

LoRaMoto: A Communication System to Provide Safety Awareness Among Civilians After an Earthquake

Roger Pueyo Centelles^a, Roc Meseguer^a, Felix Freitag^a, Leandro Navarro^a, Sergio F. Ochoa^b, Rodrigo M. Santos^c

^a*Department of Computer Architecture, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain*

^b*Department of Computer Science, Universidad de Chile, Santiago 8370456, Chile*

^c*Department of Electrical and Computers, Universidad Nacional del Sur, CONICET, Bahía Blanca 8000, Argentina*

Abstract

Civilians in areas hit by earthquakes become in particular concerned about the safety of family and friends. However, outages in communications networks following these events limit their capacity to get information. So, people drive their cars to check on their loved ones, risking their own lives and hindering the activities of first response organizations. This article proposes LoRaMoto, a communications system that allows civilians to exchange information about their status with others without hampering the first responders and minimizing their exposure to hazards. Furthermore, the information exchanged can be aggregated to support the operations of emergency management. LoRaMoto extends the LoRaWAN architecture and implements a packet-forwarding mechanism between the end nodes of the system. This helps overcome potential infrastructure blackouts after an earthquake, so it provides safety awareness information when it is most needed. We evaluated the system performance and capacity of this system by simulating its use in a realistic environment, testing a baseline configuration, and exploring several modifications to better understand its scalability and its portability to other emergency environments. The simulation results show how the LoRaMoto system can meet the communication needs of civilians, and also keep first response and emergency management organizations informed.

Keywords: Communications system, emergency response system, awareness dissemination, distributed computing, first response for civilians.

1. Introduction

Earthquakes are unpredictable natural hazards that, when they occur in populated areas, raise two big concerns for those who suffer them: first, avoiding injuries and getting to a safe location and second, learning about the safety conditions of their family and friends. Typically, earthquake safety procedures (e.g., self-protection protocols and evacuation routes) address only the safety and security issues. Once they themselves are safe, and while aftershocks may still be occurring, people must deal with the second matter, most likely under a blackout of utilities and regular communications services, such as mobile or landline telephony and Internet-based messaging. Such infrastructure is usually unavailable because of physical damage, traffic overload, or lack of electricity [1]. To avoid the risk of fire, the electricity supply is usually shut down immediately after the shock, and it is activated only after a case-by-case damage inspection, which in the event of an earthquake can take days. All these reasons contribute to the typical communications blackout that comes after an earthquake [1, 2, 3].

Figure 1 depicts a simplified and conceptual view of such a communications scenario, where most communications brokers in the affected area (e.g., cellular antennas) collapsed or were shut down. Therefore, civilians and first response organizations in the area need alternatives to interact and coordinate with each other. Typically, first response organizations (firefighters, police, emergency medical services, government agencies) already use VHF/UHF radio systems for their communications [4], but such an infrastructure is usually not available to support interactions among civilians or between them and the first responders.

In the absence of the usual means for contact in such situations, many people drive their private vehicles to reach the home of their loved ones to learn about their condition. These actions not only risk the personal integrity of the travelers, but they also hamper the work of incident response teams. Since public transportation systems are

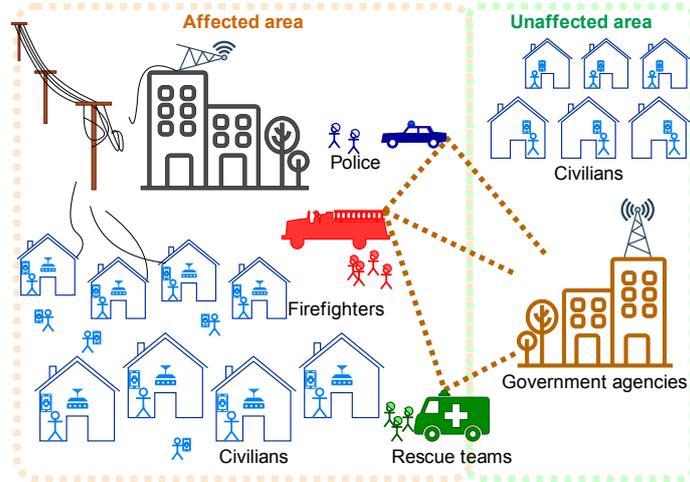


Figure 1: Typical communications scenario after an earthquake.

stopped for safety reasons, further limiting the capability of civilians to reach their family and friends, those who have no means of transportation stay at home, sheltering in place, as that is usually the meeting point assumed by families—and the approach recommended by most earthquake safety procedures, except for those areas under risk of a tsunami. So, people typically wait for others to contact them, with increasing anxiety and concern that might damage their emotional health.

Most studies of communications during and immediately after an earthquake have focused on incidence response teams or civilians that are not located in the affected area [5, 6]. However, some recent research efforts have examined whether civilians could support particular activities in the disaster response processes (e.g., damage evaluation, communicating requests for help [7, 8], and identifying the locations of people in large urban populations and steering them to safety[9]). Regardless of this progress, communication with civilians, and their participation as human sensors and information consumers, are issues that must be addressed. comprehensive approach, where human beings and their devices become active elements in such a process (i.e., it should evolve towards an Internet of People (IoP) interaction paradigm [10]).

To foster the participation of civilians in first response activities, this article proposes a communications system named LoRaMoto, aimed at helping civilians exchange information about their safety with their family and friends and also with first responders in the aftermath of an earthquake. Moreover, since such information is geolocalized and coded, it can be aggregated and then used to support decision-making by first response teams or government authorities. For instance, such information could be used to create a heat map of the affected area or to identify places where assistance is more required. In this sense, allowing the participation of civilians as human sensors is crucial for providing information almost in real time to support decision-making at higher levels.

The LoRaMoto system is based on and extends the LoRaWAN architecture, using LoRa radio technology to transmit information between the end nodes (i.e., users at their homes and workplaces) and an application hosted at such places as an emergency coordination center. This avoids the need for a cloud-based deployment. The interaction between the users and the system is performed with a small LoRa-based device installed at the user’s home, using a software application on a personal communications device (e.g., a smartphone). The software application uses the LoRa-based device as the interface to exchange messages with the LoRaMoto system. Regarding the underlying technology, we opted for LoRa because it allows the implementation of dedicated and single-purpose solutions based on the Internet of Things (IoT). This allows control of access to the network and message traffic, and it does not consume much energy. The last three features help the proposed infrastructure extend the amount of time such a service will be operational after an earthquake, compared to the typical service availability of Internet Service Provider (ISP) networks.

This proposal builds upon the authors’ previous work in [11]. In particular, this extended version includes a congestion control protocol, based on Acknowledgment (ACK) messages, that includes a feedback mechanism to

enhance end-users' interaction. Moreover, it extends previous work beyond the LoRaWAN framework to enable direct communication between end nodes. This capability includes a packet-forwarding mechanism that improves the resiliency of the system in the event of network infrastructure failures. The performance and capability of the system were evaluated through simulation. The results show that the proposed approach improves on the earlier one. It can also support the stated communication among citizens and keep first response and emergency management organizations informed.

The remainder of this article is structured as follows. The next section explores the most significant research in the domain. Section 3 describes the technical background of the LoRa radio technology and the LoRaWAN architecture. Section 4 describes the communications system and its underlying technology and architecture. Section 5 explains the modeling of the scenario in which the proposed system was evaluated. Section 6 explores different aspects and capabilities of the communications system, presenting and discussing the results of the simulation. Section 7 presents a discussion that positions this paper's contributions in the context of related work. Section 8 highlights the conclusions and provides suggestions for future work.

2. Related Work

The first hours after an earthquake are a critical period during which civilians, first responders, and government agencies need to know about the safety condition of others to decide their next actions. Throughout this period, the decision-making process is often conducted in a distributed and chaotic way, and it is also under the pressure of time—since this is usually the scarcest resource [12, 13]. In such a scenario, relying on communications and timely information is essential for making proper decisions. This section addresses the related work on these topics.

2.1. *communications Support In Emergency Responses*

Communications systems, as instruments to facilitate the exchange information and the coordination of participants in disaster relief efforts, have been recognized as the stumbling block for making emergency response activities more efficient and effective [14, 13]. This is still an open research issue, In particular when mobile digital communications are required inside an affected area [7]. Most research efforts consider only the participation of first response organizations and government agencies, but not civilians.

Typically, first responders require communications systems that are easy to deploy, support mobility, provide reliable communication, and cover a large area. These are important challenges that researchers intend to address in the design of new communications systems. In most cases, the proposals consider WiFi-based implementations, like Mobile Ad-hoc Networks (MANETs) and Opportunistic Networks (OppNets) [15, 16, 17], since they have several advantages. However, they also have limitations, mainly in terms of communications reliability, threshold, and the infrastructure and equipment needed.

In contrast, communications among government agencies (typically, units located outside the affected area) can be infrastructure-based, using networks prepared for such situations. The link between government agencies and first responders is usually implemented through gateways that exchange information between the two worlds.

Concerning the communications support for civilians, as mentioned before, most or all of the communications infrastructure is usually shut down, including mobile and landline telephony and Internet-based messaging. This encourages the physical transportation of people, putting their lives at risk and hampering the response procedures.

Several recent studies have analyzed how the LoRa technology and the LoRaWAN architecture could support activities in different application domains, including disaster relief efforts [18]. The features and capability of this technology open new opportunities to allow the participation of civilians in this communications scenario. The next section discusses the main research proposals concerning such technology.

2.2. *Using LoRa Technology To Support communications During Emergency Responses*

In the last few years, several LoRa-based systems have been proposed to support communications in scenarios comparable to first responses after earthquakes. For instance, Sciullo et al. proposed an Emergency Communications System (ECS) [19] that operates over infrastructure-less phone-based networks and guarantees long-range Device-to-Device (D2D) communications thanks to the LoRa technology. This system, named LOCATE, has two main components: a mobile application through which users can convey minimal yet vital emergency-related data, and a

dissemination protocol to spread the emergency requests over multi-hop LoRa links. In [20], the same research group extended the LOCATE system, adding a probabilistic store-and-forward mechanism derived from Delay-Tolerant Networks (DTNs) to it as a dissemination protocol. Although this is a very interesting proposal, it is not suitable for disaster scenarios, because of the typical limitations of scalability of the probabilistic store-and-forward protocols derived from DTNs [21]. LoRaWAN, which is a network protocol that builds upon the LoRa radio technology, also has these scalability limitations [22, 23], and its performance decreases [24] when there is node mobility. Our proposal aims to address these problems of scalability and performance, without node mobility, since the system is focused on civilians that exchange safety information while staying at home. Therefore, the proposed system is closer to an ad hoc network than to a DTN.

Georgiou and Raza analyzed LoRa technology through simulations [23] and found that the standard coverage probability drops exponentially as the number of end-devices grows. This is due to interfering signals using the same spreading sequence. Those researchers stated that this fundamental limiting factor was perhaps more significant for LoRa scalability than, for instance, spectrum limitations. In recent work, Liando et al. reach a similar conclusion in a different experimental environment [22]. To deal with this problem, we propose a message forwarding mechanism where end nodes with better connectivity can help their neighboring nodes to achieve successful message transmission.

Magrin et al. simulated a LoRaWAN network with an ns-3 module to study the performance of a LoRa-based IoT network in a typical urban scenario [25]. The results showed that a LoRaWAN network can scale well, achieving packet transmission success rates above 95 % when a gateway is serving a number of devices in the order of 10^4 . Through experimentation rather than simulation, Liando et al. showed that each LoRaWAN gateway can support up to 6000 nodes with a packet transmission success figure of $> 70\%$ [22]. However, while the dimensions match, the most important difference between those studies and the proposal presented here lies in the high rate of simultaneous messages imposed by the emergency scenario where the delays must be short.

Regarding the applications, it is important to mention that LoRaWAN is especially useful to support applications with asymmetric communication. Apart from a static deployment of sensors, in some cases, sensor networks might employ mobile sensor nodes too. In such a scenario, the impact of node mobility on the performance of LoRaWAN also was studied by Patel and Won [24]. They reported two key findings: (i) LoRaWAN is susceptible to mobility; and (ii) the effect of mobility worsens the performance for end-nodes in cases of bad reception conditions (e.g. when they are indoors or far from a gateway). This impact depends on the message size, the distance between nodes, and the mobility speed. For small messages and low speeds, the performance can be reduced by a 5 %.

Most solutions to this problem propose dynamic or adaptive configurations of LoRaWAN [26]. In our case, the LoRaWAN network is quasi-static, and the messages are small (payload is in the 12 to 20 bytes order), so, node mobility is not an issue. Moreover, the forwarding mechanism works as if it were dynamic. The configuration of the nodes does not change, but the messages are redirected by the nodes with better links.

On the other hand, information about the location of response teams or citizens is a valuable resource during earthquake response activities. In this sense, there are several proposals that use a LoRa network to communicate people's locations or geolocate messages. For instance, Kang et al. proposed a communications technology-agnostic message encoding for geographical location data and timestamp [27]. The geolocated messages can be used by disaster management agencies to diagnose the emergency situation or support decision-making. Provided that LoRa can be used to calculate the position in areas with poor GPS coverage, Lahouli et al. proposed a way to infer firefighters' locations and distances based on IEEE 802.15.4 standard-compliant UWB radios, and to provide updated reports of positions using a communications solution based on LoRa [28]. In the same line, Sciuillo et al. used LoRa for ad hoc message communication and also for trilateration in order to identify users' geolocation [20].

These previous efforts show a clear opportunity to use the LoRa technology to support communications in earthquake response activities and also to implement mechanisms to improve such a service (e.g., forwarding messages [29], scheduling [30, 31], synchronizing transmissions [32] and time [33, 34]). This also includes the chance to involve civilians as both information providers and consumers under a crowdsourcing scheme.

Considering these capabilities and opportunities for improvement, the proposed LoRaMoto system provides a message exchange infrastructure that includes confirmation of message reception and forwarding as a way to allow civilians to report their safety condition and get news about the status of their loved ones.

