XXXVII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS - SBrT2019, 29/09/2019–02/10/2019, PETRÓPOLIS, RJ

Performance Evaluation of LoRaWAN and RS-LoRa in Mobile Scenarios

Dayrene F. Fonseca, Rodrigo C. de Lamare and Ewerton L. Madruga

***Abstract*— LoRaWAN and RS-LoRa are two of the most recent MAC layer proposals for the Internet of Things (IoT). In this paper, we extend the performance evaluation of these protocols to mobile scenarios, using relevant metrics such as the average Packet Error Ratio (PER) and the average network delay. Different scenarios are evaluated by varying the number of nodes, gateways and the mobility model of the end devices. We also propose a pseudo-code to represent the Adaptive Data Rate (ADR) mechanism implemented by the LoRaWAN nodes. Our pseudo-code makes a more illustrative representation of how the ADR mechanism works, facilitating its comparison with potential proposals to improve the performance of LoRaWAN.**

***Keywords*— Internet of Things, MAC protocols, LoRaWAN, RS-LoRa.**

I. INTRODUCTION

The IoT applications promise to address many challenges that humanity is facing today, such as population growth, energy crisis, resource depletion, and environmental pollution. In this context, Low-Power Wide-Area (LPWA) technologies are recommended as the underlying networking solution for a variety of IoT applications, since they can offer low-power operation, low-cost and low-complexity end devices, which are also able to establish communications over a large geographical area. Recently, the research community has been interested in studying the LPWA technologies, especially in finding new solutions to improve its MAC layer, which is essential to ensure the scalability, energy efficiency and high performance requirements demanded by the IoT applications.

Among the MAC layer solutions proposed for the LPWA technologies, LoRaWAN has stood out for receiving significant attention from both industry and academia in recent years. Although LoRaWAN offers a compelling combination of long-range and low-power consumption data transmission, it still faces several challenges [1].

A. *LoRaWAN Limitations and Related Work*

The LoRaWAN limitations are fundamentally related to scalability and reliability. That is because LoRaWAN networks are vulnerable to the capture effect, which makes stronger signals survive collisions, while weaker signals get lost, reducing the performance of nodes that are far away from the gateway [2]. The research community has made significant contributions to study the performance of LoRaWAN, as well as to improve it; some examples are presented below.

A comprehensive analysis of the capabilities and limitations of LoRaWAN can be found in [1]. Haxhibeqiri *et al.* [2] investigate how the number of end devices and their throughput requirements, affect the scalability of single-gateway LoRaWAN deployments. In [3] is conducted a scalability study of LoRaWAN, focusing on the effect of confirmed versus unconfirmed messages. Reynders *et al.* [4] investigate the performance of LoRaWAN in scenarios with both single and multiple gateways, concluding that acknowledgements are not scalable and multiple gateways improve the network performance considerably. Despite the efforts, in general, the conducted studies only consider scenarios with static nodes, and less attention has been provided to the effect of the nodes' mobility over the LoRaWAN network performance. Although in [5] and [6] are carried out experimental evaluations of LoRaWAN under mobile scenarios, the experiments only consider one end device, overlooking the effect that the mobility of one node can have over the performance of other nodes and on the overall network performance.

Some contributions improving the scalability and reliability of LoRaWAN, have also been reported. A summary of research works that enhance the performance of LoRaWAN and have been published from 2015 to September 2018, is presented in [7]. Among all the proposals, one of the most recent and relevant is introduced in [8], where Reynders *et al.* propose RS-LoRa, a new MAC protocol that improves the LoRaWANs reliability and scalability through a two-step lightweight scheduling coordinated by the LoRaWAN gateways. Although authors in [8] carry out a performance study of RS-LoRa and LoRaWAN, comparing their performance in terms of the average PER, the network throughput and fairness, their analysis only considers scenarios with static nodes, and the used metrics do not contemplate an important parameter demanded by the IoT applications, such as the network delay.

