A Study of Extensive LoRaWAN Downlink Communication in a Mobility Scenario

Mads Smed Enggaard Thomassen[1], Kasper Stenholt Winkler[1], Davide Magrin[2], and Michele Albano[1(✉)]

[1] Department of Computer Science, Aalborg University, Aalborg, Denmark
{mthoma18,kwinkl18}@student.aau.dk, mialb@cs.aau.dk
[2] Department of Computer Engineering, University of Naples Federico I, Naples, Italy
davide.magrin@unina.it

Abstract. Low Power Wide-Area Networks (LPWANs) are gaining a lot of attention in use cases related to the Internet of Things . Most LPWAN communication technologies are asymmetric and have inherently better uplink communication, thus can implement ubiquitous monitoring and data collection efficiently. On the other hand, there is a growing demand for a sizable downlink channel. This work aims at providing a framework to study the feasibility of extensive downlink communication in LoRaWAN with moving End Devices (EDs). Starting from a real-world implementation of a Firmware Update Over the Air application, we experiment with two simple strategies to reduce packet loss when EDs move from within range gateway to another. Then, we extend the ns-3 LoRaWAN module to scale up the study by means of simulation. Such tests show that the chosen strategy and the deployment of the gateways have a strong impact on packet loss and on the time needed to complete the transmission of the data to the EDs. The code of the ns-3 module is available at https://github.com/madsthom/lorawan.

Keywords: LoRa · experimental · simulation · FUOTA

1 Introduction

Low power wide-area network (LPWAN) technologies offer long-range communication capabilities that consume little energy, at the cost of limited bandwidth. LPWAN is gaining prominence for use cases based on battery-powered Internet of Things (IoT) devices, for which the End Device (ED) count is predicted to grow rapidly [11].

LoRaWAN is a Media Access Control (MAC) and networking layer built on top of the LoRa [20] physical layer. The technology has gained momentum in recent years because of its low-cost EDs, the long lifetime of the devices' battery, its reliability based on its ability to send confirmed messages, and its adaptive data rate protocol [4] that allows to optimize the data rate depending on the condition of the network.

S. Longfei and P. Bodhi (Eds.): MobiQuitous 2022, LNICST 492, pp. 455–468, 2023.
https://doi.org/10.1007/978-3-031-34776-4_24

As most LPWAN communication technologies, LoRaWAN is asymmetric by design, and provides more support to the uplink channel, since it considers that most use cases focus on collecting data from the physical world. Occasionally, however, downlink communication might also be needed, for instance, to support setting transmission parameters at the EDs or when confirmed traffic is needed by the application. To this aim, LoRaWAN supports three different communication modes, one of them being Class C, which prioritizes downlink communication, and allows the EDs to switch between communication modes when the need arises. Multicast sessions are also supported by the specification, enabling the network to broadcast a single message to multiple EDs listening on the same channel. This functionality can, for instance, be used to support the Firmware Update Over The Air (FUOTA)[1] use case, where large amounts of data need to be delivered to the EDs over LoRaWAN.

This paper investigates the feasibility of extensive downlink communication with moving EDs in LoRaWAN. However, we acknowledge that mainstream network simulators lack Class C simulation capabilities and non Line of Sight (nLoS) scenarios require a validation step employing real device implementations since channel propagation is more complicated when the presence of buildings and other obstacles is taken into account. Thus, this work contributes to the state-of-the-art by means of:

- developing a ns-3 module to simulate communication for LoRaWAN Class C devices;
- identifying a suitable propagation loss model for nLoS LoRaWAN based on measurements from a physical reference application;
- showcase the implemented ns-3 module by proposing and evaluating two simple strategies, by means of both a testbed and simulations, to optimize the communication strategy of LoRaWAN with respect to the scenario at hand.

The structure of the paper is as follows; in Sect. 2 background information is laid out. In Sect. 3 the purpose and the aim of this work are explained. Section 4 presents the reference physical implementation. Furthermore, in Sect. 5 the simulation results of Class C LoRaWAN are described. Finally, in Sect. 6 conclusions are drawn and future work is proposed.

