LoRaMoto: a LoRa-based communication system for coordinated response in an earthquake aftermath (excerpt)

Roger Pueyo Centelles*, Felix Freitag*, Roc Meseguer*, Leandro Navarro*
* Universitat Politècnica de Catalunya, BarcelonaTech, Barcelona, Spain
† Max van der Stoel Institute, South East European University, North Macedonia

***Abstract*—Response of emergency units after natural disasters, such as earthquakes, has to be coordinated, fast and efficient in order to rescue and care for the victims, keeping all the population –and the units themselves– safe amidst the chaos. Outages in mobile networks, as well as fiber- or copper-based landline and Internet connections are to be expected in such situations, so alternative communication solutions must be considered. To contribute in this duty, we propose a communication system that uses the LoRaWAN architecture to allow citizens to report their status to emergency units and public authorities with simple messages and interaction mechanisms. To analyse the system performance and capabilities we model a district of Coquimbo, a harbour town in Chile, that houses approximately 28.000 people in 7.500 homes, and simulate it with a baseline configuration. We explore several modifications of the system in order to determine its characteristics and limitations, to better understand its scalability and portability to other environments, and to outline the remaining challenges to make the system attain to specific performance guarantees.**

***Index Terms*—LoRaWAN, emergency, disaster response, LoRa, earthquake.**

I. Introduction

The Coquimbo region in Chile, some 400 km north of the capital, has been repeatedly struck by earthquakes and swept by tsunamis in the last decades, and it is expected to continue being affected in the future [1], [2]. In the aftermath of severe seismic events, outages in both mobile and wired communication networks are likely and expected to appear, either because of the infrastructure being damaged, or due to power outages. This poses additional challenges to emergency units, that have to work in a coordinated, fast and efficient manner in order to reduce the impact of the extreme event on civilians, and also prevents the population to communicate with their relatives to learn about their status.

This paper proposes a communication system that provides citizens with a mechanism to report their status to emergency units and public authorities, by means of predefined messages and a straightforward user interaction. The system is based on the LoRaWAN architecture, and uses the LoRa radio technology to transmit information between the users' nodes in their homes and workplaces, and an application hosted in-the-premises. Given our interest to analyse the system performance and capabilities to provide service to scenarios such as a district in the harbour town of Coquimbo, Chile, we created a realistic environment model with the OMNet++ simulator and the FLoRa framework on which to execute diverse network experiments. Our analysis considers a baseline scenario and system configuration, that serves a the departure point to explore its scalability, capability and limitations, and to understand the effect that different parameters pose on the overall performance.

Parameter	Real	Simulation	Deviation
Population	27,794	28.000	**0.741 %**
Homes	7,515	7,500	**0.199 %**
Populated area (km^2)	3.51	3.5	**0.285 %**
Homes density (per km^2)	2,137	2,143	**0.280 %**

Table I
Deviation between real data and simplified model

II. Methodology

A. Environment modelling

In order to simulate our system in a realistic environment we model a representative part of Coquimbo, a harbour town with a population of 240.000 inhabitants. In particular, we consider the populated area of the Coquimbo Peninsula. Based on the data provided by the Chilean National Statistics Institute [1], last updated in 2017 [2], we observe that the area under consideration comprises two census districts [3]: DC-1 and DC-2. These two districts account for a total of 27,794 people living in 7,515 homes, which are distributed in an area of 5.148 km^2.

We observe, however, that the population of DC-1 and DC-2 is concentrated in a smaller area of 3.51 km^2 while the remaining space mostly holds no buildings. We estimate this area from the data available for the smaller neighbourhood units [4] inside DC-1 and DC-2. In particular, we consider UV001, UV002, UV003, UV004, UV005, UV024, UV025, UV033 and UV034, which cover altogether almost completely the populated area of DC-1 and DC-2.

In order to simplify the figures and numbers for the simulation, we approximate the area under consideration by a rectangle of 1.4 km $\times$ 2.5 km, and we slightly reduce the number of homes to 7,500. This leads to a very small deviation in the number of homes taken into account and their spatial

[1] Chilean National Statistics Institute (INE): https://www.ine.cl/
[2] Chile census 2017: https://www.censo2017.cl/
[3] In Spanish, *distritos censales*.
[4] In Spanish, *unidades vecinales*.

