

Article

Compliant and Seamless Hybrid (Star and Mesh) Network Topology Coexistence for LoRaWAN: A Proof of Concept

Laura García ^{1,2,*} , Carlos Cancimance ³, Rafael Asorey-Cacheda ^{1,2} , Claudia-Liliana Zúñiga-Cañón ³, Antonio-Javier Garcia-Sanchez ^{1,2}  and Joan Garcia-Haro ^{1,2} 

¹ Department of Information and Communications Technologies, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain; rafael.asorey@upct.es (R.A.-C.); antoniojavier.garcia@upct.es (A.-J.G.-S.); joang.haro@upct.es (J.G.-H.)

² People Oriented Smart Technologies Laboratory (POSTLab), European University of Technology, European Union

³ Research Group COMBA I+D, Universidad Santiago de Cali, Cali 760035, Colombia; carlos.cancimance00@usc.edu.co (C.C.); claudia.zuniga00@usc.edu.co (C.-L.Z.-C.)

* Correspondence: laura.garcia@upct.es

Abstract: Long-range wireless area networks (LoRaWAN) typically use a simple star topology. However, some nodes may experience connectivity issues with the gateway due to signal degradation or limited coverage, often resulting from challenging environments in sectors such as agriculture, industry, smart cities, smart grids, and healthcare, where LoRaWAN-based IoT solutions have expanded. The main contribution of this paper is the implementation of a hybrid topology for LoRaWAN networks that remains fully transparent to current spec LoRaWAN servers and IoT applications. It enables the coexistence of mesh (multi-hop) and star (single-hop) communication schemes, dynamically adapting a node's transmission mode based on physical link quality metrics. Additionally, the user interface allows for customizing network topology and parameters. Experimental proof-of-concept tests were conducted on a campus-wide testbed. Results showed that all devices successfully switched topology mode in 100% of the instances, enabling data transmission across all three scenarios under test. Network performance metrics were evaluated, with latencies ranging from 0.5 to 3.2 s for both single-hop and multi-hop transmissions. Additionally, improvements in RSSI and SNR were observed, validating the efficiency of the proposed solution. These results demonstrate the feasibility and effectiveness of our approach in extending network connectivity to areas beyond the gateway's coverage.

Keywords: hybrid topologies; LoRa; LoRaWAN; mesh



Academic Editor: Alessandro Lo Schiavo

Received: 26 February 2025

Revised: 15 March 2025

Accepted: 17 March 2025

Published: 22 March 2025

Citation: García, L.; Cancimance, C.; Asorey-Cacheda, R.; Zúñiga-Cañón, C.-L.; Garcia-Sanchez, A.-J.; Garcia-Haro, J. Compliant and Seamless Hybrid (Star and Mesh) Network Topology Coexistence for LoRaWAN: A Proof of Concept. *Appl. Sci.* **2025**, *15*, 3487. <https://doi.org/10.3390/app15073487>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid popularization of the Internet of Things (IoT) promoted the development of new wireless communication technologies with functionalities that addressed the niche left by other widespread wireless technologies like WiFi. LoRaWAN is one of these recent wireless specifications/protocols, having gained substantial recognition for the implementation of IoT applications over LPWAN (low power wide area network) technology. This was propitiated by the support of the LoRa Alliance [1], an organization tasked with providing the technical specifications, and TTN [2], one of the most accepted providers of LoRaWAN services available to both companies and the general public. The extensive advantages of LoRaWAN continue to attract thousands of users. Its long coverage ranges

allow for reaching communication distances of 5 km in urban areas and up to 20 km in rural areas [3]. The small number of control messages in the MAC layer results in reduced energy consumption and delay [4]. Furthermore, the literature indicates that data rates range from 250 bit/s to 21.9 kbit/s [5], also establishing a maximum data rate of 50 kbit/s when using FSK modulation [3]. These characteristics ensure good performance for telemetry IoT applications that do not require frequent real-time transmissions. The support of bidirectional communications opens the door to new functionalities for remote and intelligent device control and configuration [6]. The excellent performance of LoRaWAN networks not only in outdoor environments but also in indoor facilities enables its adoption as a robust wireless communication technology in factories and other industrial buildings [7]. Moreover, as depicted in the physical layer specifications, LoRa uses the ISM (industrial, scientific, and medical) band which, along with the low cost of the available certified devices, facilitates the easy deployment of LoRaWAN networks. All these advantages have led to the development of LoRaWAN-based IoT added-value solutions in multiple sectors, including smart cities [8], smart grids [9], agriculture [10], localization [11], industry [12], and medical applications [13], among many others [14]. Nevertheless, LoRaWAN as it is currently specified also has some drawbacks. One of the main disadvantages is network size limitations. This is due to the duty cycle that, in the case of Europe, is established at 1% [15]. The characteristics of conventional star topology may lead to difficulties in providing service to inaccessible areas and zones without enough power supply infrastructure. In addition, battery-operated end devices with low power availability may need to reduce their transmission power, which could result in a lack of connectivity with the gateway.

LoRaWAN specifications contemplate the star topology as the main form of network configuration for LoRaWAN end devices, but it can benefit from the implementation of other forms of network topologies, such as tree, cluster, or mesh networks. This would create the possibility of reaching more distant areas while still routing packets to the gateway, addressing, as a result, some of LoRaWAN's current issues. Deployments in cities or industrial environments can find the effective coverage to be notably reduced due to the presence of multiple obstacles [16]. The use of multi-hop network topologies is a solution that could substantially improve the performance and applicability of LoRaWAN networks in this type of scenario. Other possible benefits of using multi-hop topologies include (1) improved overall communications as a result of adapting the topology to the lossy environment caused by obstacles or interference [17], (2) decreased network deployment costs since the required number of gateways is reduced, (3) less energy consumption as gateways need more energy to operate than end devices [18], and (4) a smaller network carbon footprint due to fewer required gateways [19].

The improvement that multi-hop topologies could bring to LoRa networks has been addressed in several works that evaluate these solutions in simulated and real environments [20–23]. However, those approaches are not compatible with the LoRaWAN specifications, which clearly jeopardizes extending the functionality of LoRa networks. The LoRa Alliance has also attempted to provide a one-hop solution through a range extender depicted as a relay node that forwards uplink or downlink frames to the next hop [24]. However, the operation of the relay device is exclusively dedicated to that purpose, disallowing leveraging it for other tasks. This means that the network needs to be planned in advance or modified with the addition of relay nodes deployed by an operator, according to the identified obstacles. The result is a static network that cannot be rearranged to accommodate new needs unless an operator decides to explicitly address that issue. Therefore, there is a gap in the literature regarding flexible LoRaWAN networks with adaptive topology management enabling automatic switching between star and mesh topologies, as this is not addressed in any of the above-mentioned works. In this context, this paper presents a

straightforward adaptive topology solution, fully compatible with the LoRaWAN standard, that enables users to change from star to mesh LoRaWAN network topologies according to physical link performance metrics. Our solution can be easily adopted by conventional networks and applications on top. The contributions of this work are manifold:

- The proposal of a straightforward approach for LoRaWAN, enabling the coexistence of star and mesh network topologies (hybrid) completely compliant with the LoRaWAN standard and seamless to IoT applications.
- The provision of a topology manager to assess network link quality and the decision to change the role of end devices to single-hop or multiple-hop transmission modes.
- As a consequence of the previous point, an end device code integrates star, mesh, and hybrid topology states to perform the changes in the transmission modes.
- The addition of a control user interface to customize the parameters used by the topology manager and allow the user to change transmission mode at will.

This proposal has been implemented for LoRaWAN end devices from different vendors. A battery of tests has been conducted to validate the proposed enhanced LoRaWAN functionality in different case scenarios in a real campus environment.

The rest of the paper is organized as follows: Section 2 introduces the related work. The methodology is described in Section 3. The evaluation scenarios are detailed in Section 4. Section 5 discusses the results obtained. Finally, the conclusion and future work are presented in Section 6.

2. Related Work

Integrating mesh topologies into LoRa wireless technology is an active area of research that both academia and the LoRa Alliance are pursuing. A recent LoRaWAN specification established the guidelines for incorporating relay devices into typical LoRaWAN star topology networks to extend coverage, but it is limited to serving 16 devices per relay node, and the sole role of the relay node is to forward frames, which does not provide enough flexibility for certain type of scenarios [24]. Conversely, most mesh proposals in the literature developed for LoRa do not comply with the LoRaWAN approved specifications. Huang-Chen Lee et al. presented a mesh proposal for LoRa networks in [25]. The authors of this work carried out tests on a university campus to provide LoRa coverage in an area of 480,000 m² with 19 devices. These tests included in-building and outdoor deployments. The results showed a packet delivery ratio (PDR) of 58.7% for the traditional star topology and 88.49% for the proposed mesh system. Moreover, they stated that signal-to-noise ratio (SNR) values higher than −5 dB and received signal strength indicator (RSSI) values higher than −110 dBm are necessary to achieve PDR results above 90%. Applying mesh LoRa solutions to UAVs (unmanned aerial vehicles) has also been considered, specifically for the case of data communication in environmental monitoring systems. The medium access control (MAC) protocol proposed by Miaoxin Pan et al. was designed based on a slotted ALOHA [26]. Their proposed mesh system considered the deployment of LoRa relay devices that enabled communication between sensor nodes and the gateway. Tests were performed both in laboratory and outdoor settings. The results returned delays lower than 6 s for 1-hop transmissions, line-of-sight distances of 10 km, and data rates of 293 bit/s.

