) join-request duty cycle that may constraint join-request retransmissions.

As a perspective, we aim to explore possible solutions to alleviate the collision problem during OTAA. Especially, we need to investigate a new method to define the time back-off sequence for join-request retransmissions.

Acknowledgment

The authors gratefully acknowledge the technical support from Stanislas Pedebearn.

References

[1] YEGIN, Alper, KRAMP, Thorsten, DUFOUR, Pierre, et al. LoRaWAN protocol: specifications, security, and capabilities. In : LPWAN Technologies for IoT and M2M Applications. Academic Press, 2020. p. 37-63.

[2] LoRaWAN Specification v1.0.2, LoRa Alliance

[3] RON, Dara, LEE, Chan-Jae, LEE, Kisong, et al. Performance Analysis and Optimization of Downlink Transmission in LoRaWAN Class B Mode. IEEE Internet of Things Journal, 2020, vol. 7, no 8, p. 7836-7847.

[4] MAGRIN, Davide, CAPUZZO, Martina, et ZANELLA, Andrea. A thorough study of LoRaWAN performance under different parameter settings. IEEE Internet of Things Journal, 2019, vol. 7, no 1, p. 116-127.

[5] INGABIRE, Winfred, LARIJANI, Hadi, et GIBSON, Ryan M. Performance Evaluation of Propagation Models for LoRaWAN in an Urban Environment. In : 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE). IEEE, 2020. p. 1-6.

[6] VANGELISTA, Lorenzo et CATTAPAN, Alessandro. A new lora-compatible modulation improving the lorawan network level performance. In : 2019 IEEE Latin-American Conference on Communications (LATINCOM). IEEE, 2019. p. 1-6.

[7] REYNDERS, Brecht, WANG, Qing, TUSET-PEIRO, Pere, et al. Improving reliability and scalability of lorawans through lightweight scheduling. IEEE Internet of Things Journal, 2018, vol. 5, no 3, p. 1830-1842.

[8] TIURLIKOVA, Aleksandra, STEPANOV, Nikita, et MIKHAYLOV, Konstantin. Improving the Energy Efficiency of a LoRaWAN by a UAV-based Gateway. In : 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2019. p. 1-6.

[9] TOUSSAINT, Joel, EL RACHKIDY, Nancy, et GUITTON, Alexandre. Performance analysis of the on-the-air activation in LoRaWAN. In : 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2016. p. 1-7.

[10] TOMASIN, Stefano, ZULIAN, Simone, et VANGELISTA, Lorenzo. Security analysis of LoRaWAN join procedure for Internet of Things networks. In : 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). IEEE, 2017. p. 1-6.

[11] SANTAMARIA, Michael et MARCHIORI, Alan. Demystifying LoRa WAN Security and Capacity. In : 2019 29th International Telecommunication Networks and Applications Conference (ITNAC). IEEE, 2019. p. 1-7.

[12] DANISH, Syed Muhammad, LESTAS, Marios, ASIF, Waqar, et al. A lightweight blockchain based two factor authentication mechanism for LoRaWAN join procedure. In : 2019 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2019. p. 1-6.

[13] TSAI, Kun-Lin, LEU, Fang-Yie, HUNG, Li-Ling, et al. Secure Session Key Generation Method for LoRaWAN Servers. IEEE Access, 2020, vol. 8, p. 54631-54640.

[14] MAGRIN, Davide, CAPUZZO, Martina, et ZANELLA, Andrea. A thorough study of LoRaWAN performance under different parameter settings. IEEE Internet of Things Journal, 2019, vol. 7, no 1, p. 116-127.

[15] Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment , ETSI, Sept 2017

[16] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, LoRaWAN Specification, LoRa Alliance, Standard V1.0, 2015.

[17] [Online]. Available: https://docs.pycom.io/

[18] [Online]. Available: https://github.com/ttn-zh/ic880a-gateway/wiki.

[19] [Online]. Available: https://www.chirpstack.io/

[20] [Online]. Available: https://atom.io/

[21] Semtech, SX1272/3/6/7/8 LoRa Modem Design Guide, AN1200.13,, July 2013.