Figure 1. The rectangular area of 1.4 km × 2.5 km layered on top of the Coquimbo Peninsula satellite map, with DC-1 and DC-2 also depicted.

density, as detailed in Table I. We also consider that homes are uniformly distributed all over the area and they all have the same elevation. Figure 1 shows census districts DC-1 and DC-2 layered over the satellite image of the Coquimbo Peninsula, as well as the size of the area considered for the simulation for comparison purposes. Note that, for example, the populated areas of DC-2 left outside the white rectangular area could very well fit in the not populated areas at the top left corner DC-1, inside the white rectangle. Eventual real deployments, however, should take more accurately the population distribution into account, with a more precise model.

B. Simulation framework

We simulate the system under consideration with OMNeT++ [5], an extensible, modular, component-based C++ simulation library and framework, in combination with FLoRa [6], a simulation framework for carrying out end-to-end simulations for LoRa networks. OMNeT++ is a well-known discrete event simulator framework used by a lively academic community.

The FLoRa framework provides a complete implementation of the LoRaWAN architecture, [3] (covering all the components like end nodes, gateways, network servers, etc.) and an accurate model of the LoRa radio physical layer (including collisions, capture effect, log-distance path loss model with shadowing, urban and sub-urban environments, etc.), statistics for the transmissions' energy consumption, etc. derived from previous experimental findings. [4].

[5] OMNeT++ Discrete Event Simulator: https://omnetpp.org
[6] FLoRa - A framework for LoRa simulations: https://flora.aalto.fi/

Figure 2. Screen capture of the OMNeT++ running the FLoRa framework, layered over the Coquimbo Peninsula OpenStreetMap imagery, showing the user nodes and gateways, and the remaining components on top.

C. Simulation scope

It is of our interest in this document is to evaluate the envisioned system's performance when handling a massive user interaction in the aftermath of an earthquake. The experiments here discussed specifically focus on this period and leave apart other aspects such as the bootstrapping of the gateways and the user nodes, etc.

The simulations we execute begin with a short setup process, during which all the devices (end nodes, gateways, network servers, etc.) perform their initialisation routines. This setup is followed by an idle period during which no packets are transmitted over the air, which would correspond to the normal system operation before an earthquake. This procedure is common to all the simulations of this work. Figure 3 shows the schedule of the events and actions occurring at one of the user nodes. We consider that, at time $T_0 = 3600$ s, an earthquake takes place, and the previous idle period finishes. From that moment on, users proceed to interact with their nodes. The interaction will happen within a certain *user reaction time*, T_{UR}. We model T_{UR} as a continuous uniform variable between 0 and 120 s. The user interaction with the node immediately triggers the transmission of the first packet, at $t = T_{Tx1} = T_0 + T_{UR}$.

After the first transmission (regardless of it being successful or not), a user node will wait for some seconds before proceeding with a second transmission, scheduled at $t = T_{Tx2} = T_0 + T_{UR} + T_{W1}$, and later, also a third one, at $t = T_{Tx3} = T_0 + T_{UR} + T_{W1} + T_{W2}$. Waiting times between

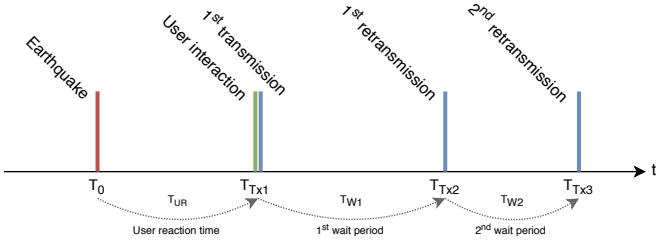

Figure 3. Timing of a user node activity in the aftermath of an earthquake.

