Departament d'Arquitectura
de Computadors
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Towards LoRa mesh networks for the IoT

Roger Pueyo Centelles

A Doctoral Thesis submitted to
the Departament d'Arquitectura de Computadors (DAC) of
the Universitat Politècnica de Catalunya (UPC)

Advisors: Roc Meseguer Pallarès and
Felix Freitag

September 27, 2021

Distributed Systems Group (DSG)
Barcelona, Catalunya

Towards LoRa mesh networks for the IoT

September 27, 2021

Roger Pueyo Centelles
roger.pueyo@upc.edu
rpueyo@ac.upc.edu

Universitat Politècnica de Catalunya (UPC)

Distributed Systems Group (DSG)
Barcelona, Catalunya

[1] TDX: `http://www.tdx.cat`, CBUC: `http://ccuc.cbuc.cat`, CSUC: `http://www.csuc.cat`, NDLTD: `http://www.ndltd.org`

Abstract

There are several LPWAN radio technologies providing wireless communication to the billions of connected devices that form the so-called IoT. Among them, LoRa has emerged in recent years as a popular solution for low power embedded devices to transmit data at long distances on a reduced energy budget.

Most often, LoRa is used as the physical layer of LoRaWAN, an open standard that defines a MAC layer and specifies the star-of-stars topology, operation, roles, and mechanisms for an integrated, full-stack IoT architecture. Nowadays, millions of devices use LoRaWAN networks in all sorts of agriculture, smart cities and buildings, industry, logistics, and utilities scenarios.

Despite its success in all sorts of IoT domains and environments, there are still use cases that would benefit from more flexible network topologies than LoRaWAN's star-of-stars. For instance, in scenarios where the deployment and operation of the backbone network infrastructure is technically or economically challenging, a more flexible model may improve certain performance metrics. As a first major contribution, this thesis investigates the effects of adding multi-hop capability to LoRaWAN, by means of the realistic use case of a communication system based on this architecture that provides a coordinated response in the aftermath of natural disasters like an earthquake. The capacity of end nodes to forward packets and perform multi-hop transmissions is explored, as a strategy to overcome gateway infrastructure failures, and analyzed for challenges, benefits and drawbacks in a massive system with thousands of devices.

LoRa is also used as a stand-alone radio technology, independently of the LoRaWAN architecture. Its CSS modulation offers many advantages in LPWANs for IoT deployments. In particular, its different SFs available determine a trade-off between transmission time (i.e., data rate) and sensitivity (i.e., distance reach), and also generate quasi-orthogonal signals that can be demodulated concurrently by different receivers. The second major contribution of this thesis is the design of a minimalistic distance-vector routing protocol for embedded IoT devices featuring a LoRa transceiver, and the proposal of a path cost calculation metric that takes advantage of the multi-SF capability to reduce end-to-end transmission time. The protocol is evaluated through simulation and compared with other well-known routing strategies, analyzing and discussing its suitability for heterogeneous IoT LoRa mesh networks.

Keywords *IoT, LoRa, LoRaWAN, mesh, multi-hop, routing*

Resum

Hi ha diverses tecnologies de ràdio LPWAN que proporcionen comunicació sense fils als milers de milions de dispositius connectats que conformen l'anomenada IoT. D'entre elles, LoRa ha emergit en els darrers anys com una solució popular per a què dispositius encastats amb pocs recursos transmetin dades a llargues distàncies amb un cost energètic reduït.

Tot sovint, LoRa s'empra com la capa física de LoRaWAN, un estàndard obert que defineix una capa MAC i que especifica la topologia en estrella d'estrelles, l'operació, els rols i els mecanismes per implementar una arquitectura de la IoT integrada. Avui dia, milions de dispositius fan servir xarxes LoRaWAN en escenaris d'agricultura, edificis i ciutats intel·ligents, indústria, logística i subministraments.

Malgrat el seu èxit en tota mena d'entorns i àmbits de la IoT, encara romanen casos d'ús que es beneficiarien de topologies de xarxa més flexibles que l'estrella d'estrelles de LoRaWAN. Per exemple, en escenaris on el desplegament i l'operació de la infraestructura troncal de xarxa és tècnicament o econòmica inviable, una topologia més flexible podria millorar certs aspectes del rendiment. Com a primera contribució principal, en aquesta tesi s'investiguen els efectes d'afegir capacitat de transmissió multisalt a LoRaWAN, mitjançant el cas d'ús realista d'un sistema de comunicació, basat en aquesta arquitectura, per proporcionar una resposta coordinada en els moments posteriors a desastres naturals, tals com un terratrèmol. En concret, s'explora l'estratègia d'afegir la capacitat de reenviar paquets als nodes finals per tal d'eludir les fallades en la infraestructura, i se n'analitzen els reptes, beneficis i inconvenients per a un sistema massiu amb milers de dispositius

LoRa s'empra també com a tecnologia de ràdio de forma autònoma, independentment de l'arquitectura LoRaWAN. La seva modulació CSS li confereix molts avantatges en xarxes LPWAN per a desplegaments de la IoT. En particular, els diferents SFs disponibles hi determinen un compromís entre la durada de les transmissions (i.e., la taxa de dades) i la sensibilitat en la recepció (i.e., l'abast en distància), alhora que generen senyals quasiortogonals que poden ser desmodulades de forma concurrent per receptors diferents. La segona contribució principal d'aquesta tesi és el disseny d'un protocol d'encaminament dinàmic vector-distància per a dispositius de la IoT encastats amb un transceptor LoRa, i la proposta d'una mètrica per calcular el cost d'un camí que aprofita la capacitat multi-SF per minimitzar el temps de transmissió d'extrem a extrem. El protocol és avaluat mitjançant simulacions i comparat amb altres estratègies d'encaminament conegudes, analitzant la seva conveniència per a xarxes LoRa mallades per a la IoT.

Paraules clau *IoT, LoRa, LoRaWAN, malla, multisalt, encaminament*

Resumen

Existen varias tecnologías de radio LPWAN que proporcionan comunicación inalám brica a los miles de millones de dispositivos conectados que forman el llamado IoT. De entre ellas, LoRa ha emergido en los últimos años como una solución popular para que dispositivos embebidos con pocos recursos transmitan datos a largas distancias con un coste energético reducido.

Habitualmente, LoRa se usa como la capa física de LoRaWAN, un estándar abierto que define una capa MAC y que especifica la topología en estrella de estrellas, la operación, los roles y los mecanismos para implantar una arquitectura del IoT integrada. A día de hoy, millones de dispositivos utilizan redes LoRaWAN en escenarios de agricultura, edificios y ciudades inteligentes, industria, logística y suministros.

A pesar de su éxito en todo tipo de entornos y ámbitos del IoT, existen casos de uso que se beneficiarían de topologías de red más flexibles que la estrella de estrellas de LoRaWAN. Por ejemplo, en escenarios en los que el despliegue y la operación de la infraestructura troncal de red es técnica o económicamente inviable, una topología más flexible podría mejorar ciertos aspectos del rendimiento. Como primera contribución principal, en esta tesis se investigan los efectos de añadir capacidad de transmisión multi-salto a LoRaWAN, mediante el caso de uso realista de un sistema de comunicación basado en dicha arquitectura, para proporcionar una respuesta coordinada en los momentos posteriores a desastres naturales, tales como un terremoto. En concreto, se explora la estrategia de añadir la capacidad de reenviar paquetes a los nodos finales para sortear las fallas en la infraestructura, y se analizan los retos, beneficios e inconvenientes para un sistema masivo con miles de dispositivos.

LoRa se usa también como tecnología de radio de forma autónoma, independientemente de la arquitectura LoRaWAN. Su modulación CSS le confiere muchas ventajas en redes LPWAN para despliegues de IoT. En particular, los distintos SFs disponibles determinan un compromiso entre la duración de las transmisiones (i.e., la tasa de datos) y la sensibilidad en la recepción (i.e., el alcance en distancia), a la vez que generan señales cuasi-ortogonales que pueden ser desmoduladas de forma concurrente por receptores distintos. En segundo lugar, esta tesis contiene el diseño de un protocolo de enrutamiento dinámico vector-distancia para dispositivos Internet of Things (IoT) embebidos con un transceptor LoRa, y propone una métrica para calcular el coste de un camino que aprovecha la capacidad multi-SF para minimizar el tiempo de transmisión de extremo a extremo. El protocolo es evaluado y comparado con otras estrategias de enrutamiento conocidas, analizando su conveniencia para redes LoRa malladas para el IoT.

Palabras clave *IoT, LoRa, LoRaWAN, malla, multi-salto, enrutamiento*

Acknowledgments

The fulfillment of this thesis was possible with the support of several people, to whom I want to express my most sincere gratitude.

First of all, I would like to thank my advisors –in no particular order, other than chronological– Leandro Navarro, Felix Freitag and Roc Meseguer. At every step of this work, they provided their guidance, offered their expertise, and helped with resolution to push it a bit further.

I would also like to thank the reviewers of this document –alphabetically– Cristina Cano Bastidas, Jorge García Vidal and José M. Barceló Ordinas, for their valuable feedback and suggestions.

The work on this thesis was easier to handle with the experience and support from the colleagues and friends at the C6, Emmanouil Dimogerontakis, Mennan Selimi and Khulan Batbayar, with a special mention to Roger Baig, who provided certainty and clarity with his mentorship.

Last, but not least, on a more personal side, I would like to thank my partner Laia, my parents, family and friends, for the support through these years. I know you will be proud of seeing this result achieved.

This work was partially funded by the Spanish government under contracts TIN2016-77836-C2-2-R and BES-2017-082490, the Spanish State Research Agency (AEI) under contracts PID2019-106774RB-C21, PCI2019-111850-2 and PCI2019-111851-2, and the Generalitat de Catalunya under contract 2017 SGR-990.

Contents

List of Figures

List of Tables

List of Acronyms

Chapter 1

Introduction

A number of Low Power Wide Area Network (LPWAN) radio technologies have emerged during the last decade providing wireless communications for IoT devices in a variety of scenarios [1, 2]. Most often, they operate in the sub-GHz region of the spectrum, using either licensed or unlicensed Industrial, Scientific, Medical (ISM) bands, where their great link budget allows nodes to transmit over several km while keeping power usage low.

LoRa, which stands for *long range*, is an LPWAN wireless communication technology owned by Semtech [3]. It employs Chirp Spread Spectrum (CSS), a proprietary modulation technique resistant to multi-path fading and suitable for noisy environments, aiming to provide low throughput communication for links of more than 10 km –outdoors, in rural areas– while maintaining power consumption low [4]. It can be used stand-alone, but most of the time it is used as part of the LoRaWAN architecture.

LoRaWAN is an open standard promoted by the LoRa Alliance that adds the Medium Access Control (MAC) layer and specifies an LPWAN protocol on top of LoRa, which provides the underlying radio communication layer. It is designed to offer secure wireless bidirectional communication between *end nodes* (i.e., tiny, resource-constrained, battery-operated IoT devices that collect data) and the application that processes and manages the data. LoRaWAN-based systems are built with a *star* or *star-of-stars* topology, where a number of end nodes upload data wirelessly to one or more gateways in *single-hop transmissions*. Then, these gateways relay the messages to an application server, that typically resides in the cloud, for data processing and decision-making.

1.1 Problem statement

The LoRa technology, especially as part of the LoRaWAN architecture, is widely adopted in a large number of IoT use cases, covering the agriculture, smart buildings and cities, industry, logistics and utilities domains, among others. While the star-of-stars topology defined by LoRaWAN is well-suited for different applications and environments, it imposes an unbalanced *nodes* $\rightarrow$ *gateway* $\rightarrow$ *network server/cloud* data path and hierarchy where some IoT scenarios do not fit well [5]. Furthermore, the radio reach of the gateways defines the network's coverage area, regardless of any possible direct links that could be established between end nodes.

These use cases that do not completely fit into the centralized LoRaWAN architecture could benefit from more flexible network topologies. Some applications could be deployed with lower ownership costs by adopting network models that allow multi-hop packet transmission between end nodes, either reach another node or a distant gateway. This could be ultimately leveraged by infrastructure-less and decentralized IoT systems that need to distribute data among nodes scattered over large areas, connected by a low-power mesh network, to perform computations at the edge [6].

1.1.1 Network coverage beyond the gateway infrastructure reach

While LoRa transmissions can cover long distances, it is difficult to provide LoRaWAN coverage for moving elements (e.g., vehicle fleets) circulating through vast areas (e.g., the open sea). There, deploying gateways can be economically or technically challenging. In these environments, multi-hop packet transmission between end nodes could facilitate communication with gateways beyond their coverage.

Figure 1.1 depicts a realistic use case where a fleet of ships from a fishing guild needs to be tracked beyond the coverage of a mainland gateway. Although technically possible, deploying additional gateways is not a feasible option for the guild. However, the usage of multi-hop message transmission can mitigate the effect of dark coverage areas and increase the network operation range. By leveraging the encounters of mobile elements and the new transmission paths they create, vessels closer to the shoreline can contribute to extend the gateway's coverage further offshore.

Similarly, in the aftermath of natural disasters, communication systems often experience downtime periods due to damage to the infrastructure (e.g., base stations) in the affected area. Under these conditions, the gateways of an IoT network (including, for example, LoRaWAN) can become single points of failure if they cease to operate, rendering part of the end nodes out of service, or even the whole of them. Direct communication between end nodes could be used to circumvent the failing infrastructure.

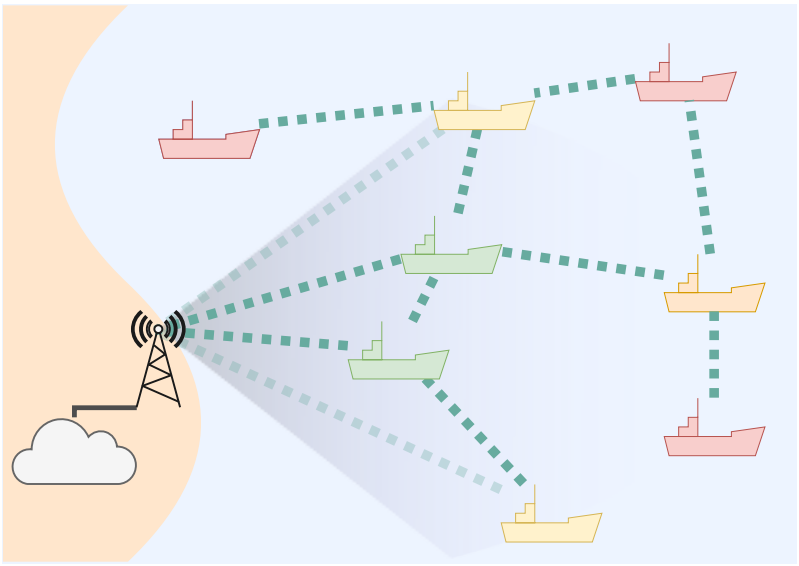


Figure 1.1: A LoRaWAN-based use case where multi-hop could help to extend the gateway coverage. The orange and red ships are out of the gateway's reach, but their messages could be relayed to the gateway by one of the green ships. Additionally, yellow ships could achieve better packet delivery ratios.

Figure 1.2 shows the architecture of a communication system for the aftermath of an earthquake, which is further developed in Chapter 4. In this envisioned scenario, people have small battery-powered devices equipped with a LoRa transceiver at their homes and workplaces. In case of emergency, they can use these devices to report their status to rescue teams or send short messages to their relatives. Gateways spread over the city receive the messages and forward them to the emergency coordination workforce. If the LoRaWAN gateways become isolated or inoperative by the adverse conditions, user nodes could take a proactive role in the system and help to route packets from and to those nodes affected by infrastructure failures.

The former and the later scenarios have fundamental differences, in terms of dimensions, mobility of nodes, predictability of the infrastructure behavior, etc. However, they share the common need of extending the gateway infrastructure's coverage beyond their physical reach.

1.1.2 Decentralized mesh networking for infrastructure-less systems

Obtaining metering data in the field is a slow, labor-intensive, and expensive task. Even if many cities worldwide, and even entire countries, have digitized readings from public utilities, etc., using wireless or Power Line Communications (PLC),

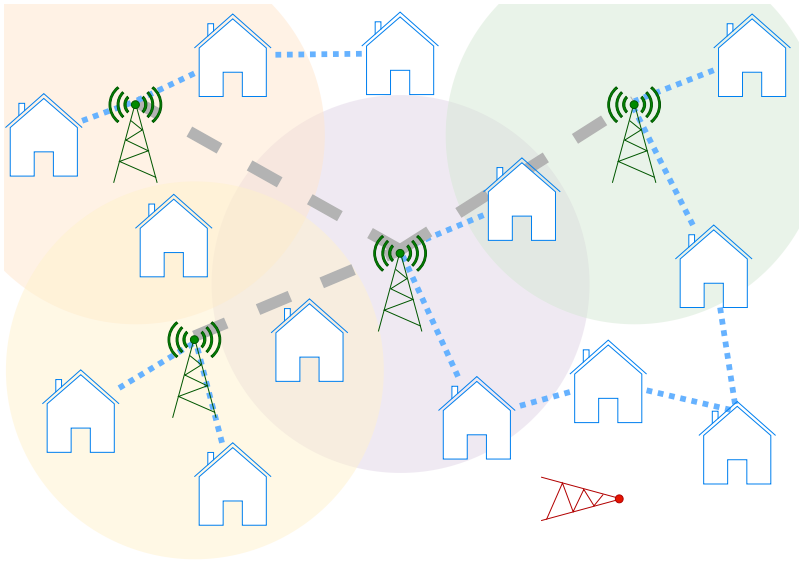


Figure 1.2: After a natural disaster, like an earthquake, central infrastructures may become unavailable. If end nodes have the ability to communicate between them, messages can still be forwarded towards the remaining data sinks –and back–.

LPWAN wireless technologies, like LoRa, can be used to facilitate this process in underserved areas. Radio transceivers can be embedded into sensing, metering or actuating devices for this purpose. Applications in this domain could also adopt mesh topologies instead of the gateway-centric model. End nodes could relay messages from one metering device to another until they reach a data sink, and also in the opposite direction. This can be of particular interest in deployments with low density of nodes, where a low nodes/gateway ratio may be economically impractical.

Figure 1.3 shows a realistic use case: a water quality monitoring system that spreads over a vast, remote mountainous region. Currently, manual intervention is required to register measurements at different locations, such as isolated settlements and farms. Due to insufficient telecommunication infrastructure, obtaining and reporting the data is time-consuming and expensive. The measurements are being automated to improve management efficiency, but the apparatus must be linked to a communication system that can report data to distant facilities. The LoRaWAN architecture does not fit this scenario, as the size of the deployment and the mountainous geography would require an elevated number of gateways. Nevertheless, as most of the nodes are in line of sight with other nodes, a multi-hop solution could instead be employed, forwarding packets between the different settlements until they reach the management facilities.

In summary, there are applications and use cases that do not completely fit under the single-hop, gateway-centric star-of-stars topology enforced by the LoRaWAN

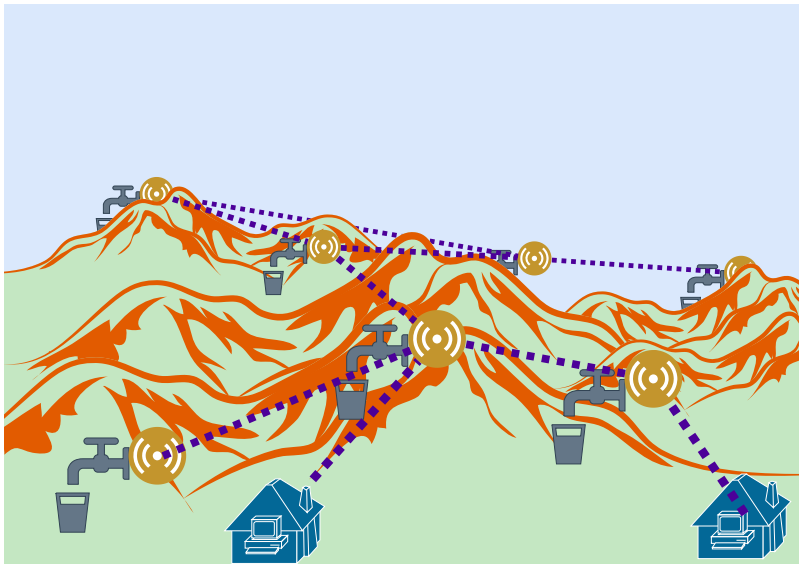


Figure 1.3: Sparsely-distributed water quality sensors can not be easily covered by a LoRaWAN gateway. However, by leveraging multi-hop communications, data can be relayed node to node until reported to the management facilities.

architecture, or that require a totally decentralized network topology that better matches their distributed nature.

1.2 Research questions

The analysis of different IoT scenarios and use cases has risen awareness regarding the applications that do not completely fit into the centralized star-of-stars, or that require a decentralized mesh topology. In the search for more flexible networking models based on the LoRa radio technology and the LoRaWAN architecture, the work presented in this thesis tries to answer the following Research Questions (RQs):

RQ1 Are topologies other than LoRaWAN's star-of-stars feasible for LoRa IoT networks?

In particular,

RQ2 How does packet forwarding between end nodes improve the different performance aspects of a LoRaWAN network?

RQ3 How can the specific features of the LoRa radio technology be used to build flat, decentralized mesh networks and provide communication between nodes of an IoT system?

RQ4 What is the performance of LoRa mesh networks, according to Key Performance Indicators (KPIs) such as scalability, reliability, throughput, latency, energy-awareness, etc.?

A positive reply to the first part of RQ1 was a prerequisite for the subsequent RQs to be worked on. This was experimentally validated at an early stage of the research, and corroborated by the related work on the topic by other authors.

1.3 Objectives

This investigation aims at providing specific solutions to answer the RQs above. In relation to them, the objectives of the work here presented are the following:

Obj1 To validate, from the technical point of view, the feasibility of using the LoRa radio technology to build a wireless mesh network for IoT systems.

Obj2 To investigate on the suitability of adding multi-hop capacity to LoRaWAN networks, with the goal of extending the gateway infrastructure's coverage, or overcome their eventual failures.

Obj3 To design, validate and evaluate a Routing Protocol (RP) for decentralized LoRa mesh networks by means of theoretical analysis, simulations and experimental deployments.

Obj4 To analyze the performance metrics of LoRa mesh networks of different conditions and scenarios (size, topology, traffic, etc.).

1.4 Contributions

The main contributions of this thesis are summarized in the following, linked to their corresponding chapters of this document and our publications in peer-reviewed magazines.

1.4.1 Survey and analysis of the state-of the art

Several authors have proposed solutions to extend LoRaWAN networks' coverage by means of multi-hop, or to build mesh networks based on the LoRa radio technology.

Chapter 3 contains an exhaustive compilation, classification, and analysis of these related works, answering RQ1 and accomplishing Obj1.

The most relevant findings of this survey were published in [MP1].

1.4.2 Evaluation of a large-scale LoRaWAN network with packet forwarding between end nodes

LoRaWAN networks are used in many applications in the agriculture, smart buildings and cities, industry, logistics, and utilities domains. They have been thoroughly analyzed and evaluated, from the theoretical and practical point of view, either in simulations or with real hardware. Their potential can also be leveraged in emergency communication networks for the aftermath of natural disasters, such as earthquakes, due to their capacity to operate on batteries for extended periods of time. However, in such critical scenarios, where the risk of network infrastructure failure is not negligible, finding alternative transmission mechanisms for nodes that may end underserved could make a significant difference. For this reason, we considered adding packet forwarding capacity to end nodes in a LoRaWAN system, in order to overcome potential infrastructure problems that would render part of the devices unconnected.

In Chapter 4 we leveraged the realistic use case of a massive emergency communication system aimed at civilians in the aftermath of an earthquake based on the LoRaWAN architecture. By means of computer simulations, we analyzed its behavior and operation with several degrees of infrastructure degradation. We then added end nodes the capacity to forward packets from their neighbors, and analyzed different KPIs, finding it can help underserved nodes' packets reach their destination, answering to RQ2 and accomplishing Obj2.

The initial works in this topic were first reported in [CP1] and later extended in [MP2].

1.4.3 A minimalistic Distance-Vector Routing Protocol for LoRa mesh networks

Besides its successful integration in the LoRaWAN architecture, the LoRa radio technology can be used independently to provide the physical Layer 1 (L1) of a communication system. In particular, because of its great link budget and low energy consumption, it is well suited for the IoT domain, where applications usually require only small data rates.

The analysis of the state-of-the-art and real use cases revealed that certain IoT applications require a more flexible topology than the star-of-stars. The examples are manifold, but can be roughly group in two categories: either the area they cover is too large or too harsh to be covered by deploying LoRaWAN gateways (or it does not make sense from the technical or economical point of view), or either the data flow of the application does not match the typical source→sink flow. Given this, in Chapter 5 in we investigated the ability to build decentralized mesh networks using the end nodes' LoRa transceiver to enable direct communication between arbitrary

pairs of nodes, without the participation of the gateways' infrastructure. For that, we designed a minimalistic Distance-Vector (DV) RP that takes advantage of LoRa's specific features, such as the SFs, as an answer to RQ3 and in line with Obj3. To answer RQ4 we analyzed the operation of the protocol by means of different KPIs, as expected by Obj4.

The work in this chapter is submitted and under review as of September 2021 in [UR1].

1.5 Methodology

The successful achievement of this thesis' objectives is based on the adoption of the design science methodology for information systems research that combines research and design, and assessment of the obtained results by experimental methods [7]. For that, the work has been divided into 6 main steps that begin with *understanding the problem and investigating the context*, continue with *conceptualizing and materializing* the proposed solutions with the constant assistance of *theoretical analysis and empirical validation* as assessment methods for the *evaluation of the proposed solution*:

1. Problem statement
2. Preliminary experimentation
3. Analysis of the context and the state-of-the-art
4. Definition of the requirements
5. Design and implementation
6. Experimental evaluation

Figure 1.4 shows the steps of the work methodology as a set of interrelated steps. The thicker lines depict the main path from the problem statement towards the outcome of the work, across different main steps. Validation and assessment steps are tightly related to the implementation, and their contribution may affect it. Additionally, dotted lines indicate that the process is not linear, but that iterations in tasks will happen as feedback is provided between the steps.

1.6 Research limitations

There are two issues that have deliberately not been taken into account in this research: energy efficiency and duty cycle compliance.

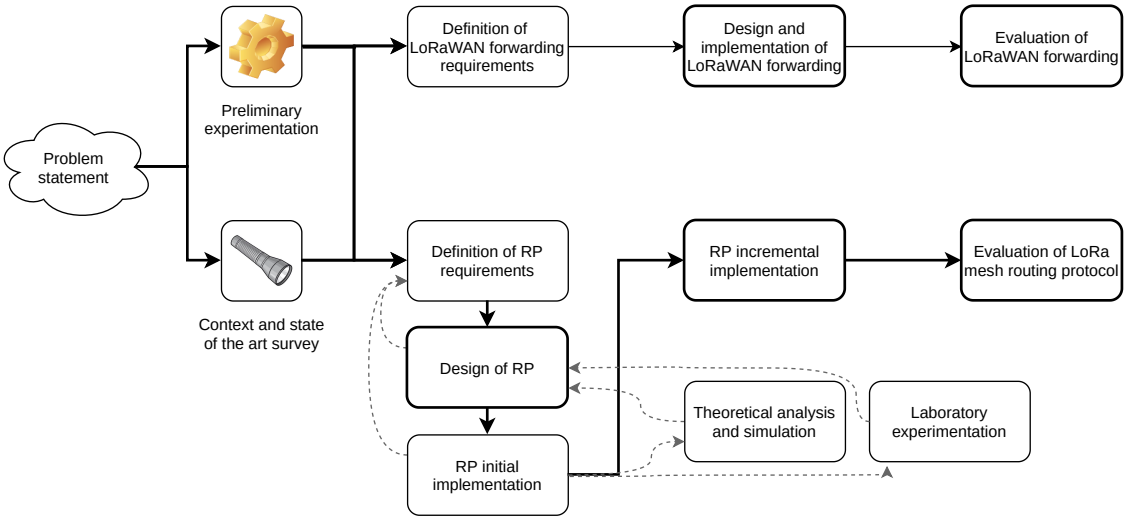


Figure 1.4: The work methodology as a succession of interrelated steps.

On the one hand, LoRa is a radio technology aimed at transmitting data at long distances with a small energy footprint, suitable for low power embedded devices. More specifically, the LoRaWAN architecture defines a series of classes for devices to adhere in their operation. Typically, end nodes conform to LoRaWAN's Class A devices operation, which allows for maximum energy efficiency as devices spend most of the time in deep sleep mode. However, adding forwarding capacity to LoRaWAN end nodes requires them to listen for the incoming transmissions from their neighbors, which avoids them entering deep sleep, incurring in higher energy consumption. Similarly, embedded LoRa devices running our RP are expected to send and receive packets at any moment in time, not making it possible for them to enter in deep sleep mode.

On the other hand, the sub-gigahertz ISM bands in which LoRa radios operate are typically affected by duty cycle restrictions. Depending on the local regulation, which may vary from country to country, a node may occupy the radio channel only a fraction of time, and remain silent otherwise to allow for other nodes to have a fair share of the radio spectrum. For example, for LoRaWAN nodes in Spain, the 868 MHz band has a 1 % duty cycle limitation (i.e., a node should stay silent 99 % of the time) and other restrictions on the maximum duration of a continuous transmission time, if no other mechanisms are provided (e.g., Listen-before-Talk (LBT)). Nonetheless, we designed our RP taking such restrictions into account, so that nodes can easily conform to these regulations. Furthermore, when performing laboratory experiments with real hardware, we used Radiofrequency (RF) attenuators to avoid generating interference to production systems in ISM bands.

Last, but not least, the performance evaluation of –first– a LoRaWAN network with end nodes capable of perform packet forwarding and –second– the DV RP for a

LoRa-based mesh network was performed by means of computer simulations (more information about the framework used is provided in Section 4.5.2). According to the published research and some benchmarks we conducted in the laboratory with embedded LoRa devices (details in Section 5.3.4), the simulator closely matches the behavior of real hardware. Despite this, there are certain aspects of the real world that the simulator may not be able to reproduce exactly, such as noise or radio interference from other networks, radio signal degradation due to complex multi-path propagation, or delays introduced by devices to handle radio packets, process data, compute routing tables, etc. Nonetheless, these aspects shall not be of enough importance to deem the simulations inaccurate.

1.7 List of publications

Journal publications

MP1 *Beyond the Star of Stars: An Introduction to Multihop and Mesh for LoRa and LoRaWAN* - **Roger Pueyo Centelles**, Felix Freitag, Roc Meseguer and Leandro Navarro. IEEE Pervasive Computing, 2021, Vol. 20, No. 2, Pp. 63-72. [JCR Q2, IF 3.175]

MP2 *LoRaMoto: A communication system to provide safety awareness among civilians after an earthquake* - **Roger Pueyo Centelles**, Roc Meseguer, Felix Freitag, Leandro Navarro, Sergio F. Ochoa, Rodrigo M. Santos. Future Generation Computer Systems, 2021, Vol. 115, Pp. 150-170. [JCR Q1, IF 7.187]

Under review

UR1 *A minimalistic distance-vector routing protocol for LoRa mesh networks* - **Roger Pueyo Centelles**, Roc Meseguer, Felix Freitag and Leandro Navarro. Submitted to the Journal of Network and Computer Applications [JCR Q1, IF 6.281].

Conference proceedings

CP1 *A LoRa-Based Communication System for Coordinated Response in an Earthquake Aftermath* - **Roger Pueyo Centelles**, Felix Freitag, Roc Meseguer, Leandro Navarro, Sergio F. Ochoa, and Rodrigo M. Santos. 13th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI), 2019.

CP2 *Evacuation Supporting System Based on IoT Components* - Gabriel M. Eggly, José M. Finochietto, Matías Micheletto, **Roger Pueyo Centelles**, Rodrigo Santos, Sergio F. Ochoa, Roc Meseguer, and Javier Orozco. 13th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI), 2019.

