electronics

MDPI

*Article*

# A Resilient LoRa-Based Solution to Support Pervasive Sensing

**Pietro Manzoni [1], Salah Eddine Merzougui [2], Claudio Enrico Palazzi [2,*] and Paolo Pozzan [2]**

[1] Departamento de Informática de Sistemas y Computadores, Universitat Politècnica de Valencia, 46022 Valencia, Spain; pmanzoni@disca.upv.es
[2] Dipartimento di Matematica "Tullio Levi-Civita", Università Degli Studi di Padova, 35122 Padova, Italy; merzougu@math.unipd.it (S.E.M.); paolo.pozzan@studenti.unipd.it (P.P.)
* Correspondence: cpalazzi@math.unipd.it

**Abstract:** Today, billions of small devices that can sense things are connected, creating the Internet of Things (IoT). This major technological step has led to ideas like smart cities, smart factories, and smart countries. One important use of this technology is pervasive sensing, which could benefit from a network that covers a wide area but does not use much power. This paper looks closely at the advantages and disadvantages of using LoRa—a network technology that can reach long distances with limited energy use—in situations like this. To this aim, we have created a holistic solution to manage the considered network enabling synchronization, routing, and reliability. In particular, we have even developed an adaptive spreading factor mechanism, simple and effective in allowing the network to cope better when the connection is not very good.

**Keywords:** LoRa; IoT; smart cities; mesh networks; sensing; resiliency

check for updates

**Citation:** Manzoni, P.; Merzougui, S.E.; Palazzi, C.E.; Pozzan, P. A Resilient LoRa-Based Solution to Support Pervasive Sensing. *Electronics* **2023**, *12*, 2952. https://doi.org/10.3390/electronics12132952

Academic Editor: Nurul I. Sarkar

Received: 31 May 2023
Revised: 29 June 2023
Accepted: 29 June 2023
Published: 5 July 2023

## 1. Introduction

The Internet of Things (IoT), characterized by interconnected devices capable of sensing, actuating, and communicating, has emerged as a transformative force in our digital age. Its escalating growth reflects its potential to redefine how we interact with our physical and digital worlds. In urban development, IoT plays an instrumental role in creating smart cities, enabling efficient resource utilization, and improving service delivery, thereby enhancing the overall quality of life. Similarly, IoT underpins the shift to smart factories in the industrial sector, transforming manufacturing through pervasive sensing and automation, thus catalyzing Industry 4.0. Beyond cities and factories, IoT has begun influencing broader national strategies, facilitating the management of resources and services at a national level. However, as IoT evolves, network connectivity, security, and privacy challenges become prominent. Given IoT's potential to generate massive volumes of data, it is critical to develop networks that can support reliable, secure, and efficient data exchange [1–3].

IoT is the required backbone to create and foster an environment where cities can improve and become smart cities [4–6]. To develop this backbone, infrastructure and communication technology are required. Since the urban environment can span over an extensive area, the positions of the sensors and actuators can vary greatly, and running cables everywhere is not always feasible; the obvious choice is wireless LPWAN technologies. Among these, the one with a rapidly increasing adoption rate is LoRa (from Long Range), a technology created and developed by the French company Cycleo, now acquired by Semtech, which still maintains it. In addition to its long transmission range (up to about 15 km) and low power consumption (as little as 25 mW to send a message), the characteristic that made LoRa one of the most popular LPWAN technologies is the fact that it is open source, the only requirement being owning a LoRa-capable device [7–9]. Other similar technologies, such as SigFox and LTE-M, are also used in the IoT environment. While they might be better in some characteristics, for example, LTE-M provides the highest throughput and lowest latency, and SigFox has, among the three, the highest maximum range, they both follow an infrastructure-as-a-service model, requiring their users to pay to send messages [10]. Additionally, since using LTE-M and SigFox relies on an existing network, they may not be available everywhere.

Overall, the current state-of-the-art presents challenges and limitations that must be addressed [11,12]. One major limitation is the focus on LoRaWAN-based networks, which may have constraints imposed by the LoRaWAN standard. Another challenge arises in achieving scalability in peer-to-peer communication, particularly in larger networks. Additionally, optimizing energy consumption requires the development of complex hierarchical routing protocols and efficient implementation strategies. The inaccurate selection of parent nodes due to inconsistent received signal strength indicator (RSSI) values further complicates the network's performance. Also, neglecting end nodes and the absence of spreading factor modulation impacts scalability and connection robustness. Furthermore, the assumption of static signal strength values and the complexity of comparing RSSI values from different devices hinder accurate evaluation in practical scenarios. To enhance the performance, scalability, energy efficiency, and practicality of LoRa mesh networks, we need to address these challenges and issues.

Our proposed solution focuses on creating a mesh network using LoRa technology to support pervasive sensing. The solution involves a series of steps and actions performed by the nodes in the network. Upon device power-up, it reads a configuration file defining its behavior. The device can connect to sensor devices, disable unnecessary features like LED lights and Wi-Fi to save power, and initialize communication with a mesh network. In summary, our holistic solution includes the following:

- Synchronization capability among nodes to ensure self-initialization and self-maintenance, including reliable message delivery within scheduled time windows;
- A multi-hop routing protocol to enable communication of the sensed data and commands beyond the transmission range;
- An adaptive spreading factor mechanism that automatically adjusts the spreading factor used by the nodes to the network conditions to maintain efficient and stable connections.

To evaluate the performance of LoRa technology, we conducted outdoor and indoor tests, considering factors such as spreading factor, distance, signal-to-noise ratio (SNR), received signal strength indicator (RSSI), packet delivery ratio (PDR), sending times, and throughput. The results demonstrated that higher spreading factors increased transmission range, leading to longer sending times and reduced throughput. The mesh solution successfully delivered messages between the sender and receiver nodes in various scenarios. However, our solution has limitations. Firstly, using high spreading factors with LoRa is, by design, associated with a lower bandwidth, thus increasing sending times and negatively affecting the network performance. Secondly, our solution lacks connectivity to the Internet, limiting data exchange and the ability to synchronize with accurate time information from the cloud. Thirdly, we did not address security concerns or encryption of data transmitted over the mesh network, leaving it vulnerable to unauthorized access and potential data breaches.

Our solution provides a framework for creating a mesh network using LoRa technology. Nevertheless, it is crucial to consider the limitations regarding throughput, internet connectivity, and security when implementing the solution in real-life scenarios.

The rest of this paper is organized as follows. Section 2 provides a background of LoRa. Section 3 reviews previous relevant literature and limitations. Section 4 presents our study and the proposed solution. Section 5 describes the devices used in this project. Afterward, it discusses our findings and results and their implications in different scenarios and contexts. Finally, Section 6 summarizes the main points, offers recommendations based on the study's outcomes, explores the study's potential limitations, and suggests avenues for future research.

## 2. Background

One of the main characteristics of LoRa is that it uses techniques derived from chirp spread spectrum (CSS) modulation to send messages. These techniques allow LoRa to have six spreading factors ranging from 7 to 12. These factors are a transmission setting that controls how fast messages are sent out. Using higher spreading factors increases the time required to send a message, but, in return, it significantly increases the sensibility of the receiver device, resulting in a longer transmission range. Figure 1 shows how higher spreading factors increase sending times. When implementing a solution that uses LoRa, choosing the best spreading factor is vital; lower ones will have higher throughput, and in comparison, higher ones will have a greater maximum transmission range. It is important to note that devices using different spreading factors cannot communicate with each other, so every device in a network has to use the same one to communicate with its neighbors.

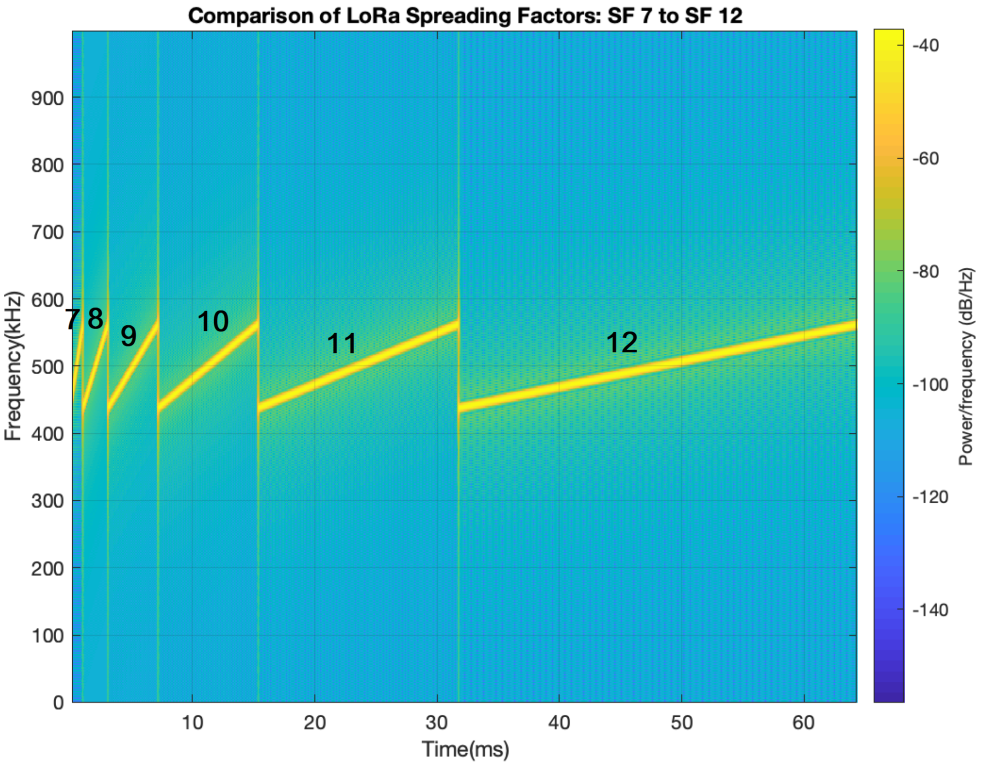

**Figure 1.** Example of time difference in sending times at different spreading factors (SF) [12].

It is also possible to increase the connection data rate by increasing the bandwidth, but the options are limited to three choices: 125, 250 and 500 kHz. Furthermore, this choice can be further limited by the country in which Lora is used.