transmissions T_{W1} and T_{W2} are modelled as continuous uniform variables between 0 and 300 s. Therefore, user nodes' transmissions take place during a 12 min period starting at the earthquake event: $t \in (T_0, T_0 + 720\,\text{s}) = (3600\,\text{s}, 4320\,\text{s})$. Since the system is meant to help emergency units provide a fast response in the aftermath of an earthquake, all the simulations terminate 1 h after the triggering event, regardless of any pending transmission, etc. These timing and operations are common to all the simulations performed, except where otherwise noted. Since some aspects of the simulations depend on random numbers (e.g. the position of the gateways on the map), there is a certain probability for components not being evenly distributed all over the are under study (for example, when only a few gateways are simulated). To overcome this effect, the presented results correspond to the average of executing each of the experiments 10 times, with different seeds for the random numbers generator. [7]

III. Results

This section discusses the results of simulating the system when different dimensions and settings are modified. We specifically focus on two performance parameters: the average Packet Delivery Ratio (PDR) of a simulation run, and the ratio of nodes that successfully send a given number of packets to the application (≥ 1 packets, exactly 1 or 2 packets, etc.). We first define a baseline scenario, by setting a series of parameters to fixed values and simulate it to obtain the performance characteristics of the system. Afterwards, we modify specific parameters of the system and the model one by one, simulate the resulting scenarios and compare them with the baseline to obtain a better understanding of their influence in the system performance. In particular, we investigate the scalability of the system (by modifying the number of user nodes and the number of gateways) and the characterisation of how different parameters (e.g., the number of retransmissions allowed per node, specific LoRa Chirp Spread Spectrum (CSS) modulation settings) to try to understand which configuration can better suit a given scenario.

A. Baseline scenario

The baseline scenario considers a realistic approach to the environment modelled in Section II-A, in which 7,500 user nodes are deployed over a rectangular area of 1.4 km $\times$ 2.5 km, following a uniform random distribution. There are 10 gateways providing coverage to the area, which are distributed using the same uniform random distribution. Other authors have concluded that areas similar to ours –and bigger– can be correctly serviced by fewer gateways [5]. However, the particularity of our scenario, where transmissions by thousands of nodes are to occur in a short period of time, suggests that a higher gateways density than in other environments should be considered beforehand. Therefore, the number of 10 gateways is taken as a departure value, to be later validated –or not– in Section III-C

Table II summarises the main results of the baseline simulation. Based on these numbers, the system does not provide a working communication channel after a disaster for about two thirds of the nodes, as only 35.77 % of them are able to successfully deliver at least one message to the application. This is a direct consequence of the low PDR, that barely reaches 25 % (averaged for all nodes).

Parameter	Value
Packets sent	22,500
Packets received (unique, app.)	5610
PDR (avg.)	**24.94**%
Nodes with 0 succ. tx.	4809 (**64.12 %**)
Nodes with 1 succ. tx.	835 (11.14%)
Nodes with 2 succ. tx.	792 (10.56%)
Nodes with 3 succ. tx.	1064 (14.19%)
Nodes with ≥ 1 succ. tx.	2691 (**35.87 %**)
Nodes with ≥ 2 succ. tx.	1856 (24.74%)

Table II
Baseline scenario results summary.

It is worth mentioning that 11.14 % of the nodes can only transmit 1 single packets successfully, while 14.19 % are able to transmit 3 packets. This means that the unreliability of the system does not equally affect all of them and, while most (64.12 %) nodes do not achieve to transmit a single packet correctly, a minority make it 3 times. Since a node sends the same information in each packet, the system globally makes a very inefficient usage of the radio spectrum: the time-on-air being occupied by redundant transmissions could be left free for other nodes transmitting, hence avoiding packet collisions to a certain extent. Therefore, unnecessary retransmissions from certain nodes should not be triggered, leaving room for other nodes. This feature should be implemented in the application component, and is one of the very first improvements it should get to improve performance.