CP3 *Extending LoRa networks: dynamic routing protocols and sub-GHz radio technology for very long range mesh networks (student research abstract)* - **Roger Pueyo Centelles**. SAC '19, 34th ACM/SIGAPP Symposium on Applied Computing, 2019, Apr., Pp, 1396 – 1397. [CORE B]

Other publications

OP1 *An IoT-based infrastructure to enhance self-evacuations in natural hazardous events*

José M. Finochietto, Matías Micheletto, Gabriel M. Eggly, **Roger Pueyo Centelles**, Rodrigo Santos, Sergio F. Ochoa, Roc Meseguer and Javier Orozco. Personal and Ubiquitous Computing, 2021, Feb., Pp. 1-18. [JCR Q3, IF 2]

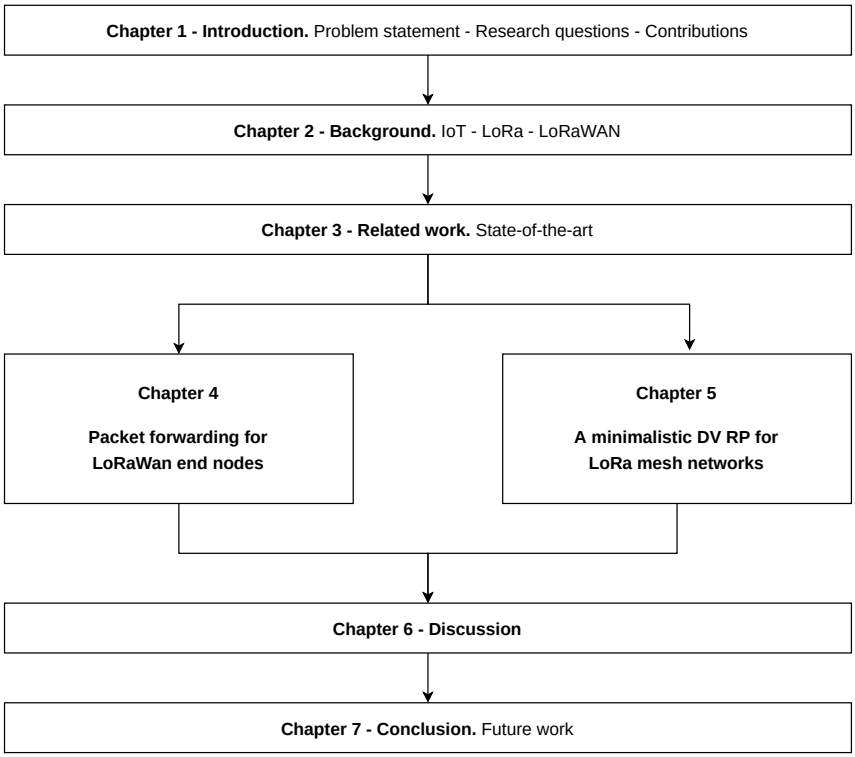


Figure 1.5: Structure of this thesis.

1.8 Structure of this document

Figure 1.8 describes the structure of this document. Following this introduction, Chapter 2 introduces the technical background for our research. The related work is exhaustively surveyed and analyzed in Chapter 3. After that, Chapters 4 and 5 contain the main work of this document, exploring packet forwarding between end nodes in LoRaWAN and decentralized mesh networks built with LoRa, respectively. Chapter 6 discusses the achievements of this thesis, as well as some common aspects that appear along the text, followed by a wrap-up conclusion in Chapter 7.

Chapter 2

Background

2.1 Wireless technologies for the IoT

With the advent of IoT and its ubiquity, several wireless communication technologies providing connectivity from few meters to several km on sub-GHz bands have appeared [1]. These LPWAN protocols are designed to connect battery-operated devices between them or to the Internet through a gateway. Among the different classification and comparison terms, we choose communication range and data rate to distinguish them. Figure 2.1 positions them on a graph, according to these properties, and groups them in three main categories: short range (less than 100 m), medium range (between 100 m and 1 km), and long range (more than 1 km). Additionally, the graph provides an approximate indication of the data rate each technology can deliver in the context of an IoT deployment.

Technologies and protocols like RFID, ZigBee, Bluetooth Low Energy and their upper layers, like the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) specification, are designed to provide short range links (from a few cm up to 100 m) at speeds between the kbps and the Mbps. Wi-Fi fits in both this and the following category, since the characteristics of the actual devices (antennae, transmission power, etc.) have a significant impact on their range.

The cellular-based 2G/3G/4G/5G evolution of the GSM technologies, as well as Wi-Fi, offer a medium communication range, with links between 100 m and 1 km and data rates well above the range of Mbps.

In the LPWAN group, Sigfox, Wize, DASH7, LoRa, nWave, Narrowband IoT (NB-IoT) and Weightless are examples of radio technologies designed to transmit data over *long-range* links, beyond 1 km, with a low power budget. Their long communication distance comes at the expense of low data rates, in the kbps order, or below. These characteristics fit many IoT applications where few data are transmitted (e.g., pe-

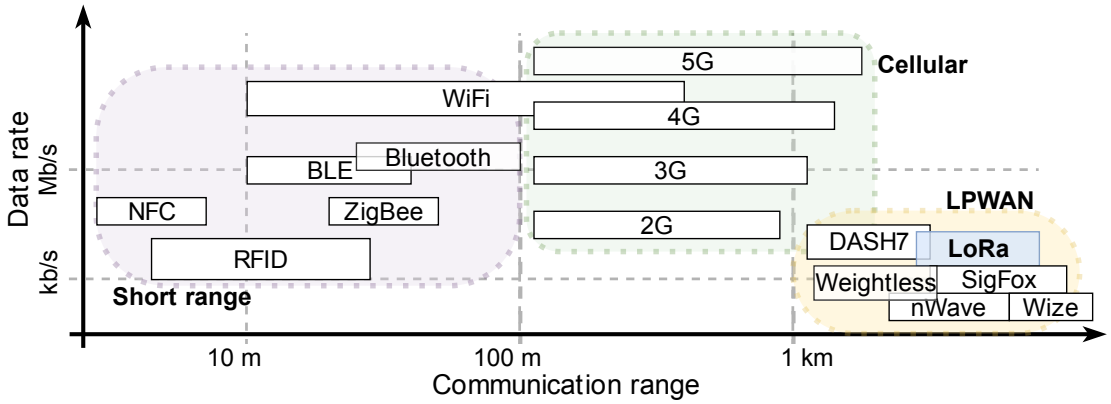


Figure 2.1: A non-exhaustive classification of radio technologies and protocols for IoT according to their communication range and data rate. LoRa provides long range links, of a few km, at up to tens of kbps.

riodical measurements) and power efficiency is required to reduce maintenance on battery-operated devices.

Sigfox is an LPWAN protocol and also a network operator that offers a global end-to-end IoT connectivity solution based on its infrastructure and technologies, with network coverage being provided by their proprietary base stations [1]. It uses Binary Phase-Shift Keying (BPSK) modulation to transmit in ultra-narrow 100 Hz channels inside unlicensed sub-GHz ISM bands. Up-link communication is limited to 140 messages per day, of maximum 12 B, while down-link communication is even more restricted, to 4 messages per day sized up to 8 B. To ensure delivery, time and frequency diversity are used, as well as duplication of transmissions.

NB-IoT was specified by the 3rd Generation Partnership Project (3GPP) project to operate in licensed bands along with Global System for Mobile communications (GSM) and Long-Term Evolution (LTE) cellular technologies [8]. It occupies 200 kHZ, and can be operated stand-alone, inside LTE guard bands or in-band. is possible in NB-IoT deployment. NB-IoT is based on the LTE protocol, reducing it to the minimum and enhancing it for IoT applications [1]. It employs Quadrature Phase-Shift Keying (QPSK) modulation, and Frequency-Division Multiple Access (FDMA) or Orthogonal Frequency-Division Multiple Access (OFDMA) for the up-link or down-link, respectively, with maximum data rates of 200 and 20 *kbps* in these directions, for a payload size of up to 1600 B.

Wize is another approach to wireless LPWAN, in this case originally aimed for industrial IoT applications [9]. The technology was first utilized and developed for gas and water metering in France. It is based on the wireless M-Bus standard at 169 MHz for utility meters, providing robust communication for hard-to-reach smart city objects over a secure data channel. Wize uses six $12,5$ kHz channels with data rates of up to $6,4$ kbps. Since it uses a lower frequency band, compared to other

technologies, it can provide better transmission range and building penetration. The technology has also been used as the physical layer for LoRaWAN.

DASH7 is another well-defined LPWAN, created by the DASH7 Alliance [10]. It is an open source wireless sensor and actuator network protocol that operates in the sub-GHz bands, based on the ISO/IEC 18000-7 standard, which defines the parameters of the active air interface communication at 433 MHz. It defines a complete communication stack from the physical level to the application layer. It uses 2-Gaussian Frequency-Shift Keying (GSFK) and up to 200 kbps (depending on the number of channels used), providing a long range (up to 2 Km), and low latency to connect moving objects in a tree topology with 2-hops simple routing.

Weightless is an open standard developed by the Weightless Special Interest Group [11]. The group initially developed three LPWAN technologies, namely Weightless-W, Weightless-N, and Weightless-P [2]. The former is aimed at leveraging the signal propagation properties of TV white spaces, while the two latter both use sub-gigahertz bands. On the one hand, weightless-N is oriented to lower power use and one-way communication, based on narrowband and Differential Phase-Shift Keying (DPSK) modulation. On the other hand, Weightless-P, provides packet acknowledgment and two-way communication, combining FDMA and Time-Division Multiple Access (TDMA) modulations.

Still other LPWAN technologies exist, although they seem not to be very popular, either because they are closed, proprietary solutions or because they are aimed at niche applications or environments. For example, nWave is an ultra narrowband communication protocol that has a focus on parking facilities in smart cities [12]. Ingenu proposed another proprietary LPWAN which operates in ISM 2.4 GHz, Telensa provides end-to-end solutions for LPWAN applications incorporating fully-designed vertical network stacks with ultra-narrow band transmissions in sub-GHz ISM bands, and Qowisio deploys mixed LPWAN that combine LoRa and ultra-narrow band communications [2].

Last, but not least, among all the abovementioned long range wireless options, LoRa is a *mature, worldwide-adopted, cheap and accessible* technology that operates in *license-exempt* ISM bands. This radio technology uses CSS modulation to achieve long range communication. On top of it, the open LoRaWAN architecture adds a MAC layer and specifies an LPWAN protocol to provide secure wireless bidirectional communications. These attributes make LoRa and LoRaWAN a very flexible solution, and therefore suitable for a wide range of scenarios and applications [1]. Details of the LoRa radio technology and the LoRaWAN architecture are further discussed below, in Sections 2.2 and 2.3, respectively.

Even if –unlike some of the above– LoRa is a proprietary radio technology, given its high availability and degree of adoption, it is relatively easy and affordable to own and deploy a private network with it (most often in the form of the LoRaWAN architecture), thus not depending on external infrastructure operators. This also

makes LoRa a convenient technology on which to perform research with, as it imposes no dependencies on third parties, allowing for a full ownership of the whole stack. On the contrary, protocols such as Sigfox or NB-IoT, are designed to work tightly coupled with gateways or base stations, which makes it difficult to use the underlying radio technologies stand-alone, and build additional layers or features on top of them (MAC, routing, application, etc.). Among those LPWAN technologies that offer the possibility to use them freely stand-alone and build higher layers upon (namely Wize, DASH7 and LoRa), the former is again the most convenient, due to its affordability and availability in the form of development boards or embedded devices ready to be programmed.

2.2 The LoRa radio technology

LoRa, which stands for *long range*, is a wireless communication technology owned by Semtech that operates in the sub-gigahertz range of the radio spectrum [3]. It employs CSS, a proprietary modulation technique resistant to multi-path fading and suitable for noisy environments, aiming to provide low throughput communication with links of more than 10 km –outdoors, in rural areas– while maintaining low power consumption. Its characteristics have been thoroughly analyzed [4, 13, 14].

Several parameters of the LoRa physical layer can be configured in order to optimize communications for a given scenario or application, or on a per-device basis: radio band and frequency, channel bandwidth, transmission power, Forward Error Correction (FEC) rate and SF. IoT deployments commonly operate on license-exempt ISM bands, which change from one geographic area to another; diverse LoRa transceivers are available to operate in any of them. Inside these bands, in turn, many channels are available for up-link and down-link communication, with thinner or narrower channel bandwidths and different maximum transmission powers allowed. In addition to the robust CSS modulation, LoRa's FEC provides protection against interference on noisy links. Table 2.1 summarizes the aforementioned configurable parameters and lists their possible values.

LoRa radio chip are available in two types: single-channel chips for end nodes and multichannel digital baseband chips for gateways (typically for LoRaWAN gateways). The former are simple radio transceivers able to send or receive on a single channel and a single SF at a time. These are typically found in devices such as sensors, which require sending and receiving limited amounts of data, and focus on simplicity and energy awareness. The latter are complex digital signal processing engines designed to operate as gateways, concentrating traffic from surrounding end nodes transmitting on different channels and different SFs at the same time. Gateway chips emulate tens of LoRa signal demodulators, and are capable of receiving up to 10 different transmissions at concurrently, as long as they use different com-

Configurable parameter	Values
Radio band and frequency	169, 433, 868, 915 MHz [a]
Bandwidth	62.5, 125, 250, 500 kHz [b]
Transmission power	14 dBm (EU), 27 dBm (USA)
Spreading Factor	6 to 12 [c]
FEC rate	4/5, 4/6, 4/7, 4/8

[a] Common frequency allocations for ISM in different regions worldwide; 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 may result in too long transmissions for specific radio bands in which duty cycle or time-on-air limitations often apply.

Table 2.1: Configurable parameters in LoRa transmissions.

binations of channel *and* SF (e.g., 4 simultaneous transmissions on, respectively, CH1-SF7, CH1-SF8, CH2-SF7 and CH2-SF8 can be correctly received).

There are several technical factors and features that make the LoRa technology especially suitable for IoT deployments. The SF is perhaps the most interesting parameter in LoRa, as it determines a direct trade-off between a link's *transmission speed* and *communication range*. Furthermore, two LoRa transmissions on the same channel using *different* SFs are quasi-orthogonal, meaning that both can be successfully demodulated by the respective receivers. This feature is actually leveraged by the gateway LoRa transceiver chips, but can also be taken advantage of by regular end nodes (granted, one channel and SF at a time).

Manufacturers currently provide LoRa chips and development boards based on different Microcontroller Units (MCUs) and Systems on a Chip (SoCs) with LoRa transceivers for prototyping and experimentation. These include popular development and Do-It-Yourself (DIY) boards (such as the Arduino [1] family and its clones) and devices based on microcontrollers and SoCs with a big open source community of users (like Espressif Systems' ESP8266 and ESP32 [2]) are readily available for less than 15 €, fostering rapid prototyping. Figure 2.2 shows a couple of them. However, encapsulated off-the-shelf devices integrating sensors and radio[3,4], ready for integration with the LoRaWAN architecture, are the most popular ones.

A drawback for LoRa (and, in fact, for any radio technology operating in the sub-GHz part of the spectrum) is the legal duty cycle limitations imposed in ISM bands, which only allows a device to transmit on a given channel for a maximum of percentage of time. Regulations for these deployments change from one geographic area

[1] Arduino - `https://www.arduino.cc/`

[2] Espressif's chips - `https://www.espressif.com/en/products/hardware/socs`

[3] CubeCell Capsule Sensor: `https://heltec.org/project/htcc-ac01/`

[4] Radio Bridge Wireless Air Temp and Humidity Sensor:
`https://radiobridge.com/products/air-temperature-and-humidity-sensor/`

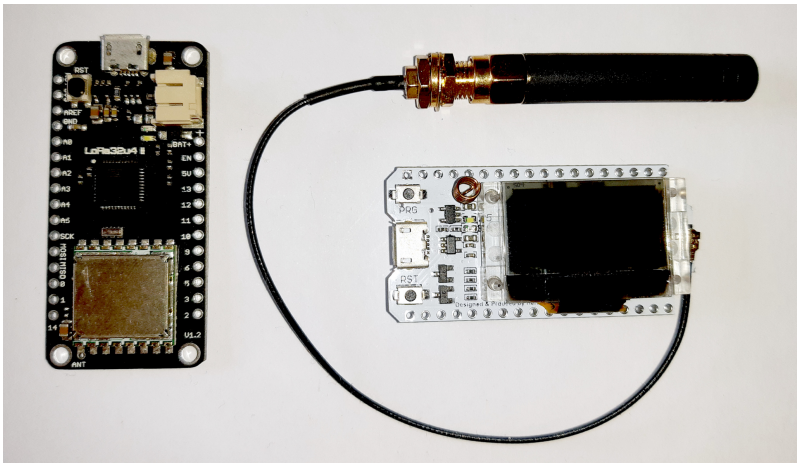


Figure 2.2: Two embedded development boards with LoRa capability. Left, a BSFrance LoRa32u4 II, featuring an Atmega32u4 MCU and an SX1276 LoRa transceiver. Right, a Heltec Wi-Fi LoRa 32 V2, featuring an ESP32 SoC and an SX1276 LoRa transceiver.

to another (e.g., in Spain, duty cycle can be as low as 1 %). Sometimes, depending on the local regulation, this can be avoided if Channel Activity Detection (CAD) mechanisms are put into place.

2.3 The LoRaWAN architecture

LoRaWAN ™ is an open standard by the LoRa Alliance [5] that defines the MAC and application layers and specifies an LPWAN protocol on top of LoRa [15]. The architecture is designed to provide secure wireless bidirectional communication between the end nodes (e.g., battery-operated IoT devices that collect data) and the application that processes the data. Its design, with the gateways at the center and the end nodes around them, has proven suitable for many and diverse IoT applications [16].

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, these applications are hosted in the cloud, so gateways actually need a connection to the Internet (via a cable, Wi-Fi access point, 3G/4G/5G modem, etc.). 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.

[5] LoRa Alliance- `https://lora-alliance.org`

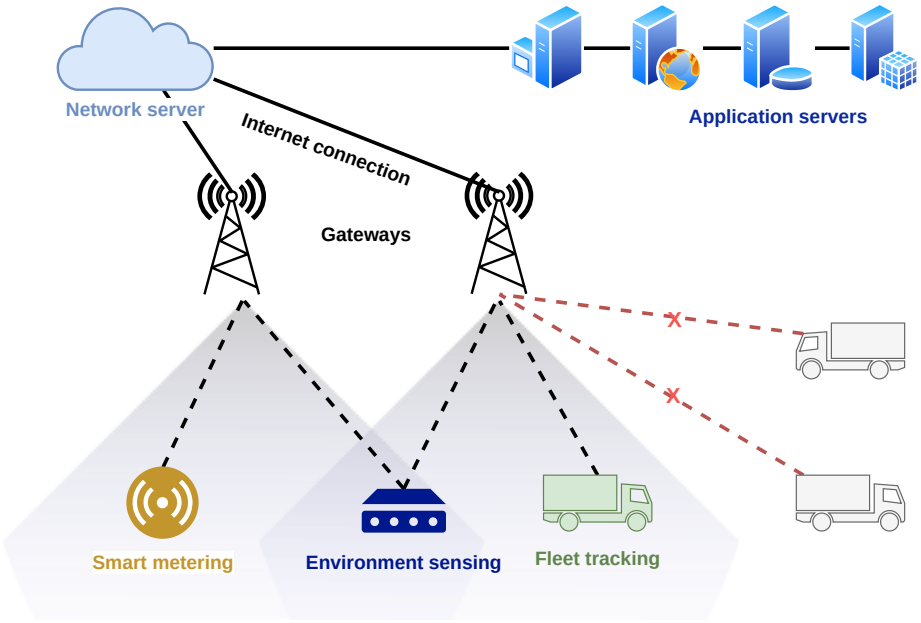


Figure 2.3: A schematic representation of the LoRaWAN architecture. The gateways, in the middle, define the star topology and the coverage area for the end nodes, at the bottom. The network and application servers, shown on top, usually reside in the cloud.

Down-link communication from the network server to the end nodes is also possible via the gateway, though some restrictions apply. Most commonly, the so-called LoRaWAN Class A end devices open two short 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. Other classes (i.e., Class B and Class C) open reception windows on a schedule basis or listen continuously for down-link messages, often requiring a constant power supply.

LoRaWAN gateways are typically built with more powerful devices than end nodes (e.g., from MIPS Wireless System-on-a-Chip (WiSoC)-based routers to ARM embedded computers and tiny x86 boards like the one pictured in Figure 2.4). Besides the higher computing power and communication capabilities, gateways usually feature multichannel LoRa transceivers that make them able to receive and successfully demodulate several incoming transmissions from end nodes simultaneously. Gateways are one order of magnitude more expensive than end nodes, but their superior network and computing capabilities allow a single gateway to service hundreds, even thousands of end nodes –depending, of course, on how many data they transmit, and how often–.

LoRaWAN has been analyzed in a number of studies for all types of applications and scenarios. By means of field measurements and simulation, for instance, it has

Figure 2.4: An indoor IMST LoRaWAN gateway based on a Raspberry Pi computer and an iC880A concentrator.

been estimated that 10^5 nodes in a dense urban area like Calgary, AB, Canada, could be serviced by just three gateways. [17]

The LoRaWAN architecture is well-suited for applications that rely on a centralized infrastructure to collect or process captured data. However, there are reasons for using a more flexible topology rather than LoRaWAN's single-hop, gateway-client scheme. For instance, to increase the network coverage, to overcome the need for additional infrastructure besides the end devices themselves for Peer-To-Peer (P2P) communication, or even to avoid data going to the cloud through an Internet connection in underserved locations. This restriction is a showstopper for deployments, particularly in terms of area coverage and throughput, as they become limited to the capacity the given gateways can provide.

Chapter 3

Related work

Several proposals regarding multi-hop, mesh and routing for both LoRa and LoRaWAN have made appearance in the recent years. Researchers have designed and implemented several solutions for diverse purposes, with different degrees of complexity and completeness. Their maturity and Technology Readiness Levels (TRLs) is heterogeneous, and range from theoretical contributions to experimentally validated proposals in testbeds or real-world deployments. Only few devices or solutions are commercially-available as end products.

In this section we analyze the most relevant proposals, classifying them depending on their application domain, the scenario they tackle, highlighting the specific features they provide for the development of more flexible network topologies. Many of these proposals were thoroughly analyzed and classified before, from different points of view: in function of the application scenarios they tackle [18], putting the focus on the LoRaWAN architecture [19], or regarding specific implementation aspects like topology and routing [20]. At the end of the section we include Tables 3.1 and 3.2, in which we summarize the most remarkable aspects of all the proposals and indicate their potential and their current limitations.

This review of the state-of-the-art and works related to multi-hop and mesh solutions for LoRa and LoRaWAN is largely based on publication MP1.

3.1 Multi-hop and mesh for the LoRaWAN architecture

The star of stars topology defined by the LoRaWAN architecture is well-suited and commonly used in many different domains and environments. However, it imposes an unbalanced *nodes* $\rightarrow$ *gateway* $\rightarrow$ *network server/cloud* data path and hierarchy where some IoT use cases do not fit well [5]. Also, the extension of a LoRaWAN

network is determined by the coverage of its gateway –or gateways– and, even if LoRa allows transmitting data over several kilometers, sometimes this is not enough to cover vast areas.

Increasing the coverage of a LoRaWAN network is sometimes hindered by the restrictions to install additional regular gateways, either for technical (lack of suitable locations, electricity or Internet connectivity, etc.) or for economic reasons (associated costs, etc.). Infrastructure ownership can also play an important role: *end users* and *infrastructure providers* might be different entities with different interests. To overcome these limitations, researchers have proposed adding multi-hop capability on top of the LoRaWAN architecture, allowing packets to travel through different devices until reaching their destination gateway. Three main strategies are found: adding multi-hop to the gateways, adding it to the end nodes, or introducing intermediate relaying devices.

Addressing the issues above, *Dias and Grilo* designed and implemented an up-link multi-hop solution to extend the coverage of LoRaWAN gateways [21]. They argue that, in some cases, deploying additional gateways is not an option (for instance, when users are not the owners of the infrastructure). Therefore, they suggest deploying intermediate nodes that relay data packets from the end nodes to a gateway, as shown in Figure 3.1a. Their proposal contains a simplified version of the Destination-Sequenced Distance Vector (DSDV) routing protocol, running on a number of intermediate *Routing Nodes*, which coordinate between them to relay up-link packets to the best available gateway. This approach could additionally mitigate network failures, since relay nodes would be able to forward messages to the closest available gateway.

Their solution was experimentally evaluated in a linear and a bottleneck topology. The observations showed that the Packet Reception Rate (PRR) decreased (around 5 %) in the former topology, as well as throughput (approx. 8 %), compared to a single-hop deployment. In the latter topology, PRR and throughput did not considerably decrease compared to the former, while network range was improved. The solution is compatible with existing LoRaWAN deployments, but only covers up-link transmissions from end nodes operating in Activation by Personalization (ABP) mode. Therefore, down-link messages transmission from the gateways to end nodes and Over-the-Air Activation (OTAA) are not supported.

Lundell et al. designed a routing protocol to provide mesh networking between gateways to extend coverage [22]. The authors claim that, in both urban and rural scenarios, gateways without Internet access could forward packets towards those with backhaul connection. They took Hybrid Wireless Mesh Protocol (HWMP) and Ad-Hoc On-Demand Distance Vector (AODV) as their starting point, adapting them to the LoRa specifics, and built a packet tunneling mechanism. Their protocol is transparent to both ends (nodes, and LoRaWAN server) and was validated in laboratory experiments and field tests, with up-link messages traversing a 4-hops network. Down-link transmission were considered, but not tested.

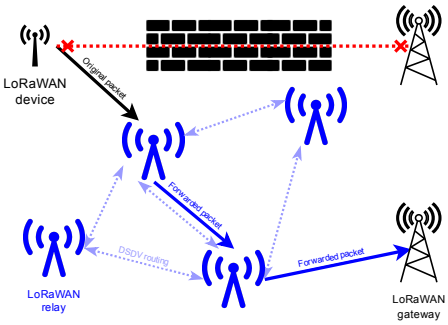
Van de Velde designed and implemented a forwarding device to provide multi-hop communication for LoRaWAN to improve range and quality [23]. Figure 3.1c contains a diagram of the solution, with the forwarding device in the middle of the communication process. According to the author, energy consumption can also be improved, since a shorter distance to cover translates into less transmission power required and less time on air for an end device. His forwarding node is implemented as an intermediate full-blown LoRaWAN gateway and network server device (i.e., it contains the full LoRaWAN gateway application components like a regular gateway would). In addition, it runs an application that is able to transmit, over the air, data packets received from the end device to the actual LoRaWAN gateway. An initial, so-called "naive", implementation is able to re-transmit up-link packets, but appears problematic for down-link transmissions. A second, more advanced implementation introduces better management of down-link packets, in particular of Acknowledgment (ACK) packets from the gateway. The author states there is a remaining challenge for those cases where the *end-device* $\rightarrow$ *forwarder* data rate is higher than the *forwarder* $\rightarrow$ *gateway* one, which has not been solved.

The solution is implemented and evaluated in a laboratory setup, where devices were only a few meters apart, with obstacles between them. Experimental results showed that the addition of a forwarding device allowed extending the communication range of a LoRaWAN deployment: the end device could transmit packets to the gateway using only one third of the previous time-on-air, which translates into less power usage. On the other hand, packet loss increased significantly with a forwarding device, although the causes remain, mostly, undetermined to the researcher. Furthermore, packet latency was also increased at least by one second, because of the delays added in the implementation.

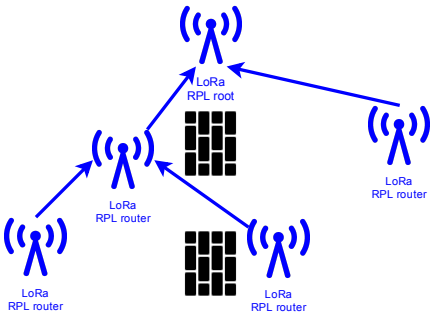
Ebi et al. implemented a synchronous LoRa mesh protocol to extend LoRaWAN networks for end nodes monitoring underground infrastructures [24]. Their approach adds repeater nodes that bridge the synchronous LoRa mesh network segment with the regular LoRaWAN gateway. The results outperform a standard LoRaWAN network with regard to the reliability of packet delivery when transmitting from range-critical locations, like underground areas. The solution enhances transmission reliability, efficiency, and flexibility, but requires a precise time reference (e.g., using GPS or DCF77 time signaling) for synchronization.

3.2 Multi-hop and mesh using the LoRa radio technology

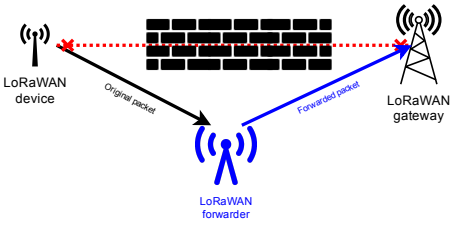
The second step in pursuing more flexible network topologies than the star-of-stars is the development of multi-hop protocols to build mesh networks, in which nodes coordinate in a decentralized manner. This way, the *gateway–end nodes* hierarchy is flattened, resulting in horizontal network deployments. There, all the nodes poten-



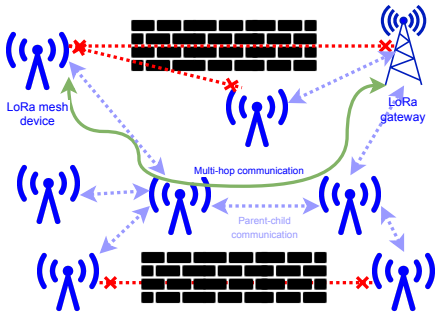
(a) Schematic representation of Dias' and Grilo's multi-hop uplink extension. In black, the components of a standard LoRaWAN system; in blue, their up-link relaying devices allow forwarding packets from a device to a gateway when they have no direct connectivity.



(b) Schematic representation of Sartori's et al.'s Routing over Low Power and Lossy Networks (RPL) multi-hop LoRa network. Nodes listen in loops to the different SFs for transmissions and use an Objective Function to create a topology with a minimum path cost.



(c) Schematic representation of Van de Velde's multi-hop forwarding node. In black, the components of a standard LoRaWAN system; in blue, his forwarding device allows retransmitting packets from a device to a gateway when they have no direct connectivity.



(d) Lee, Ke et al.'s LoRa mesh network. Nodes take the child/parent roles to form a dynamic tree topology to reach the gateway, which periodically polls the nodes for sensor measurements.

Figure 3.1: Schematic representation of different multi-hop and mesh solutions for the LoRaWAN architecture

tially have equal duties regarding network operation and organization, and traffic forwarding. This also allows room for replacing the centralized gateways and cloud server infrastructure with in-the-premises, self-managed systems that harness computation at the edge, enabling distributed, fully decentralized applications. Causes for this are diverse, ranging from technical requirements (e.g., distributed applications where the centralized data sink design does not apply because end-node devices better integrate the control loop of certain IoT requirements by means of a gateway-less mesh network) to data privacy, security, and sovereignty (i.e., independence of the network deployment from external providers).

Lee, Ke et al. designed and implemented a LoRa mesh networking system [25, 26]. The authors argue that deployments in urban areas demand a high density of gateways, to ensure that indoor nodes can communicate with the network servers. To avoid deploying more gateways, they designed a mesh networking system for IoT applications. Their design consists of a *data sink* (rather misleadingly, the authors call it *gateway*, but only from the role point of view, not referring to a device with gateway-capable LoRa hardware) that broadcasts beacons to invite nodes to join the network during an initial stage. Those, in turn, set the gateway as their parent. After this, the gateway is able to request data from the children nodes by polling them. New nodes that hear packets from the gateway or from other nodes can also join the network, choosing a suitable parent based on multiple factors (namely, Received Signal Strength Indicator (RSSI) and hop count). The gateway holds a complete view of the network topology, which can leverage to modify child to parent assignments based on its more comprehensive information. Figure 3.2a depicts their LoRa mesh devices forming a tree network topology to allow all nodes to communicate with the gateway.

The system was experimentally evaluated with a 19 nodes deployment, covering an area of $800 \times 600\,\mathrm{m}^2$ in a university campus, over a time span of 8 days. The results of the measurements showed the variations in the network topology and the impact of the mesh on each node's Packet Delivery Ratio (PDR) to the gateway. The developed mesh system improved the PDR of all nodes, especially those which were almost or completely out of the gateway's range. The authors state that while their mesh solution allows extending the coverage of a network without installing more gateways, the maximum number of nodes served would be smaller than those of a conventional star topology, because of the increased transmission times due to successive packet forwarding.

Among other challenges, *Sartori et al.* addressed the gateways' coverage issue and designed RPL+LoRA MAC (RLMAC), a MAC layer protocol that enables RPL multi-hop communications based on LoRa [27]. They argue that the star topology is convenient for ease of deployment and from a business perspective, though multi-hop could mitigate congestion issues. Moreover, multi-hop could be the only option for covering very large areas with few base stations, and could increase throughput or reduce time-on-air by using faster SFs. The authors designed a multi-hop

solution for single-channel LoRa nodes (gateways use more powerful RF chips providing multichannel reception simultaneously). They implemented the algorithms to bootstrap and operate a network using RPL by combining a slow reception loop with fast transmission loops, to ensure that nodes can receive messages from any neighbor using any SFs. Figure 3.2b depicts the proposal solution, with the RPL root device at the top and different leaf nodes below.

With the first experimental validations, the authors found that, when two nodes choose the same SF to transmit towards a third node, collisions are frequent. This was solved by adding an extra delay, randomly generated from a brief RSSI measurement. The authors state that their solution must be tested with a simulation tool for performance and power consumption before using it in a real life deployment with a large number of nodes.

Kim et al. proposed an Adaptive Spreading Factor Selection (ASFS) scheme to build LoRa mesh networks using single channel transceivers, increasing throughput and reducing costs [28]. Their proposal uses the modems' CAD capability with an iterative SF inspection and selection algorithm that allows links to operate independently at different data rates, achieving almost 100 % correct detection. This idea had been implemented already on single-channel gateways, but had not been adapted for multi-hop usage yet. The authors experimentally evaluate the proposal with up to 10 nodes, and compare three topologies (star, tree, and mesh) using Semtech's single-channel SX1272 and multichannel SX1301 transceivers. Using ASFS allows nodes to choose different and faster SFs, achieving data rates 4 to 6 times faster than without it (when all the nodes stick to a common, network-wide slower SF).