In addition, spreading factors are orthogonal to each other, so messages sent with different spreading factors and transmitted simultaneously on the same frequency channel do not interfere, meaning that multiple LoRa-based solutions can coexist in the same physical space, provided each one adopts different spreading factors.

The actual maximum range of LoRa can vary greatly by changing variables other than the spreading factor, like how powerful the devices are. According to the specifications, the technology provides a 5 km range in urban areas and 15 km in rural areas where nodes have a line of sight of each other. In 2019, researchers established a new record and transmitted a message across a distance of 766 km by using a helium balloon [13].

Since LoRa only defines the lower physical layer that enables the long-range communication link, often called LoRa PHY, the upper networking layers are lacking. LoRaWAN is one of several protocols that define the upper layers of the network stack and the one that is used more often. It acts mainly as a network layer protocol for managing communication between LPWANs gateways and end-node devices as a routing protocol. It is maintained by the LoRa Alliance, an open, non-profit association created in 2015 to support, develop and ensure interoperability between LoRaWAN devices, similar to what the Wi-Fi Alliance does [14]. Projects that use LoRa almost always use LoRaWAN, but this is not necessary as LoRa can be used by itself.

The LoRaWAN network architecture is a star-of-star where gateways relay messages between the end-devices and a central network server. This means that every end-node is connected to a more powerful device (a gateway), and, in turn, this device is connected to another one that is usually IP-capable and connected to the Internet.

Due to the topology it uses and its focus on end-node-to-gateway communication, LoRaWAN has inherently some limitations on what it can be used to do; particularly, it cannot properly have node-to-node communication, meaning no real mesh network can be implemented with it. Due to this, LoRaWAN has not been used in this work.

## 3. Related Work

A few works have already tried implementing a mesh network using LoRa. In [15], Huang and Ke propose a solution where nodes have sort of a parent/child relationship, moving away from the typical idea of a mesh network and opting for a tree-shaped topology where the gateway is the root node. When a node wants to join the network, it will choose, based on the RSSI of messages and their closeness, its parent among the nearby nodes, thus only creating the strongest possible connections. Each node can only exchange messages with either their parent or, if they exist, their children. The paper states that the gateway has global knowledge of the network, but needs to state how this information is gathered. This is likely performed if the nodes are positioned in set places and never move. This would heavily limit the possible application of the solution proposed by the paper. Additionally, the work focuses mainly on message exchanges to and from the gateway, not allowing communication between nodes not part of a parent/child relationship. This would mean that every communication between nodes, even if they were close enough to exchange messages, would have to go through the gateway, thus posing significant limitations by not allowing packets to be re-routed around the dead node. Finally, while choosing a parent node based on the strength of their connection is a good idea, it needs to address the problem discussed in the previous paper: RSSI values may not be consistent if measured by different devices.

In their work, Lundell et al. discussed a modified version of the routing protocol AODV to deliver packets from the LoRa Network to the back-end infrastructure [16]. This modified protocol is used by gateways to move messages to and from back-end-connected gateways. When a node has information to deliver to the mainframe, it forwards the message to its gateway, which will, in turn, search for a path to an internet-connected device. The paths considered in this search only consider gateway nodes and completely ignore end-node. This implementation does not resemble a mesh network since only gateways are considered for routing, and typically, they are way fewer than end nodes. The use of LoRaWAN imposes this limitation. Additionally, using a reactive routing protocol means that the network is flooded every time there is a search for a route, wasting time and decreasing the responsiveness of the network.

Huh and Kim present the implementation and advantages of a LoRa-based mesh network specifically designed for IoT applications [17]. The paper discusses the network's architecture, deployment, and its ability to provide long-range and low-power communication. It emphasizes the scalability, robustness, and energy efficiency of LoRa technology for various IoT use cases. However, the paper has limitations, including a lack of comprehensive evaluation in real-world scenarios, a lack of comparative analysis, a specific application focus that may limit generalizability, and a limited discussion of challenges and unresolved issues. Further addressing these limitations would enhance understanding and potential improvements of the proposed network.

Kirichek et al. use a matrix to represent the nodes and the connection between nodes [18]. Each value in the matrix represents the signal strength between nodes (RSSI). The path towards a node is calculated using these values and the shortest distance between nodes. Basically, this algorithm chooses the path where the chance of an error occurring during transmission is the lowest. However, the paper needs to state how the nodes share signal strength information, and it seems like it should be static, meaning that the values would not change and should be known ahead of time. Consequently, this solution cannot be applied when nodes are at least somewhat mobile. Additionally, each node already knows the position of all the others, again limiting the use of the algorithm in environments with moving nodes. Finally, although it is generally true that having a higher value is better, chip manufacturers often use different scales for RSSI making it complicated to compare values obtained from other devices.

Cotrim and Kleinschmidt's work offers a comprehensive review and classification of LoRaWAN mesh networks, focusing on multi-hop communication [19]. The study examines various approaches, protocols, and architectures employed in LoRaWAN mesh networks and categorizes them based on different characteristics. By providing insights into the LoRaWAN mesh networks' current state of the art, the review aids in understanding their advantages and limitations. However, the review primarily focuses on LoRaWAN-based mesh networks, which may have constraints and limitations imposed by the LoRaWAN standard. These limitations include restrictions on the number of gateways and limited support for multi-hop communication. Furthermore, as the field of LoRa mesh networks continues to evolve and new solutions emerge, the classification provided in the review may only cover some possible variations and approaches.

The work by Jiang et al. presents a hybrid low-power wide-area mesh network for IoT applications [20]. The network combines LoRa and BLE technologies to achieve effective and dependable communication. It also provides improved energy efficiency, expanded coverage, and support for a range of IoT devices. The effectiveness of the network in terms of coverage, power usage, and data transmission reliability is demonstrated by experimental results. However, there are limitations, such as the need for thorough scalability analysis, a lack of evaluation and comparison with alternative solutions, and a brief discussion of security and privacy issues.

Berto et al. introduce a mesh network architecture that leverages LoRa technology to enable long-range peer-to-peer communication [21]. The study focuses on establishing direct connections between nodes using LoRa devices, enabling efficient and low-power communication over extended distances. The proposed mesh network architecture offers scalability and flexibility for various IoT applications. However, we have to consider that the peer-to-peer communication approach may face limitations in terms of scalability, particularly as the network size and the number of interconnected nodes increase. Additionally, while direct communication between nodes provides benefits such as reduced latency and efficient data exchange, it may require additional routing and network coordination mechanisms to ensure reliable and optimal communication.

Hong et al. present a hierarchical routing protocol to optimize energy consumption in LoRa mesh networks [22]. The protocol considers the nodes' hierarchy and their roles in the network to enhance energy efficiency. By focusing on routing protocols, the study addresses the goal of extending the battery life of LoRa devices. However, we must note that the proposed hierarchical routing protocol introduces additional complexity regarding network management and node coordination, necessitating efficient implementation and control mechanisms. Furthermore, the effectiveness of the protocol's energy efficiency gains may vary depending on specific network conditions and application scenarios, requiring evaluation in different deployment scenarios.

Seungku et al. present an adaptive spreading factor selection mechanism to allow a LoRa network to have nodes that use different spreading factors. In their solution, each message has a preamble that is used by receiver nodes in order to perform activity detection [23]. When a node is listening for messages, it will cycle through all available spreading factors in the search for a node that is transmitting. When a node transmits a preamble using a particular spreading factor, the receiver will eventually receive it and temporarily set the same spreading factor to receive even the subsequent message right after the preamble ends. While this solution allows the sender and receiver to use the same spreading factor, it significantly increases the time required to send a message due to the need to have a preamble. Clearly, with an increase in sending times also comes an increase in power consumption. Additionally, it is also possible for the receiver to miss the preamble, thus losing the message.

Zhu et al. constructed a pipeline with multi-hop capability in order to create a LoRa network capable of going further than a single-hop scenario, especially in extreme environments [24]. Furthermore, they developed a system that uses multiple spreading factors simultaneously in order to achieve multiple access in a robust and efficient way. Here the choice of the different spreading factors used was made because they are orthogonal and, as such, do not interfere with each other, not due to some benefit that choosing one over another might have.

Croce et al. showed that it is not always possible to have independent LoRa transmission that uses different spreading factors simultaneously due to imperfect orthogonality [25]. Their testing showed that transmissions, particularly the higher spreading factors, can interfere with each other more due to the higher transmission times, resulting in data loss and the need to retransmit non-delivered messages. Knowing this, it might be more appropriate to adopt a solution that uses a single spreading factor for the whole network instead of multiple different ones at the same time, as proposed in [23].

Reynders et al. propose an algorithm that tries to identify the best possible spreading factor to use for each node of the network in order to maximize efficiency and reduce data loss [26]. They, however, fall into the same belief that simultaneous transmission that uses different spreading factors does not interfere with each other; thus, their algorithm could result in unexpected errors occurring during message exchange. Additionally, their view of an environment that is static and where nodes have no degree of movement limited the possible uses of their work.

## 4. Our Proposed Solution

Our solution employs a mesh network with no defined hierarchy, where all nodes work along with each other to maintain the network and deliver messages across it. To support this network, we have developed a comprehensive set of solutions that are summarized in the following.