B. *Contributions*

Motivated by the above state-of-the-art, in this paper, we evaluate through simulations the performance LoRaWAN and RS-LoRa in terms of the average PER and the average network delay. Our simulations create scenarios with single and multiple gateways, considering both static and mobile nodes inside a given geographical area. We also propose a pseudo-code to represent the LoRaWAN ADR mechanism, which is used to show the main differences between the legacy LoRaWAN and the recently proposed RS-LoRa protocol.

The paper is organized as follows. Section II describes the physical and MAC layers of the LoRaWAN and RS-LoRa based networks. In Section III, the proposed pseudo-code is used to introduce the ADR mechanism of LoRaWAN, and to compare it with the lightweight scheduling of RS-LoRa. Section IV presents a performance comparison between

---

Dayrene F. Fonseca and Rodrigo C. de Lamare are with the Center for Telecommunications Studies, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro 22451-900, Brazil (e-mail: dayreneff@cetuc.puc-rio.br; delamare@cetuc.puc-rio.br). Ewerton L. Madruga is with Inmetro, Rio de Janeiro 25250-020, Brazil (e-mail:elmadruga@inmetro.gov.br). This research was supported in part by the Brazilian Agency FAPERJ under Grant E-26/200.822/2019.

LoRaWAN and RS-LoRa under different scenarios and considering both static and mobile nodes. Finally, Section V gives some concluding remarks.

II. SYSTEM MODEL

A. Network Architecture

Since RS-LoRa is built upon the LoRaWAN MAC layer, networks based on these protocols have the same topology, and their nodes, have a similar network stack [8].

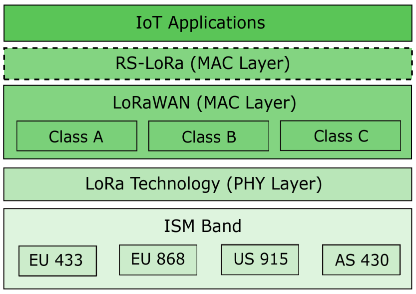

Fig. 1. Network stack of the LoRaWAN and RS-LoRa nodes. Dashed lines indicate that this layer only exists for the RS-LoRa nodes.

1) Physical Layer: As shown in Fig. 1, both LoRaWAN and RS-Lora use LoRa as physical layer technology, handling the communication between end devices and gateways in different sub-GHz frequency bands, depending on the local frequency regulations. In this paper, we address the operation in the 868 MHz ISM band.

LoRa modulation is based on the Chirp Spread Spectrum (CSS) scheme and defines the following relation between the bit rate R_b, bandwidth B, spreading factor SF and symbol rate R_s [9], [10]:

$$R_b = \frac{B \times SF}{2^{SF}} = R_s SF \quad [bits/sec]. \tag{1}$$

The symbol period T_s is defined as the reciprocal of R_s.

As can be observed in (1), a lower SF leads to a higher transmission rate and shorter transmission time, but according to the LoRa specifications, it requires a higher Signal to Noise Ratio (SNR) and corresponds to the higher sensitivity of the receiver [9].

2) LoRaWAN MAC Layer: LoRaWAN is an open source MAC protocol standardised by the LoRa Alliance that runs on top of LoRa's physical layer, as depicted in Fig. 1. A LoRaWAN network consists of the following elements: end devices, gateways, network server and application server. Among them, one of the most important is the network server, which is responsible for monitoring the gateways and end devices, forwarding incoming packets to the corresponding application server and removing duplicated messages [7], [11].

As can be observed in Fig. 1, the LoRaWAN standard defines three classes of end devices, and each one constitutes a trade-off between energy consumption and network downlink communication delay. Based on the application requirements, end devices can switch between classes A, B or C, and class A[1] must be implemented on all the nodes by default.

III. LORAWAN AND RS-LORA COMPARISON

In this section, we discuss how the LoRaWAN and RS-LoRa protocols adapt and optimize the transmission parameters of the end devices in order to increase the Packet Delivery Ratio (PDR).