2 Background Information

This section provides an overview of the LoRaWAN technology, of the network simulator 3 (ns-3), which is the tool we used to perform our simulations, and of work related to LoRaWAN extensive downlink communication.

2.1 An Overview of LoRaWAN

The growing demand for IoT-type connectivity [11] recently paved the way for various LPWAN communication technologies. Use cases usually focus on

[1] Firmware Over The Air means that an update is supplied wirelessly to the End Devices.

collecting small amount of data over large deployments, and in that sense LPWAN is a great match for ubiquitous sensing. In fact, LPWAN provides the ability to send small messages over a long distance without consuming a large amount of energy.

Among the plethora of LPWAN technologies that hit the market in the past few years, LoRaWAN [16] is arguably the most widely adopted, and the one that is pushing the LPWAN paradigm forward the most [10]. Uplink focused LPWAN technologies such as, SigFox [21] and NB-IoT [17] are the most used in IoT applications, where the EDs benefit from having a long battery life.

LoRaWAN operates on the unlicensed ISM bands between 137 and 1020 MHz, and claims a range of up to five kilometers in urban areas and up to 15 km in line of sight scenarios [20]. In Europe, the LoRaWAN protocol has duty cycle restrictions that apply to all functioning modes and must be complied with. Because of the specification's frequency layout in relation to the available regulatory sub-bands, in LoRaWAN such restrictions translate to 1% Duty Cycle for uplink communication, and 10% Duty Cycle for downlink.

One of the most important operational parameters exposed by the underlying LoRa modulation is that of the Spreading Factor (SF), which decides the "spreading" of the signal over the available frequency band. If multiple packets are transmitted simultaneously and on the same frequency but with different SF, all packets can be typically demodulated correctly (provided some restrictions on the reciprocal powers are met) [9]. Lower values of SF (with 7 being the minimum) mean that the communication happen at shorter range but packet will be transmitted at a higher data rate, while higher values (with 12 being the maximum) determine a longer range but longer on-air time to transmit the same amount of data.

LoRaWAN EDs can be configured to operate according to three different communication modes called Class A, Class B and Class C. **Class A** is primarily for uplink messages and requires the least amount of power to operate, furthermore, it is mandatory for all EDs to implement it. Devices can receive in fact downlink messages can be received only after the ED sends an uplink message. **Class B** is a beaconing mode, and compared to Class A it has a lower latency due to pre-scheduled transmission windows (ping slots), but it consumes more power. When operating under **Class C**, the EDs keep their interface on at all times, to receive packets without an uplink transmission. This mode consumes the most power of the three configurations and it is intended to be used for a short time only, with the ED going back to other modes as soon as possible. One of the most prominent use cases for Class C is to update the software/firmware of the ED.

When operating in Class A, whenever the ED sends an uplink message, it must open two receive windows, **RX1** that operates on the same frequency as the uplink message itself, and **RX2** that operates on a configured fixed frequency. The timing in which these windows are opened is configurable but normally the RX1 window opens after a one-second delay and the RX2 window opens after a two seconds delay after the uplink transmission is completed. The LoRaWAN

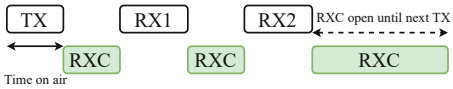

Fig. 1. Class C receive windows

specification [16] states that an ED in Class C must open extra receive windows RXC, meaning that the ED should listen to incoming RX signals whenever it is not transmitting, whose parameters are the same as RX2. The behaviour of Class C devices is shown in Fig. 1, and Class A devices operate in the same manner, except for the absence of RXC.

Targeting multiple EDs with the same packet transmission can be accomplished using **multicast**, which is a way to avoid transmitting the same packet multiple times and is possible for both Class B and C. Using multicast requires creating a multicast group of EDs and having that group perform a clock synchronization to match the timing on every ED, for example to respect the downlink slots of Class B. The tolerance regarding the timing for multicast commands to work properly is approximately one second [3].