Multi-hop solutions have been addressed in other works as a first step toward mesh networks. The authors in [20] studied multi-hop LoRa network topologies. Metrics such as coverage, packet reception ratio (PRR), and energy efficiency were evaluated for different configurations of transmission power or spreading factor (SF). Tests were conducted using outdoor end devices. The results revealed that multi-hop topologies outperformed star topologies with 50% energy savings and 35% increased coverage for a 90% PRR. Among multi-hop proposals, introducing devices to relay information to the next node is a popular

approach. Christian Ebi et al. presented a LoRa-based meshed multi-hop network intended for communications in underground environments such as sewers [27]. The proposed protocol included a new LoRa frame format for synchronous connections and the timeslot allocation method. The proposed system comprised sensor nodes, repeater nodes, and gateways. Tests were performed in two cities in Switzerland using 17 devices. The results showed better PDR performance for the mesh deployment than standard LoRaWAN. Jaime Lloret et al. developed a cluster-based protocol for LoRa multi-hop communications [28]. It included cluster-head nodes able to operate with both LoRa and WiFi, as well as aggregator nodes operating as relay nodes. The message format and message exchange for all the phases of the protocol were provided as well. Proof of concept tests were performed to validate the proposal. The results obtained on bandwidth consumption indicated that the proposed protocol complied with the most restrictive LoRa configurations. Furthermore, higher PDR values were attained for packet transmission delays of 500 ms.

With the advance in mesh solutions for LoRa, proposals and implementations of routing protocols for mesh networks began to arise. The routing protocol proposed by Shengguang Hong et al. was intended for LoRa mesh hierarchical networks with a focus on energy efficiency [29]. This protocol used contention-free concurrent transmission and minimized packet collisions. Simulations and experimental tests were undertaken as proof of concept. The results achieved a packet delivery rate of 89.8% for the normal operation phase and 95.7% for the network formation phase. The authors in [30] developed a library for the implementation of a LoRa mesh network protocol. The protocol maintains a routing table in every node that is updated through the routing messages. Tests were carried out using the proposed LoRa mesh library with ESP32 microcontroller boards. Three different topologies were evaluated to obtain PDR and latencies and control overhead. The nodes did not need to connect to a gateway to communicate among themselves. The validity of the proposal was demonstrated.

There have also been attempts to provide mesh solutions compatible with LoRaWAN specifications. Specifically, the authors in [31] presented a hybrid infrastructure that joined LoRaWAN and LoRaMesh to provide side management as a service. While the LoRaWAN and LoRaMesh networks were able to reach a coverage of 2.35 km and 600 m, respectively, the proposed hybrid solution attained coverage of 3 km for one-hop transmissions. This solution was also able to provide service to up to 1023 devices. Furthermore, Emiliano Sisinni et al. implemented a relay node to enable multi-hop LoRaWAN transmission [32]. This device, called e-node, operates transparently and is compatible with the LoRaWAN standard. The proof of concept prototype has been validated for industrial applications at the highest data rates available in the LoRaWAN specification. The results of the tests under consideration included link quality metrics (SNR and RSSI) and delay measurements.

A comparison of the existing proposals on hybrid LoRaWAN implementations has been provided in Table 1. The solution presented in this work differs from the former approaches in that it is fully compliant with LoRaWAN and also allows for coexistence and simultaneous operation in star and mesh topologies with flexible topology modifications. Topologies can be autonomously changed with a topology manager that evaluates the state of the network through the physical link quality metrics included in LoRaWAN frames, allowing for a dynamic network configuration.

Table 1. Comparison of LoRaWAN hybrid solutions.

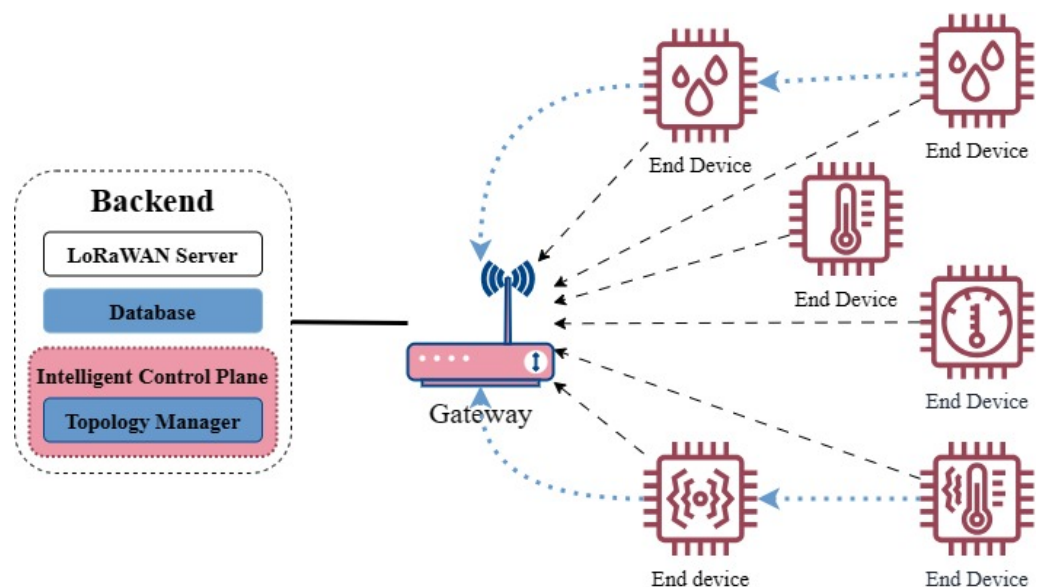
Reference	Protocol	Topology	Topology Coexistence	Dynamic Network Configuration
[31]	LoRaWAN/LoRaMesh	Star and mesh	Yes	No
[33]	LoRaWAN	Multi-hop (Relay)	No	No
[23]	LoRaWAN/LoRaMesh	Star and mesh	No (switches from only star to only mesh)	Yes (complete network)
[34]	LoRaWAN/LoRaMesh	Star and mesh	Yes	No
Our proposal	LoRaWAN	Star and mesh	Yes	Yes (device per device)

3. Hybrid Topology Architecture

This section formulates the mechanism for hybrid (star and mesh) topology coexistence in LoRaWAN networks, providing first the architecture of a generic scenario where hybrid topology configurations are executed and, later, the technical details of the implementation.

3.1. Procedure Description

LoRaWAN specifications describe a star-shaped topology as the design to be used for the wireless end of the network. That is, using point-to-point connections between end devices and gateways without any routing or alternative processes for enabling communication. In a typical LoRaWAN network, the gateway bridges the wireless end of the network, which uses the LoRa physical layer specifications, with the wired end, which utilizes the MQTT protocol to exchange uplink and downlink information with the LoRaWAN server tasked with managing the network. However, LoRaWAN specifications do not include other topology functionalities that would help address some of LoRaWAN's limitations, specifically, those related to extending the coverage of the network. Our proposal provides a solution allowing for the coexistence of star and mesh adaptable network topologies in LoRaWAN, as illustrated in Figure 1.

**Figure 1.** Architecture of a LoRaWAN network with the proposed hybrid topology procedure.

LoRaWAN networks require end devices to be activated prior to network establishment. The proposed adaptable hybrid network topology coexistence functionality sets the elements of the network to use a star topology by default with ABP (activation by personalization) as the activation method. Using the manual configuration of cipher keys

from ABP ensures the correct handling of messages even when the operation mode is changed to mesh. The end devices send data to the gateway, which is connected to the Internet and has been previously configured with the connection establishment parameters associated with the LoRaWAN server. Data is then formatted in JSON (JavaScript object notation) and transmitted to the Server using the MQTT publish-subscribe protocol. As the proposed mechanism allows multi-hop communication between LoRaWAN end devices, the MQTT packets that one end device delivers to the MQTT broker may originate from other end devices. Thus, the MQTT packets are first published in an intermediate topic at the MQTT broker. Then, data packets are sorted and published in their respective default Chirpstack topics according to their end device of origin. Data received at the server are then stored for later analysis and visualization. This new hybrid topology functionality for LoRaWAN has been designed to be fully compliant with the approved specifications defining the LoRaWAN protocol. Therefore, it can be deployed in the existing infrastructure in a simple, seamless, fast, and effective way.