A. LoRaWAN Protocol

In a LoRaWAN based network, the data rate and transmission power adaptation at the end devices is carried out through the ADR mechanism, which has two parts running asynchronously, one on the network server and the other one at the end devices[2]. If an end device wants to allow the network server to manage its transmission parameters, it will set the ADR bit in its uplink messages. Then, the network server will control the transmission parameters of the end device through the appropriate MAC commands[3]. When the network server is unable to control the data rate of the end device, or when the ADR bit is not set in uplink messages, the node can manage its parameters itself using the ADR mechanism that resides at the end device side [7], [12].

We propose the Algorithm 1 to represent how the ADR mechanism works at the nodes' side. According to the LoRaWAN specification [12], at the end devices side, the ADR algorithm will increment an ADR_ACK_CNT counter each time a new uplink packet is transmitted (repeated transmissions do not increase this counter). It also includes the ADR_ACK_LIMIT and ADR_ACK_DELAY parameters, whose values have been set to 64 and 32, respectively [12].

As depicted in Algorithm 1, when a node n wants to transmit a message, it first checks if there is a free channel (lines 4-7). We denote the set of available channels as $\mathcal{L} = \{1, 2, ..., L\}$. If a channel is found, the node must verify if an adjustment of the transmission parameters is required (lines 9-18). LoRaWAN specifications state that after ADR_ACK_LIMIT uplink messages ($ADR\_ACK\_CNT >= ADR\_ACK\_LIMIT$) without any downlink response, the end device must set the $ADRACKReq$ bit (lines 10-12), requesting the network server to respond with a downlink packet within the next ADR_ACK_DELAY uplink messages. Any downlink message received during this interval resets the ADR_ACK_CNT counter. If no reply is received after a total of $ADR\_ACK\_LIMIT + ADR\_ACK\_DELAY$ uplink messages, the end device must try to regain connectivity by adjusting its transmission parameters (lines 14-17).

As shown in Algorithm 1 (lines 20-22), the node first steps up the transmit power to the maximum value, if possible.[4] If increasing the transmission power up to the maximum value is not enough, the end device must further lower its data rate step by step every time ADR_ACK_DELAY is reached (lines 23-25). The reliability is increased by decreasing the data rate because, as explained in Section II-A.1, in LoRa modulation a lower data rate corresponds to a higher SF, and increasing the SF by 1 brings an increment of 2.5 dB in the

---

[1] In this paper, we carried out the performance evaluations considering Class A end devices.

[2] In our evaluations only the ADR mechanism at the nodes' side is enabled.

[3] A complete explanation of the LoRaWAN MAC commands can be found in the LoRaWAN specification [12].

[4] In LoRaWAN, default transmission power is the maximum transmission power allowed for the device considering device capabilities and regional regulatory constraints [12].