Implementing a LoRaWAN architecture (see Fig. 2) requires a setup including at least a **network server**, a **gateway (GW)**, an ED, and an application. The modularity in the architecture allows extending the network, for example by adding multiple GWs covering a larger area. A well-supported open-source LoRaWAN network server was implemented by ChirpStack[2]. The ChirpStack network server implements Adaptive Data Rate (ADR) [4], authentication, multicasting, device class management, GW management, logging, etc. Furthermore, it allows integration using an exposed API targeting the Network server and Application server.

2.2 Network Simulator 3

Complex network technologies such as LoRaWAN are often evaluated and tested by means of network simulation, which allows to investigate how the protocol behaves depending on its different parameters, to test it on complex scenarios at a cost lower than physical implementation, and to measure properties that are not usually readily accessible on testbeds.

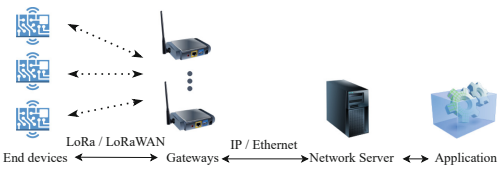

Fig. 2. LoRaWAN architecture

[2] https://www.chirpstack.io/network-server/.

Network simulators typically use the discrete-event paradigm, modelling the system's behavior as a (discrete) sequence of timed events where each change (event) in the systems happens at a specific time in the simulation timeline. This enables the simulator to jump to the next scheduled operation, with the assumption that no state of the system changes in the time in-between the events.

Network Simulator 3 (ns-3) is an open-source discrete network simulator licensed under the GNU GPLv2 license and is free for research, development, and commercial use [13]. The ns-3 platform provides models to simulate how different packet data technologies work, all driven by a general simulation engine [1]. ns-3 has a substantial [15] community which contributes to the platform and the many modules found in their ns-3 App Store.

The LoRaWAN ns-3 module [14] provides support for most of the basic functionalities of the protocol. The module supports a network server implementation with multiple GWs and Class A EDs. In [9], the authors used the module to evaluate LoRaWAN's scalability in a smart-city scenario with many EDs (up to 10^4). However, the module currently supports Class A EDs only. In [6], the authors evaluate the scalability of LoRaWAN Class B. The authors implemented an ns-3 module to simulate Class B EDs and investigated the limitations of Class B for uplink and downlink frames. They conclude that the limiting factor for Class B is the Duty Cycle Restrictions.

In this work, we will expand the ns-3 LoRaWAN module to support basic multicast operation and Class C operations.

2.3 Related Work

In [5] the authors consider downlink transmission issues for LoRaWAN Class A, and in particular the limits of downlink communication, namely, the downlink frames scheduled to be received by EDs in RX1 or RX2 receive windows (see Fig. 1). They propose three solutions to mitigate packet loss in networks with low, medium, and high amounts of downlinks: (a) using multiple GWs with overlapping coverage to be able to mitigate packet loss due to Duty Cycle Restrictions, (b) Allowing GWs to send and receive frames at the same time, and (c) implementing a balancing GW selection algorithm to select the best GW for the next downlink frame.

[2] investigates the throughput of different LoRaWAN GW deployments for downlink communication, and compares three algorithms for selecting the best GW for downlink communication. It appears that balancing the load (number of EDs) per GW increases the throughput of the system with respect to choosing a GW based on signal quality only.

In [3], the authors review the LoRa alliance's FUOTA specifications, and provide simulations of FUOTA scenarios using Class B and C. Via their simulations, they conclude that doing FUOTA via Class B takes 17% more time, but uses 550 times less energy than Class C.

In [8], authors discuss the requirements for FUOTA to work for LPWANs, and propose three key requirements, namely (i) Efficient multicasting, (ii) Packet

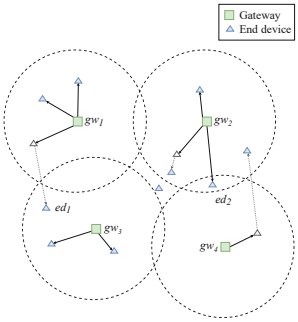

Fig. 3. A schematic overview of the reference scenario considered in this paper.

recovery, and (iii) Verification of authenticity and integrity of the update binaries. The authors developed a reference implementation of a FUOTA scenario with LoRaWAN, where EDs switch from Class A to C, perform firmware update, then switch back to Class A. However, as the authors point out, this approach is good for stationary EDs only, since data rates and device groups need to be defined before doing the update, which clashes with the dynamicity of mobility scenarios.