The hybrid topology procedure is agnostic to the LoRaWAN server and introduces a new element called the topology manager. It provides the control plane with intelligent capabilities, enabling the topology change of the wireless end of the network while being completely transparent to the Chirpstack LoRaWAN server. SNR and RSSI network link quality data are provided in LoRaWAN messages. These data are available at the LoRaWAN server, but are extracted and stored in a separate database for the topology manager to analyze them. The topology manager uses this information to assess performance and reaches a decision about the need to change the topology. Then, it sends topology change messages to be queued at the LoRaWAN server as downlink messages addressed to the end devices, indicating that their transmission mode (single-hop or multiple-hop, that is, star or mesh topology, respectively) should change. The end devices receive the messages and change their transmission mode accordingly. As a result, the topology of the network is modified and the end devices with worse performance deliver their data to other end devices with better performance for their re-transmission (multi-hop). Therefore, as shown in Table 2, end devices using a single-hop transmission mode would form a star topology. Conversely, end devices in multi-hop transmission mode could be part of a mesh topology, sending messages to other end devices, or a hybrid (star and mesh) topology, re-transmitting messages from other end devices to the gateway. The packet transmission scheme describing this process is exemplified in Figure 2. The packet format of LoRaWAN is maintained; therefore, no extension of packet specifications is required. The deployment of the proposed adaptable hybrid topology framework for LoRaWAN networks is server and application agnostic and, as a result, fully compatible with any LoRaWAN implementation.

Table 2. Relationship between transmission modes and topology modes.

Transmission Mode	Topology Mode
Single-hop	Star
Multi-hop	Hybrid Mesh

End devices can be configured for three types of topologies, as shown in Figure 3. Star topology is the standard/conventional LoRaWAN connection, where the end device is activated manually, and the topology of the network is in the shape of a star or star-of-stars. This state can be reached when the device is restarted (event 0) or a message with the request to change to star topology is received (event 3). Mesh topology is configured in the end devices that transmit their data to other end devices for their further re-transmission.

This state is reached when a message with the petition to change to mesh topology is received (event 1). Lastly, hybrid topology is configured in the intermediate end devices that receive data from other end devices and re-transmit them to the gateway. This state is reached when a request to change to the hybrid topology is received (event 2).

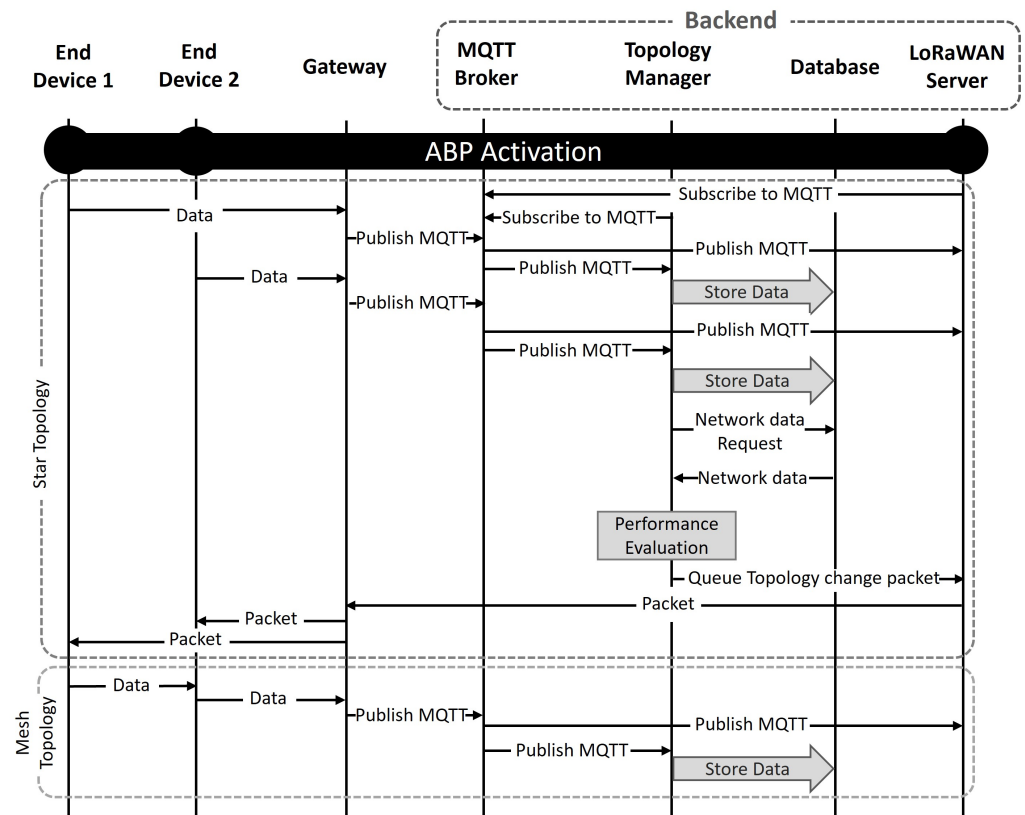


Figure 2. Message exchange among the elements of the architecture.

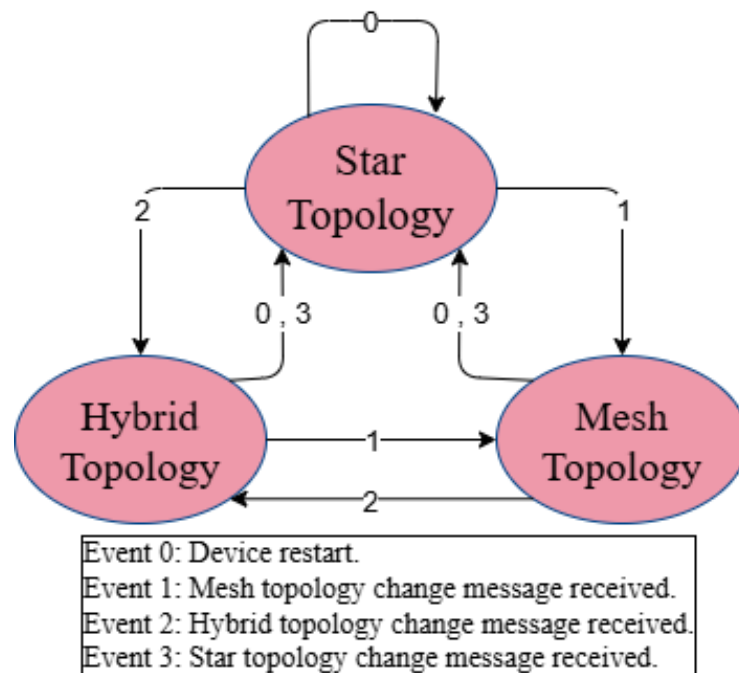


Figure 3. Diagram of the end device states.

3.2. Implementation

The technical details of the implementation of our proposed adaptable hybrid network topology for LoRaWAN, as depicted in the schema shown in Figure 4, are outlined in this subsection. All of the elements are explained at length below.

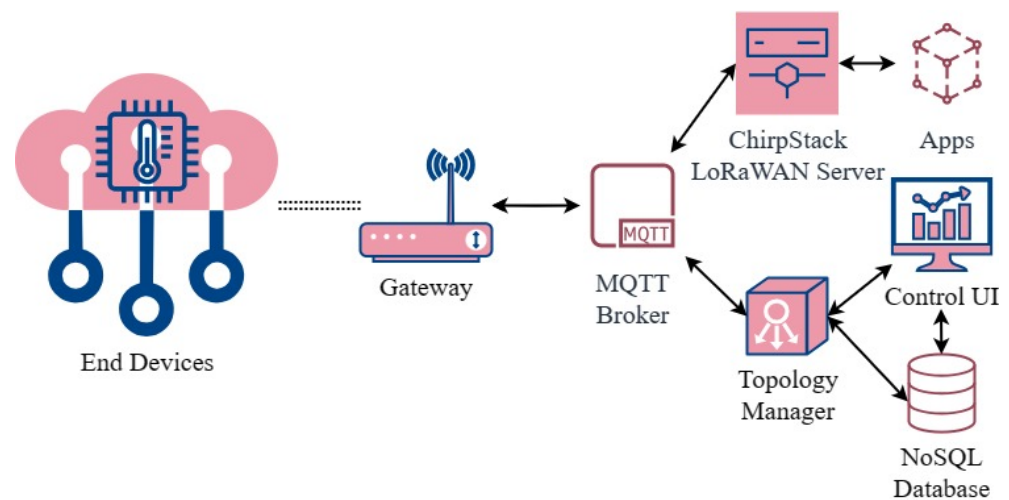


Figure 4. Elements comprising the implementation of hybrid network topology for LoRaWAN.

3.2.1. End Devices

The end devices used to implement the network are Pycom Lopy4 [30] development modules (see Figure 5). These modules support WiFi, Bluetooth, LoRa, and Sigfox, but this implementation of the end device only uses the LoRa physical layer. Each device is equipped with a Pycom Makr expansion board and a 915 MHz antenna attached with a U.FL connector. Visual studio code [35] was the integrated development environment (IDE) selected to upload the code into the Pycom devices through the serial port. This IDE was equipped with the PyMakr plugin [36] that enabled compatibility with the development modules. The programming was performed using the MycroPython (<https://micropython.org/> accessed on 24 February 2025.) programming language. This language has been optimized to operate with microcontrollers and limited or restricted environments. The programming of the end devices works with the topology manager to change the topology. This programming allows the data generated by the sensors integrated into the end device to be transmitted and listens to the requests forwarded from the LoRaWAN server, which include the transmission mode change petitions generated by the topology manager. The “LoRa” class from the “network” software (version 1) module provided for Pycom devices was used to construct and send the LoRaWAN messages. The process of identifying topology change petitions, the implementation of each operation state (star, hybrid, or mesh), and the functionality allowing for the topology changes were developed as part of the proposed solution. All the topology modes are implemented into the end devices, and therefore, the topology change petitions only indicate the mode that needs to be executed. This implementation is completely LoRaWAN server agnostic, as all the messages forwarded from the end devices follow LoRaWAN specifications.