Algorithm 1: Data rate and transmission power control at each LoRaWAN node

```
Input: $\mathcal{L}$: Set of available channels;
    $D_l$: Data rate for channel l, $\forall l \in \mathcal{L}$;
    $D\text{min}_l$ : Minimum data rate for channel l, $\forall l \in \mathcal{L}$;
    $S_l$: SF for channel l, $\forall l \in \mathcal{L}$;
    $\mathcal{P}$: Set of allowed transmission powers;
    $p\_idx$: Index to search in $\mathcal{P}$. The transmission power
    increases as $p\_idx$ decreases, achieving its maximal value
    when $p\_idx$ is equal to one;
Output: Transmission Parameters: $C_n$: Selected channel;
    $S_n$: Selected SF;  $P_n$: Selected transmission power.

 1: $Flag \leftarrow False$               # Indicates if a channel is found
 2: $Flag\_ADR \leftarrow False$ # Indicates the need to adjust the
    transmission parameters
 3: $ADRACKReq \leftarrow 0$       # Indicates if the ADR bit is set
 4: if A free channel is found then
 5:     $C_n \leftarrow l$                          # $l$ is the channel found
 6:     $Flag \leftarrow True$
 7: end if
 8: if $Flag = True$ then
 9:     if ADRACKReq = False then
10:         if $ADR\_ACK\_CNT \geq ADR\_ACK\_LIMIT$ then
11:             $ADRACKReq \leftarrow True$
12:         end if
13:     else
14:         if $ADR\_ACK\_CNT \geq$
            $ADR\_ACK\_LIMIT + ADR\_ACK\_DELAY$ then
15:             $Flag\_ADR \leftarrow True$
16:             $ADR\_ACK\_CNT \leftarrow 0$
17:         end if
18:     end if
        # Adjust the transmission parameters, if necessary
19:     if Flag_ADR = True then
20:         if $p\_idx \neq 1$ then
21:             $p\_idx = p\_idx - 1$
22:         else
23:             if $D_l > Dmin_l$ then
24:                 $D_l \leftarrow D_l - 1$
25:             end if
26:         end if
27:         $Flag\_ADR \leftarrow False$
28:     end if
29:     $S_n \leftarrow S_l$
30:     $P_n \leftarrow P_{p\_idx}$
31: end if
```

Algorithm 2: Determine transmission parameters at each RS-LoRa node

```
Input: $P_b$: RSS of the received beacon at node n;
    $\mathcal{I}$: Channel set;   $T_s$: Duration of each subframe;
    $\mathcal{S}_i$: Set of allowed SF for channel i, $\forall i \in \mathcal{I}$;
    $P'_i$: Target uplink RSS for channel i, $\forall i \in \mathcal{I}$;
Output: Transmission Parameters: $C_n$: Selected channel;
    $S_n$: Selected SF;  $P_n$: Selected transmission power;
    $T_n$: Offset time.

 1: $P_{tmp} \leftarrow 0$                     # $P_{tmp}$: a temporary variable
 2: $Flag \leftarrow False$            # Denoting if a channel is found
 3: for $i \in \mathcal{I}$ do
 4:     if $P_{tmp} < P'_i < P_b$ then
 5:         $C_n \leftarrow i$                           # Selected channel
 6:         $P_{tmp} \leftarrow P'_i$
 7:         $Flag \leftarrow True$
 8:     end if
 9: end for
10: if $Flag = True$ then
11:     $S_n \leftarrow$ randomly choose a SF from $\mathcal{S}_{C_n}$
12:     $P_n \leftarrow P_{tmp} + P_{pathloss} - 2.5S_n + P_{offset}$
13: else
14:     $S_n \leftarrow 7$                        # Select the lowest SF
15:     $P_n \leftarrow 0$ dBm
16:     $C_n \leftarrow$ arg $max_{i \in \mathcal{I}} P'_i$
17: end if
18: $T_p \leftarrow$ time-of-flight of the packet with selected SF $S_n$
19: $T_n \leftarrow$ rand(0, $T_s - T_p$)   # Select an offset time randomly
```

sensitivity of the gateway [12]. Finally, the end device will start the packet transmission over the channel found, using the SF corresponding to the selected data rate and transmission power chosen in previous steps (lines 29-30).

B. RS-LoRa Protocol

In RS-LoRa, the available bandwidth is divided into one synchronous downlink channel and $\mathcal{I} = \{1, 2, ..., I\}$ uplink/downlink asynchronous channels, which are structured in the unit of frames that last T_f seconds. A frame is further divided into subframes, and each subframe, occupying T_s seconds, starts with a beacon (sent by the gateways) that includes information such as the SFs and Received Signal Strength (RSS) values allowed for each communication channel. Nodes use this information to select the channel and SF for their data transmission, which occurs in the second part of the subframe, in an ALOHA manner. The responsible for specifying the permitted SFs and transmit powers at each channel is a lightweight scheduling mechanism, which constitutes the key innovation of RS-LoRa [8].