2.3. Participation Of Civilians In Disaster Relief Efforts

The literature reports several instances where civilians participated in a disaster response process. Most of them were spontaneous actions that people performed to help their family, neighbors, or friends. However, during the last few years, there have been many scientific reports involving civilians who used social media to support the emergency preparedness and recovery phases of the disaster life cycle [35, 36]. Social media information also helps emergency management organizations make decisions during the response phase. However, such information comes mainly from people located outside the affected area, so it is usually inaccurate and incomplete [37]. Instead, information from civilians in the field is required to improve the emergency response process, and the most feasible way to diagnose a certain area is likely to involve civilians in a crowdsourcing process. Supporting these statements, we can mention the study conducted by Murayama and Yamamoto [7] of participants in the response process after the Great East Japan Earthquake in March 2011. They showed the need to count on situation awareness, based on information provided by citizens, to make decisions during the golden hours. Similarly, Spialek and Houston [8] shows the critical role of the community to provide awareness information and clarify rumors during an emergency situation.

Although this information can be used to provide situation awareness to government agencies, Howard and Cambria [38] recommend integrating information about the actors' intentions to create a unified view of the environment. This concept, named intention awareness, can be more accurate than situation awareness in predicting the evolution of scenarios that by nature are dynamic and subject to rapid and unexpected variations, like emergency situations after an earthquake. In this sense, the information provided by civilians in the field should include their current status and intentions.

A few proposals included ways to use information reported from/to civilians in the field. Most of them proposed using mobile ad hoc communications systems (e.g., based on Twitter or ad hoc text messages [39, 17]). However, they have low reliability and short communications range, so they cannot be used to support large-scale communications among civilians in large areas [40]. Therefore, new communication systems are required to address these challenges.

Regardless of the relevance of citizens' participation in the response process, it is not formally included in the regular relief processes or in the communications systems deployed to support those processes. Therefore, LoRaMoto represents a step forward by including civilians in first response activities. In particular, the system allows citizens to exchange safety awareness information with family and friends and eventually neighbors. Because this information is coded and geolocated, it can be aggregated and processed for other purposes, such as diagnosing the status of the civil population in a zone or identifying priority areas for search and rescue.

3. Technical Background

In the aftermath of an earthquake, regular communications infrastructure such as cellular telephony networks, and wired Internet connections and landlines (either fiber or copper-based) have a high probability of suffering service outages. Seismic events can affect infrastructure in diverse ways; fibers and cables can break from excessive tension, base station towers can crumble, and precisely aligned antennas can become misaligned and cease to operate. In addition to hardware breakdowns, the absence of electricity can render most equipment unusable until electricity is restored. Lack of power can result from the destruction of utility infrastructure, but also as a preventive mechanism to avoid risks.

Given these considerations, our emergency communications system for civilians in the aftermath of an earthquake should run in parallel and independently from regular communications services, and it should use wireless technology rather than cables to overcome eventual infrastructure breakdowns. Moreover, it should be able to run on off-grid energy sources, such as Uninterruptible Power Supplies (UPSs) and batteries—at least for some hours, until service is restored. These constraints have a direct impact on the design of the system, in aspects such as its range of communication, operating band, transmission power, type of antennae, and achievable throughput.

Beyond these technical restrictions, we set an additional requirement for the LoRaMoto system that the whole infrastructure must be fully accessible, owned, and maintained by the operator. The rationale behind this design decision is the fact that any infrastructure managed by third parties may not be dependable or usable at all, or service might be downgraded. Because of these limitations in the availability, accessibility, and flexibility of the infrastructure, in addition to the reasoning conducted in Section 2, we consider LoRa radio technology to be the most suitable candidate on which to build our system. The baseline design of our proposal follows the specifications of the LoRaWAN architecture. However, LoRaMoto adds certain features, e.g., ACK message awareness and packet-forwarding between

Parameter	Value
Radio band	169, 433, 868, 915 MHz ^a
Bandwidth	62.5, 125, 250, 500 kHz ^b
Transmission power	14 dBm (EU), 27 dBm (USA)
Spreading Factor (SF)	6 to 12 ^c
Coding rate (CR)	4/5, 4/6, 4/7, 4/8

^a Common frequency allocations for ISM in different regions; LoRa may also be used in licensed bands.

^b Smaller bandwidths (7.8 to 41.7 kHz) are also supported, although rarely used.

^c Certain bandwidth and SF combinations result in transmissions that take too long for the radio bands used, where time-on-air limitations often apply.

Table 1: Configurable parameters in LoRa transmissions

user nodes, which make this system cross the boundaries of the LoRaWAN framework. In the next section, we briefly describe these technologies. Then, in Section 4, we introduce the LoRaMoto system architecture.

3.1. The LoRa Radio Technology

LoRa, which stands for “long range”, is a wireless communications technology owned by Semtech [41] that operates in the sub-gigahertz range of the radio spectrum. It employs Chirp Spread Spectrum (CSS), a proprietary modulation technique resistant to multi-path fading and suitable for noisy environments, and it aims to provide low throughput communication with links of more than 10 km—outdoors, in rural areas— while not consuming much power [42].

Several parameters of the LoRa physical layer can be configured to optimize communications for a given scenario or application: radio band and frequency, channel bandwidth, transmission power, Forward Error Correction (FEC) rate, and Spreading Factor (SF). Different license-exempt Industrial, Scientific, Medical (ISM) bands are commonly used in IoT deployments, depending on the geographic area, and diverse LoRa transceivers are available to operate in any of them.

There are several technical factors and features that make the LoRa technology especially suitable for a system like the one we propose. Table 1 summarizes the most significant ones and lists their possible values. The SF is a particularly interesting one, as it determines a trade-off between *transmission speed* and *communication range*, hence also affecting the figures for time on air and the energy consumption. Furthermore, transmissions with different SFs are orthogonal between them, so concurrent signals can be simultaneously demodulated correctly. Therefore, proper distribution of transmitting nodes in different SFs, depending on variables such as the density of nodes and gateways and distance that must be to covered, have a significant impact in the ratio of delivered packets, as we assess in Section 6.1.3.

3.2. The LoRaWAN Architecture

The LoRaWAN architecture is an open standard promoted by the LoRa Alliance [43]. It adds the Medium Access Control (MAC) layer and specifies a Low Power, Wide Area Network (LPWAN) protocol on top of LoRa. It is designed to provide secure wireless bi-directional communication between the end-nodes (e.g., battery-operated IoT devices that collect data) and the application that processes the data.

LoRaWAN-based systems are built with a star-of-stars topology, where a number of end-nodes upload data wirelessly to one or more gateways in single-hop transmissions. Gateways receive these messages and relay them, by means of an IP connection, to a network server where the IoT application resides. Very often, applications are hosted in the cloud, so gateways actually require a Wide Area Network (WAN) connection to the Internet (via cable, WiFi access point, or 4G/5G modem). In the network server, information is centrally managed and accessed by the specific IoT application components. Given that network topology, LoRa packets in the uplink direction (i.e., nodes to gateways) usually account for the majority of the data transmitted. Downlink communication from the network server to the end-nodes also is possible via the gateways, though some restrictions apply. End-nodes open two short

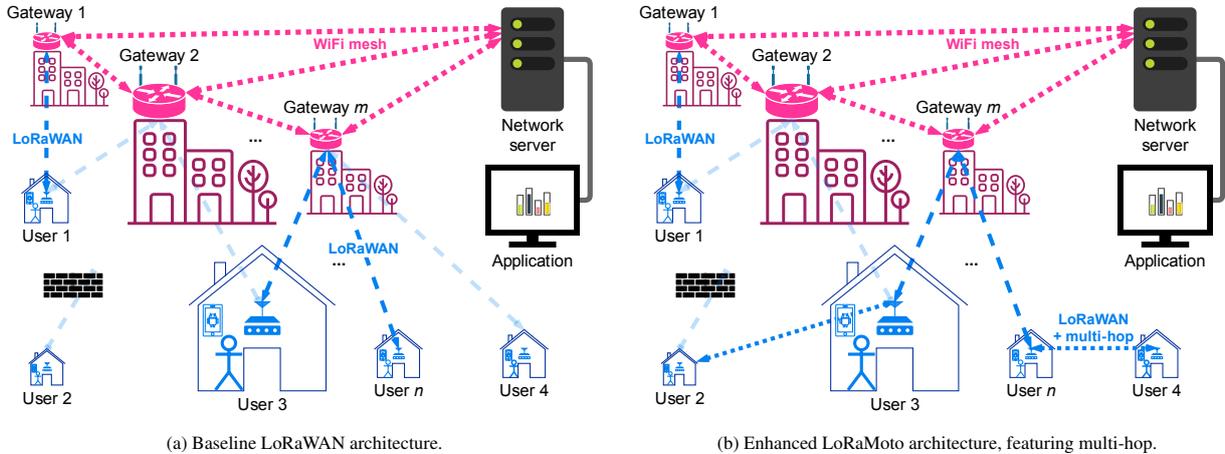


Figure 2: High-level depiction of the baseline LoRaMoto system architecture, matching the LoRaWAN specifications.

reception windows to receive any incoming message only after transmitting uplink data; afterwards, they are usually programmed to enter sleep mode to reduce energy consumption.

4. LoRaMoto System Design

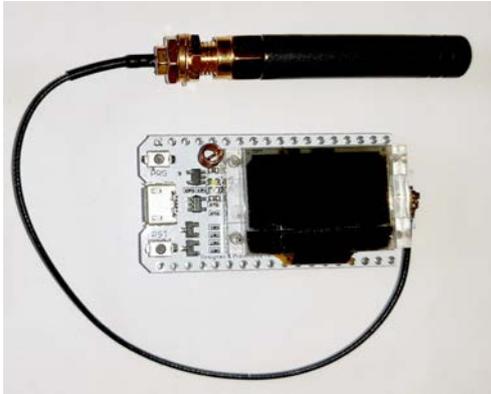
The initial LoRaMoto design was based on the LoRaWAN architecture [11, 43], but it evolved beyond it by leveraging communication between home devices and multi-hop packet transmission. The previous sections discussed the motivations and restrictions that made us opt for the LoRa technology and the LoRaWAN architecture on which to build our proposal. Here we describe the proposed architecture, its features and capabilities. Figure 2 depicts the main components of the LoRaMoto system from a high-level perspective, for both the baseline architecture that follows the LoRaWAN specifications and the one enhanced with multi-hop.

4.1. Baseline LoRaWAN Architecture

At the bottom of Figure 2a, there are several people located indoors, inside their homes fitted with a LoRaMoto end node that allows them to exchange messages with family and friends. To avoid confusion with other devices (i.e., an end-user device such as a smartphone), LoRaMoto end nodes will be referred to as *home devices* for the remainder of this paper. This equipment is to be placed at homes and workplaces, to allow both transmission and reception of messages, operating like a class A device (as defined by the LoRaWAN architecture).

The gateways in the mid-upper part of Figure 2a relay the messages from the home devices to the network server. Since a transmission from a given node may be received by more than one gateway, they generate a so-called star-of-stars network topology, as defined in the LoRaWAN architecture. Very often, gateways use an Internet connection at their location to relay packets between the nodes and the network server, for example via 3G or 4G. In the LoRaMoto system, however, since cellular networks are expected to be unavailable, gateways communicate with each other by means of attached IEEE 802.11-based wireless mesh network in order to reach the network server. Wireless mesh networks that meet LoRaMoto’s requirements in terms of topology, deployment area, and bandwidth, have already been deployed, and they are a mature technology [44]. That part of the system is independent of the LoRaWAN architecture; therefore, it is not addressed in the present work. To simplify the system design, we assume that the gateways have a good and direct connection to the network server. The gateway infrastructure, as well as the wireless routers attached to them, can be powered by batteries or UPSs, to ensure they can operate in the aftermath of an earthquake, even if a blackout occurs.

The network server, at the top right of Figure 2a, receives all the messages from the nodes that the gateways successfully demodulate, and it processes them (e.g., to remove duplicates). Typically, this component runs in the cloud, but it can also be executed on the premises, as in this scenario. The network server hosts the actual application



(a) Heltec WiFi LoRa 32 V2 development board.



(b) Prototype of a home device in a 3D-printed case.

Figure 3: Home devices are built around the WiFi-capable, low-power Espressif Systems ESP32 System on a Chip (SoC), connected to a LoRa transceiver. On the left, a development board (LoRa transceiver is soldered under the LCD display). On the right, a prototype home device inside a 3D-printed enclosure.

that manages the messages received from the nodes, processes data, and takes actions (e.g., triggering alarms or sending downlink messages). The application would provide meaningful information to emergency response teams and decision-makers to help them assist the population (e.g., charts or heat maps).

In the baseline system proposed and simulated, communication flows only in the uplink direction, from the user nodes to the central application. However, the LoRaWAN architecture allows for downlink communication from the application to the user nodes, providing richer bidirectional interaction. In this study, we leverage this resource to introduce the possibility of providing feedback to end-users (e.g., regarding whether their messages could be received by the application or not).

4.2. Packet-forwarding Beyond the LoRaWAN Architecture

The LoRaWAN architecture involves a star-of-stars topology that does not consider direct communication between nodes. This imposes system design restrictions, in particular that all communication from the user nodes to the network server must go through the gateways, even if the underlying LoRa radio technically allows it. This design is suitable for deployments in which battery-operated devices spend most of the time in a low-power sleep mode, only to wake up, perform a task (e.g., reading a sensor), transmit the collected data, and reenter the sleep mode. However, in an emergency scenario like the one we consider, where any part of the network infrastructure may unpredictably cease to function, device-to-device communication could provide some fault tolerance by allowing unaffected home devices to forward messages from others in their vicinity, but outside of the gateways coverage.

Therefore, this proposal extends the LoRaMoto baseline system’s capability beyond the LoRaWAN architecture, by allowing home devices to retransmit messages from other nodes under certain circumstances. Figure 2b shows two examples of home devices forwarding other devices’ packets. User 2’s home device is not in range of any gateway, so it cannot communicate with the network server and the application. However, it is within the reception range of user 3’s home device, so the latter can relay messages to the gateway, and back to its neighbor. Similarly, user 4’s device is nearly at the limit of the gateway’s reach. However, multi-hop transmission through user n can improve the former’s packet delivery. To achieve this operation, we assume that, in the event of an earthquake, user interaction disables the energy-savvy sleep mode, which provides long-term battery operation, in favor of more convenient system operations such as packet-forwarding capability.