When the end device is in star topology mode, its operation is performed as the LoRaWAN specifications dictate for class A devices. When a message is received, it is analyzed to determine if it is a topology change petition. If it is, the end device executes the part of the code corresponding to the operation of the new topology mode. All transmission modes can receive and process topology change petitions, and appropriately execute the required changes. Therefore, if the end device changes into the hybrid topology mode, it receives the messages from other end devices and forwards them to the gateway, and if

it changes into mesh mode, it forwards its messages to a neighboring end device. All the end devices in hybrid mode within the coverage area of an end device in mesh mode would receive the messages. However, duplicated messages at the MQTT broker are only forwarded once to the LoRaWAN server.



Figure 5. End devices used for the hybrid LoRaWAN topology procedure.

3.2.2. Gateway

The LoRaWAN gateway used for the implementation is Lorix One from Wifx [37]. This gateway is compact and includes an ARM Cortex-A5 600 MHz CPU, a 125 MB RAM, and an internal memory of 512 MB. It is also equipped with its own Operating System (OS), called Lorix OS, which is fully compatible with Chirpstack servers and other popular LoRaWAN server implementations. The gateway predominantly maintains its standard configurations. The only exception pertains to the MQTT broker.

In our proposal, the nodes that are configured in mesh topology send their data to the hybrid nodes. The data is combined into one message, which the hybrid node sends to the Gateway. Gateways are configured to post the received data in a specific MQTT topic. The default topic for Chirpstack LoRaWAN implementations is `us915_0/gateway/{ .GatewayID }/event/{ .EventType }`, where the `us915_0` is changed according to the employed band. To extract the information from the combined message so that it can be re-directed to its respective topics, an intermediate MQTT topic was created. This was done by accessing the configuration file of the gateway and changing the default topic to `csc_usc/gateway/{ .GatewayID }/event/{ .EventType }`.

3.2.3. LoRaWAN Server

LoRaWAN specifications from the LoRa Alliance consider the implementation of a Join Server, a Network Server, and an Application Server. These three servers can be deployed in the same physical device or in different locations with the option of using a Join Server from one service provider, a Network Server from another provider, and an Application Server from a third service provider. Installing an MQTT broker is also necessary as MQTT is the protocol used for communication among these servers and communication with the gateway.

The LoRaWAN server has been implemented using the open-source software from Chirpstack [38]. This software is widely used for LoRaWAN network deployments. However, it does not completely follow the specifications provided by the LoRa Alliance. In Chirpstack version 3, the network and application servers need to be installed to implement the backend of the LoRaWAN network. Therefore, the join server from the specification is not separate software and the join process is managed by the other servers. Chirpstack version 4 agglutinates the network and application servers into a single server for user convenience. This may lead to confusion due to discrepancies between the software implementation and the explanation provided in the specifications, but Chirpstack supports the functionalities of both LoRaWAN v1.0 and LoRaWAN v1.1 specifications.

To implement the proposed solution, Chirpstack v4 was employed. Thus, the network and application servers are combined into one software module that performs all the tasks stated in the specifications. We refer to this server as the LoRaWAN server for simplification. Chirpstack includes configuration files that have been modified according to the documentation to determine if the data are forwarded using plain JSON or protobuf or to determine the frequency band on which the LoRaWAN network is operating. These configurations are necessary for any LoRaWAN network using Chirpstack. In this implementation, protobuf and the US915 band are employed, with the ADR turned off and a SF of 7. The server is physically located at the Technical University of Cartagena (Spain), and it has a public IP address to enable access from Santiago de Cali University (Colombia). Our solution for adaptable hybrid network topologies is agnostic to the LoRaWAN server, and thus, no modifications are performed to the Chirpstack software [38].

3.2.4. MQTT Broker

The MQTT broker that is used to enable communication between the gateway and the LoRaWAN server was implemented using the open-source software Mosquitto version 2.0.15 from Eclipse [39]. Mosquitto does not require any additional configuration apart from the default installation. However, if the MQTT broker is going to be open to the Internet, a password using the *mosquitto_passwd* utility should be added. The implementation of the MQTT broker is located in the same computer as the LoRaWAN server. Thus, it is accessible from Colombia through the same public IP address.

3.2.5. Database Module

The proposed solution of hybrid network topology coexistence for LoRaWAN requires the installation of a database paired with its control module to allow communication with the other elements of the proposal, referred to as the database module for simplicity. As the data format in the LoRaWAN network is JSON, the popular open source NoSQL MongoDB database [40] was chosen for the implementation. This database includes free and paid versions of the software that vary according to features and storage capacity. The implementation of the database module only requires basic functionalities. Therefore, the free option meets all our requirements.

The database is used by the topology manager module. This module creates an MQTT client that connects to the MQTT broker and subscribes to the MQTT topics where the gateway posts all the messages it sends to the LoRaWAN server. In our proposed solution, this corresponds to the intermediate topic. Once subscribed, the message is decoded and the resulting JSON message is parsed. The link quality metrics stored in the message, RSSI and SNR, are extracted from the message and stored in the database. The type of event and end device identification are also stored with the link quality metrics. This information is obtained from the decoded message as well. Lastly, the messages are posted at the default MQTT topic, that is, *us915_0/gateway/{ .GatewayID }/event/{ .EventType }*. Data stored in

the database can be accessed later and requested by the topology manager to decide on the topology update.

In our implementation, the MongoDB database was installed in a local server at Santiago de Cali University. However, it could be configured to any machine as long as the connection string to the database is updated accordingly.

3.2.6. Topology Manager

This component facilitates the intelligent management of a LoRaWAN network topology updated by knowing the link quality of data transmission between end devices and the gateway at any time. This version of the topology manager is a preliminary proposal toward further development, including the use of machine learning techniques for improved/smart decision-making. The topology manager module was programmed in Python 3. Algorithm 1 represents the simplified pseudo-code of the topology decision-making procedure. First, the information from the end device, including its current topology mode, is retrieved from the database. Then, the average of the last five values from the transmission link quality parameters (SNR and RSSI) that have been stored for each end device is calculated. This allows the user to assess which end devices have good connectivity and which have more difficulties. Consequently, these data can be used to keep track of the evolution of the network link parameters and discern the moment when certain end devices would benefit from changing their transmission mode to send their data through a neighbor enabled with multi-hop transmissions within their coverage area. If there were packet loss cases, the algorithm would not update its running averages for RSSI and SNR, as it relies exclusively on received messages. Furthermore, to address potential concerns about missing values, the system compensates for connectivity issues by dynamically adjusting the network topology. This mechanism ensures network reliability despite temporal variations in link quality metrics.

Algorithm 1 Adaptive Topology Adjustment

```

1: Initialize:
2: average_rssi  $\leftarrow$  0
3: average_snr  $\leftarrow$  0
4: message_counter  $\leftarrow$  0
5: current_mode  $\leftarrow$  GetInitialMode()
6: while true do
7:   message  $\leftarrow$  ReceiveMessage()
8:   rssi, snr  $\leftarrow$  GetRSSIandSNR(message)
9:   average_rssi  $\leftarrow$  (average_rssi  $\times$  message_counter + rssi) / (message_counter + 1)
10:  average_snr  $\leftarrow$  (average_snr  $\times$  message_counter + snr) / (message_counter + 1)
11:  message_counter  $\leftarrow$  message_counter + 1
12:  if message_counter  $\geq$  5 then
13:    average_rssi  $\leftarrow$  average_rssi / 5
14:    average_snr  $\leftarrow$  average_snr / 5
15:    if average_rssi > rssi_threshold and average_snr > snr_threshold then
16:      recommended_mode  $\leftarrow$  GetRecommendedMode()
17:      if current_mode  $\neq$  recommended_mode then
18:        SendTopologyChangeMessage()
19:        current_mode  $\leftarrow$  recommended_mode
20:      end if
21:    end if
22:    message_counter  $\leftarrow$  0
23:    average_rssi  $\leftarrow$  0
24:    average_snr  $\leftarrow$  0
25:  end if
26: end while

```

Users can establish the threshold values for the SNR and RSSI to decide if an end device should change its topology/transmission mode. The thresholds are maximum RSSI, minimum RSSI, and maximum SNR. The maximum RSSI and SNR thresholds, if exceeded, are used to determine if the end device needs to change its topology mode. The minimum RSSI is established to assess when an end device should change back into star topology mode. The frequency at which this evaluation process is performed is also adjustable by the user. The process of changing topologies is carried out by sending messages to each end device with the new transmission mode (single-hop or multi-hop) they need to adopt so as to update the network topology. The end devices with worse network quality metrics should change into mesh mode, and the end devices with better network quality metrics should change into hybrid mode. As a result, the messages with a request for a transmission mode change can be forwarded to two or more end devices in the network.