In an RS-LoRa network, as in a LoRaWAN, when a node *n* needs to transmit a message, it first looks for an available channel. The most suitable channel is determined by searching over all the available channels, and selecting the one with the highest target RSS that is lower than the estimated RSS from this node to the gateway (lines 3-9) [8].

If a channel is found, a random SF is selected from the allowed SFs for this channel (line 11). Then, based on the selected channel and the target RSS at the gateway, the required transmission power is calculated (line 12). When nodes are very close to the gateway, no channel is determined (line 13). In this case, the lowest SF is selected, since it allows the minimal power consumption and is enough to reduce the amount of destructive interference (lines 14-16). Finally, ensuring that the packet does not interfere with the beacons, a random time is chosen for its transmission (lines 18-19). The RS-LoRa node will wake up at the selected time to start the packet transmission, using the parameters previously selected [8].

At the gateway side, the lightweight scheduling is responsible for coordinating uplink transmissions and for guiding the nodes to select their transmission parameters. Through the scheduling information sent by beacons, nodes are divided into groups where similar transmission powers are used to reduce de capture effect. They are also guided to use different SFs, enabling simultaneous transmissions and thus reducing the packet collisions.

IV. PERFORMANCE EVALUATION

Our performance evaluations are carried out using the Network Simulator 3 (NS-3) and the NS-3 modules proposed in [4] and [8] to evaluate the performance of LoRaWAN and RS-LoRa, respectively. We chose these modules among the existing ones, because they include features such as the possibilities of sending LoRaWAN MAC commands and evaluating scenarios with multiple gateways, which are not present in other proposals [7].[5]

A. Simulation Setup

Our simulations consider two different cases (static and mobile nodes) which are evaluated and compared under the following scenarios:

1) *Single Gateway scenarios*: Only one gateway (GW1) is enabled. It is placed at the center of the coverage area, which corresponds to the dark-blue region shown in Fig. 2, a circle of 1000 meters radius.
2) *Multiple Gateways scenarios*: Seven gateways are deployed as depicted in Fig. 2. The coverage area is the light-blue circle of 1500 meters radius, and the distance between gateways is 1000 meters.

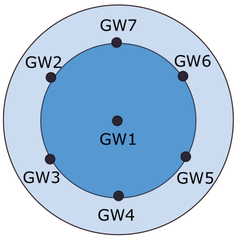

Fig. 2. Coverage area and gateways location for Single Gateway and Multiple Gateways scenarios.

Nodes and gateways are located at the height of one and 30 meters above the ground, respectively. Initially, nodes are distributed uniformly[6] over the dark or light blue area shown in Fig. 2, depending on the scenario used. In the case of mobile nodes, they change their positions following the Random Waypoint Mobility Model offered by the NS-3 simulator. Finally, we use the Rayleigh model together with the Okumura-Hata model for urban areas, as the propagation loss models.

B. Impact of the Nodes' Mobility Model on the Average PER of LoRaWAN and RS-LoRa

In this section, we use the average PER to evaluate and compare the performance of the LoRaWAN and RS-LoRa based networks under Single Gateway and Multiple Gateway scenarios, when considering both static and mobile nodes. A 95% Confidence Interval (CI) was calculated for each average PER value. The CI was plotted in a lighter color along with each one of the average PER curves shown in Figures 3 and 4.

The simulation creates different numbers of nodes for each evaluation (100, 500 and 1000), and each node will send packets of 51 bytes to the network server through the gateways, every two minutes. The frame and subframe duration of the RS-LoRa protocol, are set to 10 and 1 minutes, respectively.

1) Single Gateway Scenarios: As depicted in Fig. 3, in terms of the average PER, the performance of LoRaWAN based networks is barely affected by the mobility of the nodes. The difference, when comparing the deployments of static and mobile nodes, is not higher than 0.2% in the worst case (500 nodes). However, the nodes' mobility affects the performance of RS-LoRa based networks differently. As shown in Fig. 3, when nodes are in motion, the average PER under RS-LoRa is worse than that obtained for static nodes, and this difference starts to become more significant as the number of nodes in the network increases. The reasons behind this are as follows.