- The network is self-initialized and self-maintained through **synchronization among nodes**. This also allows to perform certain actions at specific times, like polling a connected sensor every few minutes and creating predetermined receive windows (i.e., a node will transmit only when the destination will be listening). This feature can limit data loss, retransmission, and power consumption.
- Our **routing protocol** enables devices to communicate with each other even beyond the transmission range, with messages forwarded along a specific (multi-hop) route.
- We implemented an **adaptive spreading factor** that automatically adjusts the spreading factor used by the nodes in the network in order to keep the connection between devices stable and limit errors during message exchanges.

To allow message exchange between devices, we used LoRaCTP, a MicroPython library that allows the transfer of blocks of bytes over a LoRa channel [27]. This library provides mainly two functions, one to send and one to receive data between two nodes. These functions use acknowledgments and retransmission packets to ensure messages are correctly received. Along the base sends and receives, we created two more functions: one is a broadcast function that allows us to have a device send a message to more than one node, and the other one is a receive function that uses timeouts to limit the listening window of devices (the original receive function will keep listening until a message is received, potentially blocking a device indefinitely) and can support both unicast and broadcast messages (when a broadcast message is received, no acknowledgments are sent).

Figure 2 shows a simplified flow chart for our solution.

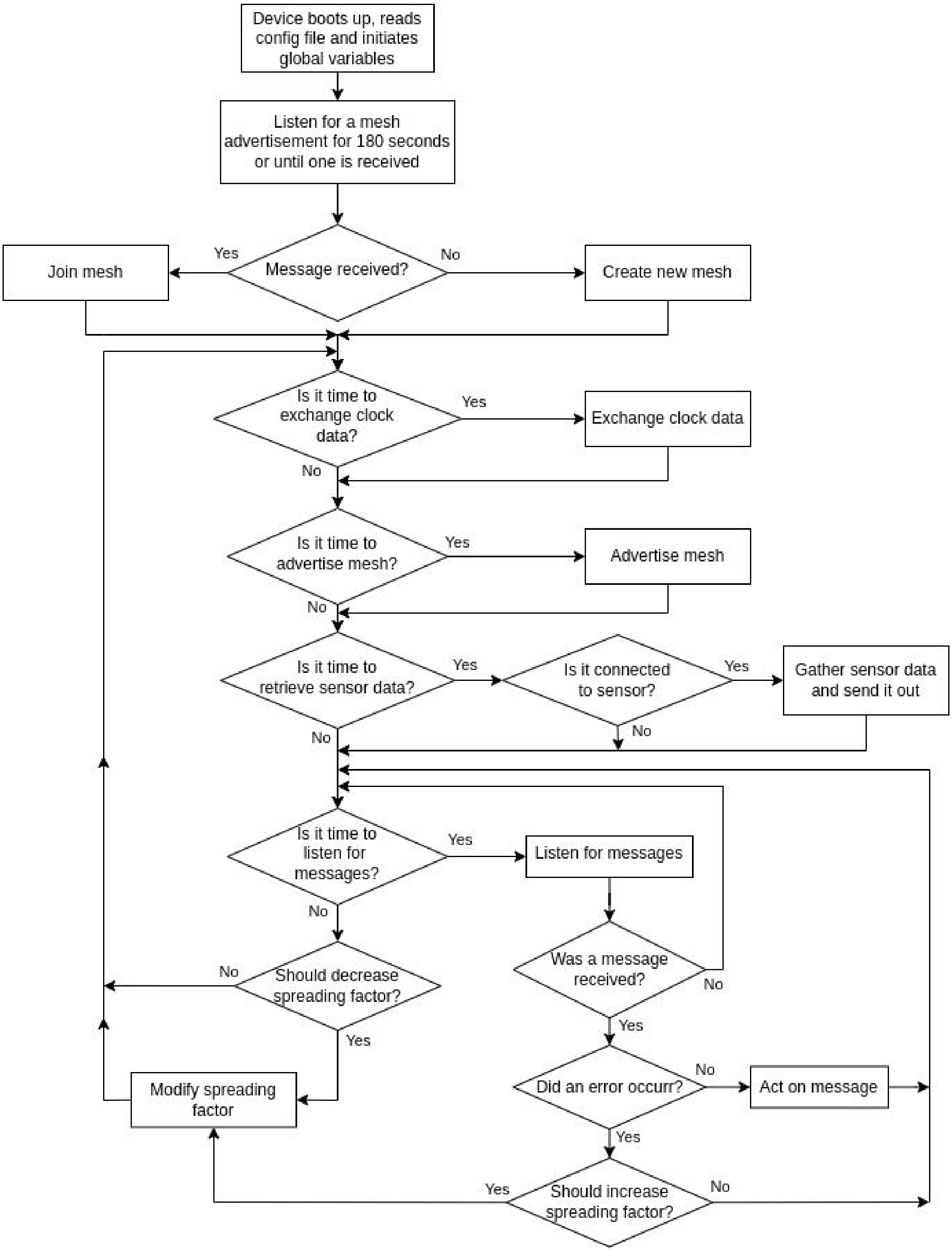
Device boots up, reads config file and initiates global variables
Listen for a mesh advertisement for 180 seconds or until one is received
Message received?
Yes
Join mesh
No
Create new mesh
Is it time to exchange clock data?
Yes
Exchange clock data
No
Is it time to advertise mesh?
Yes
Advertise mesh
No
Is it time to retrieve sensor data?
Yes
Is it connected to sensor?
Yes
Gather sensor data and send it out
No
No
Is it time to listen for messages?
Yes
Listen for messages
No
Should decrease spreading factor?
No
Yes
Modify spreading factor
Was a message received?
No
Yes
Did an error occurr?
No
Act on message
Yes
Should increase spreading factor?
Yes
No

**Figure 2.** Flow chart of our solution.

### *4.1. Synchronization among Nodes*

As previously noted, our network possesses self-management capabilities, eliminating the need for external initialization. Initially, when a device is powered up, it retrieves vital initialization information to guide its operational parameters, essentially configuring settings that influence its behavior. These data are derived from the configuration file, which dictates subsequent actions, instructing nodes on whether or not they need to establish a Bluetooth connection with a sensor device or initialize the GPS to obtain geographical coordinates. In an effort to conserve energy, the device is designed to automatically deactivate non-essential features, such as the LED light and Wi-Fi antenna, if they are not in use.

After the initial setup is finished, the node starts listening for advertisements from an existing mesh network for some time. If one is received, the node requests to join the network. Otherwise, if no advertisement is received in that window of time, the node creates a new mesh network, with itself as the only node present.

After the node is in a mesh network, it enters an endless loop running the main code of the device. Every action the nodes execute is preceded by a check on the time reported by the real-time clock (RTC) internal to the device. This clock is initialized when the node enters the mesh network: it uses an arbitrary value if the node is creating the mesh or it is sent over by the node that is bringing the device onto the existing network. Since the clock is shared with other nodes, every network device reports more or less simultaneously. This allows us to have a well-defined receive window, meaning that a node will send out a message only when it knows the receiver is listening. This synchronization allows nodes to save power by avoiding retransmitting packets that have not been received due to the other node not listening. To ensure the clock's consistency, some nodes broadcast their clock value and keep all nodes in sync. The node that creates the network initializes its internal clock with an arbitrary time and date since, in our solution, we did not focus on connecting the network to the Internet. Our implementation can easily be extended to have a node connected to the Internet and communicating with a server or a cloud infrastructure to exchange data and commands, as well as to retrieve the current time.

Inside the endless loop, each node will perform many actions, but it will mainly accomplish the following:

- Advertise the mesh network and bring in new nodes;
- Retrieve data from the connected sensor;
- Listen for messages.

Other actions a node can perform are generally tied to the ones listed above, controlled by what is in the config file, or performed to maintain the mesh network alive and reliable. For example, a node may respond to a message containing sensor data since it requires an acknowledgment, forward a message to the correct node, or retrieve its GPS coordinates. Every action requiring interaction with other nodes is performed after checking the current time in order to exchange messages only during the scheduled receive window.

### *4.2. Routing Protocol*

We created a multi-hop routing protocol to deliver messages across the network. Nodes periodically exchange network information. At first, each node broadcasts what it knows about the network topology (i.e., who its neighbors are). However, over time, the information exchanged grows with the knowledge of each node, and, at a certain point, each node will have complete knowledge of the network's topology.

With this knowledge, when a node needs to send a message to a node out of its transmission range, it will calculate all the possible paths and choose the shortest one (i.e., the one that requires the least amount of hops). Once the node chooses the path, it forwards the message and the previously calculated path to the next hop.

Our algorithm is a hybrid one. We chose not to use a reactive protocol since that would mean flooding the whole network when searching for a route, thus requiring large amounts of time and reducing the responsiveness of the network. Similarly, we did not choose a proactive protocol since it would require all nodes to calculate and store every possible route, wasting time computing routes that would likely never be used, especially in larger networks.

To keep the network representation that the nodes have consistent with the real one, it removes inactive (i.e., nodes that ran out of battery) nodes from path calculations. If the dead node is powered up again, it will act as a new node that wants to join a mesh network.

The path is calculated with a modified Dijkstra algorithm with a depth-first search. Since there are no negative weight cycles and the graph is unweighted, the algorithm has complexity $O(V + E)$, where $V$ represents the number of nodes in the network and $E$ represents the number of connections between these nodes. The algorithm returns all possible paths between the nodes and chooses the one with the least number of hops. To improve its responsiveness, our algorithm can also stop and return the first path found. This version of the routing algorithm may be used in larger networks where the high number of nodes and connections would mean that finding the best path could be consume too much time and energy.

### *4.3. Adaptive Spreading Factor*

To maintain a stable connection among nodes and limit the number of lost packets, we have created a method that allows the nodes to instruct the network to switch to a higher spreading factor to avoid further errors. In our experiments, we have chosen a threshold of three errors before the considered node increases its spreading factor and informs its neighbors to do the same. Whenever a node receives a message about increasing the spreading factor, it propagates the information to its other neighbors, thus eventually notifying all nodes. Clearly, the threshold for increasing the spreading factor can be easily modified according to the rate at which messages are sent out by nodes, or even considering other networking conditions, in order to be more reactive or more conservative.