Bottom line, all these surveyed approaches work well when selecting GWs for stationary EDs, but do not take into account mobility, and it can be argued that they would be challenged when considering moving EDs.

3 Extensive Downlink Communication in LoRaWAN

The purpose of this work is to evaluate the feasibility of extensive downlink communication with moving EDs and to study how to enhance current approaches to deliver a large quantity of data to EDs, for example, to perform FUOTA. We assume that data is subject to linear coding, and that a fixed amount of packets are needed to reconstruct the firmware update correctly. [7]

The scenario is roughly represented in Fig. 3, where multiple GWs are sending downlink messages to moving EDs concurrently. First of all, the network needs to be able to orchestrate multiple GWs to efficiently send downlink messages without interference. When moving devices are part of the picture, the quality of the communication link to the GW changes in correlation to the devices' movement. For example, in Fig. 3 the network needs to know which gateway(s) should be activated. EDs can use any packets they receive to recompose their update.

In a LoRaWAN network with unconfirmed downlink communication, the GWs do not get any indication of the signal strength perceived by the EDs, which in turn does not allow the GWs to adapt to the evolution of the scenario. Thus, to provide better support for moving EDs, we consider the need for an ED to signal the network. The network then uses the feedback regarding the

link quality to estimate which GWs are currently able to reach a given ED with their data, and if there is the need for a handover.

The communication strategies that we consider are:

- Out of Signal (OOS): EDs only send packets when their Received Signal Strength Indication (RSSI) is below a certain threshold or when they are not receiving any packets, and the network server selects which GWs to activate based on the information;
- Periodic: EDs send UL data periodically, and the network server selects which GWs to activate based on the information;
- Bulk: EDs never send UL data, and all GWs are always active.

In the simulations (described in Sect. 5), EDs move according to the `RandomWalk2dMobilityModel` ns-3 mobility model, moving within a rectangular area of side of 2000 m, and pick a random direction after either 10 min or 500 m; speed is chosen from an uniform random distribution between 2 and 6 m/s.

4 Real World Data Collection

In this section, we describe the implementation of the test setup and how the GW, network server, and application server worked together. The testbed made use of two Multitech Conduit Gateways [12] with LoRa extension cards. The GWs were running the latest firmware (5.3.3) and were configured as packet forwarders. A B-L072Z-LRWAN1 board [19] (LoRaWAN discovery board) is used as ED. The board contains a Semtech SX1276 LoRa chip [18]. The board runs Mbed OS[3] version 6.9, which implements a LoRaWAN connectivity stack complying with the specifications of LoRa [16] and supports the embedded discovery platform "B-L072Z-LRWAN1". The application executed on Mbed OS implements the following 3 steps. (i) A **Join procedure** connects the ED to the LoRaWAN network and is triggered by a power cycle or reset of the device. (ii) As soon as the ED is connected, it sends an uplink message with the ADR flag true that starts the **ADR process**. (iii) Once the optimal connectivity setting is found, the ED **switches to Class C** to receive a large quantity of data, and logs information regarding every message sent or received.

For the network to be fully functional, we installed the ChirpStack Network server and Application server from Sect. 2.1 on a central server. This implementation of the network server was chosen due to the ability to change GW based on the latest uplink transmission. The test application used to send FUOTA packets was written in python. It establishes a connection to the application server using gRPC[4], which is a high-performance universal remote procedure call framework.

The studied scenario featured an urban environment populated with buildings in Aalborg, Denmark. Given the nLoS nature of the scenario, the range

[3] https://os.mbed.com/mbed-os/.
[4] https://www.grpc.io/.