B. Number of user nodes

The number of user nodes in the system is the most important aspect of the scalability figure. According to the environment modelling discussed in Section II-A, a ratio of 1 user node per home would give a total number of 7,500 nodes. However, other scenarios in the same region of interest could be more –or less– densely populated, most probably leading to different performance figures. In this section we investigate on the system scalability by simulating it with different numbers of user nodes, while keeping the dimensions (i.e., the geographical area) and the rest of parameters equal to the baseline scenario (number of gateways, LoRa modulation settings, etc.). With this, we want to understand how the architecture scales with the number of nodes and its limits, to

[7] OMNeT++ Simulation Library: cRNG Class reference: https://doc.omnetpp.org/omnetpp/api//classomnetpp_1_1cRNG.html

be able in the future to apply different optimisation strategies that improve specific performance aspects.

Figures 4 and 5 plots the average PDR, for different number of nodes, as well as the ratio of nodes that are able to successfully deliver different number of packets to the application. The graphs outline three main behaviours regarding the system's scalability. First, to the left half, between 100 and 750 user nodes, the PDR stays approximately the same –around 55 %–; most of the nodes are able to successfully transmit all three packets to the application, while some others (between 5 and 10 %) can do it twice, and just a few (less than 5 %) can make it only once. This means that the gateways provide very good coverage to about 60 % of the nodes, while the rest of them are not covered at all. Second, in the middle-right of the graphs, when the system is simulated between 750 and 10,000 user nodes, PDR decays to a half per decade; beyond the thousand of nodes, the transmissions start generating too many collisions, which make the performance figures drop: those nose that were before able to deliver the three packets successfully now they can only do it twice or once. Third, past the 10,000 user nodes, the average PDR figure drops more steeply, below 15 %. When 24,000 user nodes are simulated, which is just over three times the nodes in baseline scenario, not even 25 % of the nodes achieve a single successful transmission.

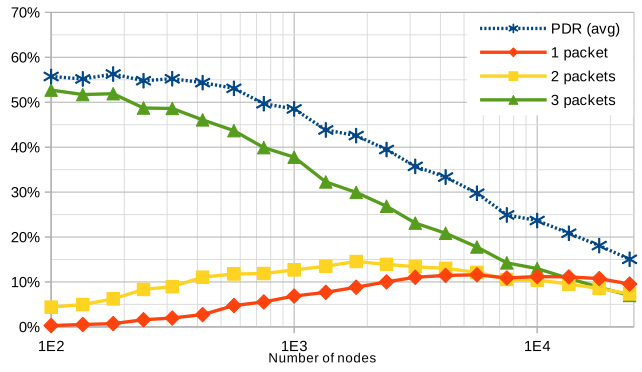

Figure 4. PDR and ratio of user nodes successfully communicating, in function of the number of user nodes (percentage of user nodes from which 1, 2 and 3 packets have been received)

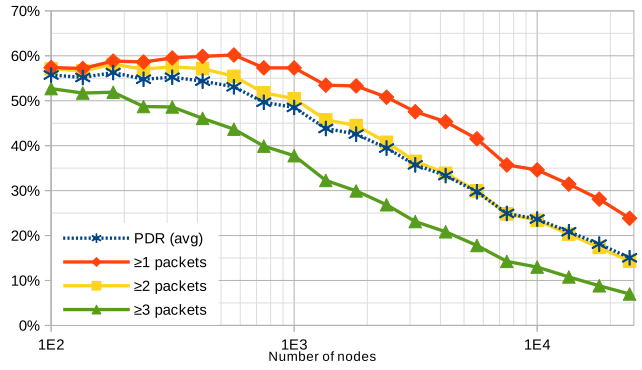

Figure 5. PDR and ratio of user nodes successfully communicating, in function of the number of user nodes (percentage of user nodes from which ≥ 1, ≥ 2 and ≥ 3 packets have been received).

The trends in Figures 4 and 5 suggest that 10 gateways are not sufficient to cover the whole area under study –at least if they are placed randomly–, since more than 40 % of the nodes are unable to transmit a single successful packet to the application way before collisions start to saturate the system. Both more gateways and a careful deployment would be required in a real deployment to properly cover the whole area. According to the figures, the system scales correctly for up to 1,000 user nodes. This turning point could possibly be pushed to a larger value by placing more gateways: more densely placed receivers should improve the chances to properly decode more packets (e.g., thanks to LoRa's capture effect [6]), improving the system's scalability.