Since one or more nodes may not correctly receive the notification to switch to a different spreading factor, our system periodically selects a node to temporarily change its spreading factor to other values to broadcast a message instructing all potential listeners to use the newer spreading factor. This system would allow defective nodes left behind to join the network again.

In our solution, nodes can only use spreading factors 7, 10 or 12 since we have empirically noticed that moving only to the next greater value, for example, from 7 to 8, does not substantially improve the connection health and would result in requiring multiple spreading factor increases, wasting additional time. Furthermore, using only a selected few options could avoid interference with other nearby applications using LoRa with different spreading factor values.

Since using a high spreading factor has drawbacks, nodes also have the ability to reduce the error counter in a similar way to what is performed for their increments; hence, the spreading factor can be decreased after a certain number of successful message deliveries or after a certain time with no errors. The notice to switch to a lower value is sent out via flooding, identically to the one for increasing it.

## 5. Results

This section describes the tests performed and the results gathered from those tests.

### *5.1. Employed Devices*

The devices used during the course of this project were microcontrollers made by Pycom. FiPy and LoPy4 are MicroPython-capable development boards that provide a wide variety of connectivity, notably LoRa. Both of these devices have an Xtensa dual-core 32-bit LX6 microprocessor, 4 MB of RAM, and 8 MB of external flash. This relatively powerful processor allows them to even support multi-threading. For LoRa communication, they use a Semtech SX1272 modem. To greatly improve transmission, an external antenna was connected to the devices. These devices were coupled with PyTrack and Pysense expansion boards. The expansion shields provide additional functionalities to the development boards, like GPS connectivity, allowing the device to retrieve its latitude and longitude coordinates and access built-in sensors like accelerometers, temperature, light, and many more.

Both FiPy and LoPy4 can be powered by an external battery through a micro-B USB port. When powered up, the device immediately executes the `boot.py` file in its flash memory. This code helps the device connect to a network or save power or even disable specific antennas if they are not being used. After that, the device runs the code that is stored in `main.py`. Additionally, each device has a `lib/` folder containing the libraries needed during execution.

The LoRa settings used while performing the tests were the following:

- Frequency: 868 MHz;
- Bandwidth: 125 kHz;
- Transmit power: 14 dBm.

We also recorded the power consumption of the devices during certain actions and reported it in Table 1.

**Table 1.** Power consumption during specific actions.

| Action | Power Consumption (mAh) |
|---|---|
| Idle (Wi-Fi on) | 100 |
| Idle (Wi-Fi off) | 40 |
| Listening for messages | 50 |
| Sending messages | 120 |
| Retrieving data from sensor | 80 |
| Regular sleep | 50 |
| Light sleep | 20 |
| Deep sleep | 10 |

On Pycom devices, different types of sleep exist:

- *Regular sleep*: All microcontroller functions keep running;
- *Light sleep*: Wireless modems are switched off, but the CPU and RAM remain active. The LoRa modem has to be re-initialized after wake-up;
- *Deep sleep*: In addition to what light sleep does, it also powers off the main CPU and RAM, leaving only a low-power coprocessor and RTC timer running. After waking up, the node will start the execution again, just like if the device was completely powered off and then back on.

### 5.2. LoRa Tests

First, a series of tests was performed to understand the limits of the LoRa technology and the devices we used.

#### 5.2.1. Outdoor Test

The first test was performed on a bike lane that went straight on for about 2.5 km. As Figure 3 shows, one device was placed, stationary and sending messages, on the white dot while another device was moving farther away and listening, stopping at the set distances indicated by the red dots. The same experiment was repeated multiple times; a higher spreading factor was used each time between the two nodes.

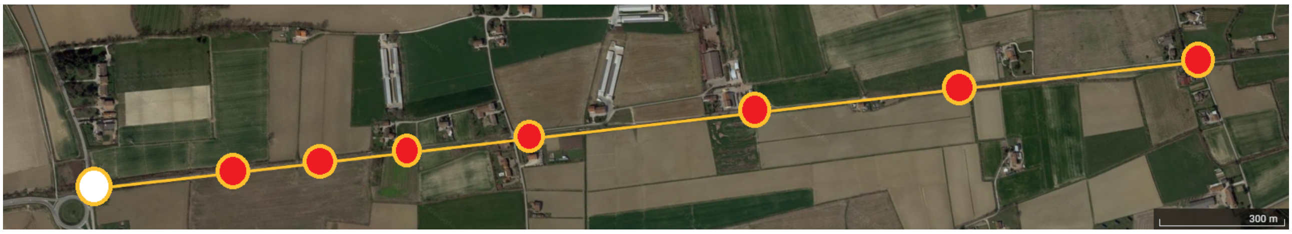

**Figure 3.** LoRa outdoor test.

During the test, the first node (white dot) sent messages in waves, one for each of the positions of the other node (red dots). The test recorded the packet delivery ratio (PDR) calculated by the following equation $\frac{P_{rec}}{P_{sent}}$ where $P_{rec}$ is the number of packets correctly received by the node on one of the red dots and $P_{sent}$ is the number of total packets sent by the node on the white dot. We also recorded the signal-to-noise ratio (SNR) and received the signal strength indicator (RSSI). SNR is a pure value obtained by the formula $\frac{P_{signal}}{P_{noise}}$, where $P_{signal}$ is the power of the signal (meaningful input) and $P_{noise}$ is the power of the background noise (meaningless or unwanted input). RSSI is expressed in decibels and indicates the power level being received by the antenna. We acknowledge the fact that RSSI values may not be as valuable for use cases that utilize different devices since the scale of values that can be obtained can vary between makers of wireless modems. However, they can still be helpful when compared with data obtained similarly. For RSSI, it is pretty much always true that a value closer to 0 indicates a stronger signal.

Figure 4 shows how the PDR varied as a function of the spreading factor used by the nodes and the distance between them. It is clearly seen that using higher values of the spreading factor significantly increases the transmission range between the two devices. When employing the spreading factor 7, there was a drop in PDR just after the 300 m mark, whereas with the spreading factor 10, it came after 700 m. Instead, the spreading factor 12 was able to maintain a perfect packet delivery ratio (100%) until the nodes were 2 km apart.

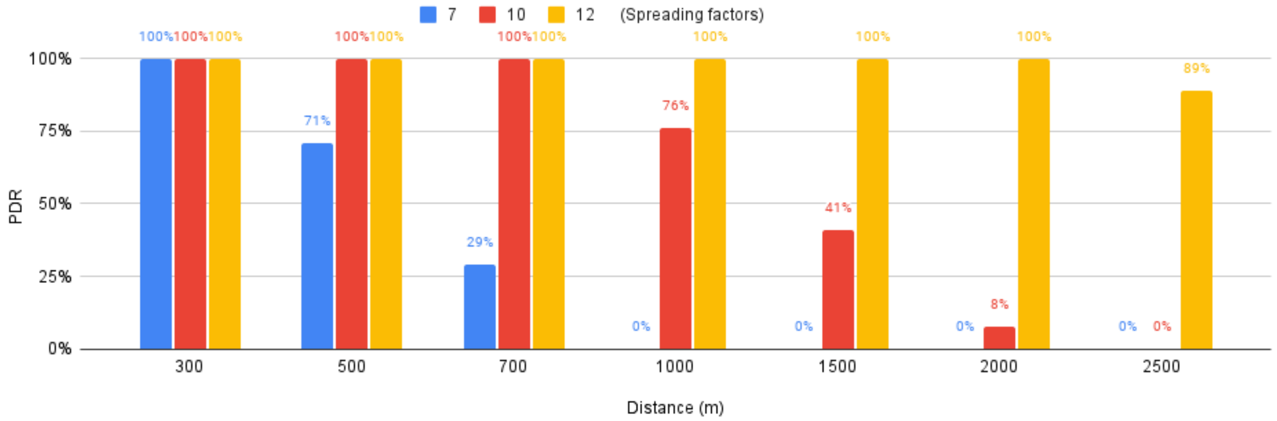

**Figure 4.** PDR as a function of the distance between nodes and spreading factor (7, 10, 12).

Figures 5 and 6 show how the SNR and RSSI varied as a function of distance. As one can expect, messages with higher SNR have significantly better PDR, since it means that the noise is much lower than the power of the signal. Similarly, messages with RSSI higher than −100 have a high probability of being delivered correctly. To know what SNR and RSSI values non-delivered packets would have had, the values for the previous and subsequent messages were taken and averaged. So, for example, if a message was not received and the previous one had an RSSI of −108, and the next one was −106, the missing packet would be assigned an RSSI of −107.

### 5.2.2. Indoor Tests

Another series of tests was performed indoors to understand the limits of the LoRa technology when used inside a building. These tests were carried out in Torre Archimede, the building housing the Department of Mathematics of the University of Padova. As shown in Figure 7, this building has seven floors, the ground floor, and two underground floors.