When changing topologies, the first end devices to perform the change are those changing into hybrid (star and mesh) mode, and the petition for a topology change directed to the end devices that need to change into mesh mode is put on hold. This way, packet loss is avoided as the hybrid end device needs to be operative before the end device changing into mesh mode performs its topology change. The topology manager will not conduct any other petition for topology changes until the previous requests are completed since the effects of the topology change cannot yet be assessed. The message for the topology change includes the end device identifier and the operating mode to which the end device needs to change. Once the petition is formed, it is added to the downlink queue at the LoRaWAN server. This information is appended to the data field in the LoRaWAN message, and thus, no change is performed in the operation mechanism of the LoRaWAN protocol. To send the topology change petition, four consecutive messages are transmitted in time intervals of 5 s so as to ensure the reception of the message. Upon message reception, all end devices read the identifier to assess if they are the recipient. Messages directed to other end devices will be discarded.

For an end device to change back into star topology mode, the topology manager sends a topology change message when the minimum RSSI threshold is exceeded. This downlink message must be received first by the end device in hybrid topology mode. Then, it will re-transmit the message to the end device in mesh mode. The topology manager also allows manual topology administration. Therefore, if necessary, a topology mode can be selected for a specific end device, and the topology manager will send the topology change petitions accordingly.

The topology manager was implemented in a local Ubuntu server at Santiago de Cali University. This implementation does not imply any changes to the software of the remaining elements of our implementations. However, it needs to be configured with the required information regarding the MQTT broker that manages the topics where the information of the end devices is published to extract the link quality data, the application ID at the Chirpstack LoRaWAN server, and the topic where the LoRaWAN server is subscribed for data.

Changing topologies in LoRaWAN networks can extend the coverage of the network and improve transmission link quality for end devices that are affected by temporary or permanent interference or obstacles. However, greater benefits can be achieved with future topology manager implementation strategies, such as reducing energy consumption, minimizing the carbon footprint of the network deployment, and even creating a network that is “alive” or “sensitive” and therefore able to respond to the dynamics of the environment.

3.2.7. Control User Interface

For a clear understanding of the operation of our proposed solution for star and mesh adaptable topology coexistence in LoRaWAN networks, a control user interface has been developed. The UI allows users to visualize the current topology mode of all the end devices in the network. The topology mode can also be manually selected by the user. The UI permits users to see the RSSI, SNR, delay, last parameter update, and other parameter data history, as well as queuing custom downlink messages and configuring some of the parameters and thresholds of the automatic mode enabled by the topology manager. This UI was implemented using Apache2 in an instance of Ubuntu server. The UI connects to the MongoDB database to consult network parameters, which requires the connection driver to be configured with MongoDB to enable data petitions.

4. Experiment Design

In this section, the experimental testbed and the evaluation scenarios for the proposed hybrid topology procedure for LoRaWAN are defined.

After the implementation of the proposed solution for network coexistence in LoRaWAN, a set of experimental tests were performed on the campus of Santiago de Cali University (Colombia). As the devices would be susceptible to tampering if placed unattended in different parts of the campus, the most viable option for the feasibility tests was to locate all the end devices inside the laboratory. The antenna was located on the roof of one of the university buildings at a distance of 90 m in a straight line from the location of the end devices.

The testbed consists of 6 end devices initially connected point-to-point, that is, in one-hop to one gateway, as presented in Figure 6. This initial state of the implementation resembles the conventional LoRaWAN star topology. The correct operation of the evaluation testbed was assessed before making topology changes. These end devices transmit data at different times. The LoRaWAN messages carrying these data include the link quality metrics from end device to gateway, specifically, the SNR and RSSI values. This information is extracted from the LoRaWAN message and stored in the database for later analysis and visualization. Another metric that is stored at the database is latency. This metric is obtained by subtracting the time of message transmission from the time of message arrival at the specific topic for the end device in the MQTT broker as automatically configured by the Chirpstack LoRaWAN server. Lastly, the information on the topology of the end device is obtained from the topology manager through the control UI, where the historical data from all the end devices can be accessed.

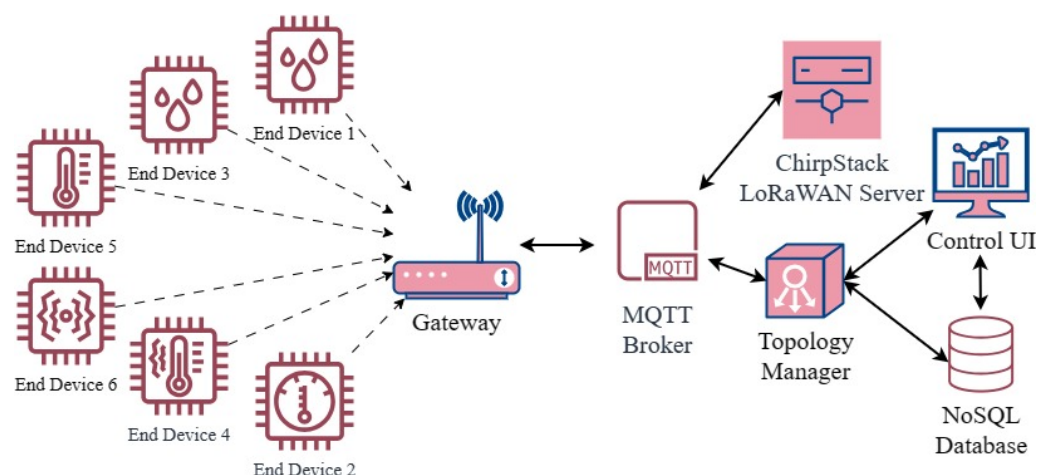


Figure 6. Graphical representation of the initial evaluation scenario.

To assess the coexistence between star and mesh topologies, three scenarios were considered for evaluation in which the communication of different end devices is changed to follow a mesh topology as represented in Figure 7. The aim of these scenarios is to demonstrate the feasibility of different topology combinations and the performance of the network according to the metrics previously described. Therefore, the topology changes in these tests were performed manually to force the desired scenario. Scenario 1 begins with the initial state of the testbed. In this scenario, two end devices send their data through a different hybrid device. This results in two devices operating in mesh topology mode and two devices operating in a hybrid state. This scenario was designed to assess the feasibility and performance of the solution with multiple hybrid nodes.

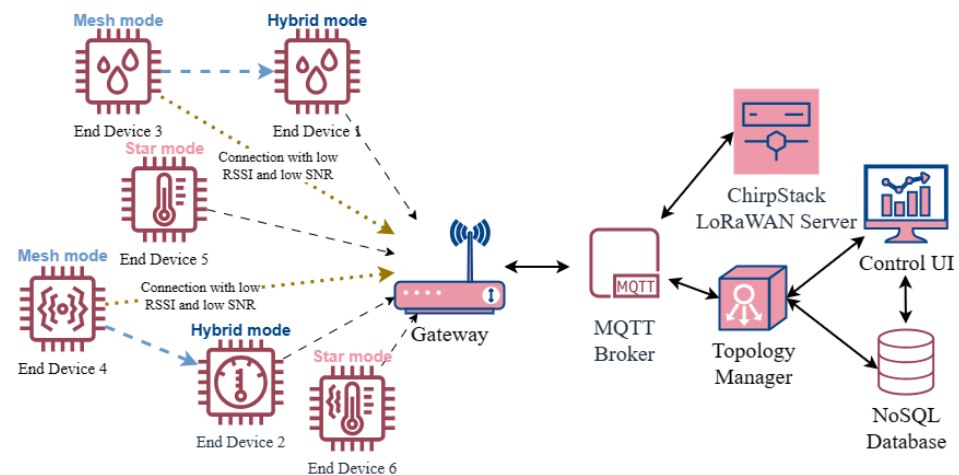


Figure 7. Graphical representation of coexisting topologies in the evaluation testbed.

Scenario 2 also starts in the initial state with all the end devices operating as star topology. Then, two of the six end devices change to mesh topology and send data through the same end device that has changed to operate as a hybrid node. That is, it receives the data from mesh devices and forwards them to the gateway. This scenario aims to verify that hybrid end devices are able to manage several mesh devices.

Lastly, scenario 3 initiates with the testbed configured in star topology. Then, a device changes into mesh node and begins transmitting through one hybrid node. After some time, the device transmits through a different hybrid node. This scenario was developed to gauge the possibility of servicing moving devices or reconfiguring the network dynamically if necessary. In this way, a mesh end device can transmit data through the hybrid device accessible at the moment.

To change the network topology from star to mesh/hybrid, the topology manager sends a message to the devices that need to change their topology. To do so, the message is added to the downlink queue for the intended device. As the mesh devices are enabled to deliver their data to other end devices, the intermediate devices located between the mesh device and the gateway also receive a topology change message from the topology manager to switch their operating mode to hybrid mode. Likewise, if a device switches from mesh mode to star mode, then the hybrid node in the intermediate hop will re-transmit the topology change message to the destination device. The process of receiving the topology change message, decoding it, interpreting it, and changing the operating mode is performed by using the software that was programmed and uploaded into all the end devices, as previously described in the implementation. The data frame forwarded to the end devices in hybrid mode includes the MAC address of the mesh devices that originate the corresponding data frame to identify the re-transmitted data and publish the information to the MQTT topic of the corresponding device. This means that the messages

from the mesh devices need to be published in the topic the Chirpstack LoRaWAN server assigned to them when the end devices were activated through ABP and not as if the message originated at the hybrid device. This process is performed seamlessly so that the server operates as if it were in a typical star LoRaWAN network.