4.3. Home Devices

In the LoRaMoto system design, it is assumed that citizens will have a small user device deployed at their homes and workplaces through which they can communicate in the aftermath of an earthquake when regular telecommuni-

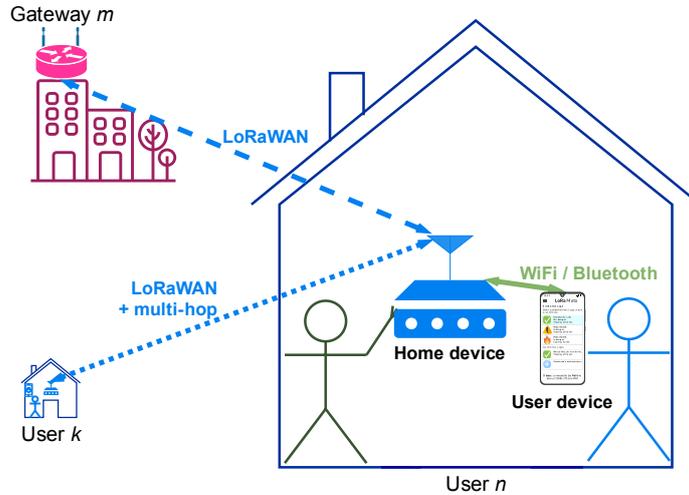


Figure 4: Typical interaction scenario with a home device.

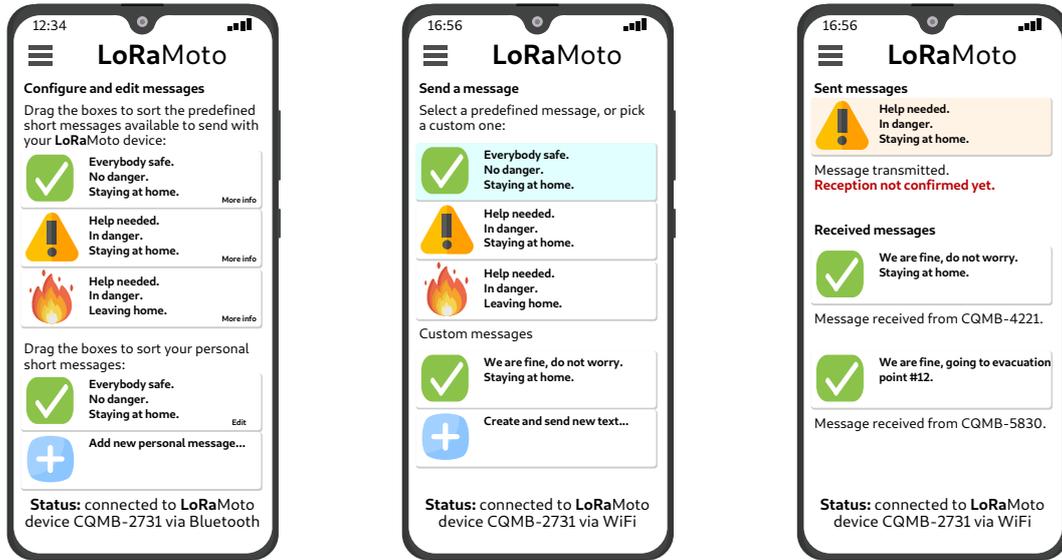
cations networks are likely to be inoperative. Home devices must meet certain requirements to fit into our system design:

- include a radio transceiver capable of sending and receiving LoRa packets
- include a WiFi or Bluetooth radio, to allow local wireless interaction
- include a screen (LCD, OLED, or e-ink), for user interaction and informative purposes
- include pushbuttons for user interaction
- have low power consumption
- have reduced physical dimensions
- be powered through a micro USB or USB-C cable, during idle operation
- be powered by batteries

There are a number of low-power Microcontroller Units (MCUs) available in the market to which a LoRa transceiver can be connected. Among them, Espressif Systems' ESP32 SoC is a popular option for embedded devices that require wireless connectivity, as it provides both 802.11bgn WiFi and Bluetooth support. Furthermore, small development boards based on that device and featuring a LoRa transceiver, like the one shown in Figure 3a, are readily available for purchase and allow a fast prototyping path. Figure 3b shows a prototype of a home device built on an ESP32-based development board, inside a 3D-printed custom case with three buttons for user interaction. These devices are easy to install and their cost is approximately USD 30.

4.4. User Interaction With a Home Device

Interaction of end-users with the LoRaMoto system occurs through the home devices deployed at the citizens' premises, as shown in Figure 4. Users can interact with the device by pushing its buttons or through an application running in their personal computers (e.g., smartphone, tablet, or laptop). Home devices are built with low-power embedded equipment based on highly integrated SoCs, with WiFi or Bluetooth capability, and they including a LoRa transceiver, as described above. On the one hand, short-range radios (WiFi or Bluetooth) are to be used locally to facilitate end-user interactions with their device, such as configuration. On the other hand, the LoRa radio is to be the long-range communication mechanism that allows end-users to send their safety status to the server. Using the same infrastructure, the server delivers the messages to their destination (i.e., a particular home device and the computers or user devices connected to it). Because the messages stored in the server are coded and their sources are geolocated, it is possible to use algorithms to aggregate such messages and quickly produce reliable information to support the planning, monitoring, and decision-making conducted by first response teams and public authorities. The way the



(a) Edition and configuration of messages before an earthquake. (b) Selection of the message to transmit in the aftermath of an earthquake. (c) An unconfirmed outgoing message and received texts in the aftermath of an earthquake.

Figure 5: Mock-up of the web page provided by the home device, to manage the list of messages it can store (5a) for later use in the aftermath of an earthquake (5b, 5c).

users can interact with the LoRaMoto system using their home device differs substantially before a natural disaster and in its aftermath. Next, we detail these two scenarios.

4.4.1. Before the Earthquake

The LoRaMoto home device is designed to allow its users to send and receive short messages, to be chosen from a list of predefined ones (e.g., “Everybody safe”, “Medical help needed”). In Addition, users can create personalized messages according to their needs (e.g., “We don’t know about John’s family”).

The short-range radio of the home devices provides a regular WiFi access point to which users can connect. Once there, a local web server running on the home device itself shows the list of messages and provides the customization options. Additionally, this configuration process can also be made with a smartphone application that uses Bluetooth to communicate with the home device. Figure 5 shows, on the left (Fig. 5a) a mock-up of the user interface for messages configuration, as displayed on a smartphone.

4.4.2. In the Aftermath

In the unfortunate event of an earthquake, people can interact with their home devices directly. This provides a simple interface to send and receive messages. Using the buttons and the LCD screen, they can choose to send a message (among the predefined ones or their custom-generated texts), and read incoming notifications. Direct interaction with the home device is the quickest way to communicate through the LoRaMoto system, but it might not provide the same experience a smartphone application will. However, in the event of a probable blackout, the home device, backed by batteries, is readily available for use.

In addition, people can interact with their home devices by using their regular personal devices (e.g., smartphones or laptops) to connect with it using WiFi or Bluetooth. As an example, Figure 5 is a mock-up of the smartphone application–web interface, connected to the home device to transmit a message using the LoRaMoto system (center, in Fig. 5b), and showing the status of sent and received messages (right, Fig. 5c).

Parameter	Real	Simulation	Deviation
Population	27,794	28,000	0.741 %
Homes	7515	7500	0.199 %
Populated area (km ²)	3.51	3.5	0.285 %
Homes density (per km ²)	2137	2143	0.280 %

Table 2: Deviation between real data and simplified model.

5. Scenario Modeling

To analyze the behavior of the LoRaMoto system in a realistic environment, we modeled a representative part of Coquimbo, Chile, a harbor town with a population of 240,000 inhabitants. Based on the data provided by the Chilean National Statistics Institute (INE), updated in 2017 [45], we observed that the area under consideration comprises two census districts (in Spanish, *distritos censales*): DC-1 and DC-2. These districts account for a total of 27,794 people living in 7515 homes, which are distributed in an area of 5.148 km². The population of DC-1 and DC-2 is concentrated in a smaller area of 3.51 km², while the remaining space mostly has no buildings. We estimated this area from the data available for the smaller neighborhood units (in Spanish, *unidades vecinales*) inside DC-1 and DC-2. In particular, we considered UV001, UV002, UV003, UV004, UV005, UV024, UV025, UV033, and UV034, which together cover almost all the populated area of DC-1 and DC-2.

To simplify the scenario for the simulation, we approximated the area under consideration as a rectangle of 1.4 km × 2.5 km, and we reduced the number of homes slightly to 7500. This led to a very small deviation, less than 1 % in the number of homes taken into account and their spatial density, as detailed in Table 2. We also considered homes to be uniformly distributed through the area and all the space having the same elevation.

Figure 6 shows, on the left, the 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 that are not in the white rectangle could very well fit in the unpopulated areas at the top left of corner DC-1, inside the white rectangle. Figure 6 also shows, on the right, the components of the LoRaWAN architecture in the simulator software, placed over the area under study. Eventual real deployments, however, should take into account the population distribution more accurately, using a precise model. Signal propagation models should account for the geographical aspects of the terrain by using height maps, and they should consider the size and location of buildings and the attenuation characteristics of internal and external walls.

5.1. Baseline Simulation Scope and Dimensioning

Given our interest in evaluating the system’s performance when handling a massive user interaction in the aftermath of an earthquake, the experiments discussed here focus specifically on this period and do not detail other aspects such as the bootstrapping of the gateways and the user nodes.

The software simulations start with a short setup process, during which all the devices (e.g., user nodes, gateways, and network server) perform their initialization routines. This setup is followed by an idle period during which no packets are transmitted over the air. This would correspond to the normal system operation before an earthquake. This procedure is common for all simulations reported in this work.

Figure 7 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 happens within a certain user reaction time, T_{UR} . We modeled 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, a user node waits for a random period of time before proceeding with a second transmission (or a *retransmission*), scheduled at $t = T_{Tx2} = T_0 + T_{UR} + T_{W1}$. After another random waiting period, a third packet transmission takes place, at $t = T_{Tx3} = T_0 + T_{UR} + T_{W1} + T_{W2}$, then a fourth transmission, and so forth. Waiting times between transmissions T_{W1} and T_{W2} are modeled as continuous uniform variables between 0 and 300 s. Since the system is meant to help emergency units provide a fast response in the aftermath of an earthquake, all the simulations are limited to one hour after the triggering event, regardless of any pending transmission. Since some



(a) Satellite view of Coquimbo.



(b) Simulation software components over the city map.

Figure 6: On the left (6a), the rectangular area of $1.4 \text{ km} \times 2.5 \text{ km}$ is layered on top of an aerial view of the Coquimbo Peninsula DC-1 and DC-2 depicted. It shows that their actual populated areas closely match the simplified area. On the right (6b) is a screen capture of OMNeT++ simulation software running the FLoRa framework, with different network components such as user nodes and gateways placed over the Coquimbo Peninsula (map © OpenStreetMap).

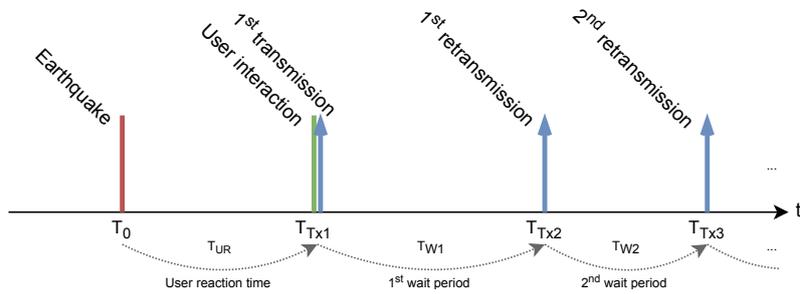


Figure 7: Timing of user node activity in the aftermath of an earthquake.

aspects of the simulations depend on random values (e.g., the position of the gateways on the map), there is a certain probability that the components might not be evenly distributed across the area under study. To overcome this effect, the results correspond to the average of executing each of the experiments 10 times but using different seeds for the random number generator (cRNG Class).

To evaluate how the proposed system worked in this realistic scenario, certain aspects and parameters are determined by the design. So, they must be dimensioned, and some technical decisions must be taken. Table 3 summarizes the values and configurations that define the baseline scenario for the simulations. First of all, the number of 7500 user nodes in the system is determined by the actual population density of the area under consideration, as discussed in Section 5. The transmission power for user nodes ranges from 10 to 22 dBm in 1 dBm steps, randomly assigned with uniform distribution, as a way to model the varying conditions in which the devices will be deployed (thickness and materials of the walls, height, etc.). The messages exchanged through the system are small (payload is in the order of 12 to 20 bytes). This size represents a compromise between the size of a predefined message (which could be encoded with a single byte) and a short and concise text message occupying a few bytes.

The LoRa SF employed by the user nodes ranges from SF7 to SF12, randomly assigned with a uniform distribution, to place a large number of devices in non-colliding transmission groups without any information of their potential interference. This distribution takes advantage of the signal orthogonality between transmissions using different SFs (as mentioned in Section 3.1), and it does not require any previous knowledge about the system or the environment. A better performing SF assignment could be made, for instance, before an earthquake happened (i.e., in Figure 7, before, T_0) using periodic communications between the nodes and the central application. Last, to reduce time on air, the bandwidth and FEC LoRa parameters have been set to 125 kHz and 4/5.

Regarding the number of gateways, two values are considered: 75 and 10. For this scenario we understand that, since many user nodes rush to transmit their messages within a short interval of time, network congestion can easily occur. One way to overcome the effects of this behavior is to deploy a much higher density of gateways than for typical LoRaWAN deployments (where end nodes transmit in a more evenly distributed fashion and a single gateway can provide good coverage for 10^4) [23, 25]. Therefore, we choose the value of 75 gateways, which corresponds to a 100 to 1 ratio of user nodes to gateways, as the baseline number of gateways. However, as mentioned in Section 4, it is likely that after an earthquake only a fraction of the gateways will remain in service so, in parallel with the 75 gateways, we also considered a downgraded system in which only 10 gateways remain in production. This allows for comparison between deployments with a high and a low density of gateways, to better understand how changing the aspects of the system affects its performance in different situations.