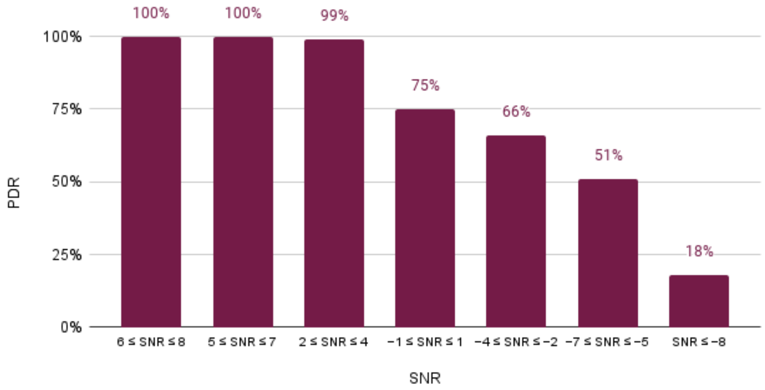

**Figure 5.** PDR as a function of SNR in LoRa outdoor test.

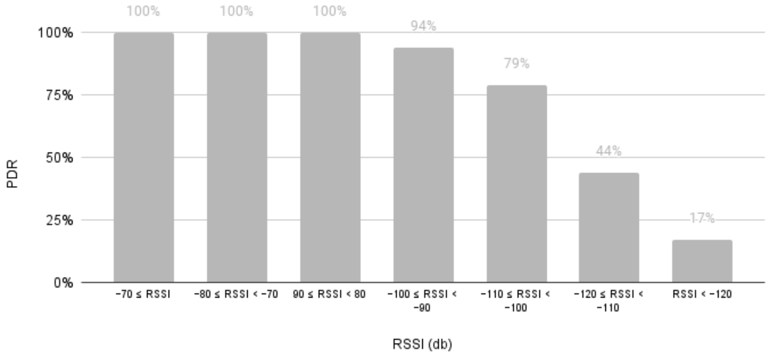

**Figure 6.** PDR as a function of RSSI in LoRa outdoor test.

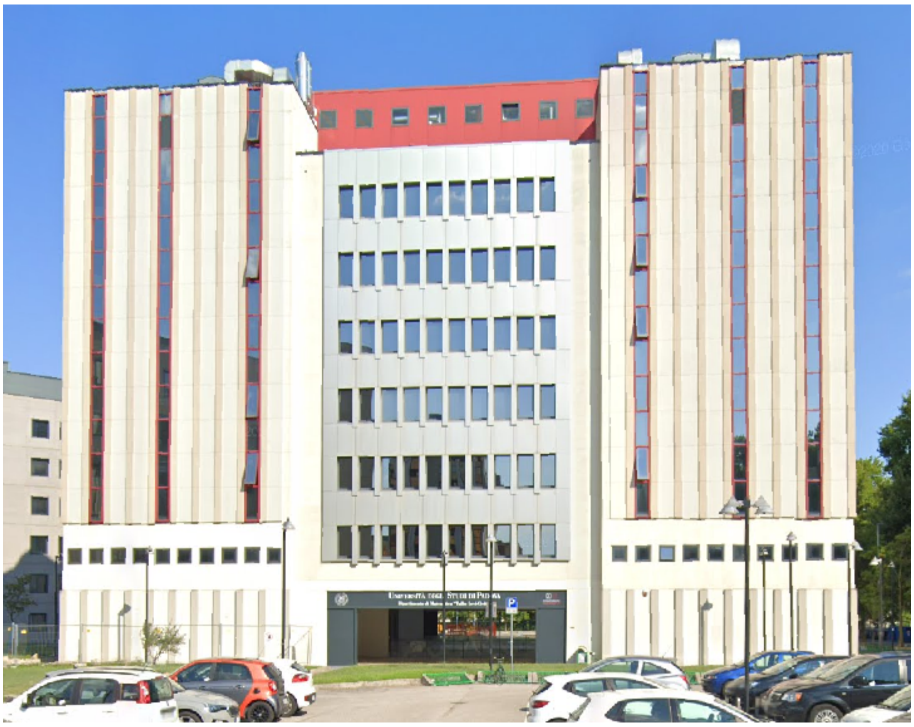

**Figure 7.** Torre Archimede, Mathematics Department of the University of Padova.

Similarly to the outdoor experiment, we recorded PDR, RSSI, and SNR for this series of indoor tests. When the first test was carried out, the first device was placed on the ground floor, and the second one was on the seventh floor, and both used a spreading factor of 7.

The first iteration of the test showed, as can be seen in Table 2, that the two nodes could communicate rather well, but some data loss occurred with a spreading factor of 7 (SF 7, in the table).

**Table 2.** Results obtained from the indoor LoRa tests.

| Configuration | PDR | RSSI (db) | SNR |
|---|---|---|---|
| Ground floor-7th floor (SF 7) | 93% | [−96; −84] | [1; 4] |
| Ground floor-5th floor (SF 7) | 100% | [−85; −76] | [3; 6] |
| Ground floor-7th floor (SF 10) | 100% | [−80; −72] | [5; 6] |

To guarantee perfect communication between the two devices, meaning a 100% PDR, two options are possible:

1. Moving the nodes closer together;
2. Using a higher spreading factor (more than 7).

For option 1, we moved the second device closer to ground level at the 5th floor. With this configuration, the nodes were able to communicate without errors. Similarly, with option 2, the nodes had 100%.

### 5.2.3. Other LoRa Tests

The greater transmission range achieved by the higher spreading factors comes at a cost: an increased time required to send messages. Figure 8 shows how sending times increase along with the spreading factor. This test was performed with messages of 244 bytes (20 bytes for the header and 224 bytes for the payload).

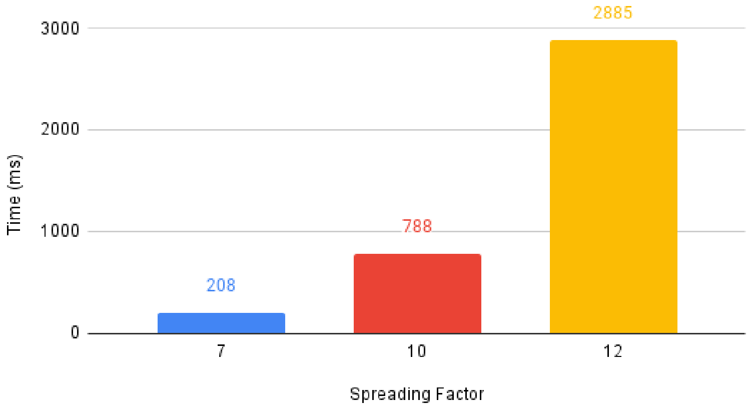

**Figure 8.** Time required to send a message as a function of spreading factor.

With spreading factor 7, sending the message takes about 200 ms, while sending the same message with spreading factor 12 requires almost 2900 ms. The increase of almost 14 times between 7 and 12 highlights the main drawback of using a high spreading factor. Having longer sending times means that fewer messages can be sent overall, partially because the channel is busy for longer amounts of time, and also since LoRa uses unlicensed frequencies, it has a 1% duty cycle (at least in Europe, where the test was performed).

The increased sending times also reflect on the connection's throughput between the nodes. Since the throughput is calculated by the formula $\frac{B_{sent}}{T}$, where $B_{sent}$ is the total amount of bytes sent during a $T$ time window. If $B_{sent}$ stays the same and $T$ increases, the resulting throughput will obviously decrease. Figure 9 shows the effect of higher spreading factors on the throughput.

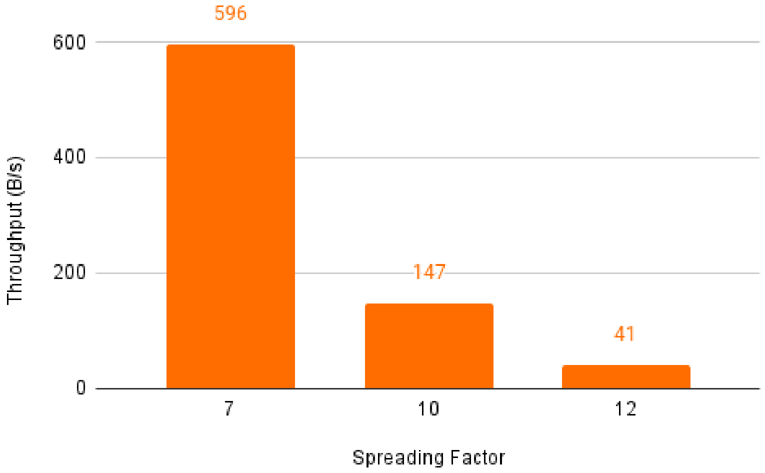

**Figure 9.** Throughput as a function of spreading factor.

Thanks to the data in Figures 8 and 9, it becomes clear that using the highest spreading factor is not always the best solution and a compromise between range (and PDR) and throughput has to be made.

### 5.3. Mesh Tests

A few different scenarios were used to test the mesh solution proposed in the previous chapter, along with the adaptively changing spreading factor. In this series of tests, one node marked with the letter *S* (for the sender) is sending out messages addressed to the node marked with the letter *R* (for the receiver). Like in the previous tests, we recorded RSSI, SNR, and, more importantly, the ratio of packets generated by the *S* nodes that are correctly delivered to *R*.

#### 5.3.1. Straight Line

In this experiment, the nodes were put along a (somewhat) straight line on a road, as seen in Figure 10. This configuration was chosen as it mimics that of street lights, as discussed in [17], where messages to control the state of each light (ON or OFF) could be sent from one node to the next one in line. While LoRaWAN could be used to cover this scenario, using it would require multiple gateway nodes to be present in order to connect all lampposts, increasing the number of devices needed to cover the same area and adding complexity.

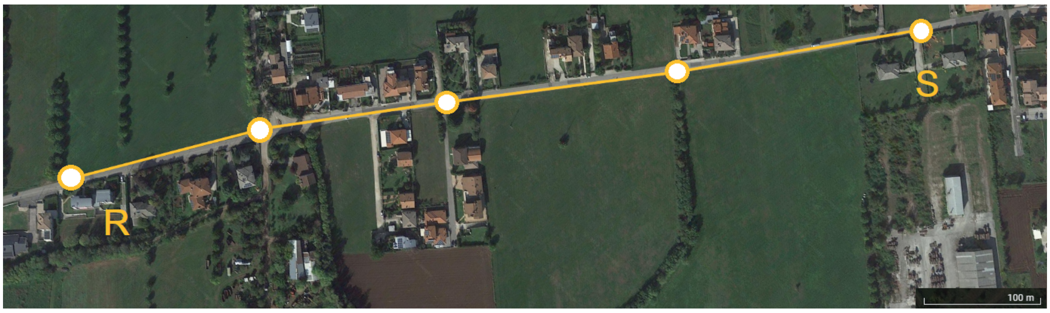

**Figure 10.** Positioning of nodes in straight line test.

All nodes were about 100/150 m from each other and used a spreading factor of 7. Table 3 contains the data gathered from this test.