Liao et al. strived to construct a multi-hop network based on LoRa in combination with Concurrent Transmission (CT) [29]. They verified that the LoRa technology is compatible with CT and that receiver performance can be improved by introducing timing offsets between the relaying packets. They propose the offset-CT method, which adds random timing delays while preventing offset from diverging over the multi-hop network. First, they assess how CT affects LoRa, considering its long symbol time, especially when the highest SFs are used. According to their analysis, CT is unlikely to fade a whole LoRa symbol. Second, they discuss if LoRa's high-order M-ary Frequency-Shift Keying (FSK) modulation and its Carrier Frequency Offset (CFO) could possibly be incompatible with CT. However, they consider that the two capture effects (frequency-domain and time-domain spreading) provided by the FSK modulation naturally benefit LoRa in a CT scenario, hence making their combination particularly suitable. Figure 3.3a depicts the proposed mesh network where packets are forwarded one relay at a time, taking advantage of the CT properties.

To validate their findings, the authors simulated the performance of a LoRa receiver with CT and then conducted real-chip experiments using a Semtech LoRa SX1272

RF transceiver[1]. They concluded that LoRa reception can survive under CT with no more additional requirements than other CT-compatible standards such as IEEE 802.15.4. Furthermore, they assessed that their proposed offset-CT can significantly improve PRR when the added timing offset delays are away from multiple values of T_S (the modulation's symbol time). Finally, they performed proof-of-concept experiments to show the feasibility of the so-called Multiple-Building Area Networks (MBANs) by means of LoRa with CT *and* offset-CT, using 18 nodes with the aforementioned RF chip in both a low-density and a high-density scenario. These last experiments showed that CT-LoRa enjoyed a very high PRR performance in an MBAN scenario already, which was significantly improved by the offset-CT method when nodes were densely deployed.

Nunez Ochoa et al. analyzed different LoRa radio parameters (SF, bandwidth, transmission power) and computed the energy consumption of the transceivers for both star and mesh topologies [30]. They analyzed the flexibility LoRa offers to configure its radio parameters for different deployment scenarios and the impact they have during both radio transmission and reception's energy consumption, and proposed various strategies to reduce it. For a star topology, the authors identified that increasing the SF had a more significant impact on energy consumption than increasing the transmission power. Therefore, they advise to first adapt the transmission power and then to increment the spreading factor to minimize energy consumption. For a mesh topology, energy consumption was optimized by applying different radio configurations for different network layouts, where nodes' density played a determinant role in coverage and number of hops.

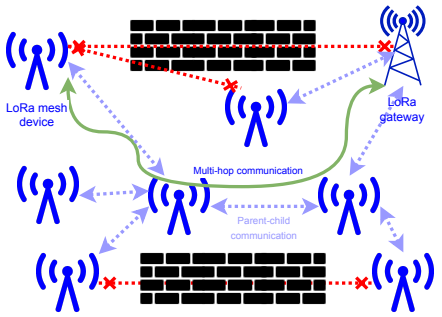
According to those findings, the authors state that a global strategy that exploits both the star and the mesh topologies can provide a trade-off between keeping energy consumption low and extending network coverage, and that it will depend on the global area to cover and the nodes' density. Figure 3.3b

3.2.1 Multi-hop linear networks

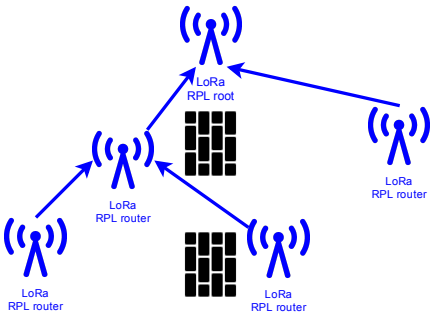
Monitoring systems for linear utilities that connect distant points (power lines, waterways, piping systems, etc.) are common among IoT deployments. Similarly, underground deployments in sewage systems, mines, etc. experience equivalent coverage issues. While LoRa provides long transmission distance, this may not suffice for systems spanning over hundred of kilometers, where gateways would need to be deployed at intervals along the utility.

The multi-hop solutions for linear network in this category are a particular case of the generic multi-hop and mesh LoRa solutions. However, given their predictable communication pattern, topology and timing, authors have provided solutions to address the problems in these specific conditions, like synchronization to allow nodes

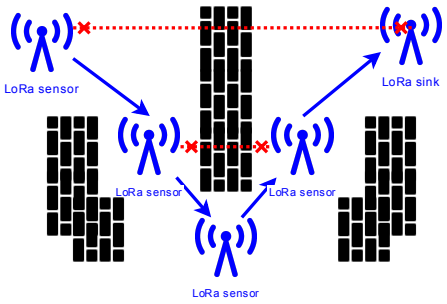
[1] Semtech SX1272 Datasheet - `https://www.semtech.com/uploads/documents/sx1272.pdf`



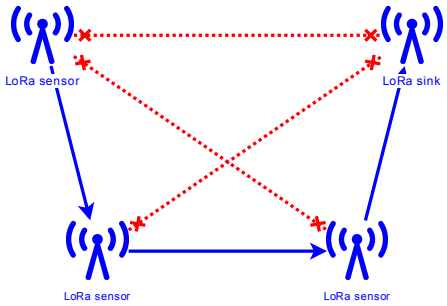
(a) Lee, Ke et al.'s LoRa mesh network. Nodes take the child/parent roles to form a dynamic tree topology to reach the so-called *gateway*.



(b) Sartori's et al.'s RPL multi-hop LoRa network. Nodes listen in loops for packets on each SF, and use an Objective Function (OF) to create a topology with minimum path cost.

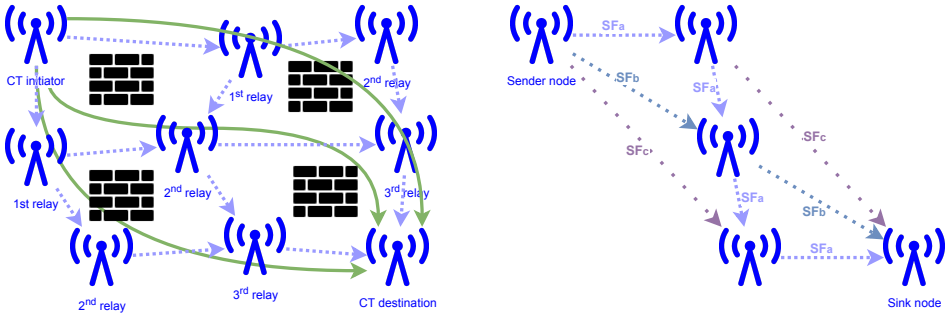


(c) Abrardo and Pozzebon's multi-hop linear LoRa network. Deployments inside tunnels reduce the radios' reach to few hundreds of meters, requiring for a synchronous wake-from-sleep and transmit chain towards a gateway.



(d) Duong and Kim's multi-hop linear LoRa network. Nodes synchronously wake up from a sleep state to receive data from their previous node, combine them with their own and forward them towards the sink.

Figure 3.2: Schematic representation of different multi-hop and mesh solutions for LoRa (I).



(a) Liao et al.'s CT-enabled multi-hop LoRa network. Concurrent transmissions sequentially propagate a packet through the network, until the flooding reaches the destination device.

(b) Nunez Ochoa et al.'s energy-efficient LoRa mesh network. Using different SFs (SF_a, SF_b, SF_c), devices modify their transmission range, providing the so-called "degrees of freedom", leading to different energy consumption figures.

Figure 3.3: Schematic representation of different multi-hop and mesh solutions for LoRa (II).

to spend most of the time in an energy-saving sleep state, waking up only to receive a packet and forward it down the line (eventually, adding a local datum or measurement).

Abrardo and Pozzebon designed a multi-hop LoRa linear network for underground environments, optimizing the nodes' sleep/wake cycles to reduce battery consumption [31]. After an in-the-premises measure campaign, the authors found that LoRa transmission was limited to around 200 m, making the classical star topology unsuitable for the pervasive monitoring of very long aqueducts which include slight curves that obstruct the line of sight. To overcome this limitation, they opted for a data propagation model where sensor nodes would form a transmission chain from the very first node towards the gateway, as shown in Figure 3.2c. Given the difficult access to an infrastructure such as an underground aqueduct, the system was optimized to minimize energy consumption, with nodes entering in sleep mode except during the data sampling and transmission time (this includes sending their own data and the forwarded data).

A prototype of the system was developed using Arduino UNO boards[2] for the sensor nodes and a Waspmote USB Gateway[3] as the data sink device. The researchers found that the internal clock of the Arduino boards used was not accurate enough to wake the devices at the right moment after long sleep cycles (i.e., lasting more than 120 s). Therefore, they developed an optimized synchronization mechanism to propagate data between pairs of nodes, starting from the first sensor and towards

[2]Arduino UNO board: `https://www.arduino.cc/en/Main/ArduinoBoardUno`
[3]Libelium Waspmote board: `http://www.libelium.com/products/waspmote/`

the gateway, allowing for longer sleep cycles, reducing power consumption to 50 % of their previous approach.

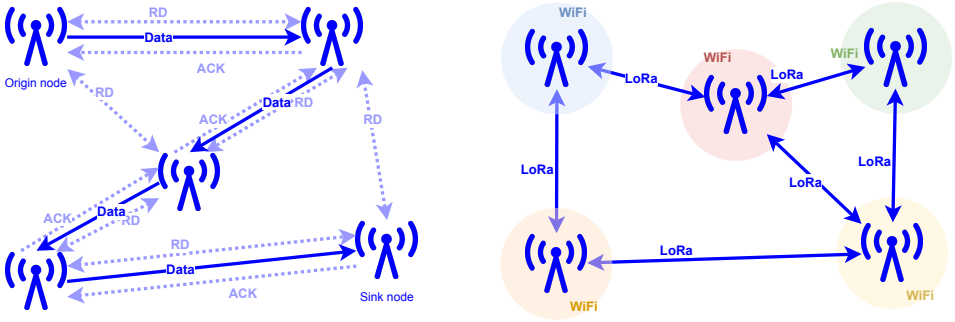
Duong and Kim designed and implemented a protocol with multi-hop communication for LoRa networks covering large distances [32]. Their solution was intended for deployments where every monitoring node is placed along a line, such as a gas pipe or a high voltage line. Figure 3.2d exemplifies this case, where a number of LoRa nodes are placed along a semicircle and create a linear topology. The system operates in two stages: an initialization period and the regular operation period. During the first one, the linear topology is formed as an initialization message is passed from node to node, starting at the data sink, to make sure that each of them is at the appropriate position and synchronized for the operation phase. During the second stage, nodes forward data in the leaf $\rightarrow$ *sink* direction. To do so, nodes are synchronized and wake up at specific moments in time to receive data packets from their neighbors, which they can combine with their own data packets and send them further across the line.

The system was deployed in a university campus, with 4 monitoring nodes placed along a straight line and one data sink, with distances between them ranging from 150 to 200 m, and evaluated using different LoRa SFs. The obtained PDR ranged from 92 to 98 % and throughput went from 185 to 28 bps, depending on the SF used. This solution is well-suited for a unidirectional up-link data flow and also takes into account the nodes' active/sleep cycles, which enables energy-savvy operation. However, down-link data transmission is not covered.

3.2.2 Software libraries and tools

McCauley developed RadioHead, a packet radio software for embedded processors, that provides an object-oriented library for sending and receiving messages via a variety of data radio technologies, including LoRa [33]. In particular, it provides the RHMesh subclass for sending addressed, optionally acknowledged, datagrams accross a network using multi-hop[4]. The class adds automatic route discovery and route signalling within a mesh of adjacent nodes. Figure 3.4a depicts a fictional mesh network where this solution could be used. It can be used in networks where the network topology is dynamic and nodes move around or become unavailable. RHMesh uses reliable hop-to-hop messages delivery, but not end-to-end acknowledgements. The author warns that programs using the RHMesh require almost 2 kB of SRAM, which is beyond the capacity of some Arduino devices. [5] Furthermore, the class does not have message queuing, which means that it can only handle one message at a time.

[4] `http://www.airspayce.com/mikem/arduino/RadioHead/classRHMesh.html`
[5] `https://www.arduino.cc/en/Products/Compare`



(a) Schematic representation of Dias' and Grilo's multi-hop uplink extension. In black, the components of a standard LoRaWAN system; in blue, their up-link relaying devices allow forwarding packets from a device to a gateway when they have no direct connectivity.

(b) Sudo Mesh and Secure Scuttlebutt's LoRa mesh network. The nodes use the Wi-Fi radio to provide a short-range access point for end-users to connect with their devices (e.g., smartphones, laptops) and the LoRa radio to create a mesh network to exchange data between them (e.g., chat messages).

Figure 3.4: Schematic representation of features provided by LoRa-based mesh libraries and tools.

The volunteers from Sudo Mesh and Secure Scuttlebutt are developing disaster.radio, an off-grid, solar-powered, long-range mesh network [34]. They have designed and prototyped an open source software and hardware solution to provide access to emergency communication for communities to effectively coordinate critical infrastructure in the event of ecological disasters. The hardware is built around the ESP8266 [6], a Wi-Fi microchip with a full TCP/IP stack and microcontroller capability, together with a Semtech SX1276 LoRa transceiver for node to node communication. Figure 3.4b shows how the nodes use their two radios to provide a short-distance Wi-Fi connection for end users and long-distance LoRa communication between them.

The software, which is a work-in-progress, is envisioned to provide applications for end users (using the ESP8266 Wi-Fi interface) and to include a distance vector mesh routing protocol [7], inspired by the Babel RP. The authors have also introduced the concept of Disaster Area Network (DAN), an heterogeneous wireless mesh network that uses LoRa for point-to-point or point-to-multi-point connections between nodes.

3.2.3 Commercial products and devices

Hester and several other contributors are working on Meshtastic [35], a project for using inexpensive development boards with GPS, battery, and a LoRa chip as

[6] `https://espressif.com/en/products/hardware/esp8266ex/overview`
[7] https://github.com/sudomesh/disaster-radio/wiki/Protocol

secure mesh communicators. Meshtastic is intended for outdoor sport activities or any other situation with no Internet access. Users create a private mesh to exchange their location and send text messages to a group chat. Devices forward packets using a flooding algorithm to reach the furthest member. The chosen hardware is based on a Wi-Fi/BLE-capable ESP32 SoC bundled with a LoRa transceiver and, optionally, a GPS receiver.

Pycom provides commercial development boards and OEM products for IoT projects in the Python language. These devices can run Pymesh, a firmware for flexible LoRa mesh networking [36]. It provides encrypted ad-hoc communication over raw LoRa, implements LBT MAC, and supports multiple node roles (leader, router, child, and border router). The firmware also has some routing capabilities, as it claims to forward packets via the best link available. Unfortunately, Pymesh can only run on Pycom's products, making it incompatible with other vendors.

NiceRF commercially offers the SV-Mesh and LoRaStar range of LoRa transceivers. These products, available as embedded boards or packaged devices, provide serial TTL, RS232, or RS482 communication over LoRa links. They consist of a low power microcontroller and a regular LoRa transceiver. The manufacturer developed the proprietary LoRa-Pro mesh networking protocol, which defines a 2 byte addressing scheme, three network roles (node, router, node plus router), and a virtually unlimited number of routes.

3.3 Multi-SF detection

Although single-channel LoRa transceivers are meant to operate on a given channel and SF at a time, a few proposals are able to use the radio chips in a smart way that allows them to automatically detect the SF on which an incoming packet is being transmitted. With this technique, a device may be able to receive transmissions on any available SF (although only one at a time), which opens the door for building more flexible multi-SF networks. This topic is discussed in more detail in Section 5.1.

M. Westenberg leveraged LoRa's CAD capability to achieve multi-SF detection in the Single Channel LoRaWAN Gateway [37]. This project allows turning a regular ESP8266/ESP32-based LoRa-capable device (e.g., end-nodes and development boards) into a LoRaWAN gateway. Although its operation is limited to a single channel and a single SF at a time (unlike full-fledged gateways which are multichannel, multi-SF capable), transmissions on different SF can be automatically tuned and received. Later, *J. Braam* ported the code of the Single Channel LoRaWAN Gateway above to the Lua language, so that it could run on the NodeMCU firmware usually found in ESP8266-based devices [38].

Heltec commercially offers the Dual Channel LoRa Gateway. It consists on an ESP32 SoC with two LoRa transceivers that leverage multi-SF detection to provide

an affordable gateway for small LoRaWAN deployments [39]. The product offers, therefore, the functionality of two single-channel gateways like the one mentioned above, packaged into a single embedded device.

Kim et al. (previously mentioned, in previous Section 3.2) used their ASFS to use multiple SF with single-channel transceivers [28]. The results of their proposal are later discussed and compared with ours, in Section 5.1.

3.4 Discussion

The considerable amount of proposals analyzed, and their diversity, not only answers RQ1 positively, but also shows the interest in the research community in tackling problems related to those described in Section 1.1.

The review of the state-of-the-art already provides answers to some of the RQs formulated in the introduction of this thesis in Section 1.2, albeit only partially. However, the literature surveyed still does not answer the main RQs regarding the performance aspects of LoRaWAN networks where end nodes are enhanced with multi-hop capacity (RQ1) above, how can the distinctive features of the LoRa technology can be taken advantage of to build communication networks for IoT systems with flexible mesh topology (RQ2) and what are their performance figures (RQ3), nor it elaborates on how these more flexible networks can be leveraged (RQ4).

It is worth noting that this work, including this chapter, focuses on the LoRaWAN architecture and the LoRa radio technology as enablers of the IoT, in particular, when it comes to flexible network models and topologies that go beyond the centralized ones. In this context, it is easy to find analogies with research in other well-established domains like Wireless Sensor Networks (WSNs). These networks are typically formed by low-power devices that monitor the environment and, with different degrees of cooperation, forward the data towards a central sink. A plethora of technologies, platforms, and architectures for WSNs have been discussed [40], including manifold proposals for routing [41, 42, 43]. Our work, however, builds upon the specifics of multi-hop and mesh topologies for LoRa and LoRaWAN and, in particular, focuses on performance aspects that are strongly influenced by the technology's specific features and properties.

Authors	**Challenges addressed**	**Application scenario**	**Approach**	**Validation**	**Potential**	**Limitations**
[21] Dias & Grilo (2018)	Coverage extension without adding more gateways	Dark spots in urban deployments	Intermediate relay nodes implementing simple DSDV routing	Prototype assessment with 4 routing nodes	Up-link extension of multi-stakeholder LoRaWAN network	No down-link transmissions
[22] Lundell et al. (2018)	Coverage extension without Internet backhaul	Large rural sensors networks, urban IoT deployments	Gateways mesh for transparent node to server tunneling	Proof-of-concept implementation with 4 gateways	Monitoring of very large areas, gateway-based backbone mesh network	No down-link transmissions
[23] Van de Velde (2018)	Range and quality of links, reduce energy consumption	Connecting nodes and gateways with blinding obstacles between them	Forwarding device between end nodes and gateway	Experimental laboratory setup	Bi-directional extension of LoRaWAN network	Increased packet loss and delay
[24] Ebi et al. (2019)	Coverage extension to reach infrastructure underground	Dark spots in isolated or difficult to access locations	Synchronous sensor nodes mesh network with routing towards a sink	Experimental and production deployment in urban-sized network	Monitoring over vast areas, underground locations or difficult to access spots	Maximum five chained sensor nodes due to LoRaWAN's payload restrictions

Table 3.1: Summary of the surveyed multi-hop and mesh proposals for LoRaWAN.

Authors	Challenges addressed	Application scenario	Approach	Validation	Potential	Limitations
[25, 26] Lee et al. (2018)	Indoor nodes' communication in dense urban areas	Campus-sized monitoring environment	Mesh system, RSSI and hop-count-based next-hop selection	Campus-sized experimental testbed with 19 nodes	Collaborative LoRa mesh networks for very large areas	Smaller density than with a star topology
[27] Sartori et al. (2017)	Large areas coverage with few base stations	Distributed or isolated IoT deployments	Extension of RPL with new OF and metrics	Building-sized experimental setup with 5 nodes	Infrastructure-less IoT, in-the-premises computing	Unbalanced network, bottleneck in RPL root nodes
[28] Kim et al. (2020)	Throughput enhancement for single-channel nodes	High traffic networks	Adaptive Spreading Factor Selection (ASFS)	Campus-sized experimental testbed with 10 nodes	Overlaid, simultaneous multi-SF operation	Complex routing algorithm required for best results
[29] Liao et al. (2017)	Combination of LoRa and Concurrent Transmission	Neighborhood- sized IoT deployments	Multi-hop packet flooding by means of Concurrent Transmission	MBAN-scale deployment simulation and experimental validation	Distributed sensors and actuators orchestration, computation at the edge	Multi-node synchronization of Concurrent Transmission
[30] Nunez Ochoa (2017)	Energy consumption optimization	Very energy-constrained end nodes	Combination of star and mesh topologies	Analytic calculations	Lifetime extension of battery-operated nodes	Real-life networks' dynamics, lack of experimental validation
[31] Abrardo & Pozzebon (2019)	Underground networks with limited range	Monitoring of tunnels and subterranean utilities	Linear origin to sink multi-hop packet forwarding	In-place measurements, laboratory tests, analytical analysis	Management of underground, linear topology infrastructure	Unidirectional communication, nodes synchronization
[32] Duong & Kim (2017)	Networks covering long distances in a linear topology	Monitoring of linear utility deployments	Linear leaf to sink multi-hop packet forwarding	Campus-sized experimental setup with 5 nodes	Management of long-distance, linear topology infrastructure	Unidirectional communication, reduced PDR and throughput
[33] McCauley (2014)	Datagrams transmission over a multi-hop network	Dynamic networks with varying topology	Automatic route discovery and signaling	Software library	Gateway-less, context-aware network deployments	Missing end-to-end ACK and message queuing
[35] Hester et al. (2020)	Very long range data broadcast	Off-grid, outdoors, emergency communication and location	Broadcast smart data flooding	Available, off-the-shelf devices	Autonomous, off-grid communication for communities	Scalability, no routing
[36] Pycom (2020)	Network flexibility, decentralization	Self-organizing, multi-gateway mesh networks	Routing protocol, multiple network roles	Development devices commercially available	OEM integration in off-the-shelf products and applications	Closed source, proprietary solution, incompatibility with other vendors
[44] NiceRF (2020)	Serial data communication over encrypted mesh	Remote control, telemetry, automation	Multiple network roles, routing protocol	Readily available commercial product	Machine to machine (M2M) communications	Closed source, proprietary solution, incompatibility with other vendors

Table 3.2: Summary of the surveyed multi-hop and mesh proposals for LoRa.

Chapter 4

Packet forwarding in LoRaWAN end nodes

The operational area of a given LoRaWAN network is determined by the radio coverage provided by its gateway infrastructure. This makes the architecture well-suited for applications that rely on a centralized infrastructure to collect or process captured data. However, there are reasons for using a more flexible topology rather than LoRaWAN's single-hop, gateway-client scheme. For instance, to increase the network coverage, avoiding the need for additional gateways, to overcome infrastructure failures, or to avoid data going to the cloud through an Internet connection in underserved locations. Figure 4.1 depicts these limitations.

In this chapter, we introduce *LoRaMoto*, a communication system to provide safety awareness among civilians after an earthquake. By means of its development, and through computer simulations, we explore the operation and performance of a large LoRaWAN network. We emulate the possible effects of a natural disaster on the gateways, and evaluate if adding end nodes the capacity to forward their neighbor's packets, as a strategy to overcome infrastructure failures, affects performance.

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 and LoRaWAN because it allows the implementation of dedicated and single-purpose solutions based on the 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

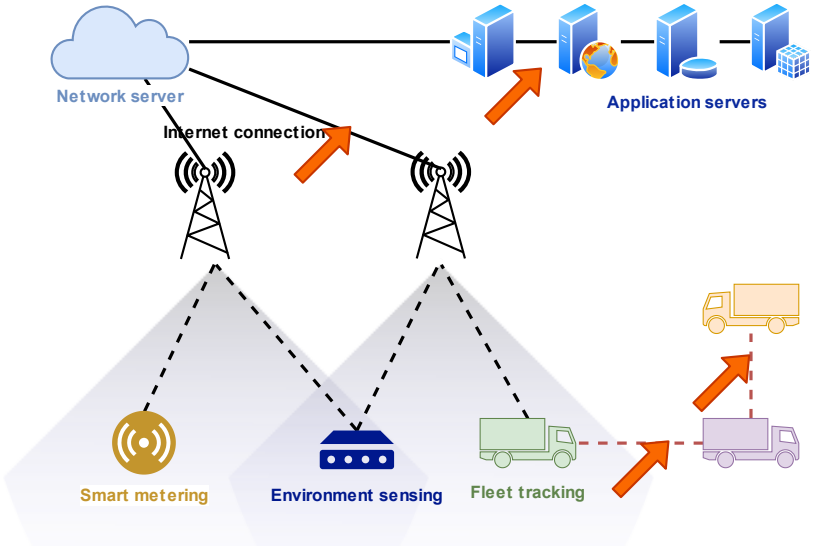


Figure 4.1: A schematic depiction of the LoRaWAN architecture, where the orange arrows indicate some of its limitations. In particular, direct communication between nodes is not supported, albeit technically feasible.

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.

In particular, this extended version includes a congestion control protocol, based on 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 capacity 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 simulations. 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 work in this chapter builds upon the authors' previous work CP1, which was later extended for MP2.

4.1 Context of emergency networks

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 are themselves 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 [45]. 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 [45, 46, 47].

Figure 4.2 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 [48], but such an infrastructure is usually not available to support interactions among civilians or between them and the first responders.

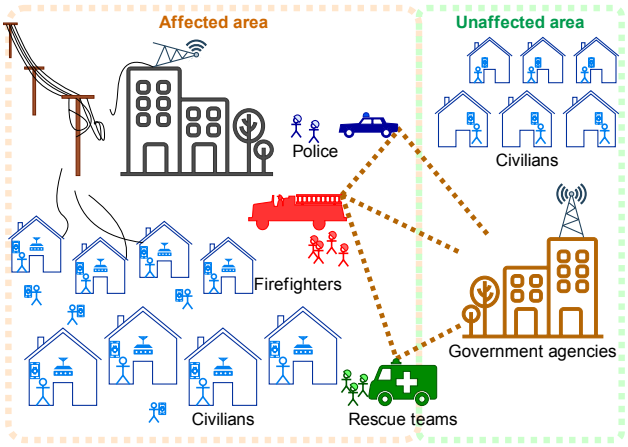


Figure 4.2: Typical communications scenario in the aftermath of an earthquake.

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 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 [49, 50]. 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 [51, 52], and identifying the locations of people in large urban populations and steering them to safety[53]). 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 [54]).

To foster the participation of civilians in first response activities, we propose a communication 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 geolocated 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.

4.2 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 [55, 56]. In such a scenario, relying on communications and timely information is essential for making proper decisions. This section offers a brief review of the related work on these topics.

4.2.1 Communications support in emergency responses

Communications systems, as instruments to facilitate the exchange of 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 [57, 56]. This is still an open research issue, In particular when mobile digital communications are required inside an affected area [51]. 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 Wi-Fi-based implementations, like Mobile Ad-hoc Networks (MANETs) and Opportunistic Networks (OppNets) [58, 59, 60], 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 [16]. 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.

4.2.2 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) [61] 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 [62], 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 [63]. LoRaWAN, which is a network protocol that builds upon the LoRa radio technology, also has these scalability limitations [64, 14], and its performance decreases [65] 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 [14] 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 [64]. 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 [66]. 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 $> 70\,\%$ [64]. 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 [65]. They reported two key findings: (i) LoRaWAN is susceptible to mobility; and (ii) the effect of mobility worsens the performance of 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 [67]. 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 [68]. 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 [69]. In the same line, Sciullo et al. used LoRa for ad hoc message communication and also for trilateration in order to identify users' geolocation [62].

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[25]., scheduling [70, 71], synchronizing transmissions [72] and time [73, 74]). 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.

4.3 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 4.2, 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 user nodes, which make this system cross the boundaries of the LoRaWAN framework. These technologies were introduced in Chapter 2. Below, in Section 4.4, we introduce the LoRaMoto system architecture.

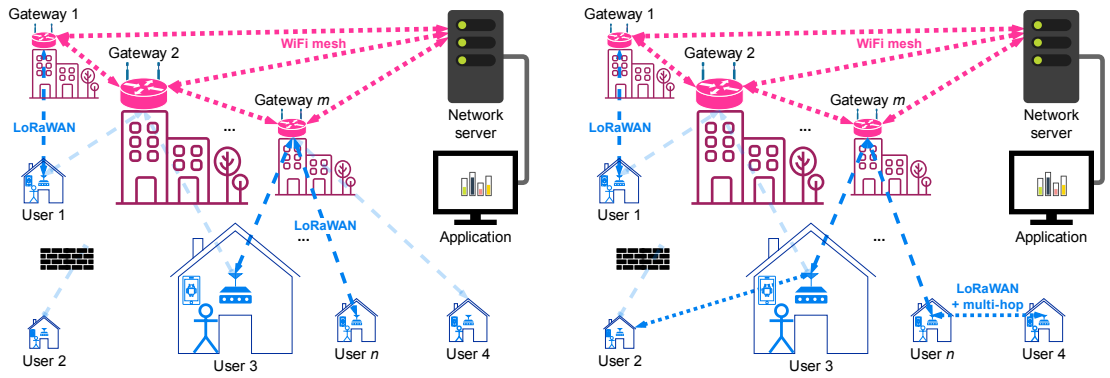
4.4 LoRaMoto system design

The initial LoRaMoto design was based on the LoRaWAN architecture [75, 15], 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 4.3 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.

4.4.1 Baseline LoRaWAN architecture

At the bottom of Figure 4.3a, 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 4.3a 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 net-



(a) Baseline LoRaWAN architecture.

(b) Enhanced LoRaMoto architecture, featuring forwarding between end-nodes.

Figure 4.3: High-level depiction of the baseline LoRaMoto system architecture, matching the LoRaWAN specifications.

work 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 an 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 [76]. 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 4.3a, 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 that manages the messages received from the nodes, processes data, and takes actions (e.g., triggering alarms or sending down-link 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 down-link 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).

4.4.2 Packet-forwarding beyond the LoRaWAN architecture

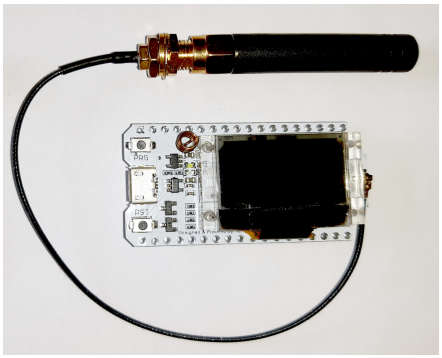
The LoRaWAN architecture involves a star-of-stars topology that does not consider direct communication between nodes. This imposes system design restrictions, in particular that all communication from the user nodes to the network server must go through the gateways, even if the underlying LoRa radio 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 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 4.3b 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.

4.4.3 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 telecommunications 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 Wi-Fi 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



(a) Heltec Wi-Fi LoRa 32 V2 development board.

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

Figure 4.4: Home devices are built around the WiFi-capable, low-power Espressif Systems ESP32 SoC, connected to a LoRa transceiver. On the left, a development board (LoRa transceiver is soldered under the display). On the right, a prototype home device inside a 3D-printed enclosure.

- 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 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 Wi-Fi and Bluetooth support. Furthermore, small development boards based on that device and featuring a LoRa transceiver, like the one shown in Figure 4.4a, are readily available for purchase and allow a fast prototyping path. Figure 4.4b 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.

4.4.4 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 4.5. 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 Wi-Fi or Bluetooth capability, and including a LoRa transceiver, as described above. On

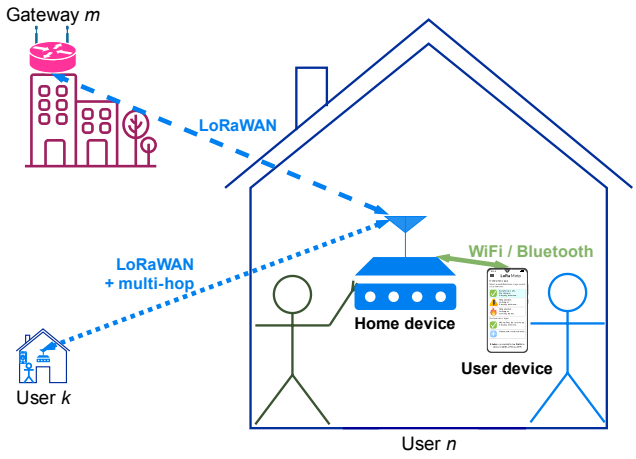


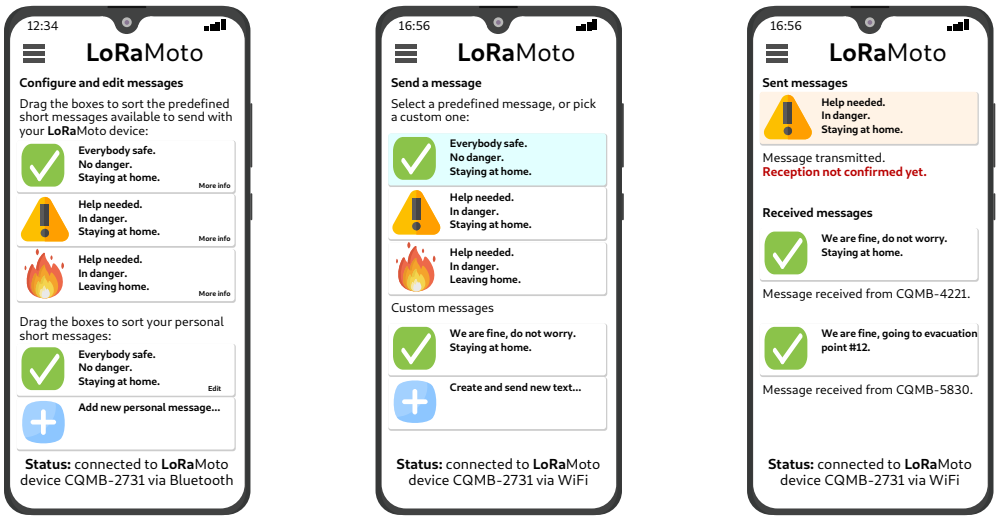
Figure 4.5: Typical interaction scenario with a home device.

the one hand, short-range radios (Wi-Fi 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 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.