Finally, we set 3 as the default number of packets that each user node sends (i.e., one transmission of the original message and two retransmissions). Looking at Figure 7, this means that no additional packet transmissions occur at any node before $T_{T,x3}$. This value of packets per node is an arbitrary decision, but it also defines the maximum time between a user interaction and the end of the transmissions of 10 min, which could be a reasonable timing for an emergency scenario. Nevertheless, the simulations in Section 6.4.1 analyze the effect of retransmissions on the system.

The settings and timing mentioned above are common to all the simulations performed and discussed in upcoming in Section 6, except where otherwise stated (for example, when the baseline system is tested with a different number of home devices, a varying number of gateways, etc.)

5.2. Simulation Framework

To analyze the capabilities of the proposed system, we use OMNeT++ [46], an extensible, modular, component-based C++ simulation library and framework, in combination with FLoRa [47], a simulation framework to carry out end-to-end simulations for LoRa networks. OMNeT++ is a well-known discrete event simulator framework used by a lively academic community. Moreover, the FLoRa framework provides a complete implementation of the LoRaWAN architecture [48] and an accurate model of the LoRa radio physical layer derived from previous experimental findings [49]. Figure 6b is a screen capture of the simulation application’s graphical interface, with home devices spread over the city, the gateways scattered between them, and the network server and its related routing modules on top, outside the area under study.

The FLoRa framework is readily available to simulate a complete LoRaWAN deployment with, e.g., end nodes, gateways, and network servers, like the one described in the baseline scenario. To adapt the framework to our earthquake aftermath use case, we have adjusted several of its parameters according to the specifications in Section 5.1.

Parameter	Value
Number of home devices	7500
Number of gateways	75
Transmission power (home devices)	uniform(10...22) dBm, 1 dBm steps
Transmission power (gateways)	14 dBm
Payload size (home devices)	12 bytes
Payload size (gateways)	7 bytes
Spreading Factor	uniform(SF7...SF12)
Bandwidth	125 kHz
Coding rate	4/8
Messages per home device	1
Packets per home device	3 (1 orig. + 2 rtx.)
Time between packets	uniform(0...300) s

Table 3: Common dimensions and parameters applied to the simulations, except where otherwise noted.

However, to simulate the LoRaMoto architecture, which includes the packet-forwarding functionality, we have modified several parts of the FLoRa framework. The source code is available as a Git repository forked from the original one at <http://gitlab.com/rogerpueyo/floramesh.git>.

Our changes apply mostly to the LoRaNode module. It is the one used to model the home devices, and it includes two nested submodules that manage the application layer (LoRaNodeApp) and the LoRa network operations (LoRaNic). The former one provides the logic for packet-forwarding, and it allows the use of different forwarding algorithms and strategies. These range from a simple FIFO queue to context-aware forwarding as a function of environment parameters (e.g., reception of ACKs, signal reception quality, and timeliness). The latter one also is a compound module. It manages the MAC layer of the home devices (LoRaMAC) and the physical layer of the LoRa technology (LoRaRadio). Figure 8 shows the different nested modules as displayed by OMNeT++’s graphical interface.

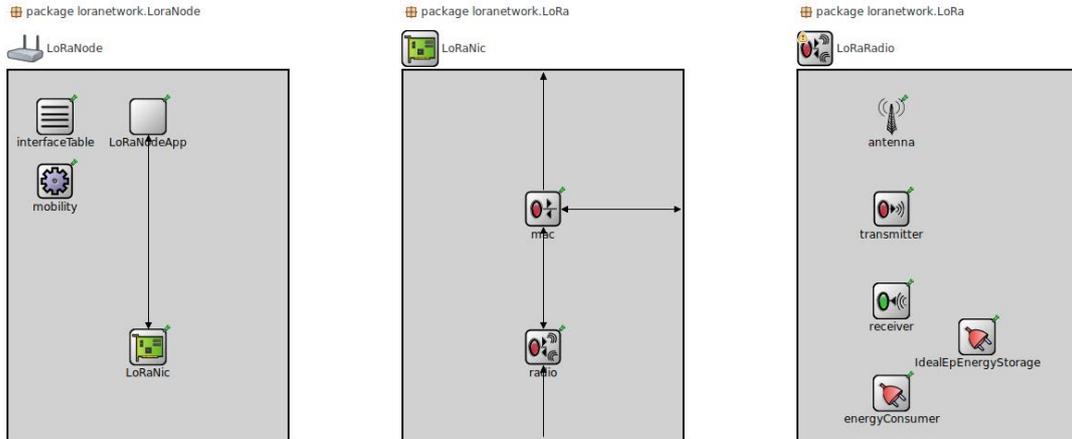
Along the same lines, the LoRaWAN gateway modules provided in the FLoRa framework have a nested structure analogous to the one in the end nodes. This structure takes into account the enhanced LoRa physical hardware capacity in terms of signal demodulation and network connectivity. Moreover, the NetworkServerApp module, which models the central application running on the network server, also has a compound nested structure like the one in the previous modules. We have modified such a module to handle packets that have been forwarded by home devices, in order to properly manage them, account for duplicates, and send downlink messages in reply.

6. Simulation Results

This section discusses the results from the simulations of the LoRaMoto system performed with the OMNeT++ framework using different configurations. We first explored the baseline architecture, in its simplest form as described in Section 4.1. It followed the LoRaWAN specifications and packets were sent only from the home devices directly to the gateways. Then we laid out the capacity of the architecture to cope with a challenging scenario like the aftermath of an earthquake, with thousands of devices transmitting in a short period of time. Then, we investigated how the modifications we included in the proposed LoRaMoto system (namely, bidirectional communication with ACKs and packet-forwarding between home devices) made it more reliable and, hence, more suitable to support communication among civilians in an emergency. Afterward, we explored how changes in specific parameters that have been predefined in Section 5.1 affected the system’s capacity to provide service to end-users.

6.1. Baseline LoRaWAN Architecture

The results and the analysis presented in this subsection offer an approximation to understand the capacity of the baseline system to be a communication mechanism in the aftermath of an earthquake. In particular, we analyze the scalability of the system—surveying its behavior with different numbers of home devices—and the impact that failures in the gateways infrastructure have on the percentage of users that can communicate successfully. In addition,



(a) The LoRaNode module, corresponding to a home device, includes the application module (LoRaNodeApp) and the network module (LoRaNic). (b) The LoRaNic module, providing the LoRa network capability, contains the MAC module (LoRaMAC) and the radio module (LoRaRadio). (c) The LoRaRadio module, providing the physical network layer, contains the physical components needed for LoRa communication.

Figure 8: Home devices are modeled in OMNeT++ with the LoRaNode compound module from the FLoRa framework (left), which includes several nested modules taking care of the application and the networking layers (center, right).

we study the effect on the system of an important LoRa transmission parameter, the SF, to facilitate comparison with the modified system architecture, simulated later.

6.1.1. System Scalability and Congestion

The density of user nodes is the most important aspect affecting the scalability of the system, followed by the density of gateways. Together they indicate the limits of the LoRaWAN architecture regarding network congestion. According to the environment modeling discussed in Section 5 for the Coquimbo Peninsula scenario, a ratio of one user node per household would give a total number of 7500 nodes (i.e., ≈ 2150 nodes/km²). However, scenarios in other geographical locations could be more—or less—densely populated, leading to different performance figures. In this section, we investigate system scalability by simulating it with different numbers of user nodes, while keeping the other dimensions and the rest of parameters untouched, equal to the baseline (geographical area, 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 be able in the future to apply different strategies to improve specific performance aspects.

Figure 9 plots the percentage of user nodes that can transmit successfully ≥ 1 , ≥ 2 , or 3 LoRa packets to the application when the system has a high density of gateways (Fig. 9a) and is fully operational. The figure also plots the percentage of user nodes when there is a low density of gateways (Fig. 9b)—only 10 of them remain unaffected after the earthquake.

The orange line in Figure 9a represents the percentage of nodes correctly transmitting ≥ 1 packets. For a number of user nodes between 100 and 1000, such a percentage remains mostly flat at 90%. This means that most of the nodes can communicate with the central application, while less than 10% cannot. Then, the success percentage slowly decreases beyond the few thousands of nodes, and it finally, drops past the $\approx 10,000$ -node boundary, when performance is remarkably degraded. The yellow and green lines (for the nodes that, respectively, transmit ≥ 2 or 3 packets) show similar trends, albeit more acute. As the number of nodes increases, so does collision probability, making it more difficult for a node to transmit successfully more than once or twice. For the low gateway density case, in Figure 9b, the trends in the graph are similar, although the reduced number of gateways provides significantly worse performance. So, at most, messages from 60% of the nodes reach the central application. The difference between the two set-ups indicates the importance of gateway density for system performance, and how their failure in the event of an earthquake can prevent a significant percentage of users from communicating their messages.

A remarkable observation from both graphs in Figure 9 is that, while some nodes cannot communicate with the central application even once, many of them can do it twice or even three times. This poses a problem of a lack of balance between nodes. Some of them keep transmitting a piece of information that has already been received, with

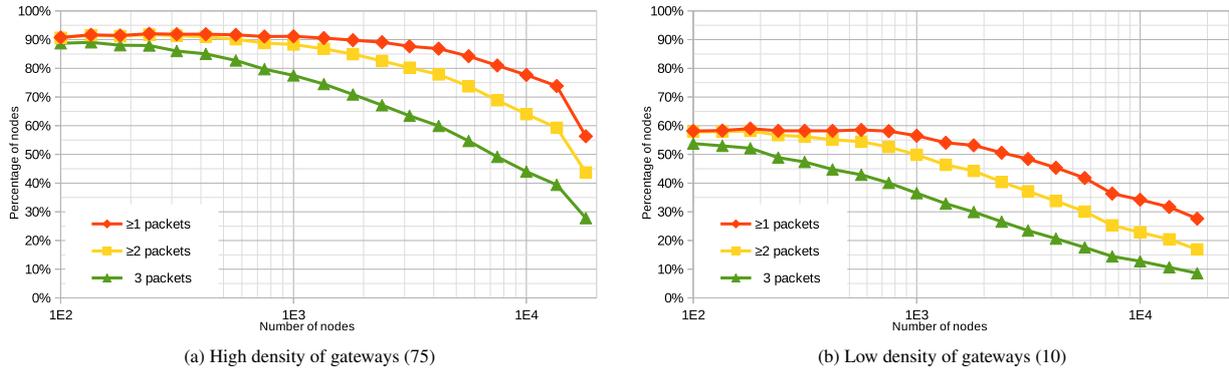


Figure 9: Percentage of user nodes transmitting successfully ≥ 1 , ≥ 2 or 3 LoRa packets to the central application, in terms of the number of user nodes in the system (ranging from 100 to 18,000). The left graph (9a) plots the full system operation with 75 gateways, while the right one (9b) corresponds to a downgraded operation in which only 10 gateways remain in operation. Note for the labels on the x-axes: $1E2 = 1 \times 10^2$, $1E3 = 1 \times 10^3$, $1E4 = 1 \times 10^4$.

their retransmissions occupying time on air and hindering gateways from receiving messages from other nodes that have not succeeded in their transmissions. One way to address this is by adding a downlink confirmation message from the central application to the nodes that have successfully transmitted their message, so that they cease retransmitting. This confirmation would be in the form of an ACK packet. Such an idea is explored in Section 6.2 as bidirectional communication is technically possible and the LoRaWAN considers the mechanism. There we evaluate its effects and analyze its trade-offs.

To sum up, the scalability of the system depends on the number of nodes and gateways (or their density per km^2), but it also depends on the pace at which the devices transmit their messages. In scenarios like this one, where a large number of nodes rush to transmit their data in a short time period, the bandwidth of the underlying LoRa radio technology may be too limited to provide a scalable solution if a higher density of nodes is considered.

6.1.2. Gateways Density and the Effect Of Infrastructure Failures

In the LoRaWAN architecture, the home devices send their data to the application running on the network server through one—or more than one—of the gateways available in the system. Therefore, the number of gateways that cover an area has a direct impact on the reception of packets. So, intuitively, increasing their density should improve the overall Packet Delivery Ratio (PDR), as that would increase the chance that one of them receives a transmission with a Signal-to-Noise Ratio (SNR) good enough to demodulate it successfully and relay it. (In simple words, there are more “ears” listening to what is being broadcast.) However, because the deploying gateways can be complex and expensive, their number must be properly dimensioned, and this raises an important trade-off. On the one hand, given a target performance (i.e., a percentage of home devices successfully reaching the application, whether it is 50%, 95%, 99%), only the required infrastructure providing it should be installed. On the other hand, in the emergency scenario considered in this analysis, the operation of the infrastructure cannot be taken for granted and certain performance degradation is to be expected. For these reasons, in this section, we modify the baseline scenario and analyze the impact of gateways density by simulating the system with a wide range of gateways (starting from a bloated number of 1000, down to only 1). By analyzing the results of the simulations, we want to understand how gateways density affects overall performance, to determine whether the choice of 75 gateways is sufficient and how the system performs in a downgraded status with fewer gateways available.

Figure 10 plots the percentage of the 7500 user nodes that can communicate successfully with the central application, and how many times they can do it (≥ 1 , ≥ 2 or 3 messages) as a function of the number of gateways deployed to receive their LoRa packets. (Note the logarithmic scale on the x axis.) The graph shows, as noted in Section 6.1.1, that the higher the number of gateways, the higher the percentage of user nodes that successfully reach the application. For a lower density of gateways, on the right side of the figure, small changes to their number have significant effects on the success ratio. However, for the central part, and as the density of gateways rises, many more gateways are required to achieve only small improvements in the system success ratio.