5. Results

This section presents and discusses the performance evaluation results obtained from the experimental proof of concept set for the three aforementioned scenarios.

5.1. Scenario 1

The latency results obtained for the transmissions between the end devices and the final topic in the MQTT broker for all the devices that performed a topology change in scenario 1 are displayed in Figure 8. With this metric, it is possible to discern if the delays that may occur by performing multiple hops negatively affect the overall operation of the LoRaWAN network and its applications. The evolution of the changes in topology and transmission mode for each of the devices in this evaluation scenario is shown in Figure 9, where end devices 4 and 5 change into mesh mode, and end devices 2 and 6 change into hybrid mode. Devices 1 and 3 remain in star mode for the duration of the complete test. This setup was performed manually through the control UI to generate the desired scenario. As can be seen from the results, the latency of end devices 4 and 5 increases when they change from star to mesh mode because an additional hop to the route of the data frame is added. Furthermore, there is an increase in latency variations for both devices, with oscillating values that remain above 0.5 s and below 2.5 s. Regarding the devices that changed into hybrid mode, the latency of device 2 increased and the latency of device 6 decreased compared to their starting point, although both remained within the range of 0.725 and 2.362 s. Although the messages from the end devices in mesh mode undergo one more hop to reach the gateway, their latency stays in the range obtained from the end devices in hybrid mode. The latency of devices 1 and 3, which continued in star mode, stayed significantly more stable compared to the devices in mesh or hybrid mode. It is also important to note that latency increases with elapsed time for all devices. This phenomenon can be attributed to network congestion, which accumulates over time. Although star devices typically experience lower delays, their performance can be impacted under high network load. Additionally, we observed that some hybrid and mesh-mode devices experienced longer queuing times due to topology changes. External interference or environmental conditions may have also played a role in latency variations, further impacting transmission reliability across all topologies. Overall, this increase in latency remains within acceptable values for typical LoRaWAN telemetry applications that do not operate in strict real-time.

The RSSI and SNR values for scenario 1 are plotted in Figures 10 and 11, respectively. After the topology mode change, devices 4 and 5 reveal more variations in RSSI, obtaining both higher and lower values compared to the average RSSI before the topology change. However, after the topology change, the average RSSI is -75.41 dBm for device 4 and -75.11 dBm for device 5, which is better than the average RSSI prior to the topology mode change of -77.45 dBm and -75.53 dBm, respectively. The rest of the devices did not experience noticeable changes throughout the test. Regarding the SNR, there are no significant changes caused by updating the topology mode at the end devices. Therefore, considering the values obtained when all the devices were in the conventional star mode with the conditions of this experiment, we can conclude that the link quality remains within usual values for current LoRaWAN networks.

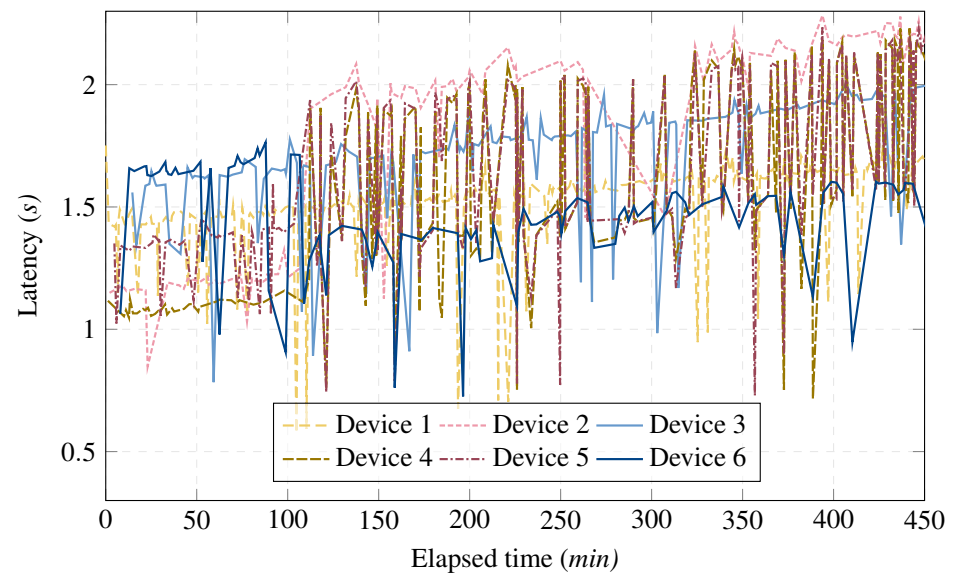


Figure 8. Latency in scenario 1.

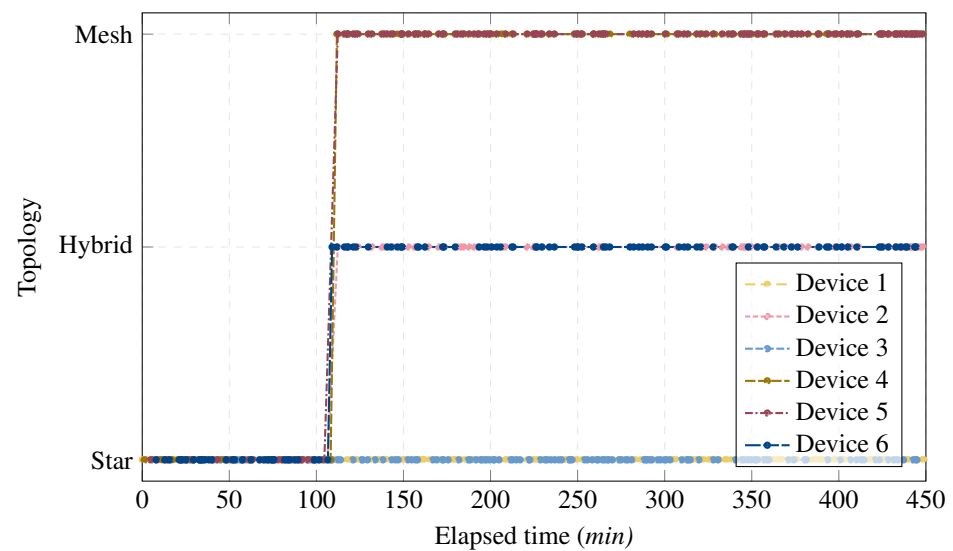


Figure 9. Topology changes of the devices in scenario 1.

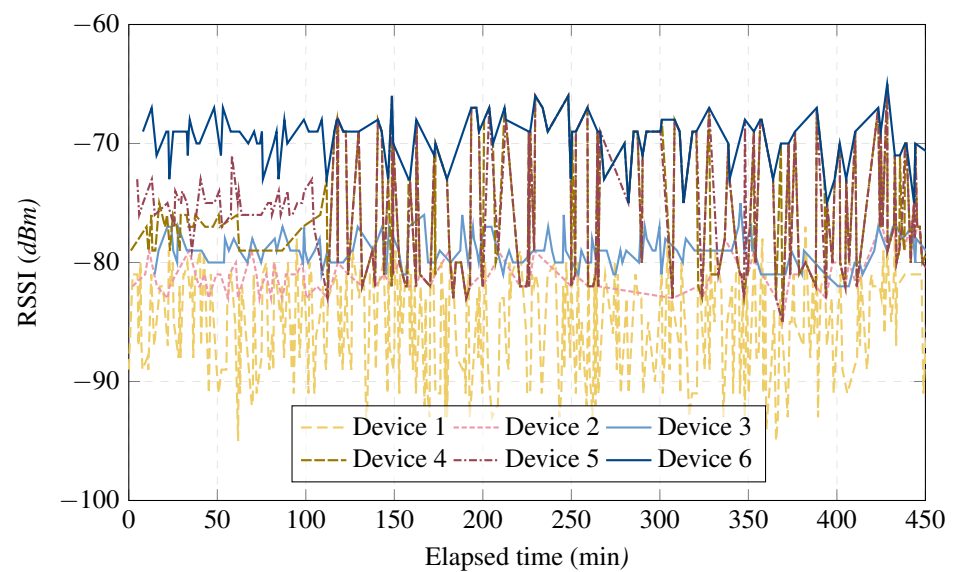


Figure 10. RSSI of the devices in scenario 1.