4.4.4.1 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 Wi-Fi 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 4.6 shows, on the left (Fig. 4.6a) a mock-up of the user interface for messages configuration, as displayed on a smartphone.



(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 4.6: Mock-up of the web page provided by the home device, to manage the list of messages it can store (4.6a) for later use in the aftermath of an earthquake (4.6b, 4.6c).

4.4.4.2 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 Wi-Fi or Bluetooth. As an example, Figure 4.6 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. 4.6b), and showing the status of sent and received messages (right, Fig. 4.6c).

Table 4.1: Deviation between real data and simplified model.

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 %**

4.5 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 [77], 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 km^2. The population of DC-1 and DC-2 is concentrated in a smaller area of 3.51 km^2, 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 $\times$ 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 4.1. We also considered homes to be uniformly distributed through the area and all the space having the same elevation.

Figure 4.7 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 4.7 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.

(a) Satellite view of Coquimbo.

(b) Simulation software components over the city map.

Figure 4.7: On the left (4.7a), the rectangular area of 1.4 km × 2.5 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 (4.7b) 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).

4.5.1 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 4.8 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 $t = T_{Tx1} = T_0 + T_{UR}$.

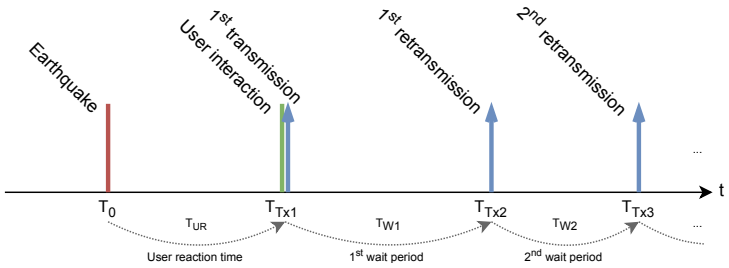


Figure 4.8: Timing of user node activity in the aftermath of an earthquake.

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 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 4.2 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 4.5. 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 2.2), 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 4.8, 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) [14, 66]. 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 4.4, 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 4.8, this means that no additional packet transmissions occur at any node before

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

T_{Tx3}. 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 4.6.4.1 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 4.6, 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.)

4.5.2 Simulation framework

To analyze the capabilities of the proposed system, we use OMNeT++ [78], an extensible, modular, component-based C++ simulation library and framework, in combination with FLoRa [79], 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 [80] and an accurate model of the LoRa radio physical layer derived from previous experimental findings [81]. Figure 4.7b 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

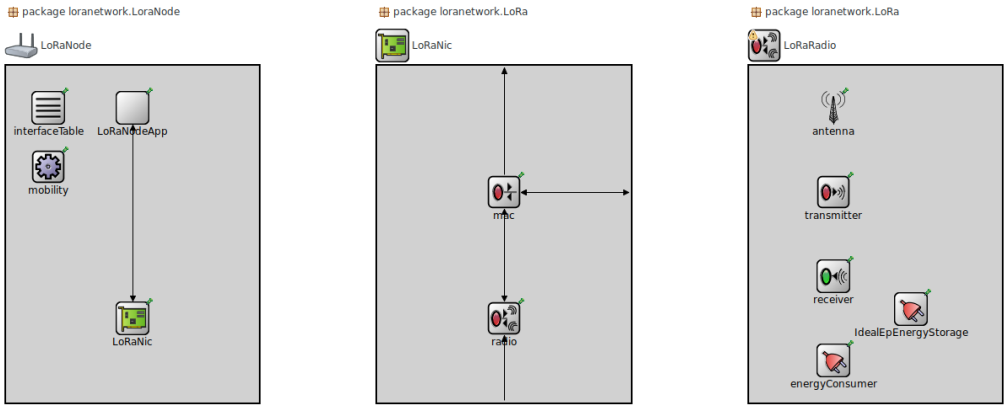
in the baseline scenario. To adapt the framework to our earthquake aftermath use case, we have adjusted several of its parameters according to the specifications in Section 4.5.1. However, to simulate the LoRaMoto architecture, which includes the packet-forwarding functionality, we have modified several 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 4.9 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 down-link messages in reply.

4.6 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 4.4.1. 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 4.5.1 affected the system's capacity to provide service to end-users.



(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 4.9: 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).

4.6.1 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, 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.

4.6.1.1 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 4.5 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/km^2). However, scenarios in other geographical locations could be more—or less—densely populated, leading to different performance figures. In

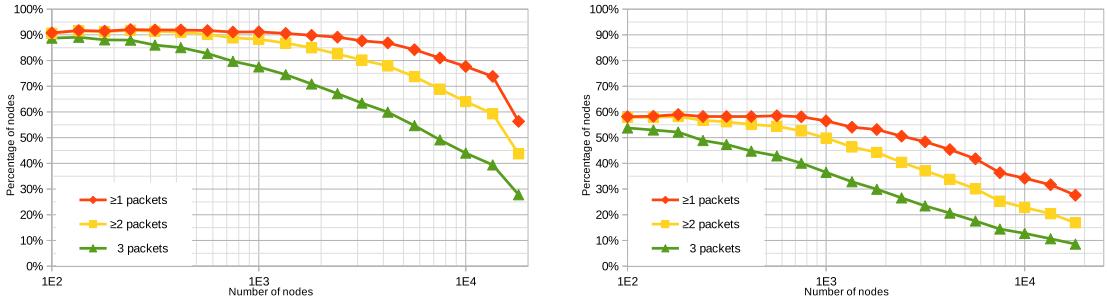
this section, we investigate system scalability by simulating it with different numbers of user nodes, while keeping the other dimensions and the rest of parameters untouched, equal to the baseline (geographical area, number of gateways, LoRa modulation settings, etc.). With this, we want to understand 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 4.10 plots the percentage of user nodes that can transmit successfully $\geq$1, $\geq$2, or 3 LoRa packets to the application when the system has a high density of gateways (Fig. 4.10a) and is fully operational. The figure also plots the percentage of user nodes when there is a low density of gateways (Fig. 4.10b)—only 10 of them remain unaffected after the earthquake.

The orange line in Figure 4.10a represents the percentage of nodes correctly transmitting $\geq$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 $\geq$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 4.10b, 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 4.10 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 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 down-link 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 4.6.2 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



(a) High density of gateways (75)

(b) Low density of gateways (10)

Figure 4.10: 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 (4.10a) plots the full system operation with 75 gateways, while the right one (4.10b) corresponds to a downgraded operation in which only 10 gateways remain in operation. Note for labels on the x-axes: 1E2 $= 1 \times 10^2$, 1E3 $= 1 \times 10^3$, 1E4 $= 1 \times 10^4$.

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.

4.6.1.2 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 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

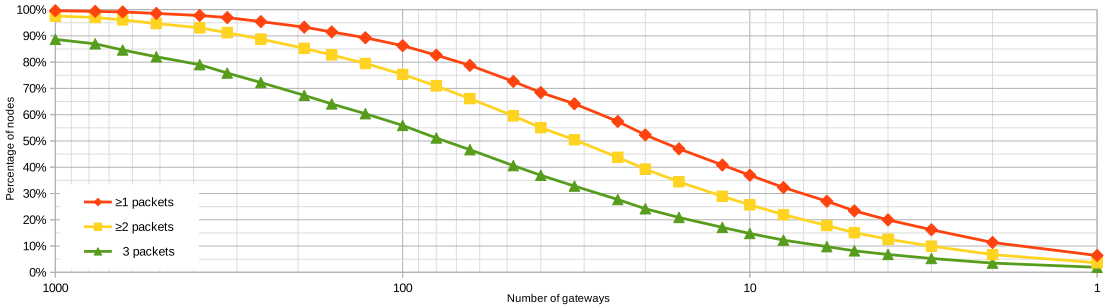


Figure 4.11: Percentage of user nodes transmitting successfully ≥1, ≥2 or 3 LoRa packets to the central application, in function of the number of gateways available in the system (ranging from 1.000 to 1).

sufficient and how the system performs in a downgraded status with fewer gateways available.

Figure 4.11 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 4.6.1.1, 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's success ratio.

While Figure 4.11 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.

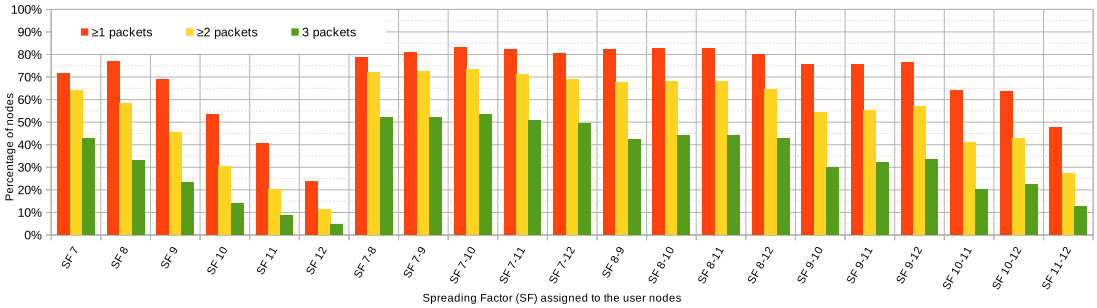
4.6.1.3 Spreading factor diversity

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.

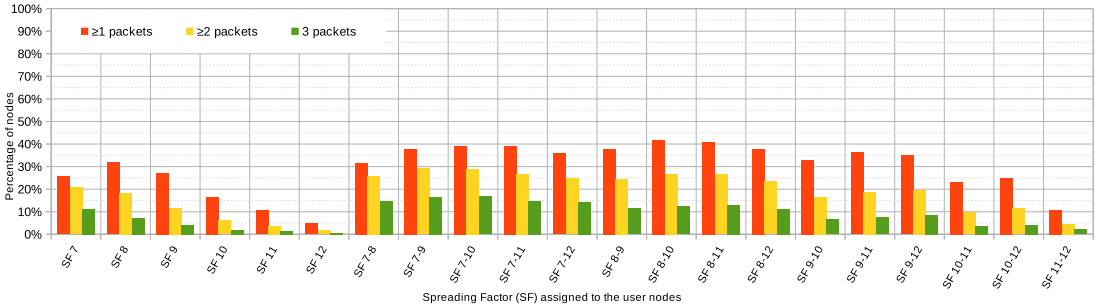
In this experiment, we changed the SFs the user nodes use to transmit data to the gateways, in order to investigate which value (or which combination of them) is more suitable for a given system, for both a high density and a low density of gateways cases. We first simulated the system with all the nodes 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 4.12 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 4.12a corresponds to the scenario of a high density of gateways, and Figure 4.12b to the lower one. The data show that choosing the smaller single SFs 7 to 9 (shorter range and time on air) provides significantly better results than bigger SFs.

Figure 4.12 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



(a) High density of gateways (75)



(b) Low density of gateways (10)

Figure 4.12: 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).

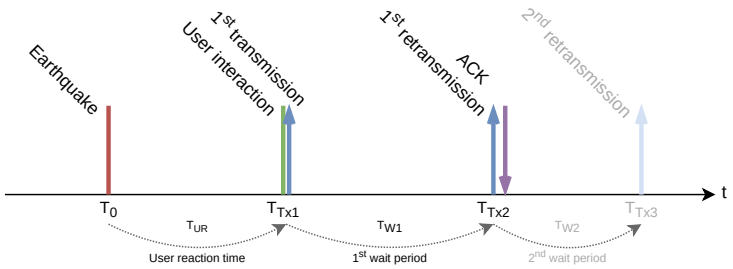


Figure 4.13: Timing of the user node activity in the aftermath of an earthquake, including ACK down-link messages.

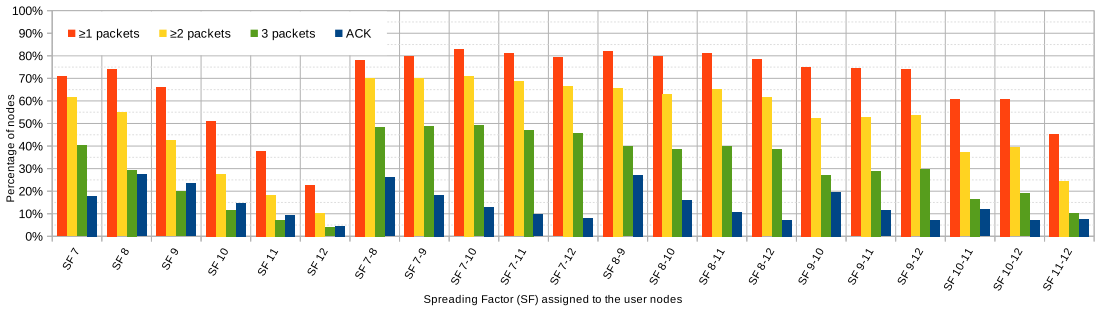
the default values predefined in Section 4.5.1 and used throughout this section, does not provide the best results. Indeed, for both higher and lower densities of gateways, choosing high SFs (11, 12) penalizes the performance figures, while avoiding them and concentrating more nodes on shorter SFs increases 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.

4.6.2 Down-link ACK from the application to user nodes

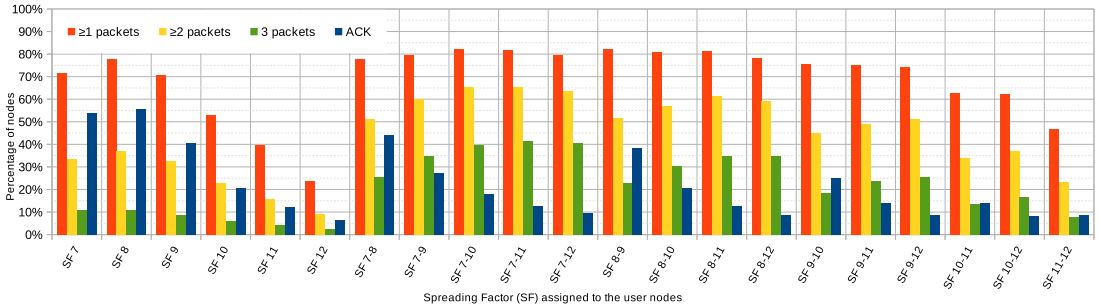
The communications system described in Section 4.3a and analyzed in Section 4.3 has a fundamental limitation: data flow is unidirectional. It goes only from the user nodes to the central application. While this design can help civilians give emergency response teams valuable information, so they can 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 down-link 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 piggybacking messages from other users (e.g., sent by family or friends). This would open the system up to more sophisticate communication patterns and alleviate users' concerns about the safety of their loved ones, mentioned in Section 4.1. This would also allow the implementation of software solutions that work under an IoP-based paradigm, in which citizens are the key actors.

Down-link 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 4.13 shows the schedule of events and actions occurring at one of the user nodes. Compared to the baseline scenario (see Figure 4.8), now a down-link 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. Down-link 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 piggybacking of actual messages sent from one user to another, bringing the LoRaMoto system closer to the IoP paradigm.



(a) Down-link ACK message, nodes continue retransmissions, high gateways density (75)



(b) Down-link ACK message, nodes stop retransmissions, high gateways density (75)

Figure 4.14: 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.

Among the different experiments we made for the baseline scenario in the previous Section 4.3, we chose here to focus on the effect of the ACK messages for various SFs. Figure 4.14 shows the percentage of user nodes that transmit successfully $\geq$1, $\geq$2 or 3 LoRa packets to the central application, as well as the percentage of nodes that correctly receive a down-link ACK message from it, in terms of the different SFs assigned to the user nodes. (This is similar to Section 4.6.1.3). For the upper graph (Figure 4.14a), the down-link ACKs messages are enabled, but user nodes keep retransmitting their packets; for the lower graph (Figure 4.14b), they stop retransmitting upon receipt of the ACK message. Comparing the results with the ones shown in Figure 4.12a, it can be seen that ACK messages cause a very small decrease in performance. (About 3 % fewer 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 4.14, 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 $\geq$1 messages 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.

4.6.3 Packet forwarding between user nodes

The analysis performed in Section 4.6.1 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 might be left out of the system. However, adding ACK messages and enabling bidirectional communication (explained in Section 4.6.2), 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

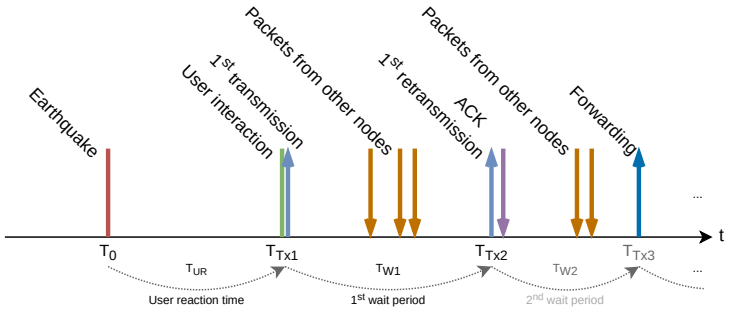
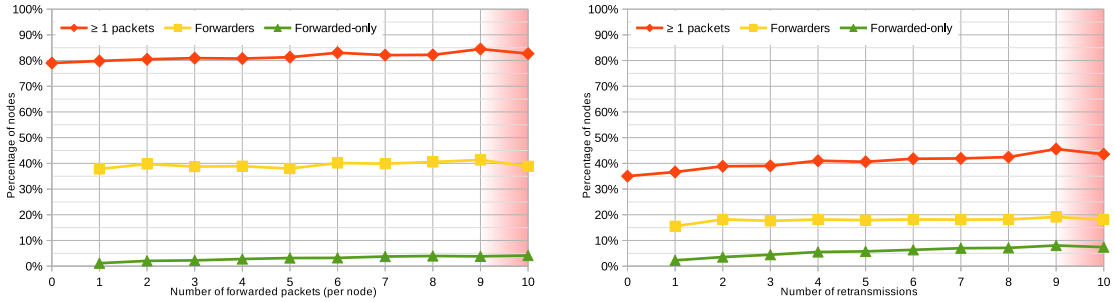


Figure 4.15: Timing of a user node activity in the aftermath of an earthquake, including ACK down-link messages and packet-forwarding.

packets (T_{W1}, T_{W2}, etc., as detailed in Section 4.5.1 and depicted in Figure 4.8), 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 down-link 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 forward from the storage buffer, without taking into account any aspect like arrival time, source node, and reception quality. Figure 4.15 shows the schedule of events and actions occurring at one of the user nodes. Compared to the baseline scenario (see Fig. 4.8), now a down-link ACK can reach the node and, depending on the configuration, will stop sending the remaining retransmissions, saving time on air for other nodes.

Figure 4.16 shows the effect of making home devices forward packets from neighboring nodes. The graphs indicate that packet-forwarding has a positive effect on sys-



(a) High density of gateways (75)

(b) Low density of gateways (10)

Figure 4.16: 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).

tem performance, as more individual home devices can reach the central application than when no packets are forwarded. For instance, on the left graph (Figure 4.16a, 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 4.16b, 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 later 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 4.16 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 should also be designed and tested to leverage the packets that have 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.

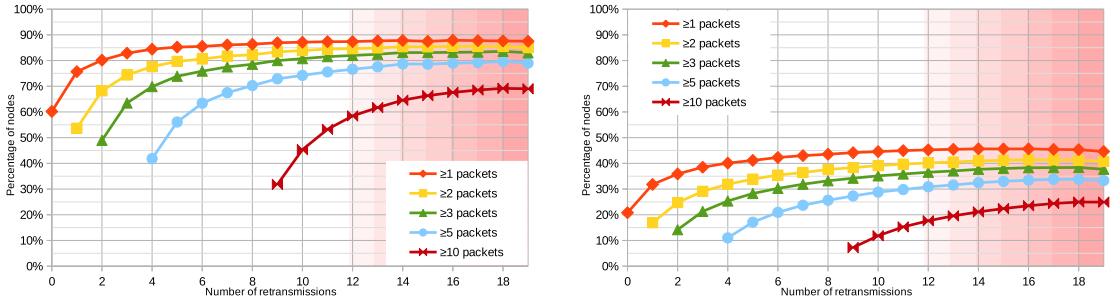
4.6.4 System sensitivity to design parameters

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.

4.6.4.1 Packet retransmissions

As detailed in Section 4.5, 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 4.8 on page 54. 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 down-link 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 4.17 plots the percentage of user nodes that can successfully transmit $\geq$1, $\geq$2, $\geq$3, $\geq$5 and $\geq$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 $\geq$1 grows. But the growth is asymptotic, never reaching the ideal 100 %. This saturated behavior is common for both a high density (Fig. 4.17a) and a low density of gateways (Fig. 4.17b) 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



(a) High density of gateways (75)

(b) Low density of gateways (10)

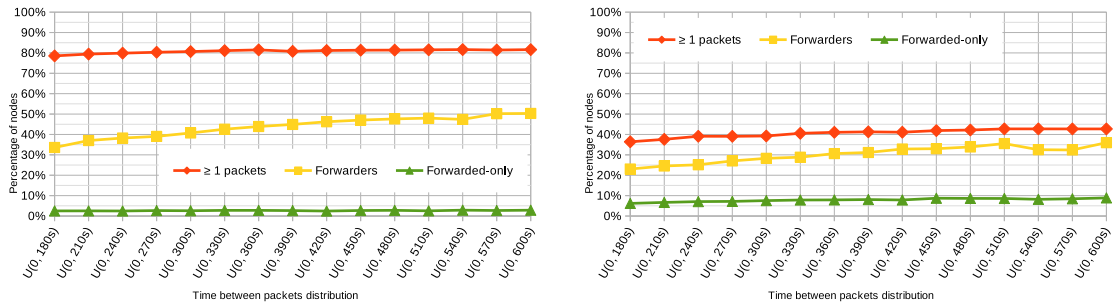
Figure 4.17: Percentage of user nodes transmitting successfully ≥1, ≥2, ≥3, ≥5 or ≥10 LoRa packets to the central application, in function of the number of allowed retransmissions per node (ranging from 0 to 19). The left graph plots the performance of a system operating with a high density of gateways (75); the right one corresponds to a low-density version with only 10 of them.

on air and bandwidth, and allowing for more retransmissions cannot help balancing that usage.

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 $3600\text{s}/max[0, 300]\text{s} = 12$. 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 4.17, where the trends suggest a slight decrease in the success ratio—hence the reddish background of the graphs.

4.6.4.2 Time between packets

As discussed in Section 4.6.1, 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 4.8. The waiting time between packets poses a trade-off in the system between its response speed (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



(a) High density of gateways (75)

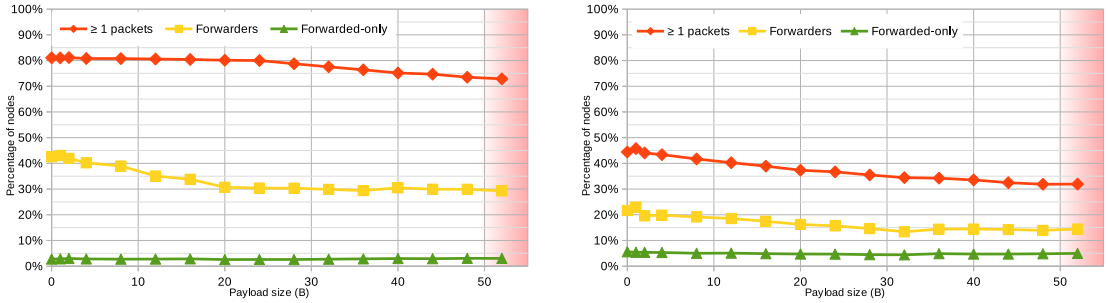
(b) Low density of gateways (10)

Figure 4.18: Percentage of user home devices transmitting successfully $\geq$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.

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 4.6.1 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 (the fastest case), and 0 and 600 s (the longest case), to understand how this parameter affects the system response.

Figure 4.18 plots the percentages of home devices that can communicate successfully with the application, for different waiting periods ranging from $U(0, 180\,\text{s})$ to $U(0, 600\,\text{s})$ (where $U(0, n\,\text{s})$ means uniform distribution from 0 s to n s). 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 4.18b) than with a high density one (Figure 4.18a). 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.



(a) High density of gateways (75) (b) Low density of gateways (10)

Figure 4.19: 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.

4.6.4.3 Payload size

When dimensioning the system parameters in Section 4.5.1, 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 Coding Rate (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 4.19 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 data points noted above) to 51 bytes (the biggest payload all SFs can accommodate [13]). 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

	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 4.3: 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).

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.

4.6.5 Summary of results

The first section of simulations results (Section 4.6.1) was a comprehensive exploration of the boundaries of the system based on the baseline scenario. Table 4.3 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, 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: down-link 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.

4.7 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 Wi-Fi or other radio technologies that have shorter ranges than LoRa (discussed in Section 4.2.1). 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 the LoRa technology to support communication among first responders in the field during emergency responses (discussed in Section 4.2). 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 [64, 14]. 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 4.4.2.

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 [14]. 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 4.6.3, 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.

4.8 Conclusions and future work

This chapter presented 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 to reduce 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.

Having shown how direct node-to-node communication and packet-forwarding can successfully contribute to system performance, we have tackled RQ2. However, 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.

Chapter 5

Mesh networks for IoT based on LoRa

The usage of stand-alone LoRa to build mesh networks, independently from the LoRaWAN architecture, has been explored by different authors (Sec. 3.2). Most of them focus their research on analyzing particular performance aspects or solving a specific problem for deploying a system in harsh environments. Only few of them address the topic from a broad and generalist point of view, but limited to experimentation with a small number of nodes.

In this chapter we present, evaluate and discuss a minimalistic DV RP for multi-hop LoRa mesh networks. This protocol has a simple design, so it can be run on resource-constraint embedded devices, and takes advantage of LoRa's radio technology features to calculate the best path between pairs of nodes. We focus on the SF, a modulation parameter that poses a trade-off between communication range and bit rate, generating quasi-orthogonal signals when different SFs are used. Under certain circumstances, this can allow for two or more transmissions to occur simultaneously on the same frequency. Furthermore, we exploit LoRa chips' CAD feature to detect radio transmissions and automatically tune the receiver to the incoming packet's SF.

We also introduce a novel Time on Air (ToA) metric, which takes advantage of the multi-SF characteristics of a LoRa mesh network, calculating the best route between two nodes based on the aggregated transmission time required along the whole path. In order to evaluate the RP and the different routing metrics, including ToA, we use the FLoRa framework [79] and the OMNeT++ simulator [78], testing it in different mesh network deployments with up to 64 nodes. We analyze key performance aspects such as PDR, throughput, latency, etc. under different set-ups, and compare our proposed metric with other well-known routing strategies.

The present chapter extends the work under review as of September 2021 in UR1.

5.1 Automatic multi-SF detection and reception with single-channel LoRa radio chips

As described in Section 2.2, LoRa radio chips are provided in two segments: single-channel transceivers for end nodes and multichannel digital baseband chips aimed at LoRaWAN gateways. Radio chips in the former category are designed to operate in a half-duplex fashion, using a single channel (i.e., a single frequency) and a single SF at a time. In the context of the LoRaWAN architecture, end nodes will typically need to transmit uplink only from time to time towards a gateway and, more rarely, receive down-link transmissions following a more or less planned frequency and SF scheme. The more powerful LoRa gateway chips, in the latter category, have more advanced Digital Signal Processing (DSP) resources that allow them to successfully demodulate several concurrent incoming transmissions on different frequencies and SFs. Indeed, they can also perform transmissions simultaneously if needed.

Leaving the LoRaWAN architecture aside, two or more end nodes with a single-channel LoRa transceiver can be used to communicate and exchange data. To do so, all nodes must be configured with the same radio settings – namely, frequency/channel, bandwidth, CR and SF. In the context of a mesh network like the one shown in Figure 5.6, nodes are typically deployed at different distances between them and with different surrounding environments. This means that each one of the possible links between pairs of nodes has different physical characteristics. This, when using LoRa, may mean that different *minimum* SFs can be required for each individual link. Therefore, to allow different nodes to interoperate in a given network, and since they are only able to use a single SF at a time, they all must be configured to use the same SF. Unfortunately, this *common minimum SF* will typically be higher than the minimum SF required on a per-link basis, forcing transmissions between nodes to be slower than they could actually be.

LoRa radio chips feature a CAD operation mode that is designed to detect incoming transmissions with minimum energy consumption. They do so by turning the receiver on to perform a quick scan of the radio channel, shutting it down and comparing the received signal with the ideal packet preamble waveform. This whole process is very fast (it takes around a symbol period to run), and is much more energy efficient than constantly attempting to receive a packet. The CAD feature can be used in a smart way by end nodes with single-channel LoRa transceivers that allows them to automatically detect, tune and successfully receive transmissions on any single SF (given the correct frequency/channel is configured). To do so, an algorithm on the end nodes continuously asks the transceiver to detect incoming transmissions and, when one is detected, starts a process that tries to perform the reception starting with the fastest SF available and, if unsuccessful, iteratively tries it on higher SFs. End nodes in a network, instead of using a fixed SF, can dynamically switch between different SFs in function of the incoming packet. As a result, links between pairs of nodes can operate at their minimum needed SF, not

limited by the requirements of the other links. For this to work reliably, however, the preamble length must be longer (around 12 symbols) than the default used by LoRaWAN (8 symbols).

The multi-SF strategy based on LoRa's CAD capability was initially exploited by *M. Westenberg* for the Single Channel LoRaWAN Gateway project, which turns a regular ESP8266/ESP32-based LoRa-capable embedded board into a LoRaWAN gateway [37]. Afterwards, *J. Braam* ported the code to Lua language so that it could run on the NodeMCU firmware usually found in ESP8266-based devices [38]. Later, as discussed in Section 3.2, *Kim et al.* proposed their ASFS scheme to enhance throughput by supporting multi-SF with single-channel LoRa modules [28]. In their implementation, the receiver repeats the preamble inspection three times in ascending order, starting with SF7. If the preamble is detected all three times in a specific SF, the receiver regards it as a candidate SF and attempts a packet reception. The authors claim their iterative solution can reduce false SF selection to 0 %, but it remains unclear whether this means that 100 % of the *detected* packets are correctly processed or 100 % of the *transmitted* packets are correctly demodulated by the receiver. Unfortunately, the source code related to the experiment is not available for replication.

5.1.1 Design and implementation

Motivated by the abovementioned proposals using multi-SF detection, we designed and tested our own proof-of-concept implementation, in order to validate whether this strategy could be leveraged at a hardware level for building LoRa mesh networks. To do so, we used several TTGO V2.1_1.6 [1] and T-BEAM V1.1 [2] embedded devices by the manufacturer LILYGO. The boards were programmed in the Arduino environment, performing the interaction with their SX1276 single-channel LoRa transceiver by means of the RadioLib library [82].

Our approach to LoRa multi-SF detection is based on an iterative process that continuously scans the radio channel using all the available SFs. Unaware of *Kim et al.*'s findings [28], instead of sequentially scanning each SF for a preamble (i.e., SF7→SF8→SF9→...→SF12→SF7...), we followed an alternating sequence which tests SFn twice as often as SFn+1 (i.e., SF7 is scanned twice the times SF8 is scanned, which in turn is scanned twice the times SF9 is scanned, etc.). The following piece of pseudocode shows the core of this simple algorithm:

```
#define MINSF m
#define MAXSF n
int SF[ARRSIZE] = {6, 7, 6, 8, 6, 7, 6, 9, 6, 7, 6, 8, 6, 7, 6, 10...
```

[1] LILYGO® TTGO LoRa32 V2.1_1.6: http://www.lilygo.cn/prod_view.aspx?Id=1270
[2] LILYGO® T-Beam V1.1: http://www.lilygo.cn/prod_view.aspx?Id=1281

```
int i = 0;
int packets = 0;

void loop() {
    lora.setSF(SF[i])s;
    preamble = lora.scanChannel();

    if ( preamble == PREAMBLE_DETECTED ) {
        String rxPacket;
        int rxState = radio.receive(rxPacket);

        if (rxState == ERR_NONE) {
            // packet successfully received
            processPacketString(rxPacket);
        }
    }

    do {
        i = (i+1) % ARRSIZE;
    } while ( SF[i] < MINSF || SF[i] > MAXSF );
}
```

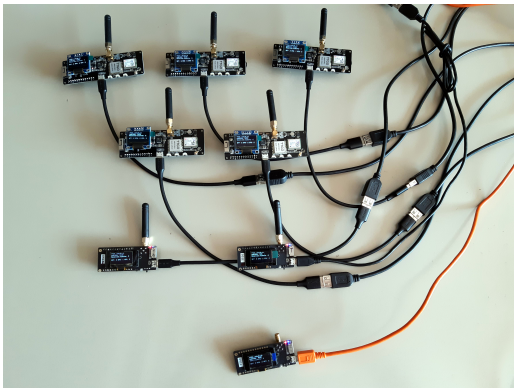
The complete source code for running the algorithm on the actual hardware using the Arduino environment is publicly available in a Git repository [3].