This test showed that it could be possible to use the solution proposed in this paper to control street lighting in a smarter way instead of relying on hard-coded timers for the ON/OFF switch or light sensors that could provide unreliable results due to nearby light pollution.

**Table 3.** Results obtained from the straight line test.

| Measurement | Value |
|---|---|
| PDR | 100% |
| RSSI (db) | [−78; −64] |
| SNR | [6; 8] |

### 5.3.2. Urban Environment

In this experiment, the nodes were positioned in an urban environment on a step shape, as seen in Figure 11.

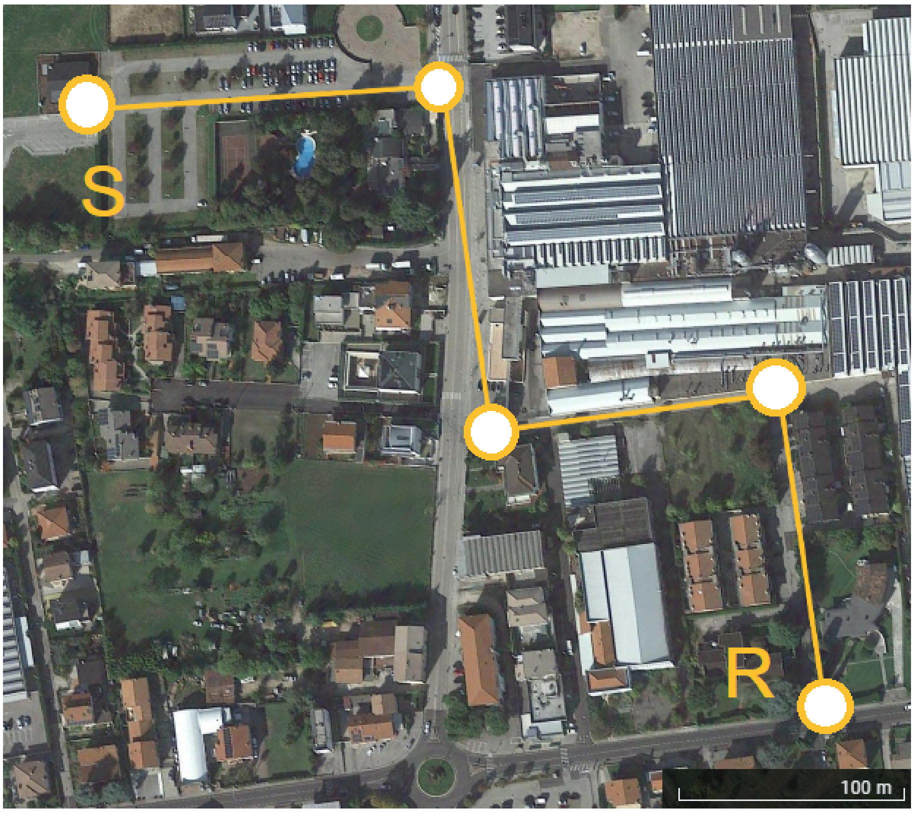

**Figure 11.** Positioning of nodes in urban environment test.

All nodes were about 100/150 m from each other and used a spreading factor of 7. Table 4 contains the data gathered from this test. This test is somewhat similar to the previous one. Still, while in the straight-line topology test more nodes could be connected to a single gateway using a higher spreading factor, this is now not the case when the devices are positioned behind corners that obstruct the line-of-sight between nodes. While nodes *S* and *R* might still communicate with each other by using a high spreading factor, this could not provide an error-less connection and, as seen before, using a higher spreading factor comes with a high cost. Having the ability to have relay nodes with our solution might be more expansive since more devices are needed. Still, it will produce a better solution where more data can be transferred between the network's two end nodes and with no packet loss.

**Table 4.** Results obtained from the urban environment test.

| Measurement | Value |
|---|---|
| PDR | 100% |
| RSSI (db) | [−88; −65] |
| SNR | [5; 7] |

### 5.3.3. Quarry

In this experiment, the nodes were put around a quarry, as seen in Figure 12. This configuration was chosen because it has nodes further apart (about 400 m) and uses spreading factor 10.

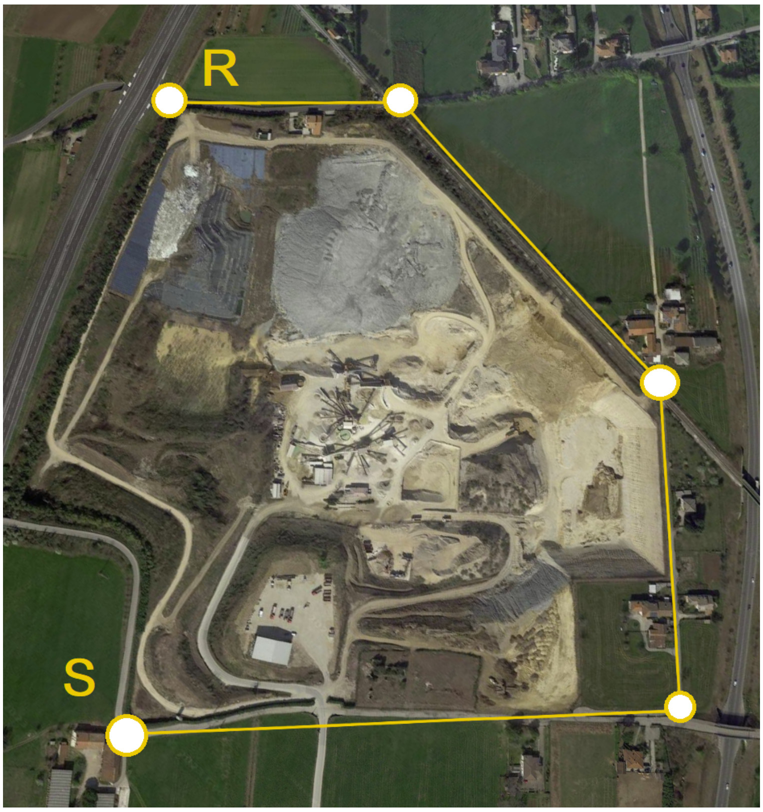

**Figure 12.** Positioning of nodes in quarry test.

Table 5 contains the data gathered from the test. The data from the shortest connection (about 100 m) was omitted as it would have skewed the RSSI and SNR values due to the nodes being closer.

**Table 5.** Results obtained from the quarry test.

| Measurement | Value |
|---|---|
| PDR | 100% |
| RSSI (db) | [−101; −89] |
| SNR | [4; 7] |

This test simulates nodes used to carry information around a potentially dangerous working environment and along higher distances. While LoRa may not be suitable for moving around highly time-critical information, especially with the high spreading factor and its increased sending times, it could still be used for other, less critical purposes, like notifying workers with new data and information gathered from sensors around the work environment.

### 5.3.4. Park

In this experiment, the nodes were put around a park, as seen in Figure 13. Nodes were using spreading factor 7 and were about 110 m from each other. In this experiment, packets could take more than one route to arrive at their destination.

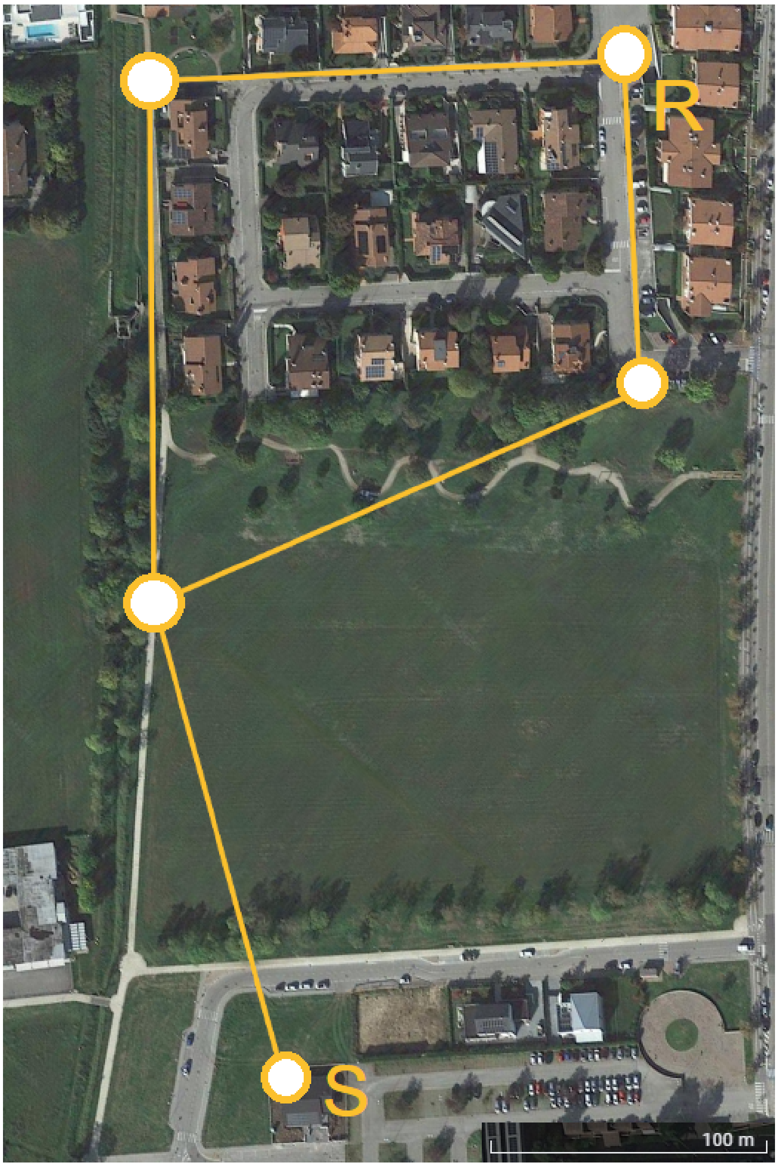

**Figure 13.** Positioning of nodes in park test.

Table 6 contains the data gathered from this test. Packets were all delivered to node *R* by the route that used the right node at the fork.