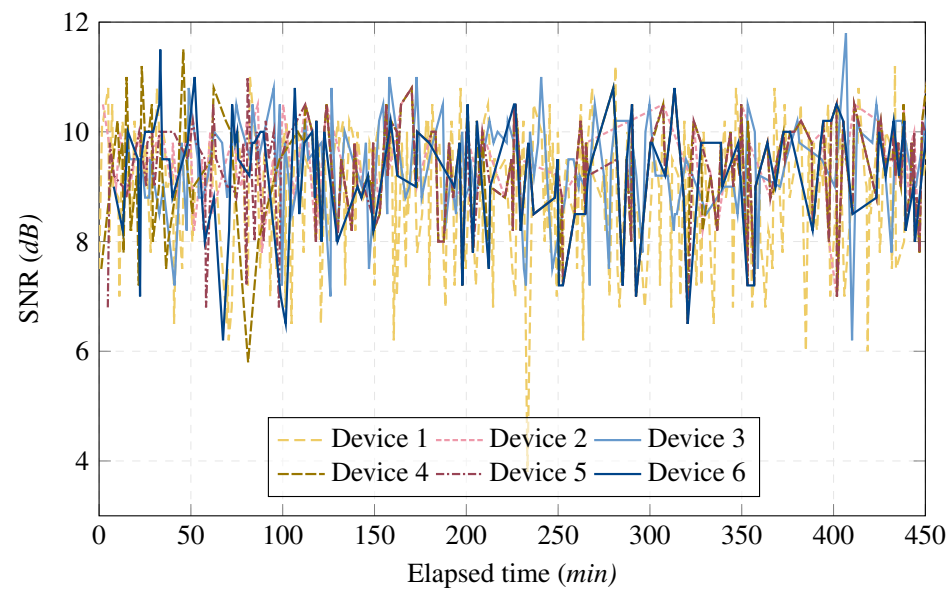


Figure 11. SNR of the devices in scenario 1.

The compilation of minimum, maximum, average, and variance of the metrics for each device in scenario 1 is provided in Table 3.

Table 3. Link quality metrics in scenario 1.

Metric	Device	Minimum	Maximum	Average	Variance
Latency (s)	1	0.6	1.8	1.5	0.03
	2	0.8	2.4	1.8	0.19
	3	0.8	2	1.7	0.06
	4	0.7	2.3	1.6	0.17
	5	0.7	2.3	1.7	0.14
	6	0.7	1.8	1.5	0.04
RSSI (dBm)	1	−95	−77	−85.4	20.15
	2	−85	−77	−80.7	3.07
	3	−82	−75	−79	1.87
	4	−85	−65	−75.7	28.68
	5	−85	−65	−75.5	27.42
	6	−81	−65	−69.8	6.22
SNR (dB)	1	3.8	11.2	8.9	1.4
	2	7	11	9.4	0.62
	3	6.2	11.8	9.3	0.87
	4	5.8	11.5	9.3	1.05
	5	6.5	11	9.3	0.95
	6	6.2	11.5	9.2	1.3

5.2. Scenario 2

The latency results obtained from the differences in time between data transmission at the end devices and data reception at the MQTT broker with two devices in mesh topology mode and one device functioning as a hybrid node are provided in Figure 12. Comparing the latency results and the topology changes (see Figure 13), we can see that all the devices that changed into mesh or hybrid topology mode (devices 2, 4, and 5) increased their latency in comparison to the conventional star mode due to a further added hop or having to handle a higher number of packets in the case of the hybrid end device. The devices that remained in star topology mode presented a slight increase in latency as the test progressed.

We can conclude that, under the conditions of the experiment, multi-hop topologies may result in an increase in latency, but this increase is not severe for the usual performance expected of LoRaWAN networks.

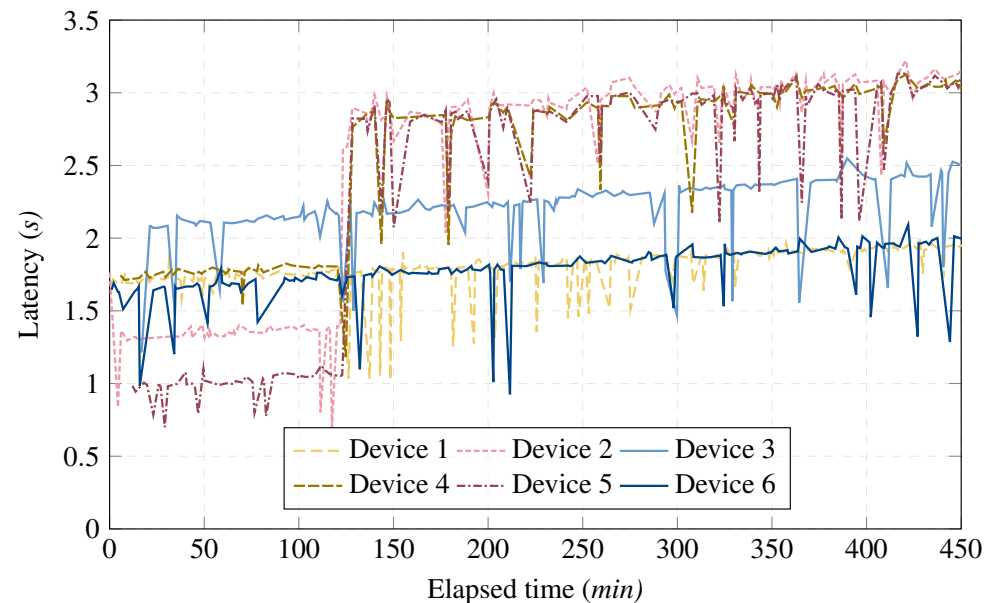


Figure 12. Latency in scenario 2.

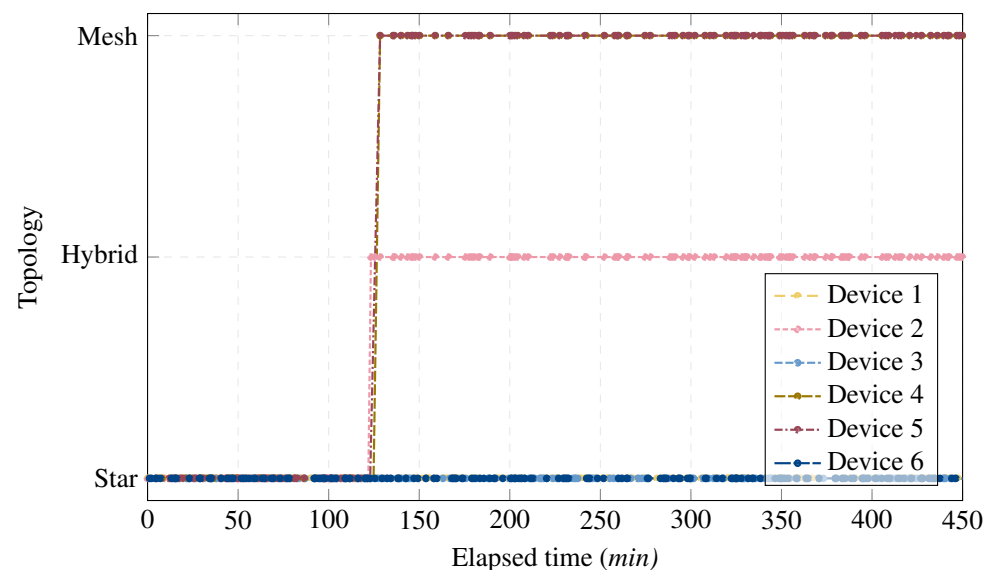


Figure 13. Topology changes of the devices in scenario 2.

The RSSI and SNR results are represented in Figures 14 and 15, respectively. For device 4, the RSSI values remain similar to those prior to the topology mode change, and they even become more stable. For device 5, there is a slight decrease in RSSI when changing topology modes, which makes the following RSSI values very similar to those of device 4. The rest of the devices present similar results throughout the test, with no significant variations after the topology change. Regarding the SNR, all devices exhibit similar results without significant changes after the topology mode update of some of the devices in the network. Although there are variations in SNR, which are expected in experimental tests, they generally remain above 6 dB, except for three instances concerning device 1. Considering the overall results over the entire test duration, Figure 15 shows consistently good SNR values for all devices, indicating minimal interference during the

tests. Therefore, the link quality results for this scenario reinforce the conclusion that was reached in scenario 1, where link quality is not significantly disturbed in mesh topologies.

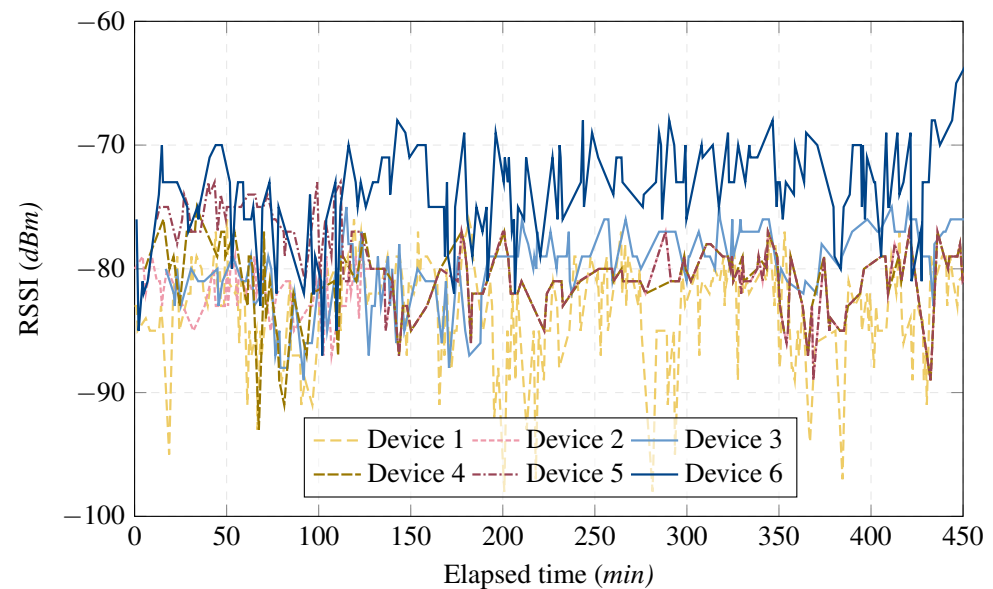


Figure 14. RSSI of the devices in scenario 2.