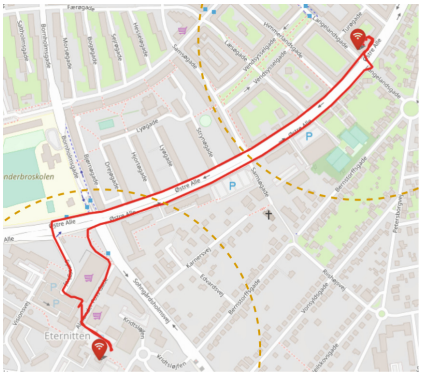

Fig. 4. Map of gateway deployment in Aalborg, Denmark

of the GWs was reduced, which resulted in a signal dead zone. The GWs were placed in two apartment buildings approximately one kilometer apart, one located on the 6th floor and the second on the 3rd floor. The GW deployment showing the approximate range and location of each GW can be seen in Fig. 4, which also shows the route taken to move by bicycle from the coverage of one GW to the other. Conceptually, we considered that the experiments involved travelling by bicycle on the same route at approximately the same speed. However, this is a limitation of the experimental setting, since the route contains four intersections that disturb the continuity of the test runs.

The test setup consisted of two scenarios where two different strategies were used for signalling to the GW that an ED is moving out of range. GPS and RSSI data are recorded as they are acquired (periodically for the GPS, with each packet transmission for the RSSI).

4.1 Test Scenario 1 - Out of Signal

The first test scenario made use of an "Out of signal" message from the ED. After receiving each downlink message, the ED (i) changes its state to an

Table 1. Test result for "Out of signal" scenario.

RSSI value:	***–140***	***–130***	***-120***
MIN SNR	–79	–82	–63
MIN RSSI	–128	–127	–128
AVG RSSI	–90,7	–87,3	–92,7
Packet loss %	42,1	39,4	6,9
Number of packets sent	140	155	188
Number of packets received	81	94	175

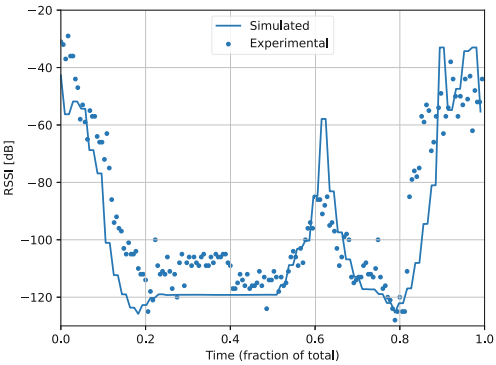

Fig. 5. Rssi for "out-of-signal" threshold: −120, and simulated RSSI.

out-of-signal state if the RSSI is under a given threshold, or (ii) changes its state back to under-coverage if the RSSI is over the threshold. When the ED is in the out-of-signal state, it sends every 10 s an uplink message to request a connection to any GW in range.

We experimented with 3 different RSSI thresholds, namely −120, −130, and −140 based on the nominal sensitivities [18] for the LoRa chip used. The results (see Table 1) show that for thresholds of −140 and −130 the minimum RSSI value is higher than the threshold, resulting in no out-of-signal packets sent and 42, 1% and 39, 4% packet loss. This suggests that the test setup is not able to yield the maximum range. However, for an RSSI value just above the minimum observed RSSI (−120), the approach yields a better result of 6, 9% packet loss. The RSSI value throughout this test run is shown in Fig. 5, which was also used to set the parameters characterizing the propagation model for the simulations of Sect. 5 (the solid line in the figure). Choosing a value even higher could result in missing some packets that the ED could have received.

4.2 Test Scenario 2 - Periodic Acknowledgement

The second test scenario consists of periodic confirmed downlink messages, having the EDs respond with an uplink acknowledgment packet allowing the network server to select the closest GW for transmitting further. This allows the network server to evaluate the signal strength between GW and EDs to select a gateway for the next downlink message. Three test runs were done with a periodicity of every 3rd, 6th, and 9th downlink packet.