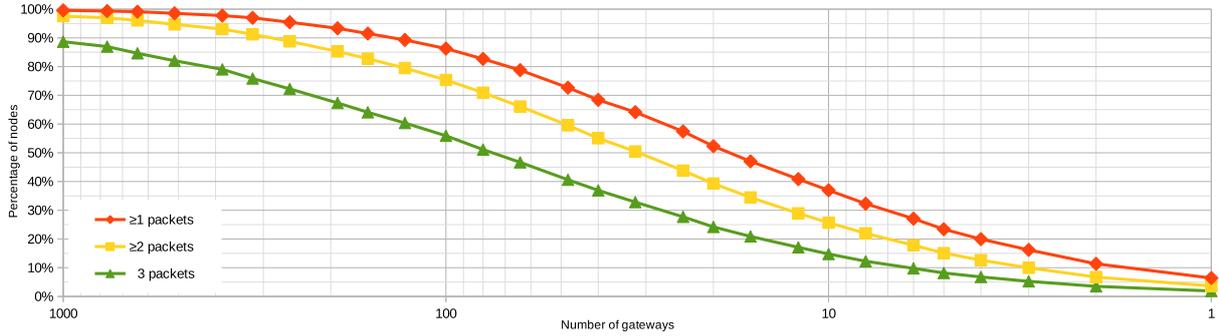


Figure 10: Percentage of user nodes transmitting successfully ≥ 1 , ≥ 2 or 3 LoRa packets to the central application, in function of the number of gateways available in the system (ranging from 1.000 to 1).

While Figure 10 suggests that the more gateways, the better success ratio, this strategy is limited by two factors: (i) as more gateways are added (hundreds, even up to a thousand of them), the success ratio figures improve only slightly (e.g., ≈ 100 gateways allow 86 % of the nodes reaching the application. Three times as many gateways —300— allow for roughly 96 % of them); (ii) the economic cost of a gateway is around 20 times as much as a user node, their deployment demands much more time and resources, and the costs of maintenance and operation also are higher.

It is difficult to determine an optimal number of gateways, since both technical issues and economic factors may play an important role in this decision. However, some statistical criteria can be applied to the decision. For example, in the baseline scenario with 75 gateways, ≈ 80 % of the user nodes can reach the application at least once. This percentage could be enough to provide emergency teams with useful data about which areas are more or less affected by the earthquake from a high-level point of view, but it would leave too many user nodes outside the system, especially as some gateways would cease to function. Furthermore, as some gateways can unpredictably become unavailable in the aftermath of an earthquake (for example, because of a building collapse), infrastructure over-building also should be considered as a countermeasure.

6.1.3. Spreading Factor Diversity

The spreading factor (SF) is a key parameter of the LoRa technology since it determines the spectral density of CSS modulation. In short, higher SFs mean longer transmission ranges and better reception sensitivity, at the expense of lower data rates and more energy usage. Lower SFs have the opposite effects, i.e., faster transmission speeds and less time on air, with reduced energy usage and, as a trade-off, shorter range due to the smaller SNR. In the preceding scenarios, user nodes were configured to randomly use 7 to 12 SFs. (These are common values in LoRaWAN deployments.) The configuration would follow a uniform distribution, so each node would transmit using any SF with equal probability. 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 election of the nodes' SF has an impact on a per-device basis but on the performance of the whole system too. For example, a node using a high SF can extend its communication range and reach more distant gateways, increasing the chances that its packets will be successfully received. However, a longer range can increase the probability of collision with the transmissions from other nodes that are using the same SF, causing a negative impact on the overall system. Furthermore, since higher SFs require longer air time, the collision probability is further increased, and this also has a negative impact from the system's global perspective.

In this experiment, we changed 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 a given system, for both a high density and a low density of gateways cases. We first simulated the system with all the nodes using the same single SF (from SF7 to SF12). Then, we defined all the possible SF combinations (SFs 7 to 12, 7 to 11, 7 to 10, etc.), and we simulated the system with nodes randomly choosing their SF from the given range, with equal probability.

Figure 11 plots the percentage of user nodes that can communicate successfully with the application ≥ 1 , ≥ 2 or 3 times, for all the possible SF combinations. In particular, Figure 11a corresponds to the scenario of a high density of gateways, and Figure 11b to the lower one. The data show that choosing the smaller single SFs 7 to 9 (shorter range

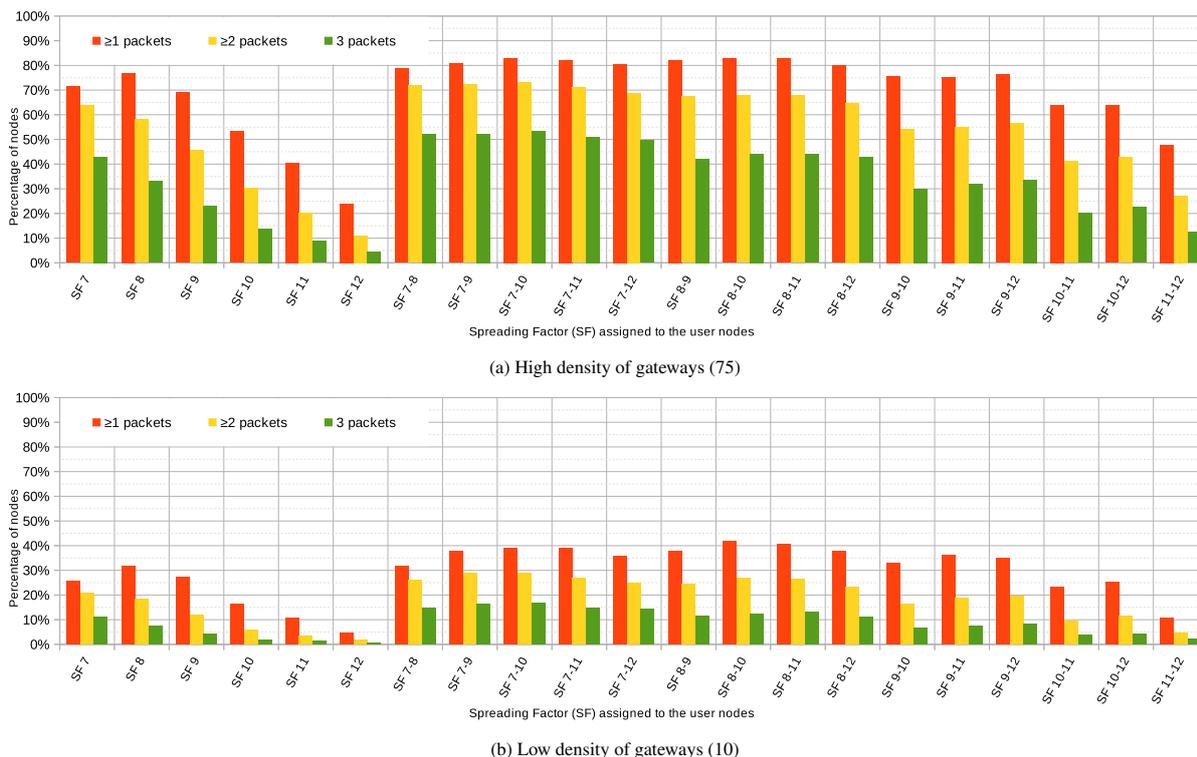


Figure 11: Percentage of user nodes transmitting successfully ≥ 1 , ≥ 2 or 3 LoRa packets to the central application, in terms of the SFs randomly assigned to them following a uniform distribution. The upper graph plots the fully operational system with 75 gateways, while the lower one corresponds to a downgraded operation with only 10 of them (y axes use 0 to 50 % and 0 to 100 % scales, respectively).

and time on air) provides significantly better results than bigger SFs.

Figure 11 also shows the effect of using different SFs in parallel. It shows that combinations of two or more SFs achieve better performance than a single SFs (i.e., combining SFs 7 and 8 is better than using only SF7 or SF8). Interestingly enough, the widest combination, ranging from SF7 to SF12, which corresponds to the default values predefined in Section 5.1 and used throughout this section, does not provide the best results. Indeed, for both higher and lower densities of gateways, choosing high SFs (11, 12) penalizes the performance figures, while avoiding them and concentrating more nodes on shorter SFs increases the success ratio of packet delivery. Nonetheless, it must be noted that the uniform SF distribution is not fair. Switching from one SF to the immediately higher one means doubling time on air, hence the latter will have a much higher collision probability than the former. A more balanced SFs distribution would take this into account.

6.2. Downlink ACK From the Application To User Nodes

The communications system described in Section 4.1 and analyzed in Section 6.1 has a fundamental limitation: data flow is unidirectional. It goes only from the user nodes to the central application. While this design can help civilians give emergency response teams valuable information so they can coordinate their activities, it does not provide any feedback to the users, which may hinder them from interacting with the system. However, making the application in the network server reply to the packets received from the home devices with downlink ACK packets would not only avoid unneeded retransmissions but, most important, provide end-users with some feedback—i.e., whether their message has been received or not. Moreover, rather than providing only a mechanism to control network congestion, ACK packets could also be used for piggy-backing messages from other users (e.g., sent by family or friends). This would open the system up to more sophisticated communication patterns and alleviate users' concerns about the safety of their loved ones, mentioned in Section 1. This would also allow the implementation of ng software solutions that work under an IoP-based paradigm, in which citizens are the key actors.

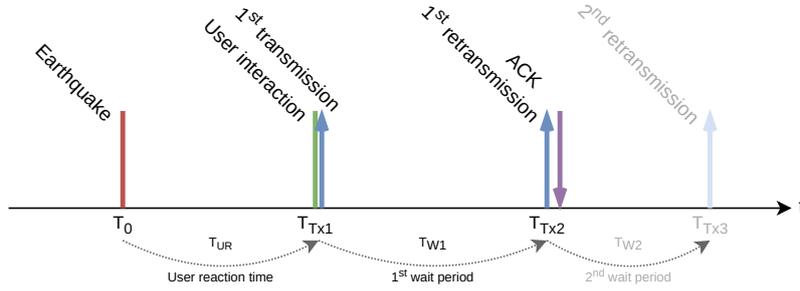


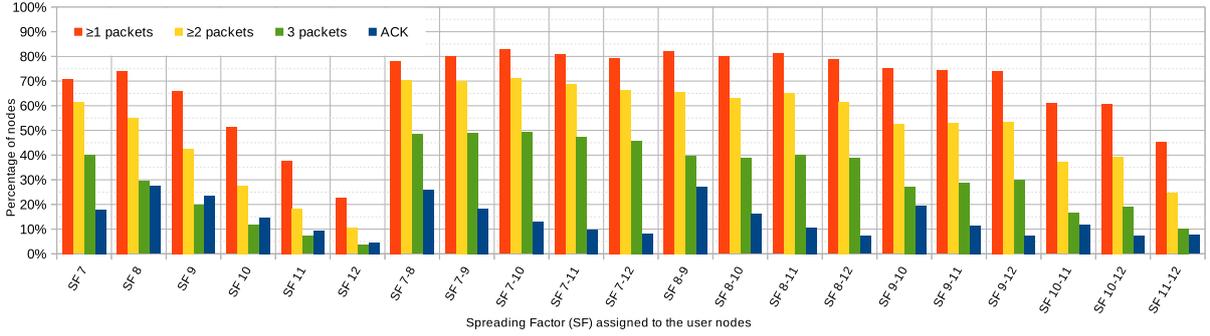
Figure 12: Timing of user node activity in the aftermath of an earthquake, including ACK downlink messages.

Downlink ACK messages are already present in the LoRaWAN architecture, although rarely used on battery-operated devices to save energy. They are easy to implement, and they can reduce the number of packets transmitted by the user nodes. This would reduce the overall probability of packet collision and increase the chances that more nodes would transmit their information successfully. However, two issues must be considered: first, ACK packets also occupy the radio channel, interfering with the uplink messages from the nodes and, second, their reception by the user nodes is not guaranteed, as they also have a certain collision probability because of interference. To understand their effect on system performance, we implemented a simple ACK mechanism, and we evaluated its impact by comparing the system (i) when ACKs are disabled, (ii) when ACKs are sent from the central application, but the user nodes keep retransmitting their messages regardless of their reception and (iii) when ACKs messages are sent and, upon receipt, user nodes stop any further retransmission. Figure 12 shows the schedule of events and actions occurring at one of the user nodes. Compared to the baseline scenario (see Figure 7), now a downlink ACK can reach the node and, depending on the configuration, the node will avoid sending the remaining retransmissions, saving time on air for other nodes. Downlink ACK packets have a payload of 12 bytes, the same as the uplink data packets sent by the home devices. By doing this, we emulate the piggy-backing of actual messages sent from one user to another, bringing the LoRaMoto system closer to the IoP paradigm.

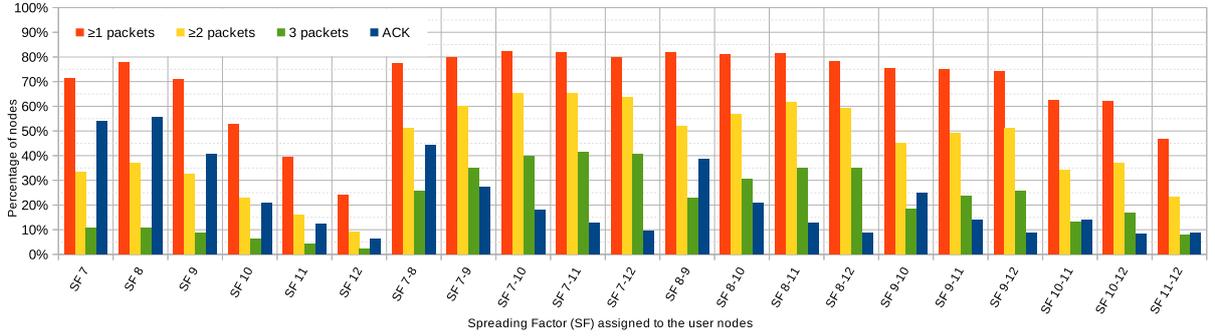
Among the different experiments we made for the baseline scenario in the previous Section 6.1, we chose here to focus on the effect of the ACK messages for various SFs. Figure 13 shows the percentage of user nodes that transmit successfully ≥ 1 , ≥ 2 or 3 LoRa packets to the central application, as well as the percentage of nodes that correctly receive a downlink ACK message from it, in terms of the different SFs assigned to the user nodes. (This is similar to Section 6.1.3). For the upper graph (Figure 13a), the downlink ACKs messages are enabled but user nodes keep retransmitting their packets; for the lower graph (Figure 13b), they stop retransmitting upon receipt of the ACK message. Comparing the results with the the ones showin in Figure 11a, it can be seen that ACK messages cause a very small decrease in performance. (About 3% less nodes can contact the central application.) This is because of the increase in the number of packets traveling through the air. However, regarding the user interaction, we now have, for the first time, the possibility of providing feedback to the users—or, at least, to some of them—after their interaction. The quantity of user nodes receiving ACK varies, though, as shown in Figure 13, in function of the SFs they use (smaller ones tend to provide better results). The big difference comes, however, when nodes cease to transmit upon receipt of an ACK message. While the number of nodes transmitting ≥ 1 message only slightly improves, much fewer nodes get to continue transmitting and, therefore, only a small percentage end up sending all 3 packets they can possibly broadcast. This actually saves a lot of time on air that, in turn, allows many more ACK packets to be successfully received by the nodes. Regarding user interaction, now at least some of the user nodes know their message has been properly received, and they can notify the end-user of this. This is a fundamental aspect for end-users, who will be waiting for feedback from the system to assuage their concerns about the conditions of their family and friends.