5.1.2 Evaluation

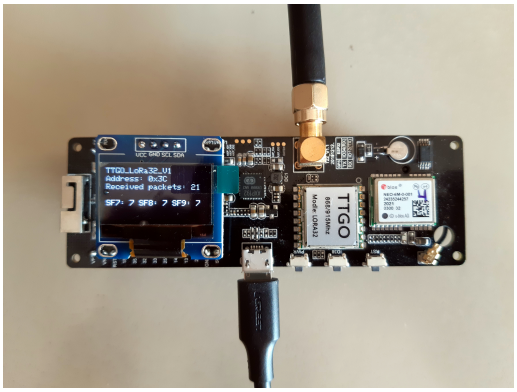
Our multi-SF detection algorithm was tested in a laboratory test with several TTGO LoRa32 and T-Beam devices. In our setup, one of the devices acted as a transmitter, sending packets in different SFs, while the rest of them acted as receivers with different configurations. Figure 5.1 shows them during one of the experiments.

The evaluation process consisted in sending packets using multiple SFs from a sender device (at least 100 packets per SF), and recording the reception statistics on the other devices. Different combinations of SFs were tested in both sender and receivers (e.g., sending on SF7 and SF8 and attempting reception on SFs 6 to 9). We observed that, when both sender and receivers were configured to transmit or listen to the same range of three consecutive SFs (e.g., SF7, SF8 and SF9), up to $99,2\,\%$ of the packets were correctly received. Also, making the receivers listen on SFs that were not being used by the transmitter, reduced the reception rate around $5\,\%$: this should be taken into account, to avoid nodes in a real LoRa mesh network drop packets by listening to unused SFs. Using a broader range of 4 consecutive SFs

[3] https://gitlab.com/rogerpueyo/arduino-loracad-with-radiolib-receiver



(a) 5 TTGO T-Beam and 2 LoRa32 devices (top) acting as multi-SF receivers, and one LoRa32 (bottom, orange cable and no antenna) acting as sender.



(b) Detail of a TTGO T-Beam device acting as a receiver. Besides the serial connection, the OLED display is used to show packet reception counters.

Figure 5.1: Experimental evaluation of the multi-SF detection algorithm in the laboratory.

Tx	Rx 1	Rx 2	Rx 3	Rx 4	Rx 5	Rx 6
SF7,8	SF7,8	SF7,8	SF7,8	SF6,7,8	SF7,8,9	SF6,7,8,9
870 pkt.	99,1 %	98,7 %	99,5 %	94,6 %	94,7 %	92,6 %

Table 5.1: Multi-SF packet reception results for a sender (Tx) transmitting on SFs 7 and 8, and receivers (Rx) listening on different SF ranges.

(e.g., SF6 to SF9) offered a slightly worse reception figure of 96,7 %. Tables 5.1, 5.2 and 5.1 summarize the results obtained in the experiments performed with different sender and receiver configurations.

The results obtained in our experiments were very similar to those published by *Kim et al.*, even if different multi-SF detection strategies used are different. A possible refinement to our strategy would include performing more than one preamble inspection before attempting a reception, similar to how it is done in the reviewed literature [28]. These two independent proposals validate, from the hardware perspective, the possibility to use multiple SFs with devices featuring single-channel

Tx	Rx 1	Rx 2	Rx 3	Rx 4	Rx 5	Rx 6
SF7,8,9	SF7,8,9	SF7,8,9	SF7,8,9	SF6,7,8,9	SF7,8,9,10	SF7,8,9,10
1997 pkt.	98,7 %	99,2 %	98,9 %	93,8 %	94,3 %	89,7 %

Table 5.2: Multi-SF packet reception results for a sender (Tx) transmitting on SFs 7, 8 and 9, and receivers (Rx) listening on different SF ranges.

Tx	Rx 1	Rx 2	Rx 3	Rx 4	Rx 5
SF7,8,9,10	SF7,8,9,10	SF7,8,9,10	SF7,8,9,10	SF6,7,8,9,10	SF7,8,9,10,11
2524 pkt.	94,0 %	96,9 %	96,6 %	95,1 %	90,7 %

Table 5.3: Multi-SF packet reception results for a sender (Tx) transmitting on SFs 7, 8, 9 and 10, and receivers (Rx) listening on different SF ranges.

LoRa transceivers. This allows for building mesh networks like the one depicted Figure 5.6, where each link between pairs of nodes will be capable of using the fastest SF possible, regardless of the rest of the links' characteristics. Furthermore, the usage of the multi-SF strategy provides an initial answer to RQ3 regarding how LoRa's specific features can be used to build flat, decentralized mesh networks.

5.2 A minimalistic Distance-Vector Routing Protocol for LoRa mesh networks

In order to deploy and assess multi-hop LoRa mesh networks, we designed and implemented a proactive, hybrid Layer 2/3 (L2/3) DV RP that takes advantage of this radio technology's specific characteristics. Our design principle was to keep complexity to the minimum needed, so that it could be run on resource-constraint embedded devices featuring a microcontroller and a LoRa radio. Besides that, the RP can benefit from LoRa's SFs different range and orthogonality properties, which allow for concurrent transmissions between different pairs of nodes. To achieve it, we also introduced a novel multi-SF-aware ToA metric that minimizes the total transmission time for a packet to reach the destination, which improves performance aspects under certain network conditions.

Our protocol's main features are:

- Distance-Vector: best routes towards any destination are calculated in a distributed way across nodes, based on the information provided by the neighbors.
- Proactive: network nodes periodically broadcast available routes, independently of data traffic, refreshing routes and keeping them up-to-date and readily available.
- Layer 2+3 hybrid: by aggregating the two layers into one, we simplify the architecture and the requirements for low-power embedded LoRa devices.
- Duty cycle-aware: for networks operating in unlicensed ISM bands, time-on-air limitations in the form of duty cycles are usually enforced; these restrictions can be embedded into the route metrics calculations.
- Lightweight, flexible and configurable: many aspects of the protocol (metric, packets timing, etc.) can be fine-tuned to fit specific use cases.

- Concurrent, overlaid networks: thanks to the orthogonality properties of LoRa's SFs, several "virtual" layered networks can operate on the same radio channel, providing higher global throughput.
- ToA metric: route costs are calculated based on the end-to-end packet transmission time, taking multi-SF capability into account.

5.2.1 Network topology and addressing

The RP builds a flat mesh network topology, with no hierarchical differentiation between nodes, regardless of their hardware characteristics or their role at the application level. To participate in the network, a node runs an instance of the RP, generates and keeps a local routing table, and exchanges routes with its neighboring nodes periodically

Figure 5.2 shows the topology of a sample LoRa mesh network running the RP. One-hop, direct communication between a given pair of nodes can usually be performed with different SFs; the diagram indicates the fastest (i.e., the lowest) one that makes the link possible. Communication between distant nodes that are not directly connected is made by multi-hop packet forwarding, using the routes calculated by the RP. While most of the links in the diagram are symmetric, a few of them require different SFs in each direction to achieve a successful communication. This can happen in scenarios with heterogeneous hardware or environmental conditions, either temporary or permanent. As a result, packets traveling between a given pair of nodes could use different routes in each direction.

To simplify the design of the RP and the applications built upon, we merge the data link (Layer 2 (L2)) and the network (Layer 3 (L3)) layers into a single one (compared to, e.g., the Wi-Fi 802.11+IP stack, where addresses of different types are used on each layer). This solution reduces overhead on network traffic and computing effort on the nodes, although it may limit direct interoperation with other networks (i.e., the Internet). In our design, network addresses take 2 bytes, ranging from 0x0000 to 0xFFFF and resulting in up to 65.536 usable addresses per network. We consider this specific addressing space size to be a convenient value between the ability to build a large mesh network, LoRa's throughput and range performance, and ease of implementation. However, it could be reduced or increased when implementing it on particular scenarios. For example, a smart metering network covering a city with hundreds of thousands of nodes may require a larger addressing space but, in this case, other strategies (e.g., network partitioning) could be more effective.

The RP supports transmission of network packets in unicast, to a given single node using its unique network address, or broadcasting to all the neighbors at one hop using a broadcast address. For the latter mode, the highest address (0xFFFF) is reserved to broadcast routing packets only. Reception acknowledgments, packet retransmissions, etc. are left to be implemented in higher layers. The RP does not

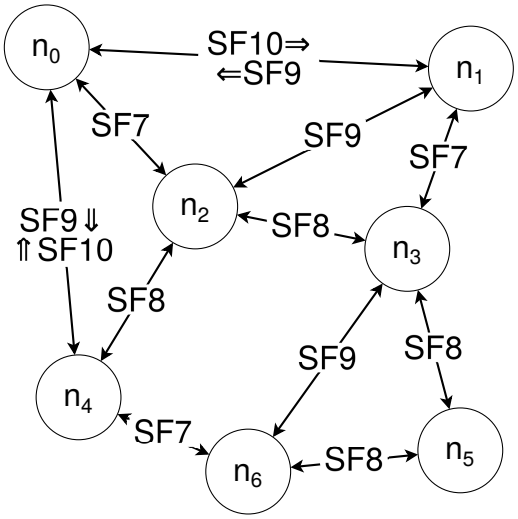


Figure 5.2: Topology of a sample LoRa mesh network. Links between each pair of nodes may use any valid SF (the diagram shows the smallest and fastest possible to create a working link). Most links are symmetrical, but others require different SFs on each direction; this may generate different routes for each direction between a pair of nodes.

handle them, to keep it as simple as possible and reduce its footprint on embedded nodes. Therefore, neither one-hop or end-to-end delivery are guaranteed at the routing level.

5.2.2 Network packets structure

Data and routing packets have a very similar structure, which is schematically depicted in Figure 5.3. Since packet size is constrained by the LoRa hardware to a maximum of 256 B, our protocol uses a minimalistic approach to reduce the overhead introduced by the header, which takes 7 B at most.

The first 4 B of a data packet hold the address of the node originating the message (*Source*, 2 B) and the address of the final recipient (*Destination*, 2 B). The next field (*Via*, 2 B) is used in multi-hop data packets to indicate the address of the next hop in its route. Its value changes from hop to hop, as the packet is forwarded by intermediate nodes. In the last hop in the route, or for single-hop data packets, the *Via* field should be the same as the *Destination*. The *Flags, Time to Live (TTL)* space (1 B) is reserved to tag packets (2 b) if required by the upper application layer, and to account for the TTL (6 b) This effectively sets a maximum number of hops to 64 to any packet. Last, the *Payload* field holds the actual application data or the RP exchanged between different nodes (see next section), holding up to 249 B.

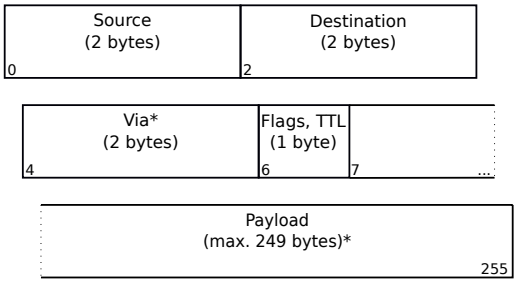


Figure 5.3: Structure of a data/routing packet. *Source* and *Destination* bytes identify the communication end points. The *Via* field is omitted when the broadcast address (0xFFFF) is set as the destination. The *Flags/TTL* byte is used to identify the packet type and watch its lifespan. Depending on the packet type, the *Payload* field can carry either the routing protocol information or the application data.

Routing packets are differentiated from regular data packets because they use the broadcast address in the *Destination* field, instead of a unicast address. Therefore, the *Via* field is not necessary, and is hence omitted to save 2 B that can be assigned to the payload, which becomes up to 251 B long.

5.2.3 Routing table

Each node in the network runs an instance of the routing protocol, creating a local routing table that is constantly updated as messages from neighboring nodes are received. Being a DV protocol, the table consists of a list of all the nodes known to be in the network, the neighbor through which they can be reached, the path cost and the route expiry time. With the information contained in the table, every node is theoretically able to communicate with any other node in the network, either directly (if they are neighbors) or indirectly using multi-hop. Furthermore, every node is also able to forward multi-hop traffic that goes through it towards its destination.

Besides the routes towards other nodes, each node also keeps track of the routes from neighbor nodes to it. Since the physical connection between two neighbor nodes may not be symmetrical, the RP also needs to feed back neighbor nodes with the information of its inbound links, so that they can correctly calculate the routes towards it.

Table 5.4 shows a snapshot of the routing table for node n_0 from Figure 5.2. In the first block, the three entries on top show the direct, single-hop routes from node n_0 to its neighbors n_1, n_2 and n_4 (indicated by *Dest=NextHop*, but using a different SF). The next entries show multi-hop routes to other routes in the network (indicated by *Dest≠NextHop*). This includes the nodes that are not directly reachable using

Dest	**NextHop**	**SF**	**Cost**	**Expiry**
0x0001	0x0001	10	-	189
0x0002	0x0002	7	-	293
0x0004	0x0004	9	-	251
0x0001	0x0002	-	4	293
0x0003	0x0002	-	3	293
0x0004	0x0002	-	3	293
0x0005	0x0002	-	5	293
0x0006	0x0002	-	4	293
0x0002	0x0001	-	4	189
0x0003	0x0001	-	12	189
0x0004	0x0001	-	9	189
...	...	-	...	...
Dest	**Source**	**SF**	**Cost**	**Expiry**
0x0000	0x0001	9	-	189
0x0000	0x0002	7	-	293
0x0000	0x0004	10	-	251

Table 5.4: Routing table for node n_0 from Figure 5.2. The first block contains the routes from n_0 to all other nodes. The second block corresponds to routes from neighbor nodes to n_0 (required to properly account for asymmetric links).

one hop (n_3, n_5 and n_6) and also those which have a better metric using a multi-hop path rather than the direct link (this is the case for n_1 and n_3). For instance, the direct single-hop path $n_0 \Rightarrow n_4$ has a higher ToA cost than the multi-hop path $n_0 \Rightarrow n_2 \Rightarrow n_4$.

The second block of entries in Table 5.4 shows inbound routes to node n_0. Node n_0 does not actually use them for its own routing decisions. However, it must keep track of them, and let its neighbor nodes learn about them. This way, neighbors can be aware of what is the minimum SF required to reach n_0 and correctly calculate their routes towards it.

Upon the successful transmission of a new routing packet, the receiving node processes it and refreshes its local routing table, updating the information accordingly. First, the link characteristics with the source neighbor node are added to the routing table, or refreshed if already present. Then, one by one, the announced routes are processed. Depending on the case, they will be added to the routing table, used to update or replace known routes, or discarded if not useful. The diagram in Figure 5.4 shows this process as a flow chart.

As exemplified by Table 5.6, a routing table may contain different entries with routes to the same node, via different next hops. While the RP will use the one with the

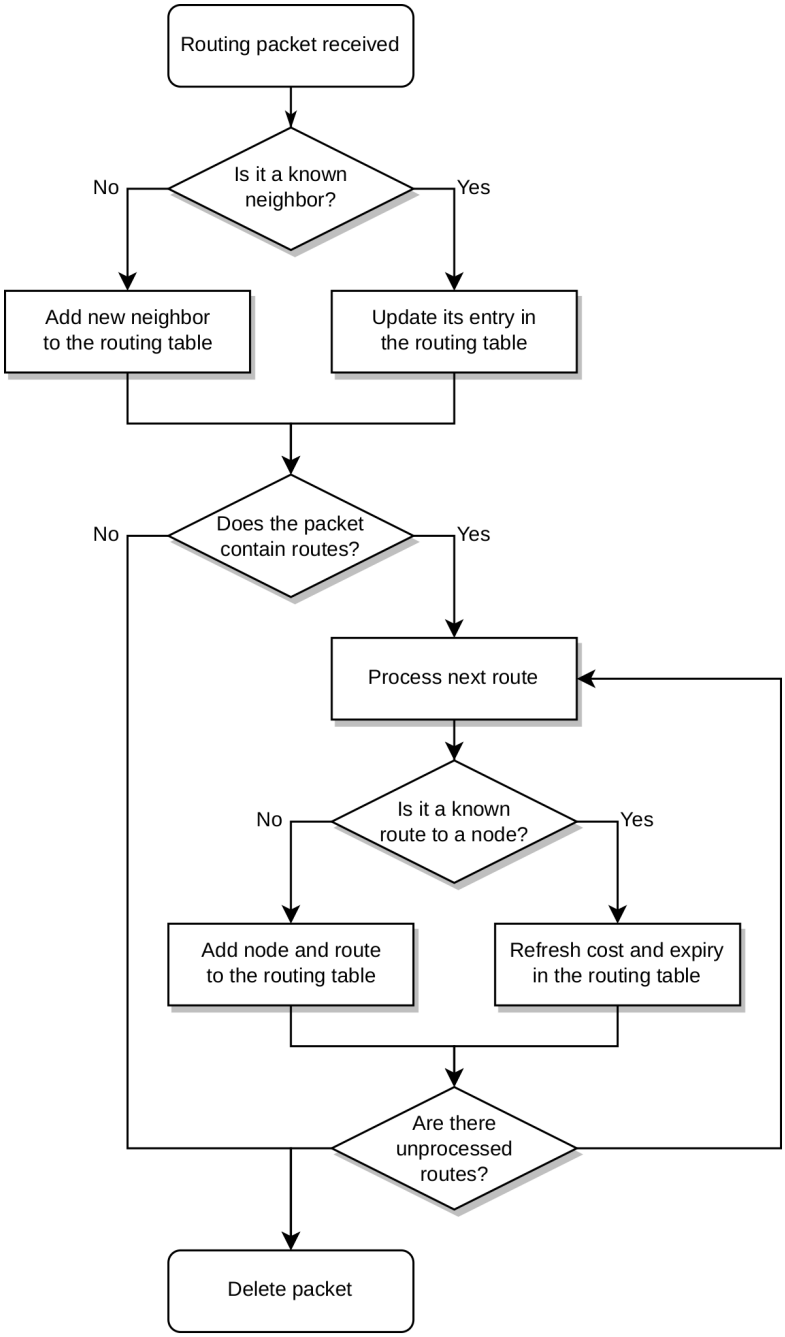


Figure 5.4: Flow chart upon reception of a new routing packet at a node.

lowest cost by default, this provides alternatives in case the main route suddenly increases its cost or expires. However, to limit the growth of the table unnecessarily, a maximum number of total and per node routes can be established.

5.2.4 Routes exchange between neighbor nodes

Nodes running an instance of the RP periodically broadcast routing packets to their neighbors. This way, they proactively generate the network topology and keep it updated, by refreshing their routing tables locally as packets from neighbor nodes are received. These packets include:

- the source node's information (i.e., its address)
- a 6 b incremental counter (instead of the TTL, which can be used to evaluate packet loss and collisions on a link, to infer its quality or its occupation)
- an excerpt of the source node routing table (i.e., a list with the best routes and their path cost to the other nodes in the network).

With this information and the details from the LoRa radio physical layer (namely, the SF used to transmit the message) the nodes that receive the broadcast packet update their local routing tables accordingly. In turn, these nodes will propagate their routes further away, to their neighbors, when they send their own broadcast messages.

The devices running the RP are expected to have single-channel LoRa radios and use the technique described in Section 5.1 to enable multi-SF reception. Therefore, they can send and receive packets on any SF available between SF_{min} and SF_{max}. The RP can use single-SF metrics, where all the nodes are expected to use a single, network-wide SF, and multi-SF metrics, where nodes may dynamically switch between SFs depending on their current status.

The transmission time for a packet when using a given SF is double the time required by the immediately lower SF. Therefore, when using multi-SF metrics, routing packets are sent using the following strategy. Broadcast messages on any given SF are sent twice as often as on the immediately higher SF (i.e., broadcast packets using SF9 are sent at double the rate than packets using SF10). This balances the cost of using different SFs, and also helps keeping routes using faster links more up to date than those using slower links. Broadcast packets on different SFs are sent in

random order (rather than sequentially), using the following probability formula:

$$p(\mathrm{SF}_n) = \frac{2^{\mathrm{SF}_{max}-\mathrm{SF}_n}}{\sum_{i=min}^{max} 2^{\mathrm{SF}_i-\mathrm{SF}_{min}}}$$

$$\text{since:} \quad \sum_{i=min}^{max} p(\mathrm{SF}_i) = 1 \qquad (5.1)$$

$$p(\mathrm{SF}_{min}) = 2^{\mathrm{SF}_{max}-\mathrm{SF}_{min}} \cdot p(\mathrm{SF}_{max})$$

Sending packets on different SFs randomly, instead of using a predefined sequence, also helps to avoid repeated collisions if nodes become synchronized.

The following pseudocode implements the random selection of a SF between SF_{min} and SF_{max} with the probability described above:

Algorithm 1: Random selection of the SF on which to broadcast a routing packet.

```
SF = 0;
while SF == 0 do
    thisSF = SFmin;
    while thisSF ≤ SFmax do
        if random(0, 1) ¡ 0.5 ) then
            SF = thisSF;
            break;
        else
            thisSF++;
        end if
    end while
end while
```

As mentioned in Section 5.2.2, routing packets are very similar to data packets, except that they are single-hop and use the broadcast address as the destination, skipping the *Via* field. Their payload contains the network routes, as advertised by the source node. Figure 5.5 show an example of a routing packet, (corresponding to node n_0 from Fig. 5.2), where the payload contains the best routes contained in the routing table from Table 5.4. In case the routing table does not fit inside the payload of a packet, routes to transmit are chosen randomly by assigning each of the routes a transmission probability, in function of its *Cost* value. This design decision makes routes to nearby nodes to be more updated than those to distant devices. Therefore, to avoid flapping, route timeouts must be configured taking into account the number of nodes and the topology of the specific deployment.

Nodes, upon reception of a new routing packet, process it and refresh their local routing table, updating it accordingly. First, the link characteristics with the source

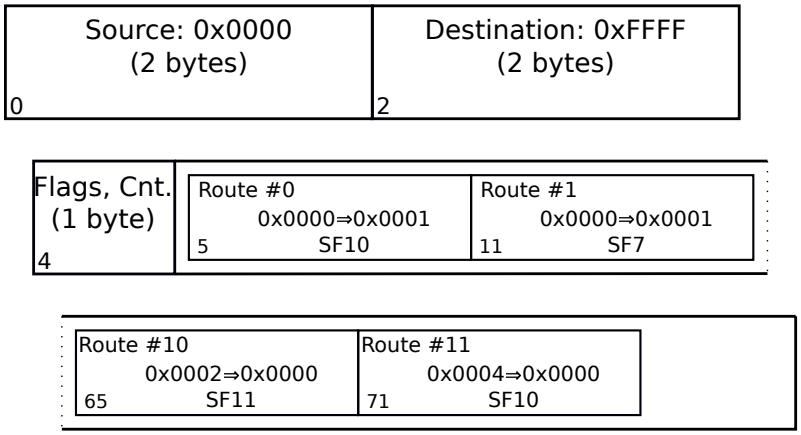


Figure 5.5: Sample routing packet as sent by node n_0 from Fig. 5.2. The *Destination* address field indicates a broadcast packet destined to its neighbors. Routes in the payload correspond to those in the routing table from Fig. 5.4.

neighbor node are added or refreshed in the routing table. Then, one by one, the received routes are processed and added or updated in the routing table. The diagram in Figure 5.4 shows this process as a flow chart.

5.2.5 Routing metrics

Our RP can be configured to use different routing strategies, in particular regarding the metric used to calculate the costs of the possible paths towards a destination. These are used to determine which node a packet will be forwarded to in order to reach its destination. The protocol can use a flooding strategy, single-SF metrics (like the well-known Hop Count (HC) or Expected Transmission Count (ETX)) or our proposed ToA multi-SF metric.

5.2.5.1 Packet flooding

The most simple multi-hop operation for the RP consists of a packet *flooding* strategy, with no actual routing involved. Nodes keep, in a finite buffer, a copy of the last packets received and broadcast, to avoid double broadcasts that would cause traffic amplification.

A *smarter* version of the *flooding* strategy, still not involving actual routing, keeps track of neighboring nodes as they send data packets, and can use unicast transmissions for the very last hop.

5.2.5.2 Single-SF metrics

Our RP implements four metrics for LoRa mesh networks where nodes use a single network-wide SF. The first and simplest one is HC, for which the cost of a given path is equal to the number of hops required to reach the destination.

The second metric is ETX, for which the cost of a given path is depends on the links along the route, proportional to the number of transmissions required to reach the destination. The quality of a link between two nodes is calculated based on the number of lost routing packets over a period of time (the more lost packets, the worse the link is and the higher its cost).

Two more single-SF metrics are available, which calculate path costs based on the physical links' RSSI. The *RSSI sum* metric uses the reception quality measure to calculate a route cost as a sum of RSSIs along the path:

$$\sum_{h=0}^{n} |min(\mathrm{RSSI}_h,\ -1)| \tag{5.2}$$

so that links with lower signal reception quality incur into a higher route cost. Likewise, the *RSSI product* metric uses the reception quality measure to calculate a route cost as the product of RSSIs along the path:

$$\prod_{h=0}^{n} |min(\mathrm{RSSI}_h,\ -1)| \tag{5.3}$$

5.2.5.3 Multi-SF ToA metric

The SF is a key element of the LoRa radio technology, as it poses a trade-off between the transmission reach and the time required to send a packet. Roughly, switching to a SF one step higher (e.g., SF7→SF8) doubles the transmission time (or halves the transmission speed, see Fig. 5.8), while increasing distance between $\times 1,25$ and $\times 1,4$ (see Table 5.5). Throughput, or transmission time, also have a direct relation with the power required to transmit a packet, which is specially critical in battery-powered devices. This is usually the case in the context of LPWAN for the IoT domain, where radio channel occupation and power (i.e., energy) are scarce resources.

Taking the abovementioned restrictions into account, we propose a Time on Air (ToA) routing metric to evaluate the cost of a path towards a destination that aims at minimizing the total end-to-end time required to transmit a packet, from source to destination, and the radio channel occupation.

The ToA metric in our RP calculates the path cost towards a node in function of the SFs used by the successive links between forwarding nodes, until reaching the destination. It ponders the cost of the hop $h_{l,m}$ between two neighbor nodes n_l and n_m as a power of 2:

$$\mathrm{h}_{l,m} = 2^{\mathrm{SF}_{l,m} - \mathrm{SF}_{min}} \tag{5.4}$$

SF	Rx sens.	P_{Tx}	BW	Dist.	Δdist.
7	−124 dBm	20 dBm	125 kHz	250 m	*n/a*
8	−127 dBm	20 dBm	125 kHz	349 m	×1, 4
9	−130 dBm	20 dBm	125 kHz	487 m	×1, 4
10	−133 dBm	20 dBm	125 kHz	679 m	×1, 4
11	−135 dBm	20 dBm	125 kHz	848 m	×1, 25
12	−137 dBm	20 dBm	125 kHz	1058 m	×1, 25

Table 5.5: Maximum reach with different SFs.

where $\mathrm{SF}_{l,m}$ is the smallest SF required to successfully transmit between nodes n_l and n_m, and SF_{min} is the minimum SF available in the system. [4] Therefore, we calculate the cost of a path between two arbitrary nodes n_i and n_j, with H intermediate hops, as:

$$\mathrm{ToA}_{i,j} = \sum_{k=1}^{H} \mathrm{h}_k = \sum_{k=1}^{H} 2^{\mathrm{SF}_k - \mathrm{SF}_{min}} \tag{5.5}$$

where SF_k corresponds to the SF used in each of the intermediate H hops in the route. In case two or more paths are available with the same metric, the one with the next hop using a smaller SF is preferred. Still, in case of tie (same metric and same SF in the next hop), the path is chosen randomly among the contenders.

The example in Figure 5.2 uses SF7 as the smallest SF available (SF_{min}). Using the ToA metric, the cost of the direct single-hop path from node n_0 to node n_1 would be $2^{10-7} = 8$. Instead, the three-hops path via nodes n_2 and n_3 would be preferred, since its ToA metric would be $2^{7-7} + 2^{8-7} + 2^{7-7} = 1 + 2 + 1 = 4$. The usage of a simple additive metric calculation is convenient in the context of resource-constraint IoT devices, which are often driven by 8 or 32 bit MCUs. This way, these operations can be easily implemented and require few processor cycles.

Section 5.4 evaluates the metric in depth in a simulation environment and compares it with other well-known metrics.

5.2.6 Loops mitigation

Our RP is subject to routing loops (e.g., in the case of a node disappearing from the network), which is a common problem in DV algorithms. To mitigate them, or their effects, we implement the following mechanisms.

[4] The smallest SF available in LoRa is SF6. For practical reasons, the minimum SF commonly used is SF7, but a higher value could be preferred.

5.2.6.1 Route poisoning

If a direct route (i.e., to a one-hop neighbor) reaches the timeout on a given node, it broadcasts this route with the maximum possible cost. This way, nodes will sequentially drop the expired route and propagate the action along the network.

Given that routing packets are transmitted without reception guarantees, this mechanism may have limited impact on loops mitigation, especially on busy networks with high packet collision probability. As a possible workaround, *poisoned* routes could be broadcast twice, or more, before being deleted locally.

5.2.6.2 Hold-down

A node, upon learning about an unreachable route (e.g., a failed neighbor), sets up a timer. During this period, the node ignores incoming information about the route, avoiding an infinite count situation.

This mechanism may be only partially effective, depending on whether a node is able to detect or receive a route's unreachability before starting an infinite count. Additionally, it may increase network convergence times.

5.2.6.3 Maximum route cost

Routes have a finite maximum cost, and those reaching this value are discarded. Therefore, in the event of an infinite count loop, affected routes are eventually discarded.

This parameter poses a trade-off between how fast loops are fixed and how big a network can be, in terms of routing (i.e., maximum number of hops).

5.2.6.4 TTL

Data packets have a maximum hop count of 64 hops. However, this limit can be set to a lower one, like the number of known nodes in the network (if smaller than 64). This way, packets entering a routing loop will be discarded sooner, as their TTL is exhausted.

5.2.6.5 Avoid duplicate transmissions

When the flooding mechanism is used (Section 5.2.5.1) instead of actual routing, nodes hold a copy of the last packets they forwarded in a finite buffer. If the packet is received again to be forwarded, it is discarded, avoiding duplicates and loops. This

Scope	Parameter	Range	Default
LoRa	Minimum SF	SF7, ... SF_{max}	SF7
LoRa	Maximum SF	SF_{min}, ... SF12	SF12
Routing	Metric	BC [5], HC, ETX, RSSI (sum/product), ToA	ToA
Routing	Average routes broadcast period (SF_{min})	$0 \ldots \infty$	60 s
Routing	Routes expiry time (SF_{min})	$0 \ldots \infty$	300 s
Routing	Max. routes to a node	$1 \ldots n$	2
Routing	Max. total routes	$0 \ldots 1024$	1024
Routing	Routing/Data traffic prior.	$n : 1$	10 : 1
Routing	Forward/Local traffic prior.	$n : 1$	10 : 1
Regulation	Duty cycle (%)	0.1 – 100	100

Table 5.6: Configuration options for the RP.

mechanism is also employed when actual routing is put into place, and complements the TTL embedded in the packets themselves to avoid them entering in routing loops.

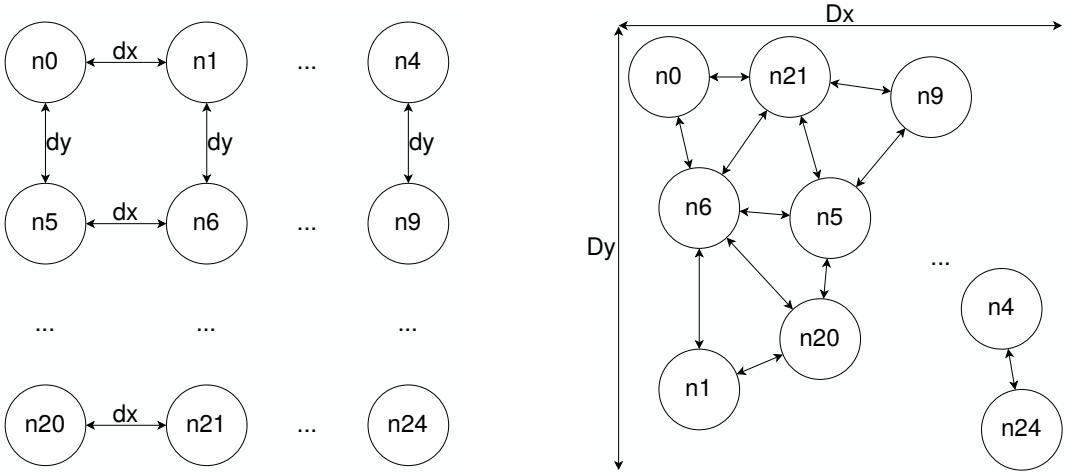
5.2.7 Protocol configuration

In order to adapt to specific use cases and conditions, many parameters of the RP can be configured. This section summarizes the most remarkable ones for the RP operation.