The results of the tests (see Table 2) show that sending confirmed messages as often as every 3rd or 6th packet yields a packet loss of 43, 4% and 26, 4%. This can be due to a collision between downlink and uplink messages. Sending confirmed messages less frequently yields a better result of 14, 9% packet loss. This can also be a matter of better timing in terms of being in and out of the signal dead zone at the right moment.

Table 2. Test result for "Periodic Acknowledgement" scenario.

Confirmed message every:	*3rd packet*	*6th packet*	*9th packet*
MIN SNR	–51	–69	–63
MIN RSSI	–122	–128	–127
AVG RSSI	–92,5	–98,3	–99,0
Packet loss %	43,4	26,4	14,9
Number of packets sent	159	121	161
Number of packets received	90	89	137

5 Simulation of LoRaWAN Class C

A LoRaWAN module [14] for ns-3 was developed, to further study the scenario at hand. To the aim of evaluating the feasibility of FUOTA-like applications with extensive downlink communication, Prior to our work, the existing LoRaWAN module implemented both PHY and MAC for Class A of LoRaWAN only. The module implements the architecture seen in Fig. 2, which consists of EDs, gateways, and a network server.

The Class C implementation was developed on top of the current LoRaWAN ns-3 module, mainly by modifying the MAC layer implementation for GWs and EDs to accommodate Class C. The Class C End Device can switch from Class A to Class C to simulate a FUOTA-like application (see Fig. 6). When the downlink communication is finished, another command is sent by the GW, which tells the MAC layer on the ED to switch back to Class A.

When in Class C, the ED behaves as per the LoRaWAN Class C specification (see Sect. 2.1) and in particular whenever a transmission (TX) is performed, the RXC is opened and closed again after the RX1 delay. If nothing is received in

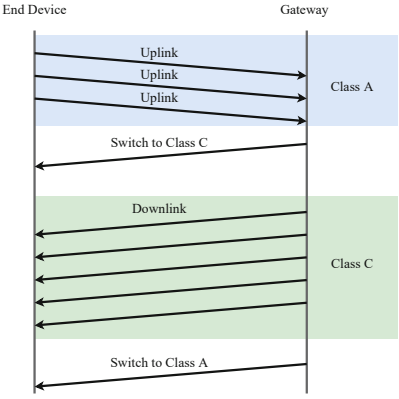

Fig. 6. Simulation scenario: Switch from Class A to Class C

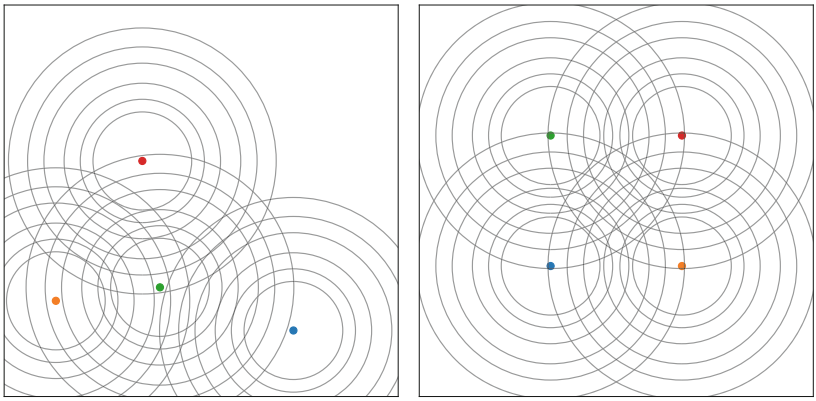

Fig. 7. Examples of a Random and Grid Gateways deployment. Dots represent Gateways, while circles represent the achievable range with each value of the Spreading Factor parameter.

RX1 the RXC is open again and is kept open until the next transmission, i.e.: the ED is always ready to receive packets whenever it is not transmitting.

As a preliminary step, we tuned the propagation loss model used in ns-3 to match our experimental data. In particular, we chose to rely on a Log-distance path loss model with $n = 4$ to capture the densely built environment of Aalborg, and a reference path loss of 60 dB at 2 m. In order to ensure this model matches the actual data, we re-created the experimental scenario in simulation, using the GPS measurements of the test route of the ED and the GW locations. The resulting path loss estimate can be seen in the solid line in Fig. 5, and the resemblance between the collected data and the simulated received power turned out to be satisfyingly close and motivated us to keep the aforementioned path loss model for the simulations described in the remainder of this work.