**Table 6.** Results obtained from the park test.

| Measurement | Value |
|---|---|
| PDR | 100% |
| RSSI (db) | [−81; −74] |
| SNR | [5; 8] |

This test shows how our solution could be used to gather data around a somewhat open environment. For example, the mesh network could be used to monitor the number of people at the park at different times through presence detection, and these data could be used to provide better services at the park and improve security measures for the citizens.

#### 5.3.5. Underground Garage

In this experiment, the sender node (node *S*) was put in an underground car park (2 floors below ground level). Another node was positioned atop the underground node, the two receivers were deployed, and the sender node would alternate between these destinations. The setup can be seen in Figure 14. Since, as explained earlier, LoRa performed rather well with nodes at 100/150 m apart, in this section we will focus on the connection between the node underground and the one above it.

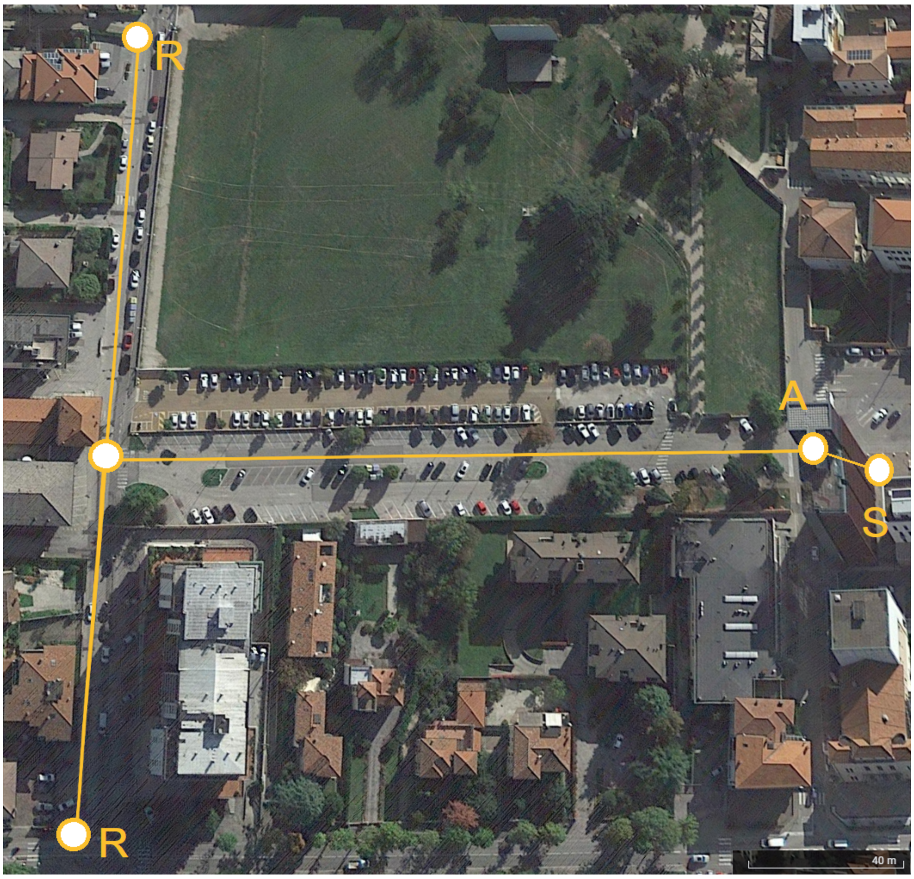

**Figure 14.** Positioning of nodes in the underground garage test.

The connection with the underground node was rather poor, and with the starting spreading factor of 7, there was a significant loss of packets. Due to this, the automatic spreading factor increase moved the nodes to 10, but even this was not very good, so another increase was performed, arriving at a spreading factor of 12. With the highest possible setting, the PDR was somewhat decent, but there was still a non-negligible packet loss. Tables 7–9 contain the data gathered from the test.

**Table 7.** Results obtained from the underground garage test with spreading factor 7.

| Measurement | Value |
|---|---|
| PDR | 50% |
| RSSI (db) | [−124; −109] |
| SNR | [−6; −4] |

**Table 8.** Results obtained from the underground garage test with spreading factor 10.

| Measurement | Value |
|---|---|
| PDR | 62% |
| RSSI (db) | [−112; −99] |
| SNR | [−2; 1] |

**Table 9.** Results obtained from the underground garage test with spreading factor 12.

| Measurement | Value |
|---|---|
| PDR | 87% |
| RSSI (db) | [−104; −99] |
| SNR | [1; 3] |

Seeing the not-so-great results with the underground node, even with the highest spreading factor, it seems unlikely that LoRa could be used when building a critical system where one (or more) node is underground and others are at ground level. It is still necessary to say that the performances can still vary greatly with the height of terrain the signal has to go through: less terrain in-between nodes will definitely provide a better PDR. Additionally, the type of terrain is also a factor that has to be considered since a reinforced concrete structure with a lot of steel would, for example, worsen the connection between nodes.

A non-critical system could surely still use this solution, for example, to build a presence detection that could provide real-time data on empty parking spaces in the underground garage with the help of acknowledgments and retransmissions to make sure packets are correctly delivered.

#### 5.3.6. Indoor Mesh Test

This test was performed inside Torre Archimede. One node was placed on the underground floor, two nodes on the third floor, and two others on the 7th floor. Figure 15 shows this.

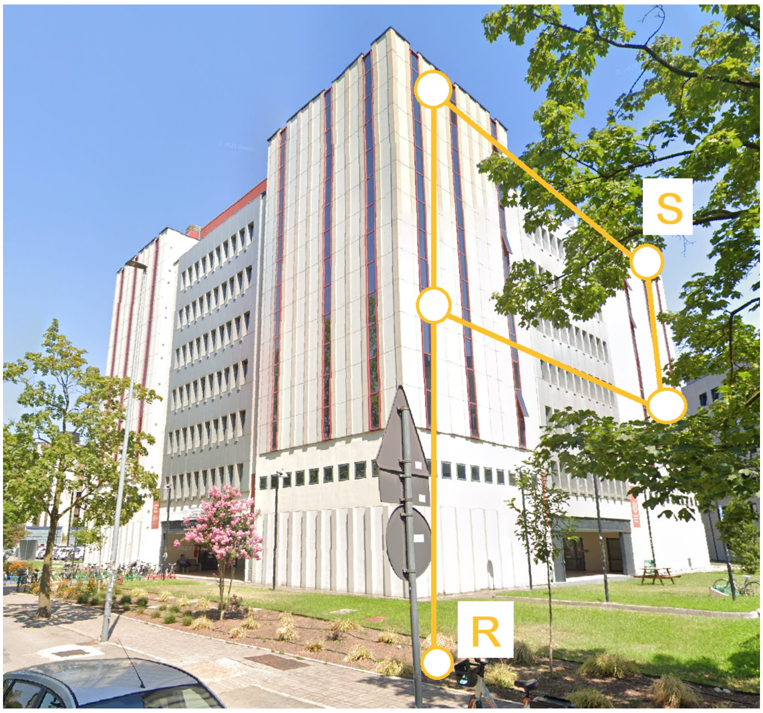

**Figure 15.** Positioning of nodes in the indoor mesh test.

The node used a spreading factor of 7 for this test. Table 10 contains the data gathered from this test.

This experiment, along with the previous indoor test, shows how LoRa and the solution we have outlined in this document could be used to implement a smart managing system for large buildings. It could be used to move data gathered from temperature and presence detection sensors to improve the overall efficiency of the building, lowering energy consumption and reducing lighting, heating, and pollution costs. Using a mesh network can make sure that, even in case a node dies, data can still be delivered to the right place.

**Table 10.** Results obtained from the indoor mesh test.

| Measurement | Value |
|---|---|
| PDR | 100% |
| RSSI (db) | [−109; −77] |
| SNR | [3; 7] |

### 5.3.7. Bike Lane Mesh Test

This test was performed along the same conditions as the test carried out on the bike lane outlined at the start of this section: one node would be still and send messages while the other node would move further away, stopping at specific distances to receive data before moving again.

The results in the first test are similar to those we obtained in this test, as seen in Figure 16. The nodes start with a spreading factor of 7, and when the message exchange starts when they are 500 m apart, the connection begins to show poor performance by dropping packets. For this reason, they move to a spreading factor of 10 to not lose any more messages. The positive results can be seen as only a slight dip in PDR at the 500 m mark, much higher than the one of the first outdoor Lora test. The nodes keep using a spreading factor of 10 up until they are 1000 m apart when the same connection issues return. At this distance, the nodes perform another spreading factor increase to use 12 and keep the PDR high. After the 2 km mark, the connection continues to show problems, but the two nodes already use the highest possible value for the spreading factor, so nothing can be done to avoid issues. The solution implemented was able to keep the connection stable and limit packet loss while, at the same time, the nodes were able to exchange messages with the highest possible throughput.