First, several LoRa-related aspects can be customized. For instance, the maximum SF to use can be changed (from the default SF12 to a smaller one, which could be required in certain regulatory domains). Instead of the default ToA metric, other well-known ones can be chosen (HC, ETX, etc.). Other aspects, like the expiry time of a learned route can be modified, to ensure the liveliness of the information contained in the routing table. Table 5.6 shows these configurable parameters. Regarding data and routing packets, their transmission priority at a node can be adjusted in an $n : 1$ ratio. Similarly, forwarded and local data packets can be prioritized according to the application needs.

5.3 Methodology

In this study we use OMNeT++ [78], an extensible, modular, component-based C++ simulation library and framework, in combination with FLoRa [79], a framework to carry out end-to-end simulations for LoRa networks. OMNeT++ is a well-



(a) Symmetric grid topology, with equal horizontal (d_{x}) and vertical (d_{y}) spacing between nodes.

(b) Random topology, with nodes distributed over a $D_{\mathrm{x}} \times D_{\mathrm{y}}$ area using a uniform density probability function.

Figure 5.6: Depiction of the network topologies used in the simulations.

known discrete event simulator framework used by a lively academic community and, together with FLoRa, it provides a complete implementation of the LoRaWAN architecture [80] and an accurate model of the LoRa radio physical layer derived from previous experimental findings [81].

We have stripped-down the LoRaWAN functionalities from the FLoRa framework and added direct communication and packet forwarding between end nodes, without the need for a gateway. This includes adding a DV routing protocol with different path cost calculation metrics. Furthermore, we have added to the framework other interesting features, like the CAD found in other implementations [28, 37]. The source code for our project, rebranded as FLoRaMesh, is publicly available on GitLab [83].

5.3.1 Network topologies

The simulations consist of a network with a variable number of nodes, arranged in two different topologies: an $N \times N$ symmetrical grid topology with equal vertical and horizontal distance (Fig. 5.6a) and a random topology with N^2 nodes uniformly distributed over a delimited square area (Fig. 5.6b).

Each of these two topologies serves a different evaluation purpose. On the one hand, the grid topology with a constant distance between nodes offers a regular and predictable environment. Therefore, once the appropriate SF and transmission power are set, single SF routing provides communication to either all or none of the

nodes, and performance will only depend on the metric properties (rather than on the network topology characteristics). This also allows checking if multi-SF routing can offer any advantage (i.e., whether if using one given SF and concurrently, in addition, higher ones, has any impact on performance, compared to using just one). On the other hand, the random topology allows measuring how multi-SF routing adapts to heterogeneous situations, and how using different SFs in different parts of the network (i.e., trying to use the fastest SF possible for each link between pairs of nodes) performs compared to using network-wide common SF that provides communication to all the participating nodes.

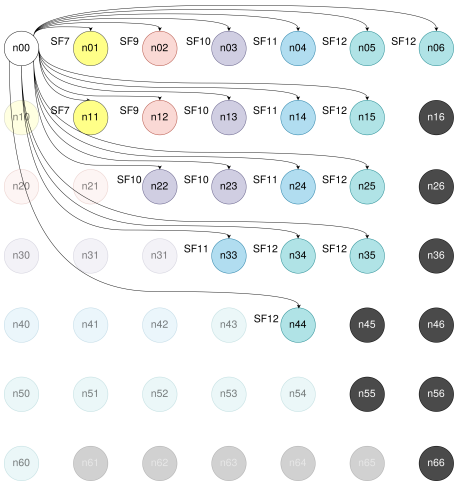
When nodes are arranged following a grid topology, four different spacing between them are used (both vertical and horizontal): 177 m, 178 m, 247 m and 248 m. These values are not arbitrarily chosen, but have a specific purpose, as depicted in Figure 5.7. A spacing between of 177 m allows nodes using the shortest-range SF7 to communicate with their adjacent nodes in horizontal, vertical, and diagonal (Fig. 5.7a). When the spacing is increased to 178 m, diagonal communication with adjacent nodes is no longer possible with SF7 (Fig. 5.7b), only vertically and horizontally. The experiments with these two spacing values allow comparing how the mesh density affects the network performance. Similarly, setting a 247 m spacing to 248 m allows communication between adjacent nodes in diagonal with SF8 or requires using the slower SF9.

When the random topology is used, nodes are uniformly distributed over the same area occupied by the grid topology. For instance, in a network with $N^2 = 36$ nodes, to compare it with a grid topology where nodes are spaced 178 m:
$5 \cdot 178\,\text{m} \times 5 \cdot 178\,\text{m} = 890\,\text{m} \times 890\,\text{m}$.
This allows for comparison between a synthetic network and a more heterogeneous deployment.

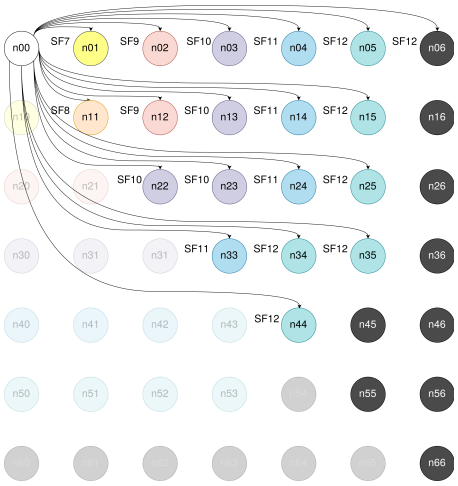
5.3.2 Nodes characteristics

All the nodes are identically configured in the simulation, using the same settings for the LoRa physical layer (e.g., transmission power, bandwidth, etc.), as listed in Table 5.7. Therefore, their behavior and performance is only affected by their position in the network, and the interaction with their neighbors. Most of the chosen configuration parameters (SF, bandwidth, preamble size, etc.) are common in real-world deployments [4]. Given these settings in the simulator, two nodes using SF7 can successfully communicate up to 250 m apart. Therefore, in order to reach nodes further away, a node either must switch to higher SFs, use packet forwarding via its neighbors, or combine both strategies. This maximum communication distance also motivates the spacing described previously in Section 5.3.1.

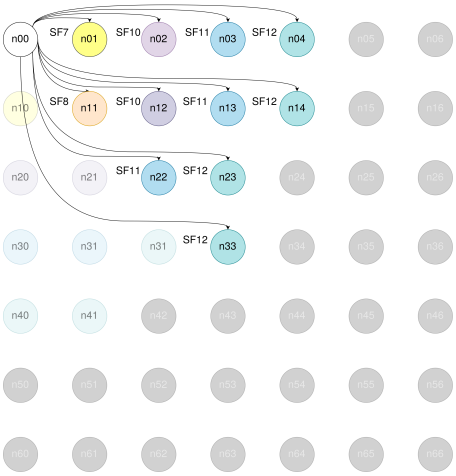
For completeness, Table 5.5 summarizes the maximum transmission distance a node can reach in the simulator, for all the available SFs. The values depend on the LoRa



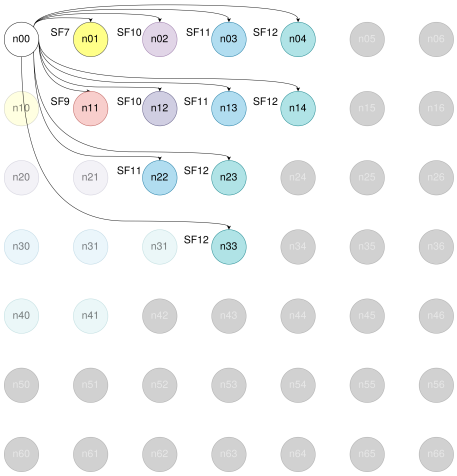
(a) $d_{\mathrm{x}} = d_{\mathrm{y}} = 177\,\mathrm{m}$



(b) $d_{\mathrm{x}} = d_{\mathrm{y}} = 178\,\mathrm{m}$



(c) $d_{\mathrm{x}} = d_{\mathrm{y}} = 246\,\mathrm{m}$



(d) $d_{\mathrm{x}} = d_{\mathrm{y}} = 247\,\mathrm{m}$

Figure 5.7: Minimum SFs required for a LoRa packet to successfully reach different nodes from node n_{00}. Changing the spacing between nodes from 177 m to 178 m disables diagonal transmission between adjacent nodes on SF7.

Configurable parameter	Values
Carrier frequency	868 MHz
Channel bandwidth	125 kHz
Spreading Factor (SF)	7, 8, 9, 10, 11, 12 [a]
Transmission power	20 dBm
Coding Rate (CR)	4/5
Preamble length	16 symbols

Table 5.7: Common node settings in the simulations.

[a] Single-SF metrics are tested with all the possible SFs, one by one. Multi-SF metrics are tested with all the possible combinations of consecutive SFs (e.g., SF7-8, SF7-9, SF7-10, etc.

settings specified in Table 5.7, but also on the radio propagation model implemented in the simulator framework (in this case, an urban environment). With a maximum of 1058 m, when using SF12, communication between nodes further away is only possible through multi-hop.

5.3.3 Traffic settings

During a simulation run, each node generates a total of $100 \times (N^2 - 1)$ unique data packets (i.e., 100 different packets for each one of the other nodes in the network) with a fixed size of 27 B (header+payload). The packets are progressively sent during the simulation, at regular intervals of 100, 10, 1 and 0.1 s (corresponding to a low, medium, high, and saturation traffic respectively), towards their destinations in random order, thus evenly distributing traffic between all the nodes in the network.

5.3.4 Preliminary simulator and LoRa hardware benchmark

We run a simple benchmark in order to verify that the devices simulated with the FLoRa framework offer a network performance comparable to real ones. The experiment consists of two nodes, one transmitting packets continuously during one hour and the other listening to them. Duty cycle restrictions are not enforced, letting the transmitter occupy 100 % of the airtime.

We perform several simulations, using different payload sizes with all the SFs, 7 to 12, and count the number of received packets to calculate the achieved throughput. In parallel, we run the same experiment with real hardware, using two TTGO ESP32 devices [6] and the RadioLib library to interact with the LoRa transceiver [82].

[6] LILYGO® TTGO ESP32: `http://www.lilygo.cn/prod_view.aspx?TypeId=50003&Id=1271&FId=t3:50003:3`

Configurable parameter	Values
Carrier frequency	868 MHz
Link bandwidth	125 kHz
Spreading Factor (SF)	7
Transmission power	20 (FLoRa), 2 dBm (TTGO)
Coding Rate (CR)	4/5
LoRa packet preamble	16 symbols
Message header size	7 bytes
Message payload size	1 to 500 bytes[a]
Time between packets	0 s
Duty cycle	100 % (not enforced)

Table 5.8: Node settings for the benchmark experiment.

[a] RadioLib allows a total message size of 256 B maximum, while FLoRa allows an arbitrarily big message size.

Table 5.8 provides the nodes configuration and settings, used in both simulated and real environments. Note that since the real devices are deployed in the lab, only 2 m apart, the lowest transmission power is used and a 15 dB attenuator is coupled to the antenna, to minimize radio emissions to the environment.

Figure 5.8 shows the achieved throughput from the transmitter to the receiver, for both the FLoRa simulation and the physical devices. Values range from 7.4 bps (the smallest 1 B payload, using SF12), to roughly 5 kbps (250 B payload size or bigger, using SF7). While simulator and real hardware provide similar results, noticeable deviations occur. In particular, achieved throughput with lower SFs is up to 6 % higher in real hardware than in the simulator. This could be attributed to the way the packet transmission time is calculated by the software. Also, the hardware devices' performance appears slightly lower with big payloads, which could be caused by the way the embedded SoC manages received packets. Despite this, we consider the FLoRa framework is accurate enough and provides an adequate simulation environment.

5.4 Experimental evaluation

This section evaluates the proposed RP for LoRa mesh networks. We conduct several experiments, using different topologies, traffic settings and nodes configurations, in order to obtain performance details about three significant KPIs: *scalability*, *throughput* and *latency*.

To get a better understanding of the protocol capabilities, we perform the set of experiments using the different multi-hop and routing strategies presented in 5.2.5:

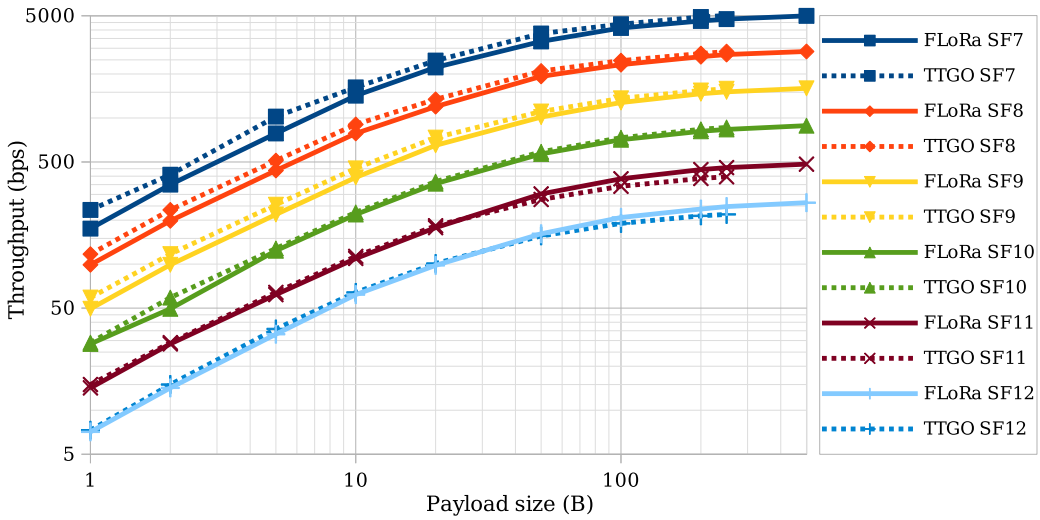


Figure 5.8: Maximum unidirectional data throughput, in bps, with different payload sizes, for SFs 7 to 12. Solid lines correspond to the FLoRa simulator, dotted lines are obtained from two TTGO ESP32 devices.

Broadcast (BC), Hop Count (HC), Expected Transmission Count (ETX), Received Signal Strength Indicator (RSSI) [7] and multi-SF ToA.

5.4.1 Scalability

The first KPI we analyze is scalability, in order to get an understanding of how a LoRa mesh network behaves as the *number of nodes* rises. A network can scale up, when adding more nodes, covering a larger area or keeping it fixed and increasing the density of the devices.

To understand the behavior of our RP with the LoRa radio technology regarding scalability, we perform two sets of experiments, for both scenarios mentioned above, and analyze the effects of each. First, we simulate a mesh network as it increases its number of nodes, keeping the nodes at a constant distance, hence covering each time a larger area. Second, we analyze a mesh network covering a predefined fixed area, with an increasing number of nodes and, hence, an increasing density.

To evaluate our multi-SF routing protocol based on the ToA metric, we test it under low, medium and high traffic conditions, and compare it with the single-SF BC, HC, ETX and RSSI routing strategies. We simulate network deployments with $N^2 = \{9, 16, 25, 36, 49$ and $64\}$ nodes, both on a grid topology or randomly distributed over

[7] Simulations with RSSI sum and RSSI product reported almost identical results, so only the first one is shown in the results for simplicity.

an equivalent area. The ToA metric is tested with all the possible SF ranges (i.e., SF7-8, SF7-9, SF7-10, SF7-11, SF7-12, then SF8-9, SF8-10, SF8-11, etc.).

5.4.1.1 Area coverage

In this section, we perform different network simulation experiments to investigate network scalability, as an increasing number of nodes covers an area that grows proportionally. We look at the end-to-end PDR performance metric of the data packets sent between any pair of nodes in the network. Our hypothesis is that, as more nodes operate on the network and, on average, the distance between a pair of nodes will be higher, end-to-end transmissions will require more hops, which will result in a higher number of collisions, hence increasing packet loss and reducing PDR. Our objective is to find whether using the multi-SF ToA metric and combine different SFs in the network improves the average PDR or not.

Initially, we simulate network deployments on a regular grid topology, keeping horizontal (d_x) and vertical (d_y) distance between nodes constant. This means that, as the number of nodes increases, so does the area they occupy. By testing different values for these distances $d_x = d_y$, we aim to understand if the routing protocol can take advantage of the multi-SF capabilities of LoRa and improve performance. Later, we perform the same network simulations except for the fact that nodes are randomly distributed, occupying an equivalent area to the previous grid topology, creating a more heterogeneous topology. Table 5.9 lists the topologies and dimensions used in these simulations.

Due to the shape of the grid topology and the border effect, the density of nodes (i.e., the nodes/area ratio) is not constant, as shown in Figure 5.9. However, as the network grows bigger, it tends asymptotically to a fixed value, which is only determined by the number of nodes and the spacing between them. We consider that, from 36 nodes onward, the border effect is negligible, so the resulting PDR will be mostly affected by the number of nodes and the area they occupy, rather than by their density.

Figure 5.10 shows the average PDR in function of the number of nodes, deployed with a grid topology, when different routing strategies are used. Similarly, Figure 5.11 corresponds to the same experiments, with nodes randomly deployed over the equivalent areas. Each of the four rows of sub-figures corresponds to a different spacing between nodes, and each of the three columns corresponds to a low, medium, and high network traffic scenarios.

The general trend shows the expected negative relation between the PDR and the number of nodes –except for the flooding-based strategies–. First, we can observe that in low and medium traffic scenarios, BC-based forwarding provides the best PDR results, with respectively 99 % and 93 % of the packets successfully delivered end-to-end. There, the redundancy of the flooding strategy, combined with the

Topology	**Nodes**	$d_x = d_y$	**Area**
Grid	9	177 m	$354 \times 354\,\mathrm{m}^2$
		178 m	$356 \times 356\,\mathrm{m}^2$
		247 m	$494 \times 494\,\mathrm{m}^2$
		248 m	$496 \times 496\,\mathrm{m}^2$
Random (uniform)	9	n/a	$354 \times 354\,\mathrm{m}^2$
		n/a	$356 \times 356\,\mathrm{m}^2$
		n/a	$494 \times 494\,\mathrm{m}^2$
		n/a	$496 \times 496\,\mathrm{m}^2$
Grid	16	177 m	$531 \times 531\,\mathrm{m}^2$
		178 m	$534 \times 534\,\mathrm{m}^2$
		247 m	$831 \times 831\,\mathrm{m}^2$
		248 m	$834 \times 834\,\mathrm{m}^2$
Random (uniform)	16	n/a	$531 \times 531\,\mathrm{m}^2$
		n/a	$534 \times 534\,\mathrm{m}^2$
		n/a	$831 \times 831\,\mathrm{m}^2$
		n/a	$834 \times 834\,\mathrm{m}^2$
...	...	...	...
Grid	64	177 m	$1239 \times 1239\,\mathrm{m}^2$
		178 m	$1246 \times 1246\,\mathrm{m}^2$
		247 m	$1729 \times 1729\,\mathrm{m}^2$
		248 m	$1736 \times 1736\,\mathrm{m}^2$
Random (uniform)	64	n/a	$1239 \times 1239\,\mathrm{m}^2$
		n/a	$1246 \times 1246\,\mathrm{m}^2$
		n/a	$1729 \times 1729\,\mathrm{m}^2$
		n/a	$1736 \times 1736\,\mathrm{m}^2$

Table 5.9: List of topologies, number of nodes, distance between nodes and total area used in the simulations.

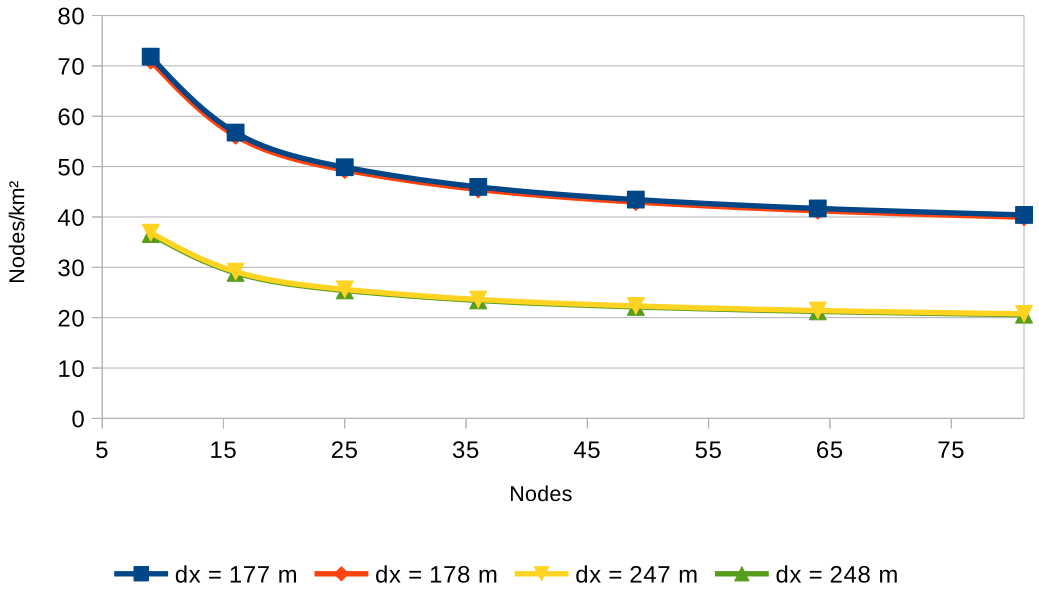
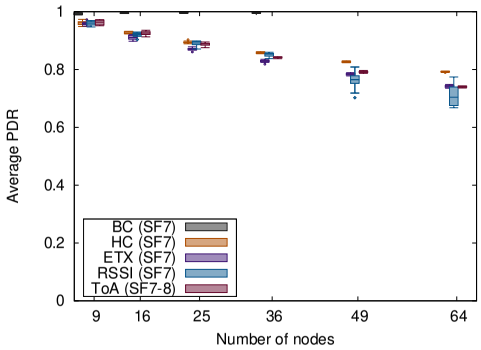


Figure 5.9: Nodes density (nodes/km^2) in function of the number of nodes simulated, for the different spacing between nodes d_x used in Sec 5.4.1.1. The graphs show an asymptotic behavior.

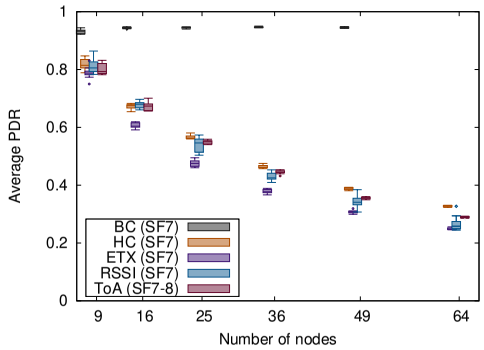
relatively free nodes, allows room for packets' duplication, which helps to ensure that at least one of the copies reaches the destination. This, however, comes at the expense of high latency, as packets are often queued in the intermediate nodes. As the traffic increases and the links begin to saturate, BC provides no significant benefits compared to other routing strategies.

Regarding the actual single-SF routing strategies, HC (the simplest metric) provides the best PDR performance in most of the experiments, closely followed by the RSSI metric. The traffic-aware ETX metric performs clearly worse in low traffic scenarios, but is almost on par with HC as the traffic load increases. The multi-SF ToA does not offer any significant improvement in terms of PDR, and its performance is in between the single-SF metrics. Furthermore, it does not take advantage of the available links in diagonal, using a higher SF that would allow using two different SFs simultaneously. Therefore, in the context of a grid topology with equal spacing between all nodes, using a simpler single-SF algorithm seems to be the most reliable solution in terms of end-to-end packet delivery. Despite not being shown in the graphs, goodput and latency performance figures are consistent with the PDR results shown here.

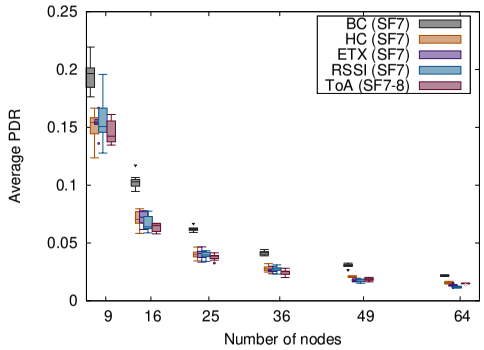
The simulations with random network topologies offer a different picture, as visible in Figure 5.11. There, we can observe that the multi-SF ToA routing metric provides better PDR performance than single-SF metrics in many of the situations, with more consistent results (i.e., less dispersion) along the different experiments. The improvement is particularly visible in the lower row (Figs. 5.11d, 5.11e and 5.11f),



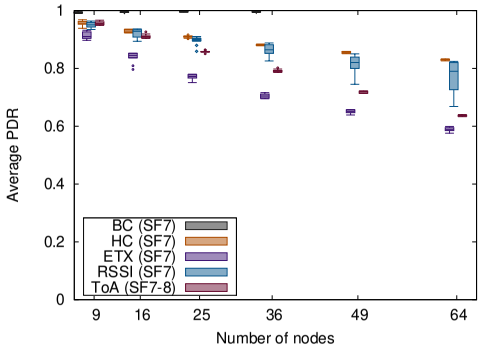
(a) $d_x = 177\,\text{m}$, low traffic load.



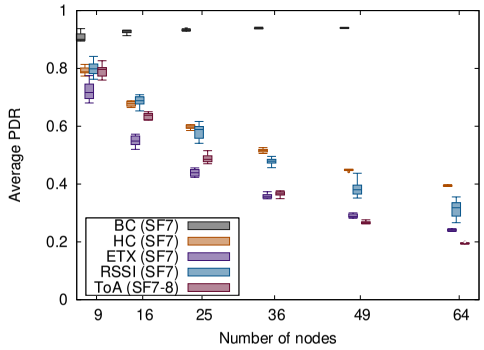
(b) $d_x = 177\,\text{m}$, medium traffic load.



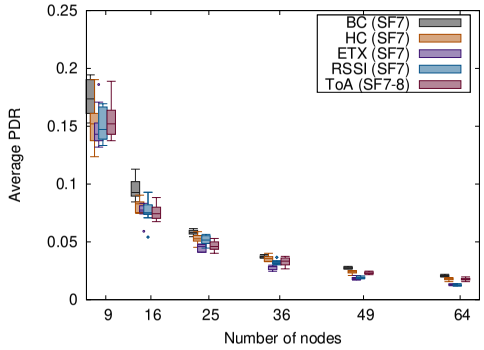
(c) $d_x = 177\,\text{m}$, high traffic load.



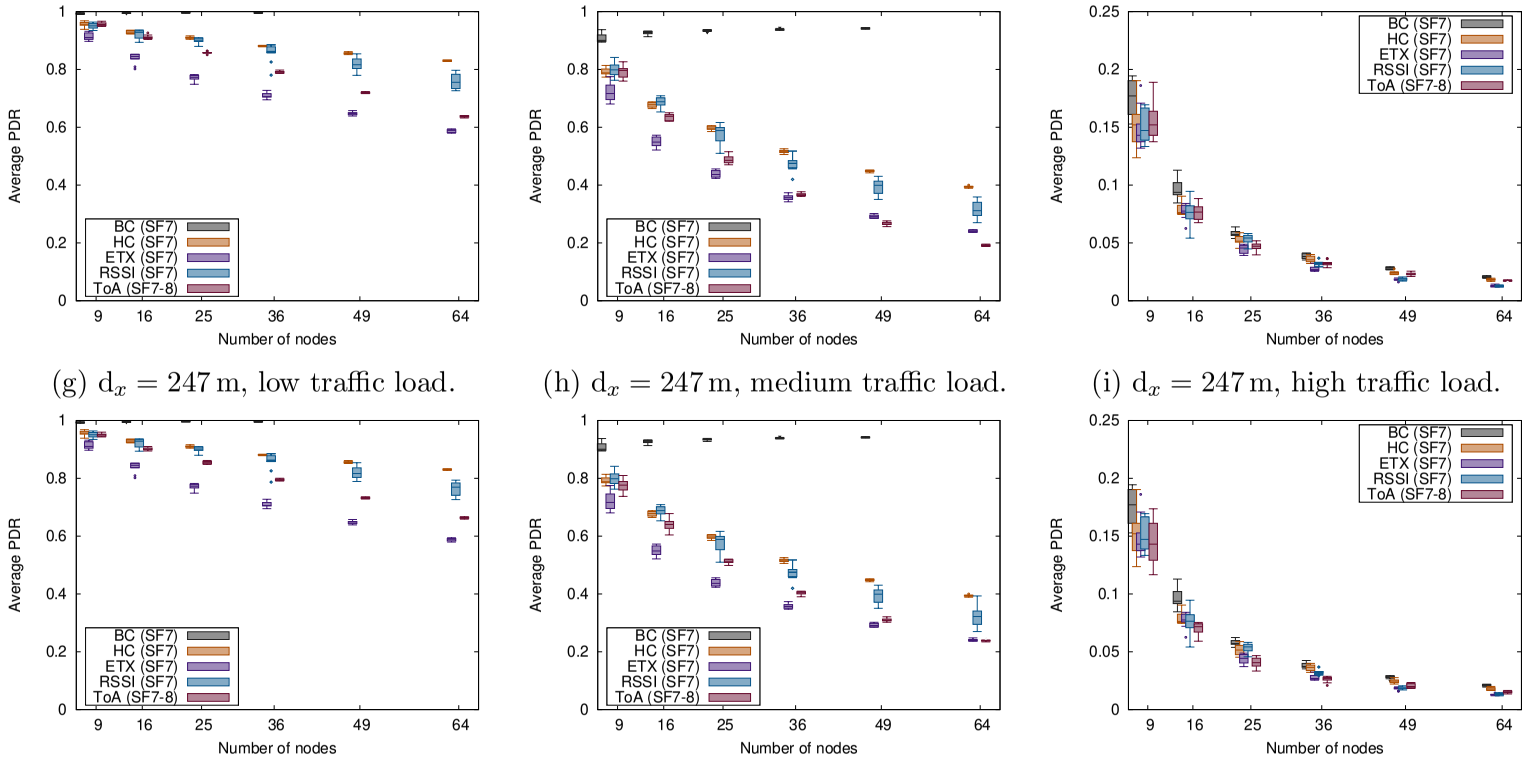
(d) $d_x = 178\,\text{m}$, low traffic load.



(e) $d_x = 178\,\text{m}$, medium traffic load.



(f) $d_x = 178\,\text{m}$, high traffic load.



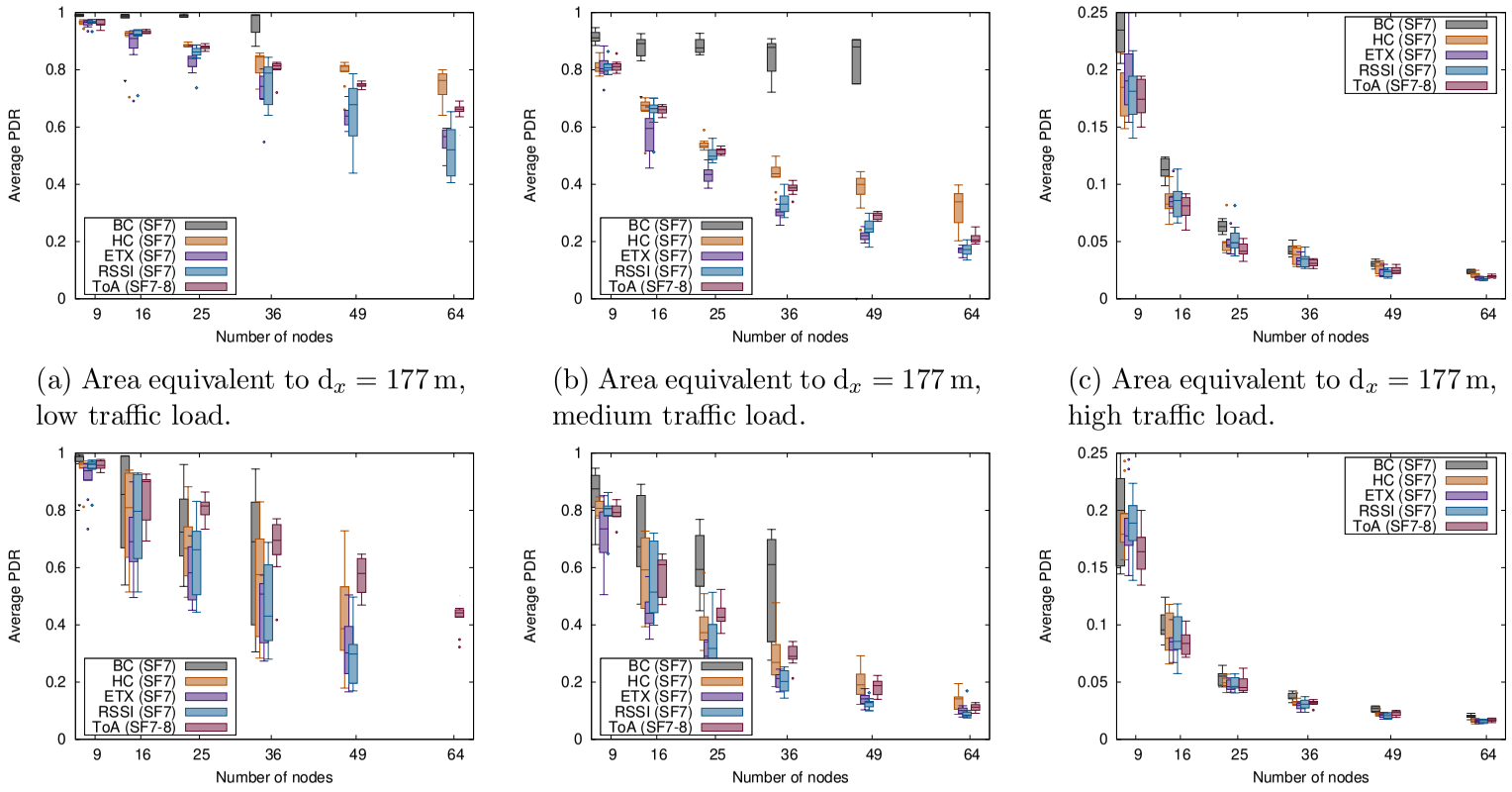
(g) $d_x = 247\,\text{m}$, low traffic load. (h) $d_x = 247\,\text{m}$, medium traffic load. (i) $d_x = 247\,\text{m}$, high traffic load.