Figure 7 shows two examples of GW deployments, the left with GWs placed randomly, the right featuring a "grid" placement. While the grid configuration clearly provides the best coverage of the entire deployment area, it might not always be possible to achieve such a GW placement in the real world. In fact, actual deployments, especially when crowdsourced as in the case of The Things Networks, might look closer to the random deployment, where some relatively large areas have no coverage. We also remark that GWs are assumed to be transmitting FUOTA packets sequentially in time, and thus any overlap in the coverage area of GWs in this work does not lead to interference between downlink packets.

The Empirical Cumulative Distribution Function (ECDF) of the time needed to download the entire update (i.e., the 100 packets representing a firmware update) is shown in Fig. 8 for both the cases of a random deployment and a grid deployment. These plots were obtained using SF 12 and the Bulk strategy to transmit the FUOTA packets. As expected, random and grid placement yield

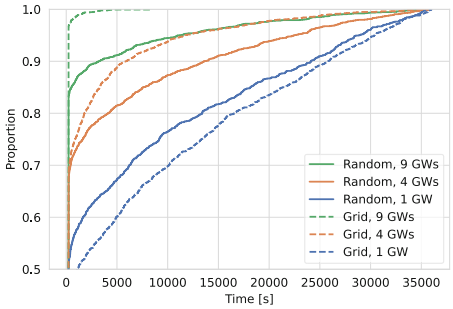

Fig. 8. ECDF of Firmware Update Download Completion Times.

similar results when a single GW is used. As the number of employed GWs grows, however, the grid placement is clearly superior: when 9 GWs are used, all EDs correctly receive the update within three hours, with 95% of the EDs within few minutes. In the random deployment with 9 GWs, instead, only 90% of the EDs are able to collect all the required packets within the first few minutes of the update, while the other 10% needs more than 7 h (25200 s) to roam back within a coverage area and be able to complete the download.

The left side of Fig. 9 compares different policies, and it shows that the Bulk strategy is the best performer for the grid deployment of 4 GWs. The right side of Fig. 9 compares the Bulk strategy in the same scenario with different SF parameter. The results hint that SF 11 is too slow, while a higher data rate allows most of the device to complete the FUOTA faster, but by having a shorter range it leads to having a small number of EDs ending up with slower updates.

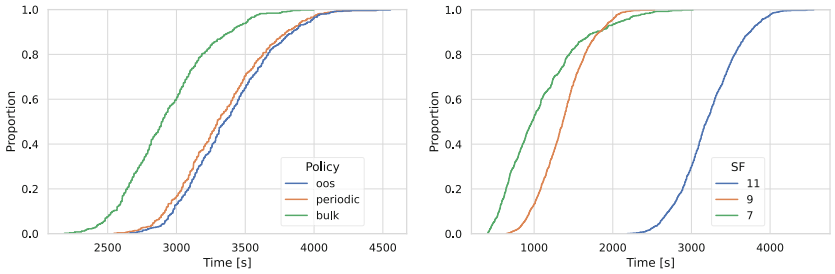

Fig. 9. ECDF of time to complete the update. On the left, we show different policies with SF 11. On the right, we use the bulk policy, various values of Data Rate.

6 Conclusion and Future Work

This paper addressed the usage of LoRaWAN Class C communication for extensive downlink communication in a mobility scenario. A Class C ns-3 component

was developed on top of the ns-3 LoRaWAN module, as well as a reference application. The resulting framework was showcased by testing two simple strategies that EDs can use to provide feedback regarding their link quality to GWs. The results showed the choice of the signalling strategy between EDs and GWs and the parameters of the LoRaWAN communication has a huge impact on packet loss.