C. Number of gateways

In the LoRaWAN architecture, all the user nodes must send their messages to the application server through one –or more than one– of the gateways available in the system. *A priori*, the intuition says that increasing the number of gateways in an area should have an impact in the overall PDR, as that would increase the chances for one of them to receive a message with a Signal-to-Noise Ratio (SNR) good enough to successfully demodulate it and relay it. In this section we modify the baseline scenario and analyse the impact of the gateways density by simulating the system with a wide range of number of gateways (starting from 1, up to 1000). By comparing the results with the baseline scenario, we want to understand how the gateway density affects overall performance, and to determine what is a reasonable required minimum amount to improve the baseline scenario.

Figures 6 and 7 plots the average PDR and the ratio of user nodes that are able to successfully communicate with the application, in function of the number of gateways deployed to receive their packets. This number ranges from a single gateway to up 1000. The graph shows, as it was predicted, that the higher the number of gateways, the better PDR and hence higher the number of user nodes transmitting at least one packet to the application. However, it is worth noting that while a PDR of 0.5 is reached with 32 or more gateways, a simulation with 1000 gateways (not in the graph) provides a PDR of 0.9487 (this is, below 95 % of successful packet delivery).

The trend in Figures 6 and 7 suggests that increasing the number of gateways in the deployment (for example, beyond the 10 gateways of the baseline scenario, up to a few tens) has a positive impact on the PDR (and, therefore, also on the ratio of nodes successfully communicating to the application). However, the effectiveness of this strategy is limited by two factors: (i) as more gateways are added (hundreds, even up to a thousand), the PDR only approaches to 95 % asymptotically; (ii) the economic cost of a gateway is an order of magnitude higher than that of a user node.

It is difficult to determine an optimal number of gateways (the economic factor may play an important role in this decision), but certain statistical criteria could be considered. For example, in the baseline scenario with 10 gateways, an average PDR of 25.41 % is achieved. According to the simulations, in order to reach a PDR of 50 % (this is, half of

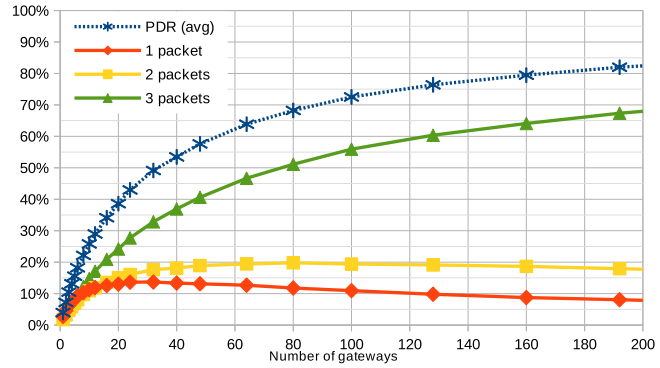

Figure 6. PDR and ratio of user nodes successfully communicating, in function of the number of gateways (number of user nodes from which 1, 2 and 3 packets have been received).

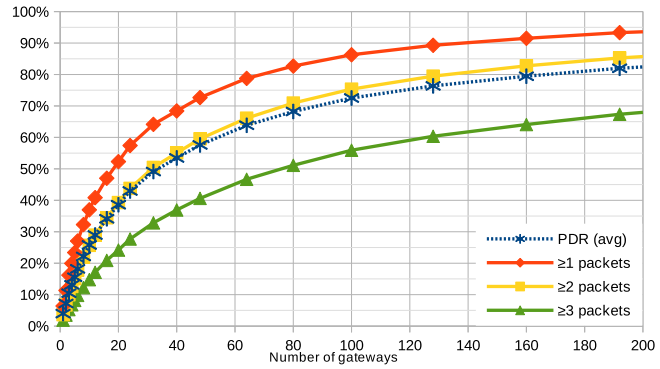

Figure 7. PDR and ratio of user nodes successfully communicating, in function of the number of gateways (number of user nodes from which ≥ 1, ≥ 2 and ≥ 3 packets have been received).

the packets are successfully processed, the other half are not), 32 gateways would be needed (PDR would be 50.41 %). This number would provide a reasonable ratio between the quantity of user nodes and gateways, but still roughly one third (66.2 %) of the nodes would not be able to successfully transmit any message. To improve this figure, other strategies to modify the system are in the following.