6.3. Packet-forwarding Between User Nodes

The analysis performed in Section 6.1 suggested that, given the dimensions of the scenario under consideration, the baseline LoRaWAN system cannot provide a reliable communication channel for all the users involved. In particular, if part of the infrastructure is not available in the aftermath of an earthquake, a considerable proportion of user nodes



(a) Downlink ACK message, nodes continue retransmissions, high gateways density (75)



(b) Downlink ACK message, nodes stop retransmissions, high gateways density (75)

Figure 13: Percentage of user nodes transmitting successfully ≥ 1 , ≥ 2 or 3 LoRa packets to the central application, and receiving an ACK from it, in function of the SFs randomly assigned to them following a uniform distribution, in the high gateways density deployment. The upper graph plots the results when ACK messages are sent but retransmissions by the user nodes continue; the lower graph, when ACK messages are sent and user nodes receiving them cease to send further retransmissions.

might be left out of the system. However, adding ACK messages and enabling bidirectional communication (explained in Section 6.2), not only provides end-users with feedback to their home devices, but it also adds a congestion control mechanism and it opens up the possibility of implementing a more sophisticated system architecture.

Leveraging the results from the previous experiments, for this section we add to the user nodes the capacity to forward to the gateways packets from other nodes. The objective here is to increase the percentage of user nodes that can successfully reach the central application. Now, each of the user nodes, during the wait times before packets (T_{W1} , T_{W2} , etc., as detailed in Section 5.1 and depicted in Figure 7), listens for other nodes' transmissions and stores their messages in a buffer. Afterward, when its own messages are sent, some—or part—of the stored packets are transmitted. This strategy should give those nodes at the edge of the system, or outside of the coverage of a gateway, additional chances for their data to reach the central application. This is especially true when the procedure includes a congestion control mechanisms, such as the ACK strategy from the previous section. It is worth noting that the proposed system design goes beyond the specifications of the LoRaWAN architecture, which does not consider direct communication between nodes, even if the underlying LoRa radio technology allows it.

To evaluate whether forwarding other nodes' packets has any impact on system performance, we simulated the baseline scenario adding the 7500 nodes with forwarding capacity, in both a high and a low density of gateways (75 and 10 of them, respectively), and enabling downlink ACK packets from the central application to the home devices. The user nodes first send their messages, with up to three transmissions, until an ACK message is received. During that time, they also store any packet received from the home devices around—given it is possible in terms of the LoRa technology (e.g., good reception signal, no collision). Only when their own messages have been sent—or an ACK has been received—do user nodes forward other nodes' packets. To assess the effect of the forwarding mechanism, we performed a simulation parameter sweep in which nodes were allowed to forward from 1 to 10 packets. (We also allowed 0 packets, i.e., no forwarding, for comparison). For the sake of simplicity, nodes pick packets at random to

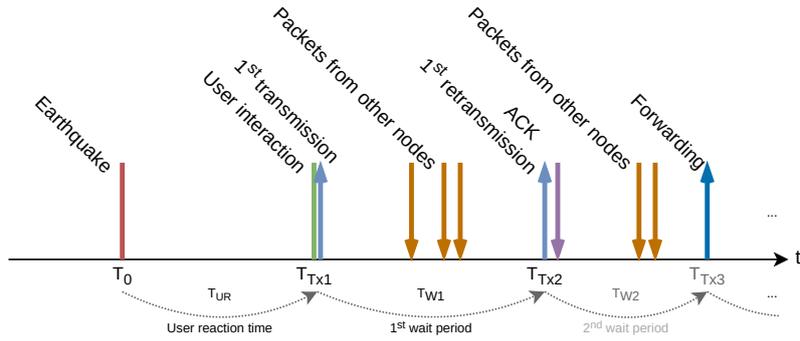


Figure 14: Timing of a user node activity in the aftermath of an earthquake, including ACK downlink messages and packet-forwarding.

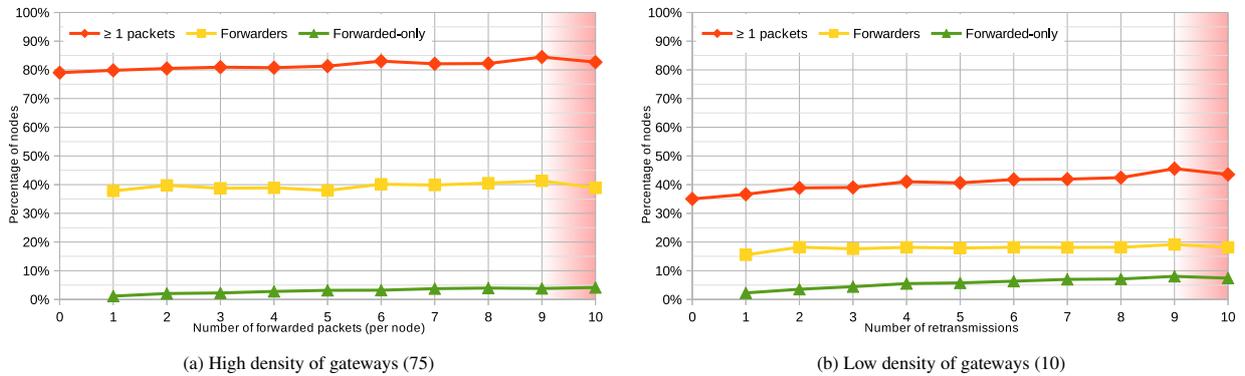


Figure 15: Percentage of user nodes transmitting successfully ≥ 1 packets to the central application (orange), percentage of user nodes working as forwarders of other nodes' messages (yellow) and percentage of user nodes that only reach the central application via forwarding (green).

forward from the storage buffer, without taking into account any aspect like arrival time, source node, and reception quality. Figure 14 shows the schedule of events and actions occurring at one of the user nodes. Compared to the baseline scenario (see Fig. 7), now a downlink ACK can reach the node and, depending on the configuration, will stop sending the remaining retransmissions, saving time on air for other nodes.

Figure 15 shows the effect of making home devices forward packets from neighboring nodes. The graphs indicate that packet-forwarding has a positive effect on system performance, as more individual home devices can reach the central application than when no packets are forwarded. For instance, on the left graph (Figure 15a, with a high gateways density setup), the percentage of home devices reaching the application rises from 79 % to 84.5 %. On the right graph (Figure 15b, with a low gateways density setup) the percentage increases from 35 % to 45.5 %. The graphs also show that, as more packets are forwarded, there is an increasing percentage of user nodes that reach the central application only through forwarding—i.e., thanks to another home device. Without the forwarding mechanism, those nodes would not have been able to communicate with the central application, so the system does benefit from this feature. As a downside, however, forwarded packets appear in the system at a longer time after the earthquake, after the regular packets are transmitted. This means that the information they carry will reach the central application with an additional delay.

This experiment has been conducted as a preliminary assessment to investigate whether forwarding packets between nodes has any impact on the system, and of what magnitude. The results in Figure 15 show the effect of the forwarding mechanism. They indicate that such a mechanism can improve certain performance metrics, such as increasing the number of unique nodes reaching the central application. However, more complex forwarding strategies need to be designed and tested, leveraging the available information at the user nodes, in order to use the bandwidth and time on air more efficiently and maximize the percentage of nodes successfully participating in the system. In addition, an advanced ACK messages algorithm also should be designed and tested to leverage the packets that have

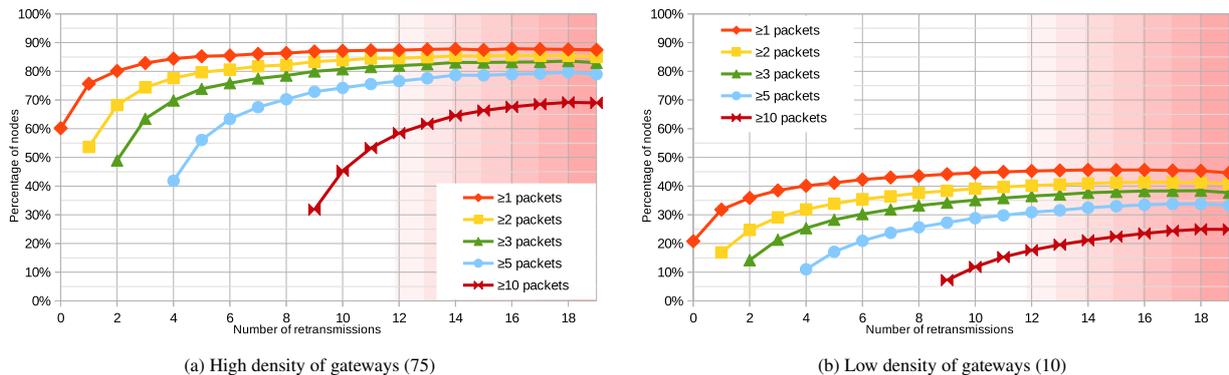


Figure 16: Percentage of user nodes transmitting successfully ≥ 1 , ≥ 2 , ≥ 3 , ≥ 5 or ≥ 10 LoRa packets to the central application, in function of the number of allowed retransmissions per node (ranging from 0 to 19). The left graph plots the performance of a system operating with a high density of gateways (75); the right one corresponds to a low-density version with only 10 of them.

been forwarded or even to combine messages from neighboring nodes to reduce the number and size of transmissions, minimize time on air, and increase the reliability of the communications.

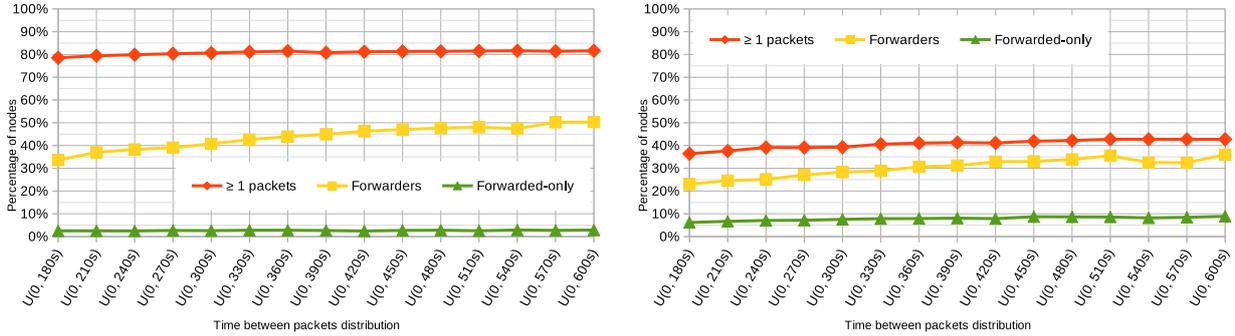
6.4. System Sensitivity To Design Parameters

In the modeling and the definition of the baseline system architecture, several important design parameters were set to specific values (e.g., the packets' payload size or the time between retransmissions). These values have been chosen as reasonable options for the context of the scenario (e.g., the payload size cannot be 0 nor exceed the network Maximum Transmission Unit (MTU), the time between retransmissions must provide a compromise that provides a quick response and avoids network saturation). However, other numbers could have certainly been used. For this reason, in this section we analyze the sensitivity of the LoRaMoto architecture to changes in important system parameters, to better understand how these changes affect performance and the system's ability to provide people with a means of communications to learn whether their family and friends are safe in the aftermath of an earthquake.

6.4.1. Packet Retransmissions

As detailed in Section 5, after the user interaction, a 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 7 on page 12). Retransmissions have a positive impact on the system, as they provide additional chances for end nodes to deliver their messages to the central application. However, they also have a cost in terms of time on air, network congestion, signal interference, and energy (albeit this last one is not considered in this paper). Not having a downlink response in the form of an ACK message from the application simplifies the system design, but it also prevents user nodes from knowing whether their retransmissions are required or not. In this experiment, we modified the number of retransmissions the user nodes perform, to understand their effect on the system and on the performance metric (percentage of unique user nodes reaching the application). We simulated the system with all the nodes retransmitting their packets from 0 to 19 times (i.e., each user node sending between 1 and 20 packets).

Figure 16 plots the percentage of user nodes that can successfully transmit ≥ 1 , ≥ 2 , ≥ 3 , ≥ 5 and ≥ 10 LoRa packets toward the central application, as a function of the number of retransmissions allowed per node. The graphs clearly show that, while allowing more retransmissions allows more nodes to reach the application, the percentage of nodes doing it ≥ 1 grows. But the growth is asymptotic, never reaching the ideal 100%. This saturated behavior is common for both a high density (Fig. 16a) and a low density of gateways (Fig. 16b) with 75 or 10 nodes. This shows that as the nodes send more packets, the unbalance between them remains: some of them almost always succeed in transmitting all the packets to the application, while others cannot make it regardless of how many times they try. Similar to the baseline, this means that, on average, nodes are misusing the available time on air and bandwidth, and allowing for more retransmissions cannot help balancing that usage.