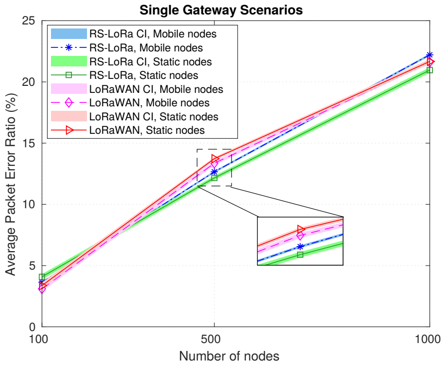

Fig. 3. Average Packet Error Ratio of the LoRaWAN and RS-LoRa based networks under Single Gateway scenarios.

When static nodes are considered, the lightweight scheduling of RS-LoRa can reduce the network capture effect by decreasing the PER of nodes that are far from the central gateway, which results in an improvement of the overall network performance, compared to LoRaWAN [8]. On the other hand, when the RS-LoRa nodes are in motion and need to transmit a message, after selecting the transmission parameters, they will wait for a random time before starting the packet transmission (line 19, Algorithm 2). Hence, they will be in a different location at the transmission instant, where the previously selected transmission parameters may not be suitable according to the lightweight scheduling of RS-LoRa.

Fig. 3 also shown that with fewer nodes in the network (100 nodes), LoRaWAN has a better performance than RS-LoRa. The reason is that in RS-LoRa all nodes close to the gateway are using the same channel, while in LoRaWAN, they can choose between three different channels, resulting in fewer packet collisions compared to RS-LoRa networks.

2) Multiple Gateways Scenarios: Fig. 4 shows the average PER of LoRaWAN and RS-LoRa based networks under Multiple Gateways scenarios when considering both static and mobile nodes. As can be observed, under the studied numbers of nodes, the variations introduced by the nodes' mobility on the average PER of LoRaWAN and RS-LoRa, are not significant. In the worst case (RS-LoRa with 1000 nodes), the difference between the static and mobile nodes deployments is not higher than 0,7%.

---

[5] The selected modules have some small implementation flows that differ from the LoRaWAN standard. Although these variations are not going to make a significant difference in our evaluations, we list them as follows. The MAC layer at the network side is implemented in the gateway instead at the network server; at the gateway, the downlink traffic is delayed if any uplink traffic is being received; and nodes should wait for a random time before transmissions, which does not really happen in ALOHA based networks.

[6] To guarantee a uniform distribution of the nodes over a circular area, we use the UniformDiscPositionAllocator Class offered by the NS-3 simulator.

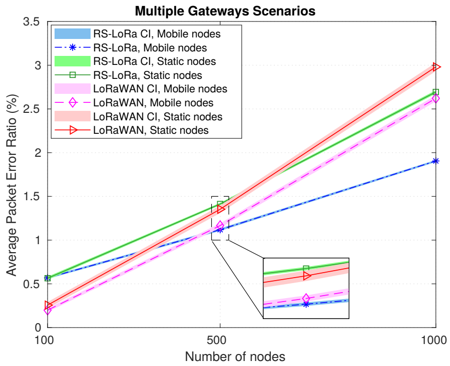

Fig. 4. Average Packet Error Ratio of the LoRaWAN and RS-LoRa based networks under Multiple Gateways scenarios.

Comparing the results presented in Fig. 3 and Fig. 4, we can see that both protocols have a significant performance improvement under Multiple Gateway scenarios. Furthermore, it should be noticed that both LoRaWAN and RS-LoRa based networks have better performance when nodes are in motion. That is because in Multiple Gateways scenarios, mobile nodes can take advantage of the fact that a packet can reach more than one gateway; thus, collisions can be solved by different gateways.