D. Number of retransmissions

As detailed in Section III-A, after the user interaction, an user node transmits its message three times. The first transmission begins right after the interaction and the two retransmissions occur after random waiting periods (i.e., T_{W1} and T_{W2}, as depicted in Figure 3. Retransmissions have a positive impact in the system, as outlined in the numeric results in Table II from the baseline scenario: while user nodes have average PDR of 25.41 %, almost 48 % of them are able to transmit at least one packet correctly. However, the way the system is designed, all retransmissions will occur regardless if a node has been able to successfully communicate with the application. This implementation allows keeping things simple (e.g., there is no need for down-link messages to be sent to the user nodes), but comes at the price of an inefficient usage of the spectrum: unnecessary retransmissions will occupy the spectrum and create further collisions without reporting

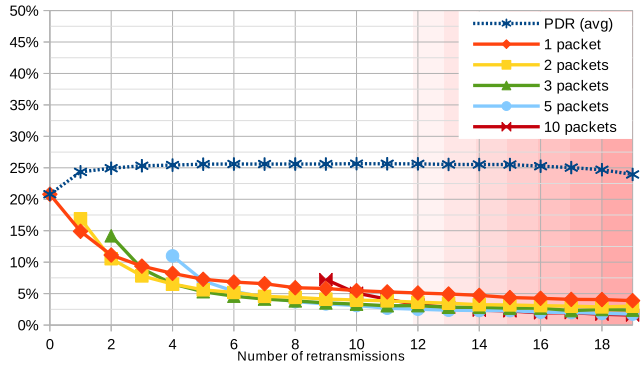

Figure 8. PDR and ratio of user nodes successfully communicating, in function of the number of allowed retransmissions per node (number of user nodes from which 1, 2 and 3 packets have been received).

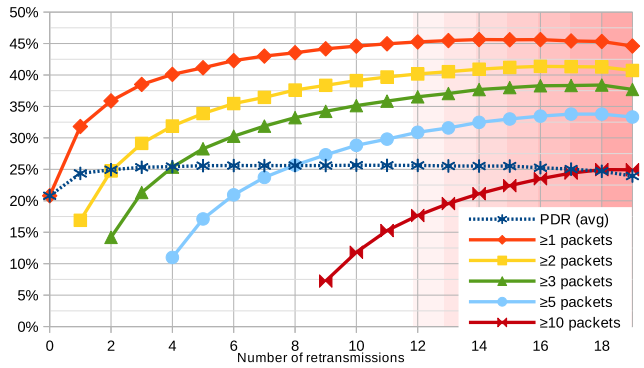

Figure 9. PDR and ratio of user nodes successfully communicating, in function of the number of allowed retransmissions per node (number of user nodes from which ≥ 1, ≥ 2 and ≥ 3 packets have been received).

any new information. In this section we modify number of retransmissions user nodes perform, in order to investigate which is the trend as more –or less– of them are allowed. We simulate the system with all the nodes retransmitting their packets from 0 to 19 times.

Figures 8 and 9 plot the ratio of user nodes that are able to successfully communicate with the application, in function of the number of retransmissions allowed for each node. The graph shows that, beyond one retransmissions, the average PDR does not significantly improve and stays constant just above 25 %. This behaviour is explained by the fact that all the nodes perform their initial transmission at a random moment during a 120 s, but the next ones are distributed over a longer period of 300 s, reducing the collision probability. Regarding the number the packets successfully transmitted per node, it is interesting to see that most of them belong to a minority of nodes that always succeed: for example, when 4 retransmissions are allowed, 12 % of the nodes succeed in delivering *all* of their packets, while fewer achieve it once, twice or three times. Similarly to the baseline, this means that these nodes are misusing the available time-on-air, and allowing more retransmissions does not help balancing this usage.