(j) $d_x = 248\,\text{m}$, low traffic load. (k) $d_x = 248\,\text{m}$, medium traffic load. (l) $d_x = 248\,\text{m}$, high traffic load.

Figure 5.10: Average network PDR for different number of nodes in a grid topology with a constant horizontal and vertical node spacing of $177, 178, 247$ and $248\,\text{m}$, using different routing strategies. Notice the y-axis scale change.

where nodes are more separated apart. These results indicate that the multi-SF routing strategy has a direct impact on nodes that, because of their random placement, are more isolated (e.g., at the edges of the network) and therefore have less communication opportunities than those at the center of the network area. In other words, the ToA metric better adapts to heterogeneous network topologies and balances the PDR performance between nodes with different network centrality. These benefits are also visible in terms of the goodput KPI, which is analyzed later in Section 5.4.2.

In conclusion, multi-SF routing strategies like our proposed ToA metric may have a positive impact on PDR, which is a main indicator of the scalability, depending on the network topology. They can better cope with the network links' heterogeneity than other routing strategies, benefiting overall performance, but they do not provide any advantage to networks with very regular topologies like a grid one. Since diversity in nodes and links is expected to occur in real-world systems, with nodes placed in diverse locations, subject to different environmental conditions (attenuation, interference, number of neighbors, etc.), the multi-SF ToA metric may ease the deployment of LoRa mesh networks and improve their performance –at least, in terms of end-to-end PDR–.



(a) Area equivalent to $d_x = 177\,\text{m}$, low traffic load.

(b) Area equivalent to $d_x = 177\,\text{m}$, medium traffic load.

(c) Area equivalent to $d_x = 177\,\text{m}$, high traffic load.

(d) Area equivalent to $d_x = 248\,\text{m}$, low traffic load.

(e) Area equivalent to $d_x = 248\,\text{m}$, medium traffic load.

(f) Area equivalent to $d_x = 248\,\text{m}$, high traffic load.

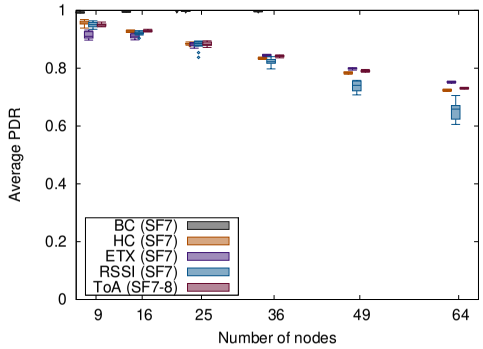
Figure 5.11: Average network PDR for different number of nodes in a random topology with uniform nodes' distribution, using different routing strategies. Notice the y-axis scale change.

5.4.1.2 Density of nodes

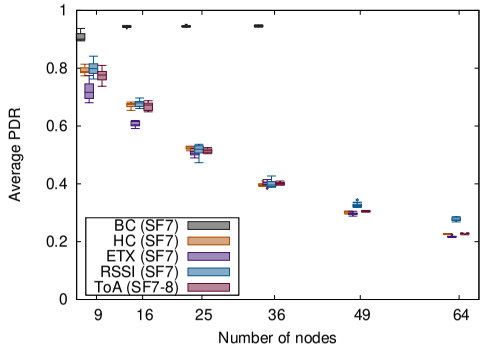
In this second part of the section, we conduct several experiments to analyze scalability as the number of nodes grows and their density hence increases. We use the end-to-end PDR, averaged among all the nodes in the network, as the performance metric. As more nodes are placed on the same area and radio transmissions become more frequent, the PDR is expected to degrade, due to the higher collision probability. Our objective is to find whether the simultaneous usage of multiple SFs reduces collisions, hence improving the average PDR.

We simulate different network deployments, with $N^2 = \{9, 16, 25, 36, 49 \text{ and } 64\}$ nodes on a fixed area of $500 \times 500\,\text{m}^2$. As the number of nodes grows on each iteration, while the area is kept constant, the density of nodes becomes higher proportionally. To evaluate our multi-SF routing protocol based on the ToA metric, we test it under low, medium and high traffic conditions, and compare it with the BC, HC, ETX and RSSI routing strategies. The ToA metric is tested with all the possible SF ranges (i.e., SF7-8, SF7-9, SF7-10, SF7-11, SF7-12, SF8-9, SF8-10, SF8-11, etc.).

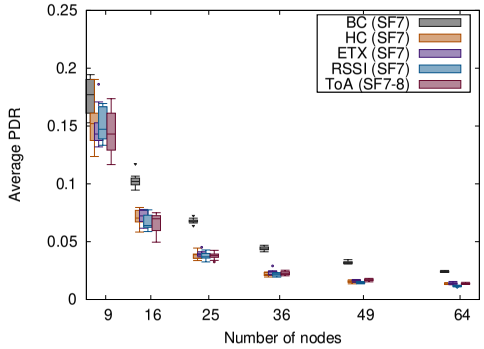
Figures 5.12 and 5.12 show the average PDR in function of the number of nodes used in the simulations, when different routing strategies are used. In the former, nodes are deployed on a grid topology; in the latter, nodes are randomly deployed with uniform distribution (details about it are in Sec. 5.3.1). Each figure contains three sub-figures that correspond to the low, medium, and high network traffic conditions. As we would expect, their general trend shows a negative relation between the average PDR and the number –and density– of nodes. For the case of the ToA metric, the SFs range providing the best performance results (e.g., SF7-9) is plotted (rather tan all the combinations).



(a) Low traffic load.

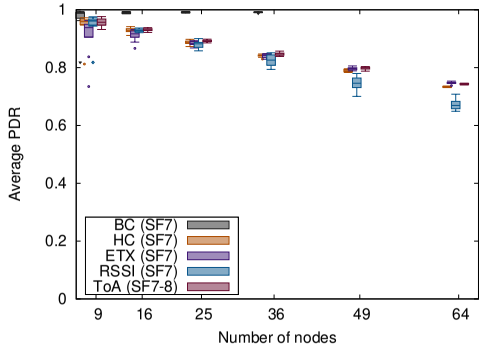


(b) Medium traffic load.

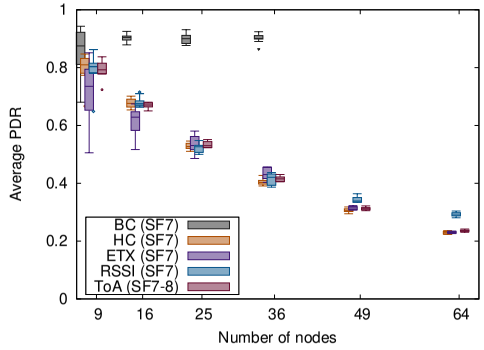


(c) High traffic load.

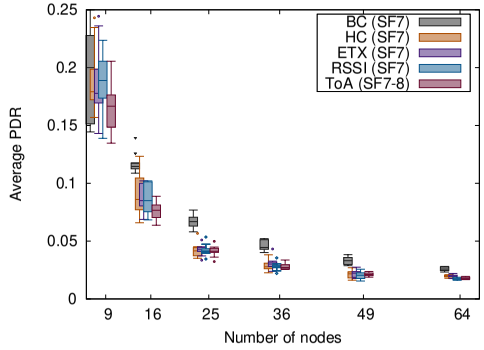
Figure 5.12: Average network PDR, for different number of nodes, in a grid topology on a fixed area of $500 \times 500\,\mathrm{m}^2$, using different routing strategies. Notice the y-axis scale change.



(a) Law traffic load.



(b) Medium traffic load.



(c) High traffic load.

Figure 5.13: Average network PDR, for different number of nodes, in a random topology with uniform distribution on a fixed area of $500 \times 500\,\mathrm{m}^2$, using different routing strategies. Notice the y-axis scale change.

For the low and medium traffic scenarios (Figs. 5.12a, 5.12b, 5.13a and 5.13b), BC strategy provides the best PDR results. Since packets are broadcast and replicated on each hop, flooding the network, the chance for any of the copies to arrive at the destination are very high, mostly compensating for any additional collisions. The actual routing strategies (HC, ETX, RSSI and ToA) offer a quasi-linear relation between the number of nodes and the PDR. Their results fluctuate in function of the number of nodes and the topology, but are very consistent, revealing there is no evident performance difference between any of them. As the traffic increases (Figs. 5.12c, 5.13c, the network becomes each time more congested, affecting the collision probability and hence PDR. The difference between BC and the routing strategies becomes smaller, as the congested network does not allow room for packets to flood the whole network and reach the destination.

According to the PDR results, there is no clearly observable benefit in using the multi-SF approach and the ToA instead of any other single-SF strategy. Our simulations indicate, however, that combining different SFs does help to increase PDR results in very congested networks, where nodes are constantly sending packets one after the other. In this scenario –regardless of the topology–, single-SF routing can barely provide a PDR of 0.001 (i.e., one packet out of a thousand reaching its final destination) in a 16-nodes network, while multi-SF ToA achieves an approximate PDR of 0.01 (i.e., one packet out of a hundred). The effect of multi-SF is measurable, but useless for such a regular and homogeneous grid topology.

5.4.2 Throughput

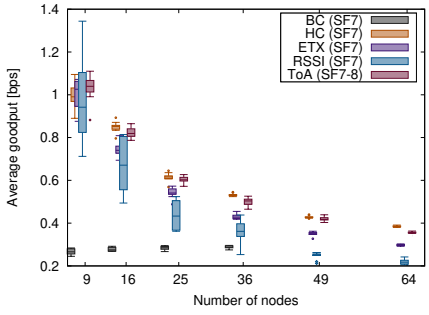
Besides the scalability and the PDR analyzed above, *throughput* is an important KPI to take into account, in order to understand the amount of data a LoRa-based mesh network can handle. In Section 5.3.4 we ran a simple benchmark that indicated the maximum throughput two LoRa nodes can achieve in ideal conditions (i.e., continuous unidirectional transmission, no duty cycle restriction, no collisions). For a payload sized the same as the one in our simulations, the throughput ranged between 2500 bps and 100 bps, depending on the SF used (roughly, each SF step up halves the speed). Our LoRa mesh network experiments, however, are different from the deployments in the benchmark. First, in the absence of a scheduler to organize transmissions, packet collisions will occur. Their frequency will mostly depend on the number of nodes and their topology, and the packets' egress rate. Second, a fraction of the available airtime will be used by the nodes to broadcast routing messages, instead of data packets. Third, multi-hop communication between arbitrary pairs of nodes requires the participation of different intermediate forwarders, spending their available time to route other nodes' traffic. For these reasons, the expected throughput measurements will be well below the numbers obtained in the ideal conditions benchmark.

For our experiments, we define the network's throughput as the total amount of valid data (i.e., payload) transmitted by any node *and* correctly received (i.e., no collisions) by the receiver node specified in the header, measured over a given period of time. Throughput is a good KPI to understand how many data can be handled by the network, but it may provide an incomplete picture of its performance, since packets being forwarded by different nodes may account for more throughput than single-hop communication between adjacent nodes. Also, a multi-hop packet colliding halfway to its destination would have generated throughput without reporting benefit in terms of end-to-end communication. To this end, we also define the network's *goodput* as the total amount of valid data transmitted end-to-end between pairs of nodes, measured over a given period of time. This magnitude provides a more precise picture about how *effective* the network is at transmitting valid data from one node to any other. To this end, in this section, we use the goodput figures to evaluate the different routing metrics.

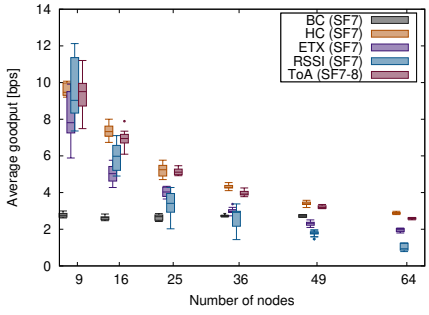
Figure 5.14 shows the average goodput achieved by the network, averaged to all the participating nodes. Nodes are deployed using a grid topology with a constant horizontal and vertical spacing of 177 and 248 m between them, in a low, medium, high and saturation traffic scenarios. Each of the four traffic scenarios reveals a different network behavior that helps to understand the throughput KPI in a LoRa mesh network. In the low and medium traffic scenario, both single-SF and multi-SF metrics provide similar results, with HC and ToA offering comparable figures. The rest of single-SF metrics tend to offer a slightly worse performance. It is worth noting that flooding-based BC and Single Board Computer (SBC), which gave the best PDR performance in Section 5.4.1.1, did it at the expense of significantly bad goodput results. As the traffic increases to a high volume, the multi-SF ToA offers slightly better goodput results than single-SF metrics. We observe that this metric can benefit from diagonal transmissions using the immediately higher SF when comparing the topologies with a spacing of 177 and 178 m between nodes.

In the high traffic scenarios, where links start to become saturated, the benefits of using single-SF or multi-SF are less obvious, as some of the experiments show a relatively small advantage and others a disadvantage. This suggests that the additional complexity of multi-SF ToA may not match the very simple and regular grid topology. However, in the saturated traffic scenario (where all the nodes try to transmit packets as frequently as possible), the multi-SF ToA metric provides nodes a mechanism to deal with some of the packet collisions. The traffic is spread on the overlaid networks with different SFs, partially desaturating the spectrum and avoiding part of the collisions This way, each node can still correctly transmit a few bps, while the single-SF metrics provide close to zero a goodput.

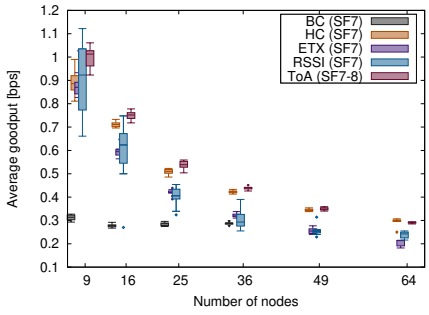
It is worth mentioning that even the best average goodput results (approx. 16 bps on SF8) are much lower than the ideal benchmark results from Section 5.3.4. However, they are consistent with the radio technology used and the challenges this demanding topology and traffic pattern poses.



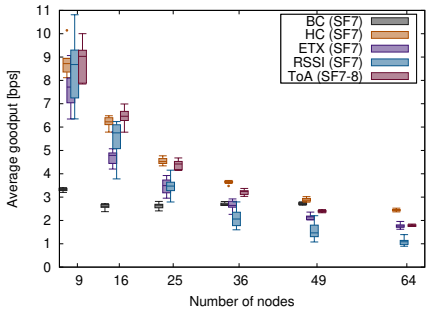
(a) $d_{x,y} = 177\,\text{m}$, low traffic.



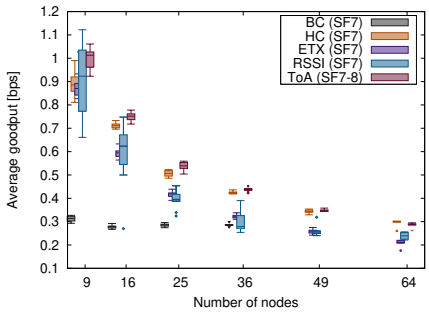
(b) $d_{x,y} = 177\,\text{m}$, medium traffic.



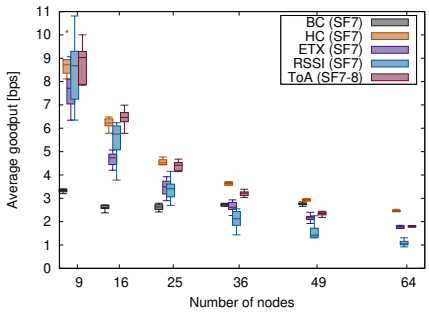
(e) $d_{x,y} = 178\,\text{m}$, low traffic.



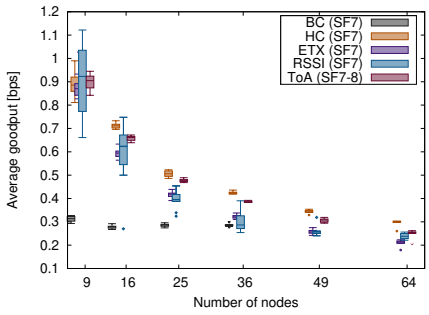
(f) $d_{x,y} = 178\,\text{m}$, medium traffic.



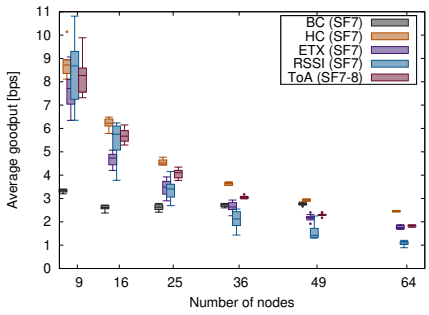
(i) $d_{x,y} = 247\,\text{m}$, low traffic.



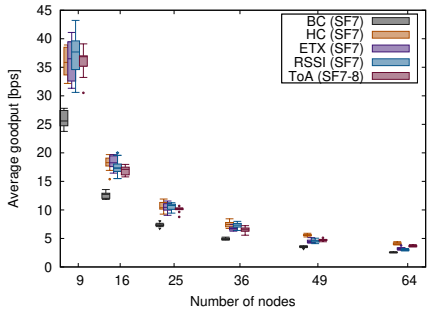
(j) $d_{x,y} = 247\,\text{m}$, medium traffic.



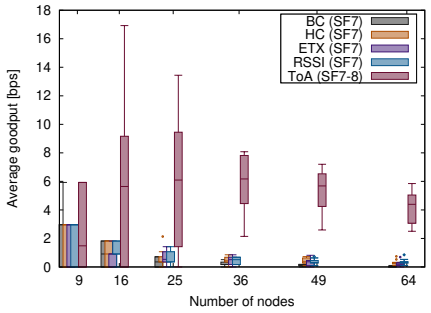
(m) $d_{x,y} = 248\,\text{m}$, low traffic.



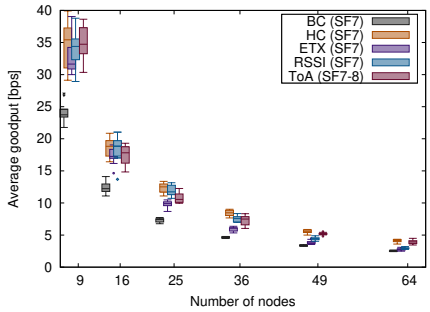
(n) $d_{x,y} = 248\,\text{m}$, medium traffic.



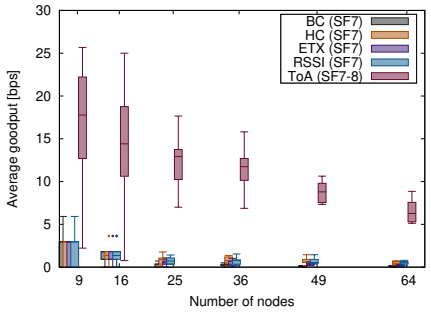
(c) $d_{x,y} = 177\,\text{m}$, high traffic.



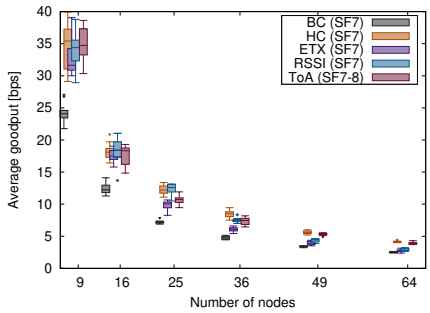
(d) $d_{x,y} = 177\,\text{m}$, saturated traffic.



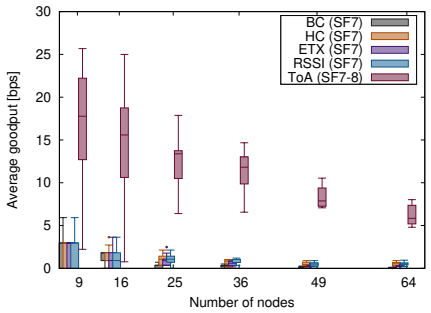
(g) $d_{x,y} = 178\,\text{m}$, high traffic.



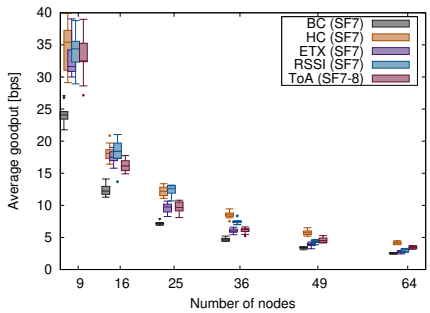
(h) $d_{x,y} = 178\,\text{m}$, saturated traffic.



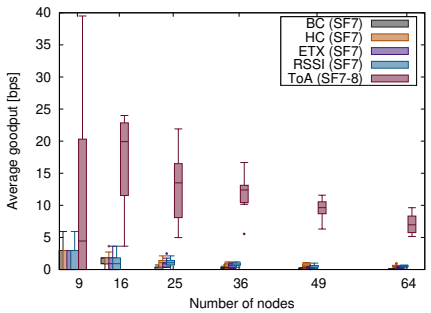
(k) $d_{x,y} = 247\,\text{m}$, high traffic.



(l) $d_{x,y} = 247\,\text{m}$, saturated traffic.

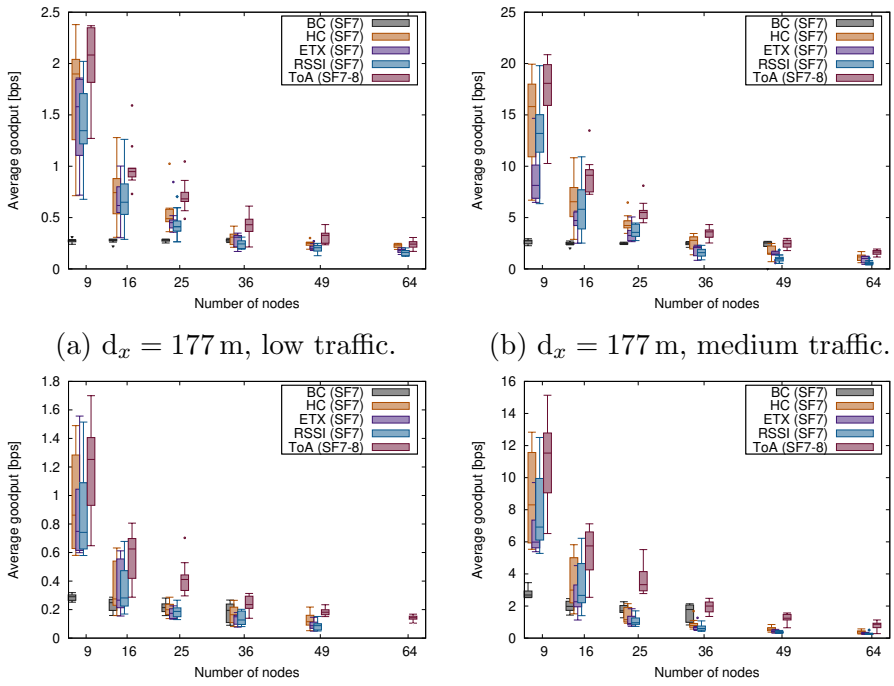


(o) $d_{x,y} = 248\,\text{m}$, high traffic.



(p) $d_{x,y} = 248\,\text{m}$, saturated traffic.

Figure 5.14: Average network goodput, in bps, for different number of nodes in a grid topology with a constant horizontal and vertical node spacing of 177 and 248 m, using different routing strategies. Notice the y-axis scale changes.



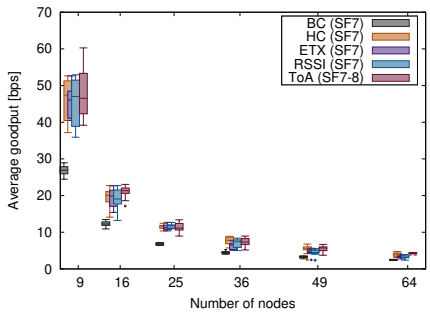
(a) $d_x = 177\,\text{m}$, low traffic.

(b) $d_x = 177\,\text{m}$, medium traffic.

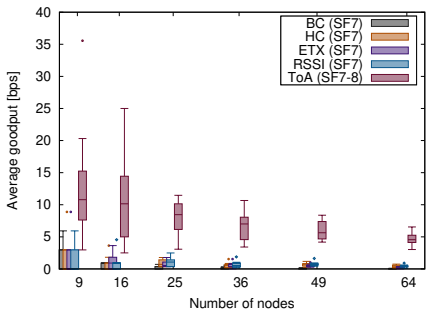
(e) $d_x = 248\,\text{m}$, low traffic.

(f) $d_x = 248\,\text{m}$, medium traffic.

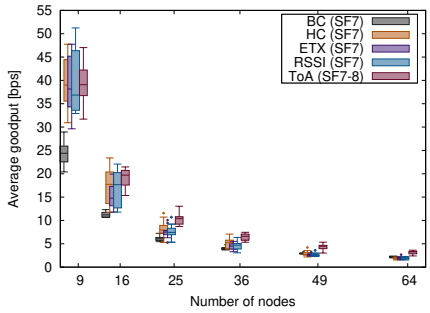
Figure 5.15 shows the average goodput results achieved by the network, averaged to all the participating nodes, when these are deployed randomly over an area the same size as the previous experiments. Now, the multi-SF ToA routing metric is able to adapt to the heterogeneity of each of the nodes, which has a positive impact on the performance. In most of the experimented cases, the best goodput figures are achieved by the ToA metric, whether we consider a low, medium or high traffic load. It is worth comparing the ratio between this figure and the one for the throughput (Fig. 5.14) to see how efficient the network is in providing end-to-end communication between pairs of nodes. In the low traffic scenario for a grid topology with 9 nodes, 177 m apart, using the ToA metric with two SFs (8 to 9), nodes achieve an average throughput of $1,85\,\text{bps}$ (Subfig. 5.14a), but only deliver a goodput just below $1,2\,\text{bps}$. Under good PDR conditions (i.e., few packet losses), this means that each end-to-end packet transmission requires an average of 1.5 hops, which is in tune with the network dimensions. In any case, this ratio varies in function of the number of nodes, their spacing, the traffic load and the routing strategy. A ratio closer to 1 may indicate that nodes use fewer hops to perform end-to-end communications (e.g., using longer-reaching SFs), while higher rates would suggest that each successful end-to-end communication requires more packet transmissions, indicating either shorter-reaching SF or higher packet losses. Still, Subfigures 5.14d and 5.14p, corresponding to a saturated traffic scenario, reveal that nodes using the multi-SF ToA take advantage of LoRa's orthogonality between SFs, delivering goodput figures in the tens of bps order.



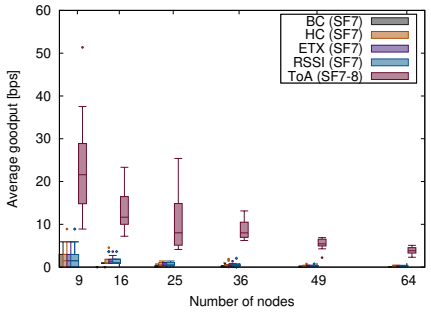
(c) $d_x = 177\,\text{m}$, high traffic.



(d) $d_x = 177\,\text{m}$, saturated traffic.



(g) $d_x = 248\,\text{m}$, high traffic.



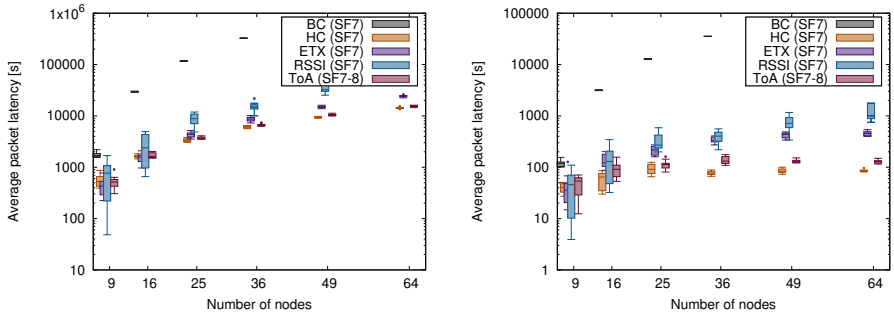
(h) $d_x = 248\,\text{m}$, saturated traffic.

Figure 5.15: Average network goodput, in bps, for different number of nodes in a random topology with an equivalent constant horizontal and vertical node spacing of 177 and 248 m, using different routing strategies. Notice the y-axis scale changes.

The analysis of the throughput and goodput performance of the network, with different number of nodes, topologies and traffic loads, shows that the data rates achieved by the nodes are much smaller than the ideal benchmark results, between two and three orders of magnitude below. Additionally, we can acknowledge that using multiple SFs simultaneously has a limited impact for the low, medium and high traffic loads, but dramatically improves throughput and goodput in traffic saturation scenarios. This suggests that, in order to maximize network performance, multi-SF operation would be preferred.

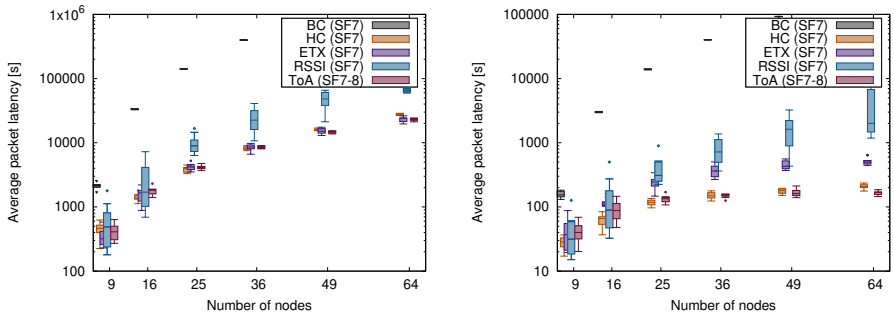
5.4.3 Latency

The last KPI analyzed for our RP is packet latency. Compared to the LoRaWAN architecture, where packet transmissions are single-hop only and have a negligible latency, it is an important KPI that must be considered in the context of multi-hop mesh networks.



(a) $d_x = 177\,\text{m}$, low traffic.

(b) $d_x = 177\,\text{m}$, medium traffic.



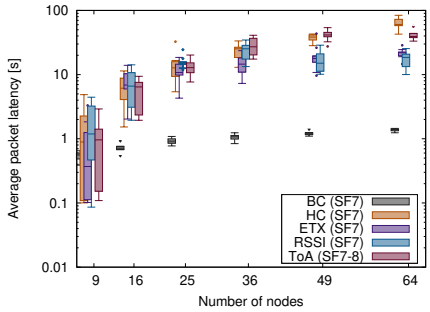
(e) $d_x = 248\,\text{m}$, low traffic.

(f) $d_x = 248\,\text{m}$, medium traffic.

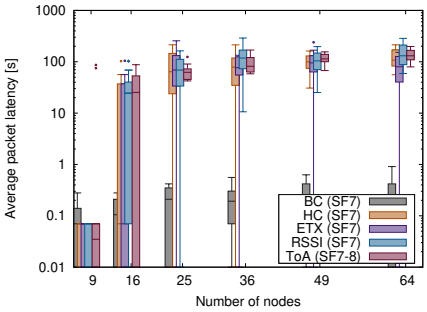
Figure 5.16 shows the average latency experienced by packets crossing the network with a grid topology, with two different node spacing (177 and 248 m). In particular, only those packets successfully delivered end-to-end are considered, and not those which experimented a collision and got lost. Similarly, Figure 5.17 offers the results for the experiments with nodes randomly distributed over an area the same size.

The figures for both topologies show that single-SF and multi-SF metrics perform very similarly, even if some particular differences may be observed. First, flooding-based BC and SBC provide much higher latency (even three orders of magnitude or mode), as they often force links saturation and filling of the nodes' buffers, which indicate they are very poor multi-hop strategies. Second, single-SF-based metric provide consistent results –even if some metrics have slightly better performance in some traffic conditions, and other metrics in other environments-. Third, multi-SF ToA provides the best latency performance in low and medium traffic loads, especially for the random topology, again taking advantage of the nodes and links heterogeneity.