C. Average Network Delay

Fig. 5 shows the average network delay under LoRaWAN and RS-LoRa based networks when Single Gateway scenarios and mobile nodes are considered. The delay of each node corresponds to the average time elapsed since a packet transmission until the reception of the corresponding acknowledgement. As can be observed, the average network delay under RS-LoRa is significantly higher than that under LoRaWAN, regardless of the number of nodes. These results can be explained as follows.

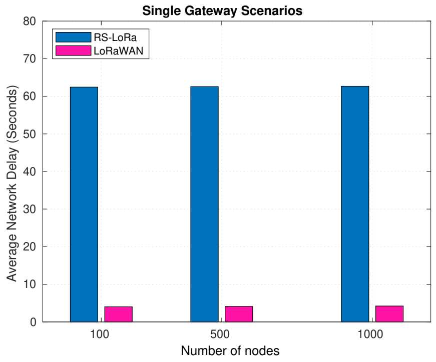

Fig. 5. Average network delay of LoRaWAN and RS-LoRa based networks under Single Gateway scenarios with mobile nodes.

In both LoRaWAN and RS-LoRa, when an uplink message arrives at the gateway, two slots for a potential downlink transmission are reserved. If acknowledgements or other data messages have arrived from upper layers to the scheduled node, before its transmission, the gateway should verify if there are any incoming messages on the current channel or any outgoing packets. If the answer is negative, the message is transmitted; otherwise, the gateway will wait for one second and then transmit the potential acknowledgement or data message in the next reception slot [4], [8].

On the other hand, the RS-LoRa gateways are also responsible for sending beacons, which are scheduled regardless of the number of uplink messages on the channel. Thus, the RS-LoRa gateways should also check if the transmission of potential acknowledgements or data messages will interfere with any beacon transmissions. In this way, in RS-LoRa, the probability of delaying the transmission of acknowledgements is higher than in LoRaWAN.

V. CONCLUSIONS

This paper evaluates how the nodes' mobility affects the performance of the LoRaWAN and RS-LoRa based networks. Our simulations in NS-3 show that, under Single Gateway scenarios, the performance of RS-LoRa gets worse when mobile nodes are considered, while the LoRaWAN performance remains the same regardless of the nodes' mobility model. We also expose that under Multiple Gateways scenarios, when considering both static and mobile nodes, LoRaWAN and RS-LoRa have a similar performance in terms of the average PER. Finally, we demonstrate that despite the number of nodes, the average network delay under RS-LoRa is significantly higher than that under LoRaWAN. This is a limitation of RS-LoRa that could be improved and should be taken into account, especially for IoT applications with requirements related to the network delay.

REFERENCES

[1] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[2] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa Scalability: A Simulation Model Based on Interference Measurements," *Sensors*, vol. 17, no. 6, p. 1193, May 2017.

[3] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017.

[4] B. Reynders, Q. Wang, and S. Pollin, "A LoRaWAN module for ns-3: implementation and evaluation," in *Proceedings of the 10th Workshop on ns-3 - WNS3 '18*, Surathkal, India, 2018, pp. 61–68.

[5] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, Mar. 2017.

[6] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. Skarmeta, "Performance Evaluation of LoRa Considering Scenario Conditions," *Sensors*, vol. 18, no. 3, p. 772, Mar. 2018.

[7] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A Survey of LoRaWAN for IoT: From Technology to Application," *Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018.

[8] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018.

[9] Semtech Corporation, "AN1200.22 LoRa Modulation Basics." 2015.

[10] R.C. de Lamare, "Joint iterative power allocation and linear interference suppression algorithms for cooperative DS-CDMA networks," *IET Communications*, 2012.

[11] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J. Prevotet, "Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs standards and Supported Mobility," *IEEE Communications Surveys & Tutorials*, 2018.

[12] N. Sornin, A. Yegin (2017). "LoRaWAN v1.1 Specification". [Online]. Available: https://lora-alliance.org/resource-hub/lorawantm-specification-v11.