By design, the system is aimed at providing all the results in the lapse of one hour after the earthquake. After that time, the simulations are finished regardless of any pending packet trans-

mission. Therefore, if retransmissions are scheduled at random intervals between 0 and 300 s, since $3600\text{s}/max[0,300]\text{s} = 12$, it is then possible that, when more than 12 retransmissions are allowed, the simulation is terminated while some nodes may still have not sent all their packets. This can be seen in Figure 9, where the trends suggest a decrease in the performance.

E. Spreading Factor (SF)

The SF is a key parameter of the LoRa technology, since it determines the CSS modulation density. Higher SFs mean higher range and better sensitivity, at the expense of a lower data rate. In the baseline scenario (Section III-A), user nodes are configured to randomly use SFs 7 to 12 (these are common values in LoRaWAN deployments), following a uniform distribution. This way, an important property of the CSS is exploited: concurrent transmissions using different SFs can coexist and be successfully demodulated by a gateway. Therefore, the SF has a significant impact not only on the transmissions of a single node, but also on the global system. For example, a node using a high SF can extend its communication range and reach more distant gateways, hence increasing the chances for its packets to be successfully received. However, a longer range can increase the collision probability with other nodes' transmissions using the same SF, causing a negative impact on the overall system. Furthermore, since higher SFs require longer air time, collision probability is further increased, which also has a negative impact from the system's global perspective.

We modify the SFs the user nodes use to transmit data to the gateways, in order to investigate which value (or which combination of them) is more suitable for the given system. We first simulate the system with all the nodes using the same SF (from SF7 to SF12). Then, we define all the possible SF ranges (SFs 7 to 12, 7 to 11, 7 to 10, etc.) and we simulate the system with nodes randomly choosing the SF from the given range.

Figure 10 plots, to the left, the average PDR and the ratio of user nodes that are able to successfully communicate with the application ≥ 1, $\geq$ or 3 times, when all of them use the same SF. According to the data in Figure 10, smaller SFs 7 to 9 (which shorten the transmission range and the time-on-air, and give faster data rates) provide a better average PDR than the bigger ones, as shorter time-on-air of smaller SFs leads to smaller collision probabilities. As the SF is increased, the PDR gradually decreases (from almost 20 % with SF7 to barely 2.5 % with SF12). Slower transmissions with longer ranges and time-on-air increase the collisions probability and, on average, make the system have worse average PDR. It is worth noting that the best PDR is achieved with SF7, but employing SF8 allows more nodes to communicate with the application at least once –which is desirable–. Here, the shorter transmission range and time-on-air reduces the collision probability, but most likely makes some of the nodes unable to reach any of the gateways.

To the right, Figure 10 plots the simulations where user nodes choose from a set of different SFs values. The widest range, SF7 to 12, corresponds to the baseline scenario in Section III-A. The goal here is to asses the positive impact of using more than one SFs simultaneously, benefiting from the