(a) High density of gateways (75)

(b) Low density of gateways (10)

Figure 17: Percentage of user home devices transmitting successfully ≥ 1 packets to the central application (orange), percentage of home devices acting as forwarders of other home devices' packets (yellow) and percentage of home devices that only reach the central application via forwarding (green), in function of the waiting time between packets.

The LoRaMoto system aims at providing a communications mechanism for the population in the aftermath of an earthquake, in particular during the first hour after the earthquake. Therefore, allowing user nodes to retransmit indefinitely not only wastes time on air, bandwidth, and energy, but it also means that, at some point, messages will arrive a long time after the earthquake. Thus, with the time between packets defined as a uniform random variable between 0 s and 300 s, since $3600s/\max[0, 300]s = 12$. Therefore it is possible that, if more than 12 retransmissions are allowed, the simulation terminates while some nodes still have not sent all their packets. This can be seen in the right part of the plots in Figure 16, where the trends suggest a slight decrease in the success ratio—hence the reddish background of the graphs.

6.4.2. Time Between Packets

As discussed in Section 6.1, after the user interaction, a home device transmits its message up to three times. The first transmission begins right after the user interaction, while the two retransmissions occur after waiting periods T_{W1} and T_{W2} , as depicted in Figure 7. The waiting time between packets poses a trade-off in the system between its speed its response (nodes sending packets at a slower or faster cadence) and the reliability of the transmissions (leading to more or less probable collisions). In certain scenarios, faster but less accurate data might be desired in order to start sketching an aftermath response plan. In other cases, more reliable transmissions might be desired to provide a slower but trustworthy communication channel for end-users.

Our system model considers a waiting time between packets with a uniform probability distribution. This way, all the transmissions are equally distributed over time, maximizing the probability of successful communication. The baseline scenario in Section 6.1 used random waiting times uniformly distributed between 0 and 300 s, and so do subsequent simulations. This means that, with three packets to be sent, it takes a node an average of 300 s between the first and the last transmission. In this section we simulate the LoRaMoto architecture using shorter and longer waiting periods between packet transmissions for home devices, using a uniform probability distribution between 0 and 180 s (fastest case), and 0 and 600 s (longest case), to understand how this parameter affects the system response.

Figure 17 plots the percentages of home devices that can communicate successfully with the application, for different waiting periods ranging from $U(0, 180s)$ to $U(0, 600s)$ (where $U(0, ns)$ means uniform distribution from 0 s to ns). The graphs show that as the waiting periods get longer and transmissions are more sparsely distributed, collision probability decreases and the percentage of home devices achieving successful communication increases. It is worth noting that the waiting time between packets has a more prominent impact in a scenario with a low density of gateways (Figure 17b) than with a high density one (Figure 17a). This effect should be taken into account in real deployments, where the gateway infrastructure may experience failures in unpredictable ways. Therefore, dynamically adjusting the waiting time between packets at the home devices as a function of the density of gateways in the vicinity could maximize the throughput of the system overall. Such a decision could be easily made by the central application based on the redundancy of the packets received from the home devices to the gateways and reported as additional information inside ACK messages.

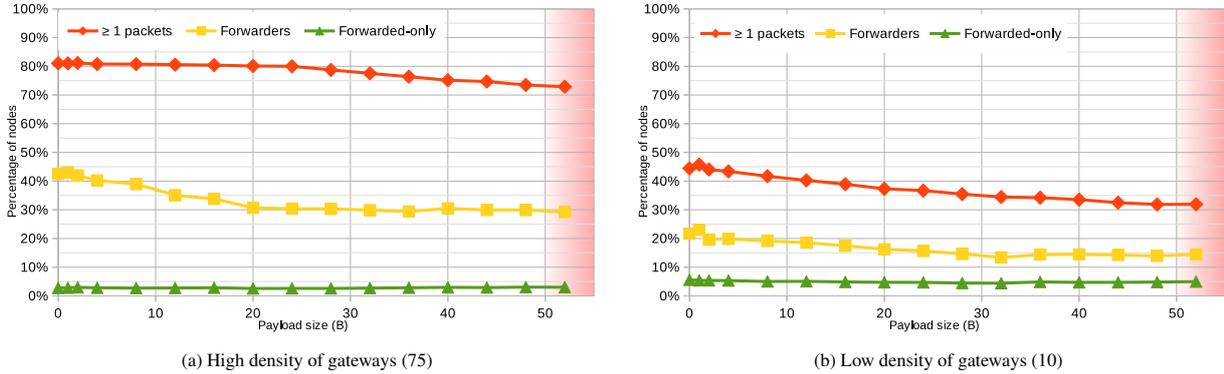


Figure 18: Percentage of user home devices transmitting successfully ≥ 1 packets to the central application (orange), percentage of home devices acting as forwarders of other home devices' packets (yellow) and percentage of home devices that only reach the central application via forwarding (green), in function of the home nodes' packet payload size.

6.4.3. Payload Size

When dimensioning the system parameters in Section 5.1, a fixed payload size of 12 bytes was set, and this value was used through all the previous simulations. As discussed there, this number represents a compromise between the size of a predefined message (which could be encoded with a single byte) and a short and concise text message occupying a few bytes. In addition to the actual data being sent by the users, transmitted packets also include the MAC layer headers, LoRa modulation CR bits, etc., but these values are constant and do not depend on the users' actions. In the following simulations, we analyzed the impact of setting different payload sizes on the home devices' capacity to transmit their packets successfully. This is because the capability of the system to transmit user-generated messages of arbitrary content and size is critical to fulfilling the communication needs of users in the context of an IoP scenario in the aftermath of an earthquake. To this end, we tested the LoRaMoto architecture using different payload sizes, to visualize the impact of this parameter on the system.

Figure 18 plots the percentages of home devices that can communicate successfully with the application, for different message payload sizes ranging from 0 bytes (i.e., an empty message but including headers and other datapoints noted above) to 51 bytes (the biggest payload all SFs can accommodate [50]). The graphs show that, when a high density of gateways is operative, the payload size has a limited effect on the percentage of home devices that can communicate with the central application. (There is around 5% difference between the two extreme values.) However, increasing the payload size has a more remarkable effect in a scenario with a low density of gateways. In that case, where the percentages are already low, the difference between the extreme values of the payload size reduces performance by approximately 15%. Therefore, home devices' reception and forwarding of packets from neighboring devices are more affected by the payload size when fewer gateways are operative. Leaving the density of gateways aside, home devices are less likely to contribute to the system by forwarding packets as the payload size increases. This effect should be taken into account by the users of the LoRaMoto system when creating custom messages, as longer ones may delay communications compared to shorter ones.

6.5. Summary Of Results

The first section of simulations results (Section 6.1) was a comprehensive exploration of the boundaries of the system based on the baseline scenario. Table 4 summarizes the most remarkable performance figures. For a high density of gateways, 80% of the nodes can communicate with the central application hosted on the premises of the emergency management agency. With a low density of gateways, only about one-third of the nodes succeed in doing this. These results are used as a baseline against which to compare the effect of modifying certain aspects of the system on the overall performance.

This exploration revealed that the proposed system can scale well when the total number of nodes is below a few thousand. Up to that point, the metric to be taken into account—i.e., the percentage of user nodes reaching the central application—stays almost constant around 80%. However, beyond a few thousand nodes, the percentage is reduced,

Baseline (LoRaWAN)			LoRaMoto	
	Parameter	Value	Parameter	Value
75 GWs	Packets received by app. (all)	14.751	Packets received by app. (all)	15.228
	Nodes with ≥ 1 succ. tx.	5.997 (80,0 %)	Nodes with 1 succ. tx.	6.158 (82,1 %)
	Nodes with 0 succ. tx.	1.503 (20,0 %)	Nodes with 0 succ. tx.	1.343 (17,9 %)
10 GWs	Packets received by app. (all)	6.328	Packets received by app. (all)	7.240
	Nodes with ≥ 1 succ. tx.	2.695 (35,9 %)	Nodes with ≥ 1 succ. tx.	2.944 (39,3 %)
	Nodes with 0 succ. tx.	4.805 (62,1 %)	Nodes with 0 succ. tx.	4.556 (60,7 %)

Table 4: Baseline (LoRaWAN) and LoRaMoto architecture results summary, for both a high (75 GWs) and a low (10 GWs) density of gateways. Numbers correspond to home devices transmitting three packets in all cases, either theirs (left, for the LoRaWAN baseline) or including forwarded messages from neighboring devices (right, LoRaMoto).

and eventually it drops when there are more than ten thousand nodes. This means that there is a strong technological limitation regarding system scalability to cope with the rush of packets being transmitted at the same time.

The number of gateways in the system also plays a significant role in performance. When simulating the system with varying numbers of gateways, we noticed a direct relationship with the percentage of nodes sending messages to the central application correctly. In particular, we found that a large number of gateways could provide very good performance. However, the strategy of using many gateways has clear limitations in real life because of the costs to deploy and maintain the devices.

Despite this, there are other strategies to improve the system performance that do not require additional hardware or deployment costs. Two of these strategies were explored: the number of maximum retransmissions per node, and the distribution of possible SFs to the nodes. In particular, we found that the former strategy could complement the two main suggested modifications to the baseline system: downlink ACKs and packet-forwarding. The addition of acknowledgment messages from the central application to the user nodes when data transmissions are successful had a very small negative impact on system performance (about 2% fewer end nodes transmit successfully). This implies that more messages can be sent on the same transmission medium. However, using ACKs as a congestion control mechanism to avoid sending unnecessary messages over the air increased packet transmission success around 3%. This may be a small gain, but the main benefit of using ACK messages goes to the users. It enriches the users' experience when interacting with the system, and it serves as a basis on which to build a more sophisticated communications mechanism to enable the transmission of messages from one user node to another. The main motivation for this paper, improving users' awareness about the safety condition of their family and friends, can greatly benefit from this strategy.

The addition of packet-forwarding functionality to the end nodes requires a system design that is more complex than the LoRaWAN architecture, but it provides a complementary method to improve performance. We have observed that, combined with the aforementioned ACK mechanism, a significant percentage of the user nodes that end up out of the gateway coverage—for whatever reason—can still have their messages reach the central application, as other user nodes mediate as packet forwarders. Furthermore, the effect of this strategy is more visible as fewer gateways in the system remain operational. This is very positive for the earthquake aftermath scenario.

Having performed these several analyses of the system in different scenarios and with diverse features, we conclude that the proposed system can greatly help end-users meet their needs for communications in the aftermath of an earthquake. However, since this is a very sensitive scenario, warranties for a good enough communication performance should be given. This challenging requirement can be achieved only by combining all the improvements and strategies explored above.

7. Discussion

The literature reports several approaches that share features with the LoRaMoto system. However, these approaches show limitations of scalability or performance to meet communication needs, especially in areas with a high density of nodes, such as an urban area affected by an earthquake.

Some of those communication systems were based on WiFi or other radio technologies that have shorter ranges than LoRa (discussed in Section 2.1). Those underlying communication technologies limit the capability of the

systems to scale and address large areas, like those involved in urban emergency responses in the aftermath of an earthquake. Moreover, these systems often use message or packet sizes of hundreds of Kb, that are much bigger than those used by the LoRaMoto system. In addition, they consider only deployments with fewer than a hundred nodes. This is almost two orders of magnitude less than those required to address urban emergency responses.

Several other systems use LoRa technology to support communication among first responders in the field during emergency responses (discussed in Section 2.2). Given the nature of the activities performed by those users, most of those systems are designed to support node mobility. That is not the scenario addressed by LoRaMoto where the nodes (i.e., civilians staying at home) are stationary or quasi-stationary. Moreover, solutions based on regular LoRa have limited scalability because communication coverage drops exponentially as the number of end-devices grows [22, 23]. Therefore, they are not suitable to address communication scenarios like those described in this article, i.e., with a high number of civilians exchanging messages in a short time period. To deal with this problem, LoRaMoto proposes a message forwarding mechanism described in Section 4.2.

There are also some proposals based on LoRaWAN, where a single gateway can successfully serve thousands of devices in stationary or quasi-stationary scenarios, as in smart city environments [23, 25]. However, those solutions involve a high node-to-gateway ratio but not necessarily low message delays. Those approaches are not aligned with the requirements for an urban emergency response in the aftermath of an earthquake. In this latter scenario, the gateways must route messages from areas with a high density of nodes in a short time period with a high message delivery rate.

The comparison of performance and scalability between LoRaWAN and LoRaMoto in a simulated emergency response, presented in Section 6.3, shows that the latter provides more network coverage and successful message delivery than the former. This is a consequence of the capability of LoRaMoto to manage a high density of gateways and nodes, and the packet-forwarding mechanism embedded in that system.

The analysis of the prior research indicates that none of those communications systems address all challenges imposed by an urban emergency response for civilians. This is not surprising because those proposals were not designed for such a purpose. In this sense, LoRaMoto represents a step forward in the design of communications systems that support and involve civilians in urban emergency responses.

8. Conclusions and Future Work

This work presents LoRaMoto, a communications system to allow civilians to exchange information with family and friends about their safety conditions in the aftermath of an earthquake. These interactions are to be performed mainly through predefined short messages that help people be informed while staying home. Furthermore, because the messages are coded and geolocated, they can be aggregated and related using crowdsourcing mechanisms, and thus generate knowledge that first response and emergency management organizations can use in their diagnosing, planning, and decision-making activities. In this scenario, civilians using LoRaMoto become human sensors that feed the system, providing quick and updated information. This extends the typical emergency response scenario towards an Internet of People paradigm, where civilians play a key role and also help reducing risks.