Several future works can be considered as immediate follow-ups on this work. First, more complex strategies can be proposed to cope with the scenario at hand. Moreover, the LoRaWAN ns-3 module can be used to simulate a large network of mobile devices, to study the system's behavior when the network scales to hundreds of EDs. Since the firmware update is broadcast, we will investigate rateless coding schemes that could work better than linear coding. Furthermore, the ns-3 LoRaWAN module should be expanded with a Class B implementation to compare its performance to Class C, and also to create a module that can simulate all three classes of the LoRaWAN protocol. To be able to test different GW selection algorithms, the network server implementation in ns-3 should be expanded with the ability to"install" different applications to control the GWs. Finally, we plan to experiment with more complex GW placements to investigate if they can yield better results, for example with and without overlap in GW ranges.

Acknowledgment. This work was partly funded by the ERC Advanced Grant LASSO; and by the Villum Investigator Project "4OS: Scalable analysis and Synthesis of Safe, Small, Secure and Optimal Strategies for Cyber-Physical Systems".

References

1. ns 3 consortium: ns-3 tutorial. Tech. rep., ns-3 consortium (January, 2009)
2. Abboud, S., El Rachkidy, N., Guitton, A., Safa, H.: Gateway selection for downlink communication in Lorawan. In: 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE (2019)
3. Abdelfadeel, K., Farrell, T., McDonald, D., Pesch, D.: How to make firmware updates over lorawan possible. In: 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), pp. 16–25. IEEE (2020)
4. Coutaud, U., Heusse, M., Tourancheau, B.: High reliability in lorawan. In: 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–7. IEEE (2020)
5. Di Vincenzo, V., Heusse, M., Tourancheau, B.: Improving downlink scalability in lorawan. In: ICC 2019–2019 IEEE International Conference on Communications (ICC), pp. 1–7. IEEE (2019)
6. Finnegan, J., Brown, S., Farrell, R.: Evaluating the scalability of lorawan gateways for class b communication in ns-3. In: 2018 IEEE Conference on Standards for Communications and Networking (CSCN), pp. 1–6. IEEE (2018)
7. Fujimura, A., Oh, S.Y., Gerla, M.: Network coding vs. erasure coding: Reliable multicast in ad hoc networks. In: MILCOM 2008–2008 IEEE Military Communications Conference, pp. 1–7. IEEE (2008)

8. Jongboom, J., Stokking, J.: Enabling firmware updates over lPWANs. In: Embedded World Conference(2018)
9. Magrin, D., Centenaro, M., Vangelista, L.: Performance evaluation of LoRa networks in a smart city scenario. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–7. IEEE (2017)
10. Mekki, K., Bajic, E., Chaxel, F., Meyer, F.: Overview of cellular lPWAN technologies for IoT deployment: Sigfox, Lorawan, and NB-IoT. In: 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (Percom Workshops), pp. 197–202. IEEE (2018)
11. Meulen, R.: Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016. Gartner Letzte Aktualisierung **7** (2017)
12. Multitech gateway. https://www.multitech.com/models/94557602LF (July, 2021)
13. ns-3 network simulator. https://www.nsnam.org/ (July, 2021)
14. An ns-3 module for simulation of lorawan networks. https://apps.nsnam.org/app/lorawan/ (July 2021)
15. ns-3 statistics. https://www.nsnam.org/about/statistics/ (July 2021)
16. N. Sornin, A. Yegin, e.a.: Lorawan 1.1 specification. Tech. rep., Lora Alliance (October 2017)
17. Popli, S., Jha, R.K., Jain, S.: A survey on energy efficient narrowband internet of things (NBIoT): architecture, application and challenges. IEEE Access **7**, 16739–16776 (2018)
18. Semtech sx1276. https://www.semtech.com/products/wireless-rf/lora-core/sx1276 (July 2021)
19. Stm32l0 discovery kit lora, sigfox, low-power wireless. https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html (July 2021)
20. Wenyan, Z.: Technical overview on LORA physical layer and mac layer. Mob. Commun. **2017**, 17 (2017)
21. Zuniga, J.C., Ponsard, B.: Sigfox system description. LPWAN@ IETF97, November 14th 25 (November 2016)