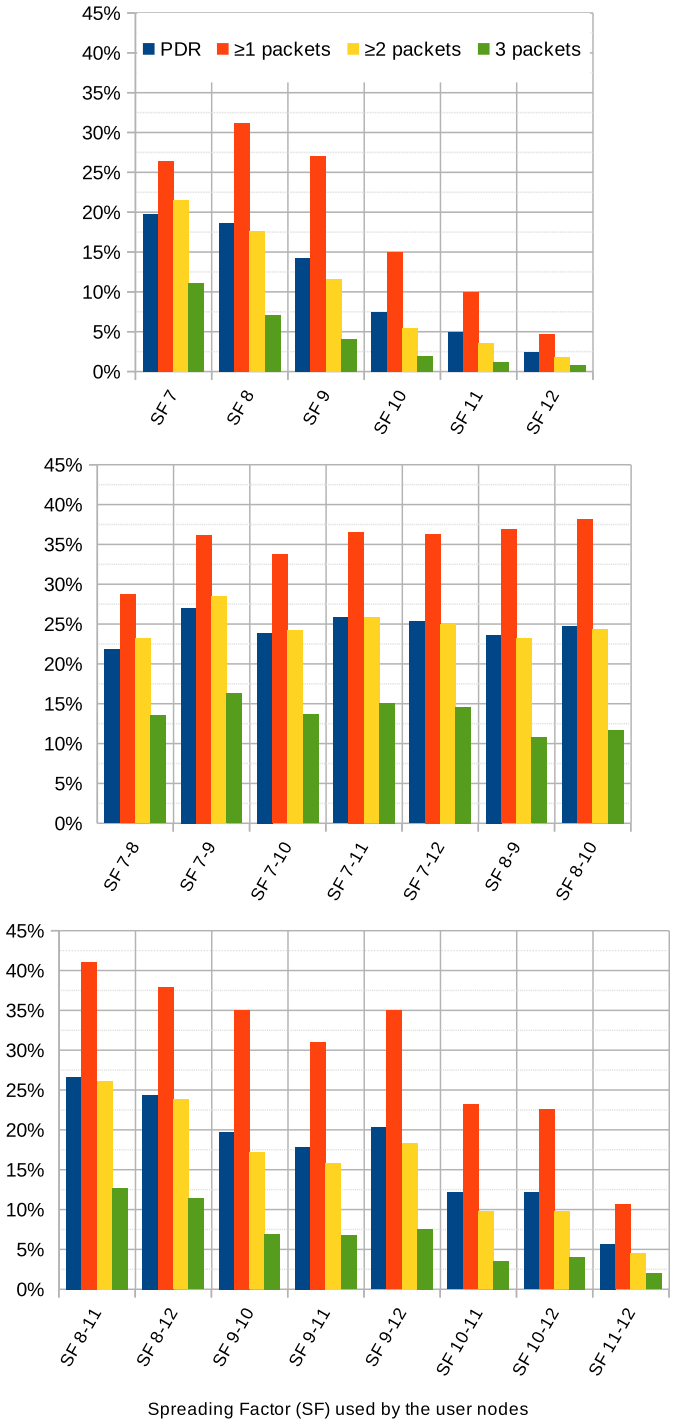

Figure 10. PDR and ratio of user nodes successfully transmitting ≥ 1, ≥ 2 and 3 packets, in function of the chosen SF.

fact that a gateway can successfully receive and demodulate two or more packets from different nodes concurrently if they all employ different SFs. Just above, we have determined that SFs 7 to 9 provide the best overall results, while SFs 11 and 12 perform significantly worse. Here, the results show that the best figures correspond to the combinations using the smaller SFs *and* a wide range of them. Regarding the PDR, for instance, the simulation employing SFs 7 to 9 provides the best performance (27 %). However, the optimal SFs selection in terms of the number of nodes able to communicate at least once corresponds to SFs 8 to 11. In this case, 41 % of the nodes reach the application at least once, compared to the 38.1 % of the baseline scenario.

ACKNOWLEDGMENT

This work was supported by the Spanish Government under contract TIN2016-77836-C2-2-R, and by the Generalitat de Catalunya as Consolidated Research Group 2017-SGR-990.

REFERENCES

[1] M. M. Switt, , and A. B. Oyarce, "Investigación identifica área de potencial terremoto sobre 8 en región de valparaíso," December 2018. [Online]. Available: http://www.dgeo.udec.cl/blog/2018/12/28/investigacion/identifica-area-de-potencial-terremoto-sobre-8-en-region-de-valparaiso/

[2] M. Moreno, S. Li, D. Melnick, J. Bedford, J. Baez, M. Motagh, S. Metzger, S. Vajedian, C. Sippl, B. Gutknecht *et al.*, "Chilean megathrust earthquake recurrence linked to frictional contrast at depth," *Nature Geoscience*, vol. 11, no. 4, p. 285, 2018.

[3] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense iot deployments," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–9.

[4] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of lpwans: range evaluation and channel attenuation model for lora technology," in *2015 14th International Conference on ITS Telecommunications (ITST)*, Dec 2015, pp. 55–59.

[5] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, April 2017.

[6] U. Noreen, L. Clavier, and A. Bounceur, "Lora-like css-based phy layer, capture effect and serial interference cancellation," in *European Wireless 2018; 24th European Wireless Conference*, May 2018, pp. 1–6.