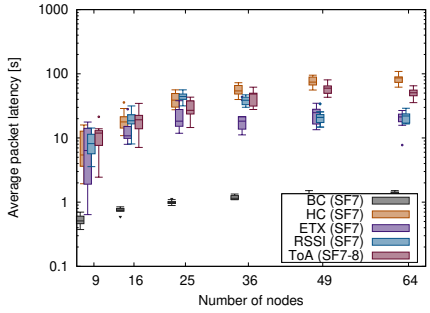
It is worth mentioning that, because of the very slow packet transmission rate of the low traffic scenarios (nodes send, at most, one packet every 100 s), messages incur in very high latency numbers, in the order of tens of thousands of seconds. This is because of both the number of hops needed to reach the destination and also because of the waiting time at the intermediate nodes' buffers. As discussed later in 5.4.4, our RP may require a modification to ensure that packets received by intermediate



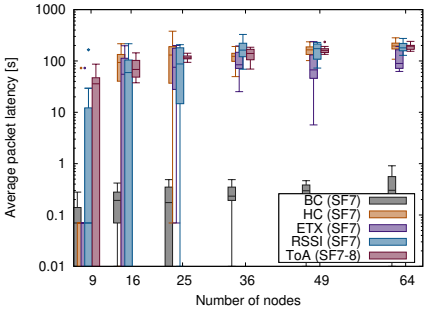
(c) $d_x = 177\,\text{m}$, high traffic.



(d) $d_x = 177\,\text{m}$, saturated traffic.

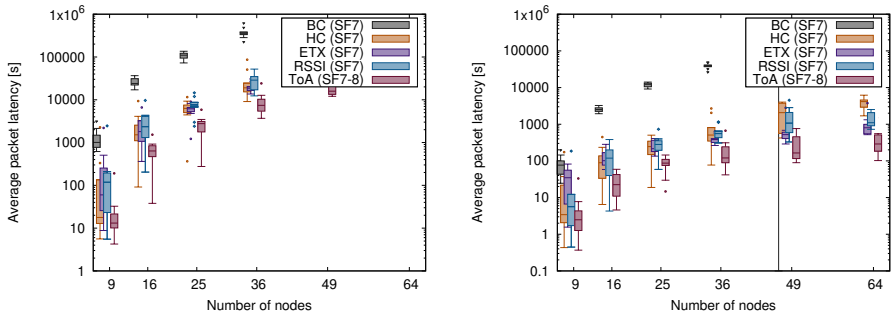


(g) $d_x = 248\,\text{m}$, high traffic.



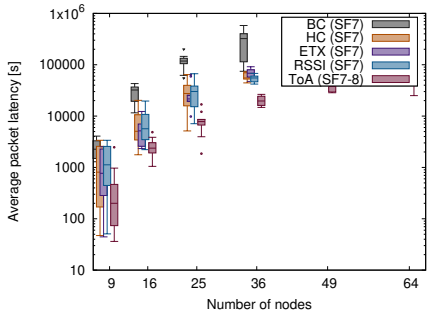
(h) $d_x = 248\,\text{m}$, saturated traffic.

Figure 5.16: Average network latency, in bps, for different number of nodes in a grid topology with a constant horizontal and vertical node spacing of 177 and 248 m, using different routing strategies. Notice the y-axis scale change.

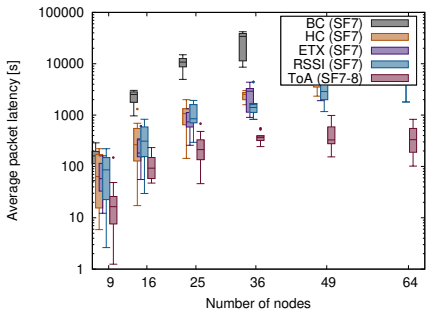


(a) $d_x = 177\,\text{m}$, low traffic.

(b) $d_x = 177\,\text{m}$, medium traffic.



(e) $d_x = 248\,\text{m}$, low traffic.



(f) $d_x = 248\,\text{m}$, medium traffic.

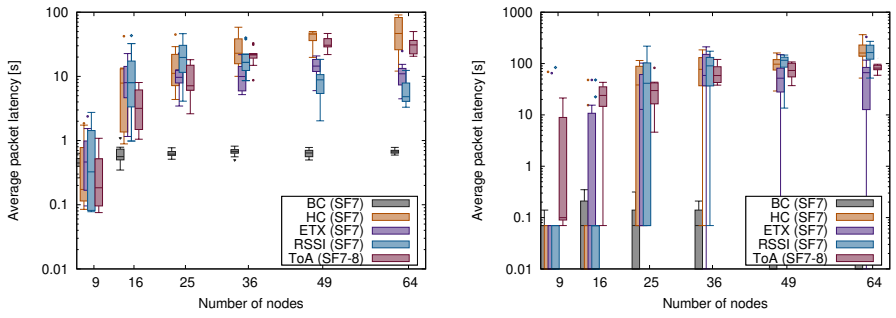
nodes are dispatched as soon as possible, with a rate different and independent of the data generation rate.

5.4.4 Conclusions

In this chapter, we presented the design and evaluation of a DV RP for LoRa mesh networks, which includes a novel multi-SF ToA metric, that adapts to networks with heterogeneous topologies. The approach followed, and the results obtained, provide the answer to RQ3 regarding how to take advantage of LoRa's specific features, and also to RQ4 regarding the performance according to different KPIs.

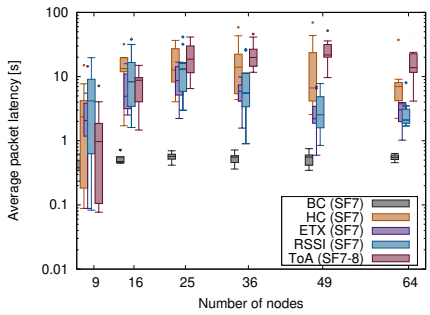
Our proposed RP is designed with a minimalistic set of features, in order to reduce memory and computing footprint, making it suitable for embedded devices composed of a microcontroller and a LoRa radio chip. It merges L2 and L3 addressing and uses a proactive broadcast mechanism for nodes to exchange routes with their neighbors, keeping the routing tables up-to-date and propagating changes over the network. Its simplicity comes at the expense of lack of certain features, such as multicast, route discovery, node-to-node or end-to-end transmission reliability.

A novel aspect of our RP is that it takes into account LoRa's capability to transmit and receive with different SFs. In combination with the radio chip's CAD features, this allows working with nodes in a mesh network using different SFs simultane-

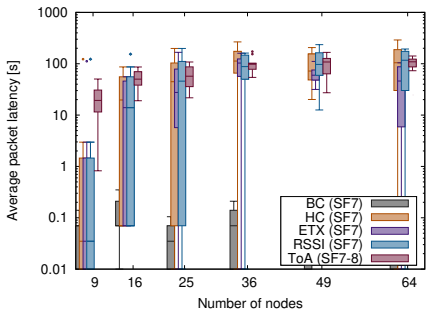


(c) $d_x = 177\,\text{m}$, high traffic.

(d) $d_x = 177\,\text{m}$, saturated traffic.



(g) $d_x = 248\,\text{m}$, high traffic.



(h) $d_x = 248\,\text{m}$, saturated traffic.

Figure 5.17: Average network latency, in bps, for different number of nodes in a random topology with an equivalent horizontal and vertical node spacing of 177 and 248 m, using different routing strategies. Notice the y-axis scale change.

ously, with packets potentially being forwarded using multiple SFs along their path. The ToA metric we present takes advantage of this feature, and has proven suitable for LoRa mesh networks with heterogeneous links and topologies. For these situations where nodes have different characteristics in terms of placement, number of neighbors and distance to them, etc. our proposal achieves better PDR, goodput and latency results than single-SF routing strategies using other metrics like HC or ETX. These contributions should be useful in real-world deployments, where nodes are expected to operate in diverse and heterogeneous environmental conditions.

Future work will evaluate our DV RP and the proposed multi-SF ToA metric with embedded devices featuring a LoRa radio, in a controlled laboratory testbed environment and on a realistic outdoors deployment.

Chapter 6

Discussion

This thesis explored the transition from a centralized LPWAN architecture for the IoT, LoRaWAN, to a distributed mesh network environment. In Section 1.2, four RQs were introduced, which were then tackled in the subsequent chapters. First, the feasibility of topologies other than LoRaWAN's star-of-stars was confirmed (RQ1), through a survey of the state-of-the-art that showed the current research efforts in this domain. Second, the effect of adding packet forwarding capacity to end nodes in the LoRaWAN architecture was analyzed (RQ2), by means of the realistic use case of a massive communication system for the aftermath of a natural disaster. Third, the specific features of the LoRa radio technology that could be used to build flat, decentralized mesh networks were investigated (RQ3), emphasizing the opportunities that multi-SF communications offered. Last, these type of networks built with LoRa were analyzed, in a simulation environment, in order to obtain metrics about their different KPIs.

While the LoRaWAN architecture and our network model are both based on the same underlying radio technology, LoRa, their approaches and scopes are radically different. On the one hand, LoRaWAN is an open standard that defines a network architecture for the IoT, in which the gateways relay messages between end nodes and a central network server. Gateways are connected to the network server via standard IP connections and act as a transparent bridge, converting radio packets into IP packets, and vice versa. The wireless communication between end nodes and gateways is built using the LoRa physical layer, by means of single-hop links between the end nodes and one or more gateways. On the other hand, our model introduces a mesh networking topology backed by a hybrid L2/L3 DV RP. Rather than being a complete architecture, it lays the foundations for decentralized IoT applications to be built upon, on the upper layers. In this context, the network hierarchy is flattened in logical terms, since all the nodes operate at the same level by running an instance of the RP, regardless of their role in terms of data generation and consumption, etc.

Our decentralization proposal is aimed at IoT applications or scenarios in which the deployment of a gateway-based, centralized topology such as LoRaWAN does not match their requirements or make it impractical. Causes for this are manifold, as discussed in previous chapters: the size of the area to cover, low ratio of end nodes per gateway, lack of Internet connection for the concentrators, distribution of data generation and consumption, etc. Ditching a centralized architecture model in favor of a decentralized one can have a positive impact in lowering the requirements of certain aspects of IoT deployments, for example regarding concentration of computational resources or energy usage. However, this will most likely come at the price of distributing this computational and networking burden among the different participating nodes. In terms of our proposal, this translates as nodes in a deployment will need to perform additional networking tasks (i.e., forwarding other nodes' data packets), to run an RP that will require additional computation efforts and energy consumption. Furthermore, by using only the end nodes' single-channel radio transceivers, which is less capable than the gateways' one, the aggregate attainable bandwidth will also be reduced.

In Section 2.1, several LPWAN were briefly analyzed (Sigfox, NB-IoT, Wize, etc.). The goal of this section was not to perform an exhaustive survey of the available technologies (there is already excellent literature doing so [1, 2, 10, 12]), but to highlight their most remarkable features and identify the differences between them. Besides the more technical details such as the modulation employed, spectrum usage or attainable data rate, we wanted to highlight aspects like *availability* –i.e., how easy it is to obtain devices using a given radio technology–, *ownership* –i.e., whether the whole stack of devices required in a real deployment using a certain radio technology can be owned or if there is always a dependency on a third party providing some of the resources (base stations, data processing, licensing, etc.)– and *affordability* –not only economical, but also technical. These aspects, therefore, were of high importance when it came to experiment with a radio technology outside its usual ecosystem, and tilted the balance in favor or LoRa.

Having this said, LoRa has a singular aspect, from the technical perspective, compared with other LPWAN radio technologies: simultaneous transmissions on the same channel, but using different SFs, can coexist and be successfully demodulated, thanks to their quasi-orthogonality property. This offered the possibility to experiment with solutions exploring this feature and taking advantage of it.

During the early stage of the present work, research on multi-hop networking for LoRaWAN [21, 22, 23, 24] or with LoRa [25, 26, 29] had already started. Some of the proposals also put the focus on the SF's properties, although from a different approach to ours [27, 28, 84]. In relation to the existing publications for multi-hop built around the LoRaWAN architecture, our proposal named LoRaMoto analyzed the topic at a large scale, while the others focused on smaller deployments of up to a few tens of nodes. By means of the simulations performed with OMNeT++ and the FLoRa framework, we could emulate the behavior of a pure LoRaWAN

network with 100 gateways and 7.500 end nodes, which we then modified beyond the LoRaWAN architecture specifications to implement packet forwarding capability between nodes. This gave us an insight on how this could be used to improve certain network performance aspects, such as PDR, and the trade-offs with other metrics such as latency or energy consumption.

In parallel to our work, *Kim et al.* published their research on multi-SF selection aimed at building multi-hop LoRa networks. In their work, the authors developed an iterative algorithm to allow single-channel to detect and tune the SF being used by an incoming transmission automatically. To do it, they extensively used the CAD feature of LoRa radio transceivers. Our approach to this problem also did, but using a different algorithm, as described in Section 5.1. The technical validation at a hardware level emerging from both theirs and our work provided the basis to further develop the work. *Kim et al.* evaluated an experimental deployment consisting of 9 devices and a gateway deployed over a campus facility. However, our interest focused on studying the properties of a LoRa-based mesh network in function of different variables like number of nodes, density and traffic load, and obtain metrics for throughput, latency, PDR, etc. For this, and since we intended to investigate our model at a larger scale, we opted for a simulation-based experimentation with OMNeT++ and the FLoRaMesh framework, which we forked from FLoRa to develop our own DV RP proposal for a LoRa mesh scenario. To the best of our knowledge, only *Kim et al.* have leveraged LoRa's simultaneous multi-SF properties to build mesh networks and, so far, no other researchers have studied it in a large-scale deployment (real or simulated).

As mentioned above, one of the costs of decentralization in the context of the networks analyzed in this work comes in the form of energy consumption. LoRaWAN is usually praised for its energy savviness. This is possible due to the usage of a low-power radio technology, but also due to the definition of three end node types or *Classes* (A, B and C) which have different roles, properties, and capacities. In particular, devices following the Class A specification are known to operate on batteries for months or years, due to their limited usage of the radio transceiver and the fact that they spend most of the time in low power consumption sleep mode. The addition of packet forwarding to end nodes in LoRaMoto or the operation of an RP in a mesh network breaks the energy savviness of LoRaWAN's Class A devices, and imposes a different set of requirements in terms of resources consumption. This aspect has deliberately been left outside the scope of this thesis, but is not of minor importance. For example, the long-lasting life expectancy of a battery-powered end node would be reduced to a few days, or even hours, if running continuously a RP and forwarding packets from its peers. While these devices still have a small energy footprint (well below 1 W at full CPU + LoRa transceiver load), they will require a redesign of their power sources to adapt them to the new computation and radio usage requirements. This could be in the form of bigger batteries and their combination with solar panels, or any other stable source of energy in the long-term.

Nevertheless, some of the reviewed literature already tackled this issue [30, 85, 31], and their proposals could be integrated into our LoRa mesh networking solution.

The proposed DV RP, in the form it was implemented and evaluated in Section 5.2, is the result of certain design decisions with the objective of providing a basic set of features, keeping its simplicity to the maximum. These features are, namely, routing between any pair of nodes with a multi-SF aware ToA metric to calculate the cost of the routes. Additionally, other metrics were added to the RP codebase to allow for comparison in the experiments, as well as mechanisms to keep the routing tables' health (e.g., loops mitigation). This focus on simplicity, however, comes at the expense of the RP not providing some functionalities which might be desirable in certain cases or environments. For example, neither end-to-end nor hop-to-hop acknowledgments are supported (which makes the RP lightweight but may reduce its reliability) nor there are explicit protections against Byzantine nodes (a malicious device could inject erroneous routes to redirect traffic through it). These problems are not new in the RPs domain, and solutions used in other scenarios can be adopted here to mitigate them, as long as their overhead (in terms of computation and additional network traffic) matches the limiting characteristics of underlying LoRa radio technology.

Security and privacy may be important concerns in those real-world IoT deployments were sensible or private data are transmitted. The LoRaWAN architecture ensures these properties by using AES-128 encryption and a set of keys (per-device and per-application), plus frame counter checks, on the end nodes and the network server. This way, even if packets cross gateways and Internet infrastructure, their data remain private, and its tampering should be noticed. Data encryption is not part of our RP and was not intended to be part of the present work. From the point of view of network performance, metrics like throughput, latency, etc. might be affected by the addition of security and privacy mechanisms; depending on the chosen strategy, the extra computational effort required may affect other processes in devices with constraint resources. Fortunately, SoCs of current embedded devices, like the ones used in Section 5.1 for experimentation, typically feature cryptographic hardware acceleration. Therefore, at least a very basic layer of security could easily be added to it by making all the participating nodes encrypt their data using a shared key. A higher level of security could be achieved in the routing layer by using private/public key pairs to encrypt, decrypt and verify packets, either network-wide or on a per-device basis. Additionally, the application layer may also use its own mechanisms to protect data privacy and security end-to-end, from the data source to the sink.

Real-world IoT deployments using LoRa mesh networking, beyond the gateway-centric architecture, are technically feasible. The hardware used in Section 5.1 for the multi-SF experiments has characteristics and resources equivalent to those devices found in IoT deployments, and is capable of running our DV RP in parallel to performing its own specific tasks. Therefore, LoRa mesh networks allow deploying ad-hoc systems that adapt to applications requiring coverage of large areas, flexible

or varying topologies and heterogeneous data transmission patterns. These benefits, however, will pose a trade-off with bandwidth, or data rate, and energy consumption, as certain networking tasks that were before concentrated, or new ones, will now be distributed among all the participating nodes.

Chapter 7

Conclusion

The work presented in this thesis focused on providing IoT networks that use the LoRa LPWAN radio technology with more flexible topologies than LoRaWAN's star-of-stars. After reviewing the existing literature, this was explored, analyzed and evaluated from two different perspectives: first by leveraging packet forwarding capacity between LoRaWAN end nodes and, second, by building a DV RP to create LoRa mesh networks. The former strategy allowed building systems with greater tolerance to gateway coverage issues (for instance, in the event of infrastructure failures), while still adhering to a centralized network model. The latter provided the layer on which to build decentralized systems where arbitrary pairs of nodes could possibly exchange information in a distributed way.

The proposed LoRaMoto system was built based on the LoRa radio technology and extended the LoRaWAN architecture with packet forwarding for end nodes. To determine its scalability and the effect of its parameters on the overall performance, we conducted several computer simulations to reproduce its operation in the aftermath of an earthquake in a mid-sized town. We analyzed LoRaMoto by running several simulations that explored different aspects of the system. We observed, regarding scalability, that the system performed consistently with hundreds of nodes, and it could scale up to a few thousand, but completely saturated beyond ten thousand nodes. This trend indicated an upper bound in the density of nodes that were part of the system. Tightly related, the density of gateways had 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.

Our proposed RP was designed with a minimalistic set of features, in order to reduce memory and computing footprint, making it suitable for embedded devices composed of a microcontroller and a LoRa radio chip. It merged L2 and L3 addressing and uses a proactive broadcast mechanism for nodes to exchange routes with their neighbors, keeping the routing tables up-to-date and propagating changes over the network. Its simplicity came at the expense of lack of certain features, such as multicast, route discovery, node-to-node or end-to-end transmission reliability.

A novel aspect of our RP was that it took into account LoRa's capacity to transmit and receive on different SFs. In combination with the radio chip's CAD features, this allowed working with nodes in a mesh network using different SFs simultaneously, with packets potentially being forwarded using multiple SFs along their path. The ToA metric we presented takes advantage of this feature, and has proven suitable for LoRa mesh networks with heterogeneous links and topologies. For these situations where nodes have different characteristics in terms of placement, number of neighbors and distance to them, etc. our proposal achieved better PDR, goodput and latency results than single-SF routing strategies using other metrics like HC or ETX. These contributions should be useful in real-world deployments, where nodes are expected to operate in diverse and heterogeneous environmental conditions.

Along this document, we identified certain aspects that required further investigation beyond the results achieved, which we would like to discuss as they provide indications for future work to be performed.

In the communication system for emergency response in the aftermath of an earthquake, we developed a strategy to allow end nodes to forward packets from their neighbors. There, node-to-node communications were leveraged, but only to overcome failures in the centralized star-of-stars topology. However, such a system spends most of the time in an "idle" mode previous to the disaster event. A strategy to improve its performance (in terms of packet delivery efficiency, etc.) should take advantage of this period to optimize different system aspects, such as transmission power for efficient usage of the radio spectrum.

We showed that this strategy improved certain aspects of the system performance, but further investigation should take into account the combination of both those 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. We also identified that 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.

The research we performed on decentralized LoRa-based mesh networks showed that our DV RP and the ToA metric took better advantage from heterogeneous network topologies rather than from regular ones. Further research could analyze more challenging and adverse network scenarios, including those with a low degree of interconnection between nodes and those with unbalanced bottlenecks. The results obtained from the simulator could be compared with experimental values obtained with real hardware in testbed environments, in order to detect any incoherence or aspect that the framework might not reproduce accurately.

In both cases above, and as recognized in the limitations stated in Section 1.6, energy consumption by end nodes was left outside the discussion in this work. The adoption of a flexible mesh network topology comes at the price of requiring more effort (i.e., energy) by end nodes to receive and forward packets at any moment in time (unlike LoRaWAN, where end nodes typically spend most of the time in an energy-efficient sleep mode). A few of the surveyed proposals from the state-of-the-art addressed this problem, and suggested energy-savvy mechanisms to coordinate end nodes for data forwarding. These strategies could possibly be combined with our DV RP, depending on the application requirements (i.e., data rate and latency) and achieve long battery operation times, comparable to those of LoRaWAN end-nodes.

Bibliography

[1] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.

[2] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 855–873, 1 2017.

[3] Semtech, "LoRa technology overview," 2021. Accessed: 2021/06/22.

[4] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, 2016.

[5] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680 – 698, 2018.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[7] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[8] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3gpp narrowband internet of things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, 2017.

[9] P. M. Evjen, "The Wize protocol, LPWAN for smart cities," Tech. Rep. WP016, Radiocrafts AS, Sandakerveien 64, NO-0484 Oslo, Norway, 2016.

[10] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J.-C. Prévotet, "Internet of mobile things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs standards and supported mobility," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1561–1581, 2019.

[11] J. P. Queralta, T. Gia, Z. Zou, H. Tenhunen, and T. Westerlund, "Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: beyond LoRa and LoRaWAN," *Procedia Computer Science*, vol. 155, pp. 343–350, 2019. The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.

[12] S. Kartakis, B. D. Choudhary, A. D. Gluhak, L. Lambrinos, and J. A. McCann, "Demystifying low-power wide-area communications for city iot applications," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, WiNTECH '16, (New York, NY, USA), p. 2–8, Association for Computing Machinery, 2016.

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

[14] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?," *IEEE Wireless Communications Letters*, vol. 6, pp. 162–165, April 2017.

[15] LoRa Alliance Technical Commitee, "LoRaWAN 1.1 Specification," 2017. Accessed: 2020-01-22.

[16] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, pp. 1–38, 2018.

[17] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, "Throughput, coverage and scalability of lora lpwan for internet of things," *IEEE/ACM International Symposium on Quality of Service (IWQoS 2018)*, June 2018.

[18] R. Pueyo Centelles, F. Freitag, R. Meseguer, and L. Navarro, "Beyond the star of stars: An introduction to multihop and mesh for LoRa and LoRaWAN," *IEEE Pervasive Computing*, pp. 1–10, 2021.

[19] J. R. Cotrim and J. Kleinschmidt, "LoRaWAN mesh networks: A review and classification of multihop communication," *Sensors*, vol. 20, no. 15, 2020.

[20] A. Osorio, M. Calle, J. D. Soto, and J. E. Candelo-Becerra, "Routing in LoRaWAN: Overview and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 72–76, 2020.

[21] J. Dias and A. Grilo, "LoRaWAN multi-hop uplink extension," *Procedia Comput. Sci.*, vol. 130, no. C, pp. 424–431, 2018.

[22] D. Lundell, A. Hedberg, C. Nyberg, and E. Fitzgerald, "A routing protocol for LoRA mesh networks," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–19, 2018.

[23] B. Van de Velde, "Mul-hop lorawan: including a forwarding node," Master's thesis, University of Antwerp, 2018.

[24] C. Ebi, F. Schaltegger, A. Rüst, and F. Blumensaat, "Synchronous LoRa mesh network to monitor processes in underground infrastructure," *IEEE Access*, vol. 7, pp. 57663–57677, 2019.

[25] H.-C. Lee and 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*, vol. 67, pp. 2177–2187, Sep 2018.

[26] K. Ke, Q. Liang, G. Zeng, J. Lin, and H. Lee, "Demo abstract: A lora wireless mesh networking module for campus-scale monitoring," in *Information Processing in Sensor Networks (IPSN)*, pp. 259–260, 2017.

[27] B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken, and K. Steenhaut, "Enabling RPL multihop communications based on LoRa," in *WiMob*, pp. 1–8, 2017.

[28] S. Kim, H. Lee, and S. Jeon, "An adaptive spreading factor selection scheme for a single channel LoRa modem," *Sensors*, vol. 20, p. 1008, 2 2020.

[29] C. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-hop lora networks enabled by concurrent transmission," *IEEE Access*, vol. 5, pp. 21430–21446, 2017.

[30] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, "Evaluating lora energy efficiency for adaptive networks: From star to mesh topologies," in *WiMob*, pp. 1–8, Oct 2017.

[31] A. Abrardo and A. Pozzebon, "A multi-hop LoRa linear sensor network for the monitoring of underground environments: The case of the medieval aqueducts in Siena, Italy," *Sensors*, vol. 19, no. 2, p. 402, 2019.

[32] C. T. Duong and M. K. Kim, "Multi-hop linear network based on LoRa," *Advanced Science and Technology Letters*, vol. 150, pp. 29–33, 2018.

[33] M. McCauley, *RadioHead Packet Radio Library for embedded microprocessors*, 2014.

[34] Sudo Mesh and Secure Scuttlebutt, *disaster.radio - a disaster-resilient communications network powered by the sun*, 2018.

[35] K. Hester *et al.*, *Meshtastic - An opensource hiking, pilot, skiing, secure GPS mesh communicator*, 2020.

[36] Pycom, *Pymesh - LoRa full-mesh network technology.*, 2020.

[37] M. Westenberg, *Single Channel LoRaWAN Gateway*, 2020.

[38] J. Braam, *A LoRaWan Gateway in Lua*, 2016-2019.

[39] Heltec, *HT-M00 Dual Channel LoRa Gateway*, 2020.

[40] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[41] S. K. Singh, M. Singh, D. K. Singh, *et al.*, "Routing protocols in wireless sensor networks–a survey," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, no. 2, pp. 63–83, 2010.

[42] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[43] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[44] NiceRF, *SV-MESH Mesh Network Modules*, 2020.

[45] A. Miaoudakis, N. Petroulakis, and I. Askoxylakis, "Communications in emergency and crisis situations," in *International Conference on Distributed, Ambient, and Pervasive Interactions*, vol. 8530, pp. 555–565, 06 2014.

[46] J. C. Araneda, H. Rudnick, S. Mocarquer, and P. Miquel, "Lessons from the 2010 chilean earthquake and its impact on electricity supply," in *2010 International Conference on Power System Technology*, pp. 1–7, 2010.

[47] P. S. Paul, K. Hazra, S. Saha, S. Nandi, S. Chakraborty, and S. Das, "Generating crisis maps for large-scale disasters: Issues and challenges," in *Wireless Public Safety Networks 3* (D. Câmara and N. Nikaein, eds.), pp. 67 – 98, Elsevier, 2017.

[48] R. M. Santos, J. Orozco, S. F. Ochoa, R. Meseguer, and D. Mosse, "Providing real-time message delivery on opportunistic networks," *IEEE Access*, vol. 6, pp. 40696–40712, 2018.

[49] D. Fischer, O. Posega, and K. Fischbach, "Communication barriers in crisis management: A literature review," in *ECIS 2016 Proceedings*, pp. 1–19, 06 2016.

[50] T. Girard, "Barriers to communicating disaster response information to the public during disaster situations," in *PhD Thesis, Karlsruhe Institute of Technology - KIT*, pp. 1–161, 2017.

[51] Y. Murayama and K. Yamamoto, "Research on disaster communications," in *IFIP Advances in Information and Communication Technology*, pp. 1–11, vol 516. Springer, Cham, 2019.

[52] M. L. Spialek and J. B. Houston, "The influence of citizen disaster communication on perceptions of neighborhood belonging and community resilience," *Journal of Applied Communication Research*, vol. 47, no. 1, pp. 1–23, 2019.

[53] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, "Modelling and simulation of Opportunistic IoT Services with Aggregate Computing," *Future Generation Computer Systems*, 2019.

[54] M. Conti, A. Passarella, and S. K. Das, "The internet of people (iop): A new wave in pervasive mobile computing," *Pervasive and Mobile Computing*, vol. 41, pp. 1 – 27, 2017.

[55] H.-C. Jang, Y.-N. Lien, and 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, (New York, NY, USA), p. 623–627, Association for Computing Machinery, 2009.

[56] S. F. Ochoa and R. Santos, "Human-centric wireless sensor networks to improve information availability during urban search and rescue activities," *Information Fusion*, vol. 22, pp. 71 – 84, 2015.

[57] B. Manoj and A. H. Baker, "Communication challenges in emergency response," *Commun. ACM*, vol. 50, p. 51–53, Mar. 2007.

[58] R. Sugihara and R. K. Gupta, "Path planning of data mules in sensor networks," *ACM Trans. Sen. Netw.*, vol. 8, Aug. 2011.

[59] A. Martín-Campillo, J. Crowcroft, E. Yoneki, and R. Martí, "Evaluating opportunistic networks in disaster scenarios," *J. Netw. Comput. Appl.*, vol. 36, p. 870–880, Mar. 2013.

[60] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A survey on multihop ad hoc networks for disaster response scenarios," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 647037, 2015.

[61] L. Sciullo, F. Fossemo, A. Trotta, and M. Di Felice, "Locate: A lora-based mobile emergency management system," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Dec 2018.

[62] L. Sciullo, A. Trotta, and M. D. Felice, "Design and performance evaluation of a LoRa-based mobile emergency management system (LOCATE)," *Ad Hoc Networks*, 2020.

[63] J. Orozco, R. Santos, S. F. Ochoa, and R. Meseguer, "A stochastic approach for modeling message dissemination in opportunistic networks," *Wireless Personal Communications*, vol. 97, no. 2, pp. 2207–2228, 2017.

[64] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," *ACM Transactions on Sensor Networks*, vol. 15, pp. 1–35, feb 2019.

[65] D. Patel and 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)*, pp. 1–5, June 2017.

[66] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of lora networks in a smart city scenario," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2017.

[67] N. Benkahla, H. Tounsi, Y. Q. Song, and M. Frikha, "Enhanced ADR for LoRaWAN networks with mobility," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1–6, 2019.

[68] J. J. Kang, I. Khodasevych, and 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, pp. 1–6, IEEE, 2017.

[69] R. Lahouli, M. H. Chaudhary, S. Basak, and B. Scheers, "Tracking of rescue workers in harsh indoor and outdoor environments," in *International Conference on Ad-Hoc Networks and Wireless*, pp. 48–61, Springer, Cham, oct 2019.

[70] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE Internet of Things Journal*, vol. 6, pp. 3097–3109, apr 2019.

[71] J. Ortin, M. Cesana, and A. Redondi, "Augmenting lorawan performance with listen before talk," *IEEE Transactions on Wireless Communications*, vol. 18, pp. 3113–3128, jun 2019.

[72] C. Gomez and J. Crowcroft, "Multimodal Retransmission Timer for LPWAN," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[73] G. Smart, N. Deligiannis, R. Surace, V. Loscri, G. Fortino, and Y. Andreopoulos, "Decentralized Time-Synchronized Channel Swapping for Ad Hoc Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8538–8553, 2016.

[74] C. G. Ramirez, A. Sergeyev, A. Dyussenova, and B. Iannucci, "Longshot: Long-range synchronization of time," in *Information Processing in Sensor Networks (IPSN)*, pp. 289–300, apr 2019.

[75] R. Pueyo Centelles, F. Freitag, R. Meseguer, L. Navarro, S. F. Ochoa, and 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*, vol. 31, no. 1, 2019.

[76] L. Cerdà-Alabern, A. Neumann, and P. Escrich, "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, (New York, NY, USA), p. 23–30, Association for Computing Machinery, 2013.

[77] Chilean National Statistics Institute (INE), "Chile census," 2017. Accessed: 2020-01-22.

[78] O. Ltd., "OMNeT++ discrete event simulator," 2020. Accessed: 2020-01-22.

[79] G. Slabicki, M; Premsankar, "FLoRa—a framework for LoRa simulations," 2018. Accessed: 2020-01-22.

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

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

[82] J. Gromeš, *RadioLib – Universal wireless communication library for Arduino*, 2020.

[83] R. Pueyo Centelles, M. Slabicki, and G. Premsankar, "FLoRaMesh—a framework for LoRa mesh simulations (forked from flora)," 2020. Accessed: 2020-12-01.

[84] G. Zhu, C.-H. Liao, T. Sakdejayont, I.-W. Lai, Y. Narusue, and H. Morikawa, "Improving the capacity of a mesh LoRa network by spreading-factor-based network clustering," *IEEE Access*, vol. 7, pp. 21584–21596, 2019.

[85] M. Anedda, C. Desogus, M. Murroni, D. D. Giusto, and G. Muntean, "An energy-efficient solution for multi-hop communications in low power wide area networks," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–5, 2018.