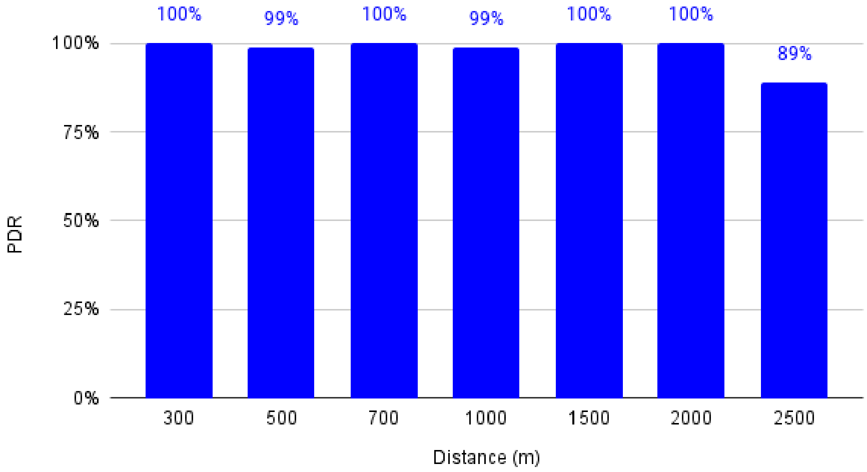

**Figure 16.** Positioning of nodes in the park test.

## 6. Conclusions

LoRa is a very powerful LPWAN that is becoming increasingly used. It offers a long transmission range and low energy consumption, making it perfect for sensor networks and the IoT world. While LoRa is generally used along with LoRaWAN, a robust protocol that can often enhance the physical layer, it requires connections to be only between a node and a gateway, heavily limiting the possible uses of the underlying technology. Mesh networks can provide a solution with a low installation time and help deliver messages, even if one of the middle-node suddenly becomes unavailable.

Combining LoRa and mesh networks in this project, we created a solution for exchanging data and commands that can be used for many purposes, including pervasive sensing supporting smart cities. In our solution, nodes can be easily added to the mesh network and connected to a sensor, the data gathered can be delivered to the destination through a multi-hop routing protocol, and the spreading factor can be adapted to reduce data loss while maintaining the throughput between the nodes as high as possible.

A future expansion of this work could focus on improving the routing algorithm by exploiting the RSSI to calculate the better path rather than choosing the one with the least number of hops. Some work should be put into developing an algorithm to normalize RSSI values obtained from different devices. The nodes currently do not use any form of encryption to secure messages, which may be helpful for certain applications. While small, battery-powered devices typically have limited computing capabilities, ways to secure the connection between nodes exist, like certificates. A future iteration could work on implementing this feature.

**Author Contributions:** Conceptualization, P.M., S.E.M., C.E.P. and P.P.; Methodology, P.M., S.E.M., C.E.P. and P.P.; Software, P.M., S.E.M., C.E.P. and P.P.; Validation, P.M., S.E.M., C.E.P. and P.P.; Formal analysis, P.M., S.E.M., C.E.P. and P.P.; Investigation, P.M., S.E.M., C.E.P. and P.P.; Writing—original draft, P.M., S.E.M., C.E.P. and P.P.; Writing—review & editing, P.M., S.E.M., C.E.P. and P.P.; Supervision, P.M. and C.E.P.; Funding acquisition, P.M. and C.E.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors of this paper acknowledge the support of MUR (Italian Ministry of University and Research) under the PON ex DM 1061 and the PNRR CN-MOST initiatives.

**Data Availability Statement:** Data presented in this paper are available upon request to the authors.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| IoT | Internet of Things |
| LPWAN | Low Power Wide Area Network |
| CSS | Chirp Spread Spectrum |
| SF | Spreading Factor |
| PDR | Packet Delivery Ratio |
| SNR | Signal-to-Noise Ratio |
| RSSI | Received Signal Strength Indicator |

## References

1. Bujari, A.; Furini, M.M.; Reoli, F.; Martoglia, R.; Montangero, M.; Ronzani, D. Standards, Security and Business Models: Key Challenges for the IoT Scenario. *Mob. Netw. Appl.* **2018**, *23*, 147–154. [CrossRef]
2. Rescio, G.; Manni, A.; Caroppo, A.; Carluccio, A.M.; Siciliano, P.; Leone, A. Multi-Sensor Platform for Predictive Air Quality Monitoring. *Sensors* **2023**, *23*, 5139. [CrossRef] [PubMed]
3. Bardi, S.; Palazzi, C.E. Smart Hydroponic Greenhouse: Internet of Things and Soilless Farming. In Proceedings of the 2022 ACM Conference on Information Technology for Social Good, Limassol, Cyprus, 7–9 September 2022; pp. 212–217. [CrossRef]
4. Ceccarini, C.; Mirri, S.; Salomoni, P.; Prandi, C. On exploiting Data Visualization and IoT for Increasing Sustainability and Safety in a Smart Campus. *Mob. Netw. Appl.* **2021**, *26*, 2066–2075. [CrossRef]
5. González-Zamar, M.D.; Abad-Segura, E.; Vázquez-Cano, E.; López-Meneses, E. IoT Technology Applications-Based Smart Cities: Research Analysis. *Electronics* **2020**, *9*, 1246. [CrossRef]
6. Bujari, A.; Gaggi, O.; Palazzi, C.E.; Ronzani, D. Would Current *Ad-Hoc* Routing Protocols be Adequate for the Internet of Vehicles? A Comparative Study. *IEEE Internet Things J.* **2018**, *5*, 3683–3691. [CrossRef]
7. Liu, Y.; Liu, L.; Liang, J.; Chai, J.; Lei, X.; Zhang, H. High-Performance Long Range-Based Medium Access Control Layer Protocol. *Electronics* **2020**, *9*, 1273. [CrossRef]
8. Yao, F.; Ding, Y.; Hong, S.; Yang, S.H. A Survey on Evolved LoRa-Based Communication Technologies for Emerging Internet of Things Applications. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 4–19. [CrossRef]
9. Cardenas, A.M.; Nakamura Pinto, M.K.; Pietrosemoli, E.; Zennaro, M.; Rainone, M.; Manzoni, P. A Low-Cost and Low-Power Messaging System Based on the LoRa Wireless Technology. *Mob. Netw. Appl.* **2020**, *25*, 961–968. [CrossRef]

10. Aldahdouh, K.A.; Darabkh, K.A.; Al-Sit, W. A Survey of 5G Emerging Wireless Technologies Featuring LoRaWAN, Sigfox, NB-IoT and LTE-M. In Proceedings of the International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 21–23 March 2019; pp. 561–566. [CrossRef]
11. Almuhaya, M.A.; Jabbar, W.A.; Sulaiman, N.; Abdulmalek, S. A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions. *Electronics* **2022**, *11*, 164. [CrossRef]
12. Kim, D.H.; Lee, E.K.; Kim, J. Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed. *Sustainability* **2019**, *11*, 1917. [CrossRef]
13. Lora Alliance Website on New Maximum Range Record. Available online: https://lora-alliance.org/lorawan-news/lorawanr-distance-world-record-broken-twice-766-km-476-miles-using-25mw-transmission/ (accessed on 27 June 2023).
14. LoRaWAN Specifications on LoRa Alliance Website. Available online: https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification (accessed on 27 June 2023).
15. Lee, H.C.; Ke, K.H. Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2177–2187. [CrossRef]
16. Lundell, D.; Hedberg, A.; Nyberg, C.; Fitzgerald, E. A Routing Protocol for LoRa Mesh Networks. In Proceedings of the 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Chania, Greece, 12–15 June 2018; pp. 14–19. [CrossRef]
17. Huh, H.; Kim, J.Y. LoRa-based Mesh Network for IoT Applications. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 524–527. [CrossRef]
18. Kirichek, R.; Vishnevsky, V.; Koucheryavy, A. Analytic Model of a Mesh Topology based on LoRa Technology. In Proceedings of the 2020 22nd International Conference on Advanced Communication Technology (ICACT), Phoenix Park, Republic of Korea, 16–19 February 2020; pp. 251–255. [CrossRef]
19. Cotrim, J.R.; Kleinschmidt, J.H. LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication. *Sensors* **2020**, *20*, 4273. [CrossRef] [PubMed]
20. Jiang, X.; Zhang, H.; Yi, E.A.B.; Raghunathan, N.; Mousoulis, C.; Chaterji, S.; Peroulis, D.; Shakouri, A.; Bagchi, S. Hybrid Low-Power Wide-Area Mesh Network for IoT Applications. *IEEE Internet Things J.* **2021**, *8*, 901–915. [CrossRef]
21. Berto, R.; Napoletano, P.; Savi, M. A LoRa-Based Mesh Network for Peer-to-Peer Long-Range Communication. *Sensors* **2021**, *21*, 4314. [CrossRef] [PubMed]
22. Hong, S.; Yao, F.; Ding, Y.; Yang, S.H. A Hierarchy-Based Energy-Efficient Routing Protocol for LoRa-Mesh Network. *IEEE Internet Things J.* **2022**, *9*, 22836–22849. [CrossRef]
23. Kim, S.; Lee, H.; Jeon, S. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. *Sensors* **2020**, *20*, 1008. [CrossRef] [PubMed]
24. Zhu, G.; Liao, C.H.; Suzuki, M.; Narusue, Y.; Morikawa, H. Evaluation of LoRa receiver performance under co-technology interference. In Proceedings of the IEEE Annual Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018. [CrossRef]
25. Croce, D.; Gucciardo, M.; Mangione, S.; Santaromita, G.; Tinnirello, I. Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance. *IEEE Commun. Lett.* **2018**, *22*, 796–799. [CrossRef]
26. Reynders, B.; Meert, W.; Pollin, S. Power and spreading factor control in low power wide area networks. In Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]
27. Nakamura, K.; Manzoni, P.; Zennaro, M.; Cano, J.C.; Calafate, C.T. LoRaCTP: A LoRa based Content Transfer Protocol for sustainable edge computing. In Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN), Tokyo, Japan, 17–19 December 2020; pp. 539–545. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.