The proposed system is built on LoRa radio technology and extends the LoRaWAN architecture with packet-forwarding for end nodes (i.e., home devices). To determine its scalability and the effect of its parameters on overall performance, we conducted several simulations that considered a representative part of Coquimbo (Chile), a harbor town affected by two earthquakes during the last decade. We analyzed LoRaMoto by running several simulations that explored different aspects of the system. As a result of these scenarios, defined two metrics of interest: the percentage of user nodes that can correctly send a packet to the central application, and the percentage of user nodes that receive a confirmation ACK message from the application. We observed, regarding scalability, that the system performs consistently with hundreds of nodes, and it can scale up to a few thousand, but completely saturates beyond ten thousand nodes. This trend would indicate an upper bound in the density of nodes that are part of the system. Tightly related, the density of gateways has a crucial impact on the percentage of end nodes able to communicate successfully. In particular, we observed that a nodes to gateways ratio of 100:1 could provide reasonable system performance, but we deemed that a relation of 750:1 would provide too low a percentage of user nodes covered by the system.

Since infrastructure blackouts in the aftermath of an earthquake could render some gateways out of order, leaving part of the user nodes without service, we extended our system beyond the LoRaWAN architecture and implemented

a proof-of-concept packet-forwarding mechanism between end nodes. This way, home devices out of coverage from the gateways could still have means to transmit messages to the central application, by relying on other nodes. The simulation of the system enabled with this feature revealed that a significant percentage (around 10 %, depending on different factors) of the nodes affected by gateway blackouts could still be part of the system, improving its usefulness for end-users.

We have identified a number of potential issues that should guide our future efforts to improve the performance metrics of the system. These include the way that user nodes are assigned LoRa SFs, to maximize the network capacity. A possible strategy would consist in implementing a warm-up mechanism—before the earthquake—during which the user nodes and the central application could optimize system aspects, such as SF and transmission power, with the objective of ensuring efficient usage of the LoRa radio spectrum.

Beyond this, having shown that direct node-to-node communication and packet-forwarding can successfully contribute to system performance, we would like to further investigate the combination of infrastructure-based (i.e., gateways) and infrastructure-less (i.e., gateway-less) solutions to foster LoRaMoto 's capacity to expand communications between civilians in the aftermath of an earthquake.

Acknowledgement

This work was partially supported by the European Regional Development Fund and the Spanish government (TIN2016-77836-C2-2-R and PID2019-106774RB-C21), by the Spanish State Research Agency (AEI) under contracts PCI2019-111850-2 and PCI2019-111851-2, and the Generalitat de Catalunya (2017 SGR-990).

References

- [1] A. Miaoudakis, N. Petroulakis, I. Askoxylakis, Communications in emergency and crisis situations, in: International Conference on Distributed, Ambient, and Pervasive Interactions, Vol. 8530, 2014, pp. 555–565. doi:10.1007/978-3-319-07788-8_51.
- [2] J. C. Araneda, H. Rudnick, S. Mocarquer, P. Miquel, Lessons from the 2010 Chilean earthquake and its impact on electricity supply, in: 2010 International Conference on Power System Technology, 2010, pp. 1–7.
- [3] P. S. Paul, K. Hazra, S. Saha, S. Nandi, S. Chakraborty, S. Das, Generating crisis maps for large-scale disasters: Issues and challenges**this publication is an outcome of the r&d work undertaken in the itra project of media lab asia entitled. “post-disaster situation analysis and resource management using delay-tolerant peer-to-peer wireless networks (disarm)”., in: D. Câmara, N. Nikaen (Eds.), *Wireless Public Safety Networks 3*, Elsevier, 2017, pp. 67 – 98. doi:https://doi.org/10.1016/B978-1-78548-053-9_50004-4. URL <http://www.sciencedirect.com/science/article/pii/B9781785480539500044>
- [4] R. M. Santos, J. Orozco, S. F. Ochoa, R. Meseguer, D. Mosse, Providing real-time message delivery on opportunistic networks, *IEEE Access* 6 (2018) 40696–40712. doi:10.1109/ACCESS.2018.2848546.
- [5] D. Fischer, O. Posega, K. Fischbach, Communication barriers in crisis management: A literature review, in: *ECIS 2016 Proceedings*, 2016, pp. 1–19.
- [6] T. Girard, Barriers to communicating disaster response information to the public during disaster situations, in: PhD Thesis, Karlsruhe Institute of Technology - KIT, 2017, pp. 1–161.
- [7] Y. Murayama, K. Yamamoto, Research on disaster communications, in: *IFIP Advances in Information and Communication Technology*, vol 516. Springer, Cham, 2019, pp. 1–11.
- [8] M. L. Spialek, J. B. Houston, The influence of citizen disaster communication on perceptions of neighborhood belonging and community resilience, *Journal of Applied Communication Research* 47 (1) (2019) 1–23. doi:10.1080/00909882.2018.1544718.
- [9] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, M. Viroli, Modelling and simulation of Opportunistic IoT Services with Aggregate Computing, *Future Generation Computer Systems* (2019). doi:10.1016/j.future.2018.09.005.
- [10] M. Conti, A. Passarella, S. K. Das, The internet of people (iop): A new wave in pervasive mobile computing, *Pervasive and Mobile Computing* 41 (2017) 1 – 27. doi:https://doi.org/10.1016/j.pmcj.2017.07.009. URL <http://www.sciencedirect.com/science/article/pii/S1574119217303723>
- [11] R. Pueyo Centelles, F. Freitag, R. Meseguer, L. Navarro, S. F. Ochoa, R. M. Santos, A lora-based communication system for coordinated response in an earthquake aftermath, *Proceedings of the 13th International Conference on Ubiquitous Computing and Ambient Intelligence* 2019 31 (1) (2019). doi:10.3390/proceedings2019031073.
- [12] H.-C. Jang, Y.-N. Lien, T.-C. Tsai, Rescue information system for earthquake disasters based on manet emergency communication platform, in: *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, IWCMC '09*, Association for Computing Machinery, New York, NY, USA, 2009, p. 623–627. doi:10.1145/1582379.1582514.
- [13] S. F. Ochoa, R. Santos, Human-centric wireless sensor networks to improve information availability during urban search and rescue activities, *Information Fusion* 22 (2015) 71 – 84. doi:10.1016/j.inffus.2013.05.009.
- [14] B. Manoj, A. H. Baker, Communication challenges in emergency response, *Commun. ACM* 50 (3) (2007) 51–53. doi:10.1145/1226736.1226765.
- [15] R. Sugihara, R. K. Gupta, Path planning of data mules in sensor networks, *ACM Trans. Sen. Netw.* 8 (1) (Aug. 2011). doi:10.1145/1993042.1993043.

- [16] A. Martín-Campillo, J. Crowcroft, E. Yoneki, R. Martí, Evaluating opportunistic networks in disaster scenarios, *J. Netw. Comput. Appl.* 36 (2) (2013) 870–880. doi:10.1016/j.jnca.2012.11.001.
- [17] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou, N. Bessis, A survey on multihop ad hoc networks for disaster response scenarios, *International Journal of Distributed Sensor Networks* 11 (10) (2015) 647037. doi:10.1155/2015/647037.
- [18] J. Haxhibeqiri, E. De Poorter, I. Moerman, J. Hoebeke, A survey of lorawan for iot: From technology to application, *Sensors* 18 (11) (2018). doi:10.3390/s18113995.
- [19] L. Sciuillo, F. Fossemo, A. Trotta, M. Di Felice, Locate: A lora-based mobile emergency management system, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–7. doi:10.1109/GLOCOM.2018.8647177.
- [20] L. Sciuillo, A. Trotta, M. D. Felice, Design and performance evaluation of a LoRa-based mobile emergency management system (LOCATE), *Ad Hoc Networks* (2020). doi:10.1016/j.adhoc.2019.101993.
- [21] J. Orozco, R. Santos, S. F. Ochoa, R. Meseguer, A stochastic approach for modeling message dissemination in opportunistic networks, *Wireless Personal Communications* 97 (2) (2017) 2207–2228.
- [22] J. C. Liando, A. Gamage, A. W. Tengourtius, M. Li, Known and unknown facts of LoRa: Experiences from a large-scale measurement study, *ACM Transactions on Sensor Networks* 15 (2) (2019) 1–35. doi:10.1145/3293534.
- [23] O. Georgiou, U. Raza, Low power wide area network analysis: Can lora scale?, *IEEE Wireless Communications Letters* 6 (2) (2017) 162–165. doi:10.1109/LWC.2016.2647247.
- [24] D. Patel, M. Won, Experimental study on low power wide area networks (lpwan) for mobile internet of things, in: 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), 2017, pp. 1–5. doi:10.1109/VTCSpring.2017.8108501.
- [25] D. Magrin, M. Centenaro, L. Vangelista, Performance evaluation of lora networks in a smart city scenario, in: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–7. doi:10.1109/ICC.2017.7996384.
- [26] N. Benkahla, H. Tounsi, Y. Q. Song, M. Frikha, Enhanced ADR for LoRaWAN networks with mobility, in: International Wireless Communications and Mobile Computing Conference (IWCMC), 2019, pp. 1–6. doi:10.1109/IWCMC.2019.8766738.
- [27] J. J. Kang, I. Khodasevych, S. Adibi, A disaster recovery system for location identification-based low power wide area networks (LPWAN), in: International Telecommunication Networks and Applications Conference (ITNAC), Vol. 2017-January, IEEE, 2017, pp. 1–6. doi:10.1109/ATNAC.2017.8215359.
URL <http://ieeexplore.ieee.org/document/8215359/>
- [28] R. Lahouli, M. H. Chaudhary, S. Basak, B. Scheers, Tracking of rescue workers in harsh indoor and outdoor environments, in: International Conference on Ad-Hoc Networks and Wireless, Springer, Cham, 2019, pp. 48–61. doi:10.1007/978-3-030-31831-4_4.
URL http://link.springer.com/10.1007/978-3-030-31831-4_4
- [29] H.-C. Lee, K.-H. Ke, Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation, *IEEE Transactions on Instrumentation and Measurement* 67 (9) (2018) 2177–2187. doi:10.1109/TIM.2018.2814082.
- [30] J. Haxhibeqiri, I. Moerman, J. Hoebeke, Low overhead scheduling of LoRa transmissions for improved scalability, *IEEE Internet of Things Journal* 6 (2) (2019) 3097–3109. doi:10.1109/JIOT.2018.2878942.
- [31] J. Ortin, M. Cesana, A. Redondi, Augmenting lorawan performance with listen before talk, *IEEE Transactions on Wireless Communications* 18 (6) (2019) 3113–3128. doi:10.1109/TWC.2019.2910512.
- [32] C. Gomez, J. Crowcroft, Multimodal Retransmission Timer for LPWAN, *IEEE Internet of Things Journal* (2020) 1–1doi:10.1109/JIOT.2020.2971609.
- [33] G. Smart, N. Deligiannis, R. Surace, V. Loscri, G. Fortino, Y. Andreopoulos, Decentralized Time-Synchronized Channel Swapping for Ad Hoc Wireless Networks, *IEEE Transactions on Vehicular Technology* 65 (10) (2016) 8538–8553. doi:10.1109/TVT.2015.2509861.
URL <http://ieeexplore.ieee.org/document/7360239/>
- [34] C. G. Ramirez, A. Sergeyev, A. Dyussenova, B. Iannucci, Longshot: Long-range synchronization of time, in: Information Processing in Sensor Networks (IPSN), 2019, pp. 289–300. doi:10.1145/3302506.3310408.
- [35] J. B. Houston, J. Hawthorne, M. F. Perreault, E. H. Park, M. Goldstein Hode, M. R. Halliwell, S. E. Turner McGowen, R. Davis, S. Vaid, J. A. McElderry, S. A. Griffith, Social media and disasters: a functional framework for social media use in disaster planning, response, and research, *Disasters* 39 (1) (2015) 1–22.
- [36] M. T. Niles, B. F. Emery, A. J. Reagan, P. S. Dodds, C. M. Danforth, Social media usage patterns during natural hazards, *PLOS ONE* 14 (2019) 1–16.
- [37] K. Eismann, O. Posegga, K. Fischbach, Decision making in emergency management: The role of social media, in: ECIS 2018 Proceedings, 2018, pp. 2–20.
- [38] N. Howard, E. Cambria, Intention awareness: improving upon situation awareness in human-centric environments, *Human-centric Computing and Information Sciences* 3 (9) (2013) 1–17. doi:10.1186/2192-1962-3-9.
- [39] T. Hossmann, F. Legendre, P. Carta, P. Gunningberg, C. Rohner, Twitter in disaster mode: Opportunistic communication and distribution of sensor data in emergencies, in: Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition, ExtremeCom '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 1–6. doi:10.1145/2414393.2414394.
- [40] C. Reuter, Communication between power blackout and mobile network overload, *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)* (accepted) 6 (01 2014). doi:10.4018/ijisram.2014040103.
- [41] Semtech, Lora technology overview, Accessed: 2020-01-22 (2020).
URL <https://www.semtech.com/lora>
- [42] A. Augustin, J. Yi, T. Clausen, W. M. Townsley, A study of lora: Long range & low power networks for the internet of things, *Sensors* 16 (9) (2016). doi:10.3390/s16091466.
- [43] LoRa Alliance Technical Committee, Lorawan 1.1 specification, Accessed: 2020-01-22 (2017).
URL <http://lora-alliance.org/lorawan-for-developers>
- [44] L. Cerdà-Alabern, A. Neumann, P. Eschrich, Experimental evaluation of a wireless community mesh network, in: Proceedings of the 16th ACM International Conference on Modeling, Analysis & Simulation of Wireless and Mobile Systems, MSWiM '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 23–30. doi:10.1145/2507924.2507960.

- URL <https://doi.org/10.1145/2507924.2507960>
- [45] Chilean National Statistics Institute (INE), Chile census, Accessed: 2020-01-22 (2017).
URL <https://www.censo2017.cl>
- [46] Omnet++ discrete event simulator, Accessed: 2020-01-22 (2020).
URL <https://omnetpp.org>
- [47] G. Slabicki, M. Preamsankar, Flora—a framework for lora simulations, Accessed: 2020-01-22 (2018).
URL <https://flora.aalto.fi/>
- [48] M. Slabicki, G. Preamsankar, 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.
- [49] J. Petajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, 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), 2015, pp. 55–59. doi:10.1109/ITST.2015.7377400.
- [50] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, T. Watteyne, Understanding the limits of lorawan, IEEE Communications Magazine 55 (9) (2017) 34–40.