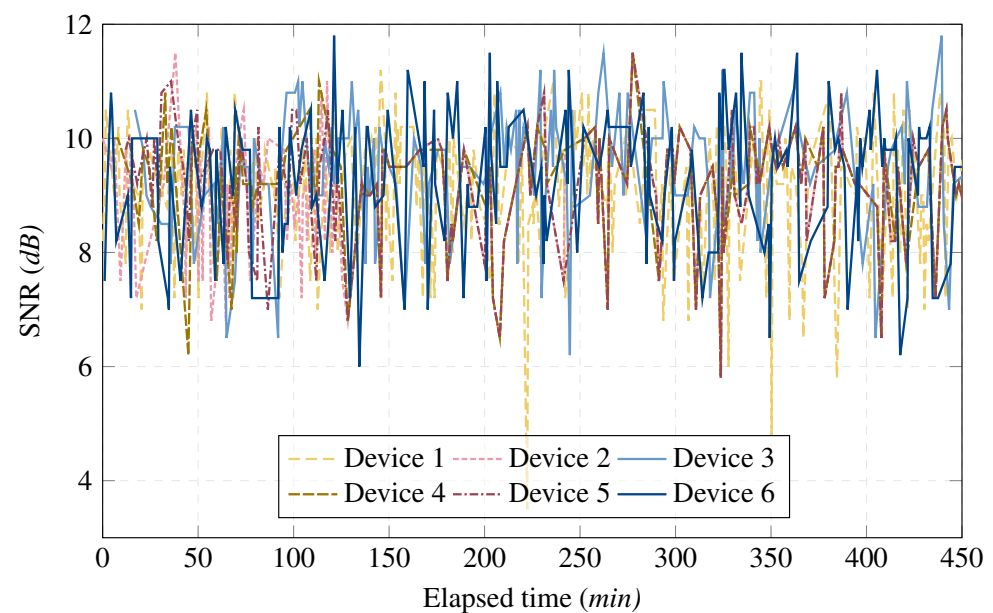


Figure 15. SNR of the devices in scenario 2.

The compilation of minimum, maximum, average, and variance of the metrics for each device in scenario 2 is provided in Table 4.

Table 4. Link quality metrics in scenario 2.

Metric	Device	Minimum	Maximum	Average	Variance
Latency (s)	1	1	2.1	1.8	0.02
	2	0.7	3.2	2.4	0.62
	3	1.2	2.6	2.2	0.04
	4	1.2	3.1	2.6	0.3
	5	0.7	3.1	2.3	0.77
	6	0.9	2.1	1.8	0.04

Table 4. Cont.

Metric	Device	Minimum	Maximum	Average	Variance
RSSI (dBm)	1	−98	−76	−83	17.24
	2	−89	−77	−81.2	6.04
	3	−89	−75	−79.7	9.68
	4	−93	−75	−80.9	9.74
	5	−89	−73	−79.6	11.06
	6	−87	−63	−73.7	17.56
SNR (dB)	1	3.5	11	9	1.37
	2	5.8	11.5	9	1.15
	3	6.2	11.8	9.5	1.13
	4	5.8	11.5	9.1	1.08
	5	5.8	11.5	9.2	1.05
	6	6	11.8	9.3	1.47

5.3. Scenario 3

Lastly, the latency for the devices in scenario 3 is plotted in Figure 16, and the respective topology changes are provided in Figure 17. In this case, the devices were instructed through the control UI to change topology modes so that the mesh end device could transmit first through one hybrid device and then through another. Device 5 is the end device that changes its topology mode to mesh. Its latency shows a small increase when transmitting through the first hybrid device (device 2) and another increase when changing the hybrid device to device 4. Nonetheless, device 4's latency remains below that of the star topology mode devices through the complete duration of the test. Regarding the hybrid mode devices, device 2 improves its performance when changing to hybrid topology mode and improves again when going back to star topology mode. Under the conditions of this test, device 4 continued with predominantly stable latency results with a slight increase as the test progressed. All the devices in star topology mode obtained the highest latency results and did not register significant changes.

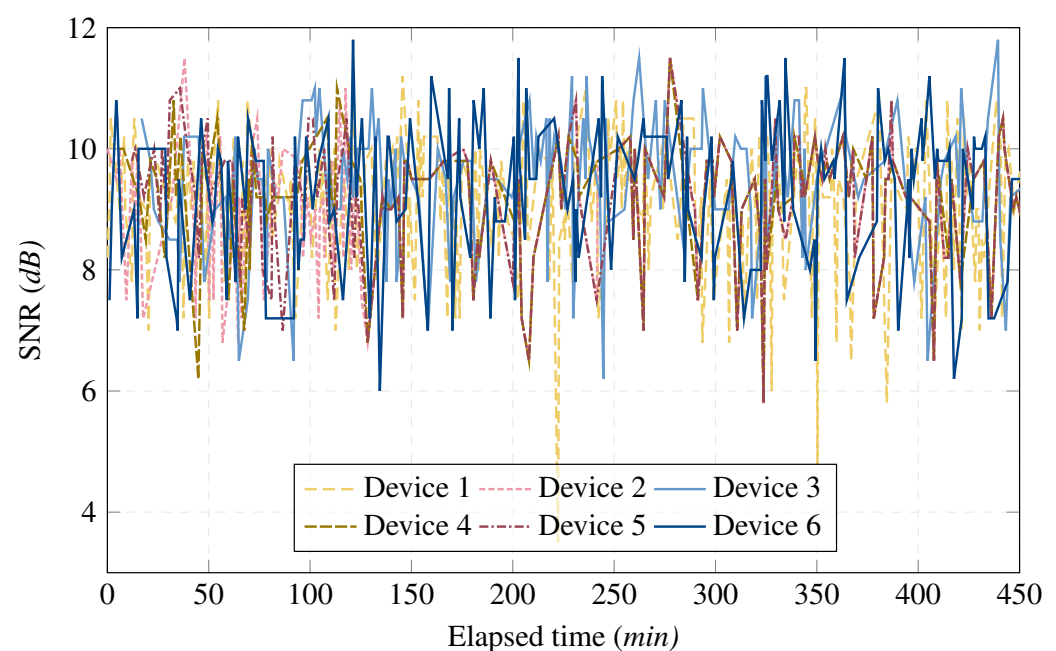


Figure 16. Latency in scenario 3.

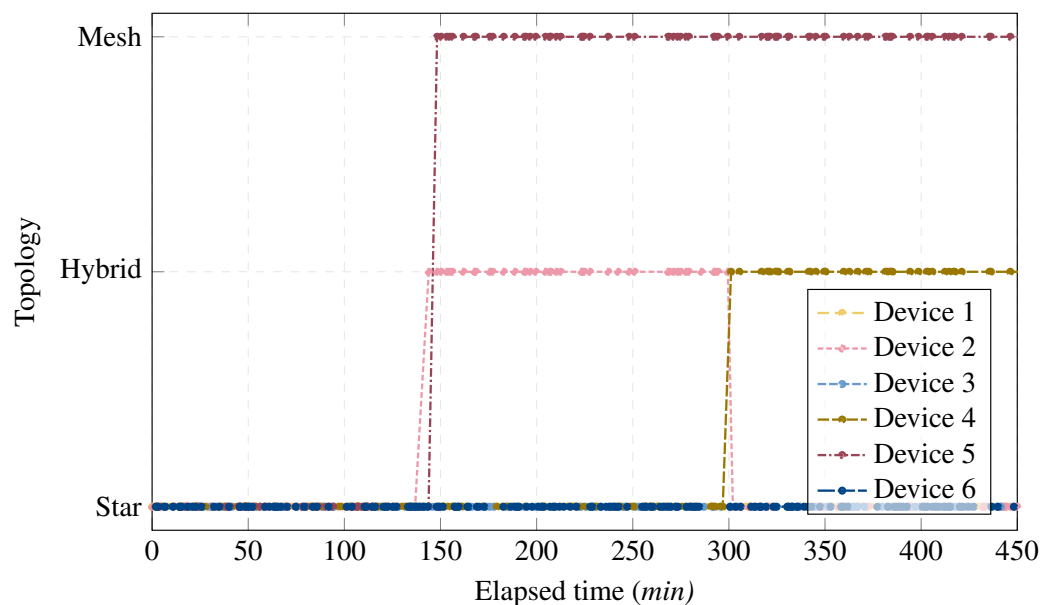


Figure 17. Topology changes of the devices in scenario 3.

The corresponding RSSI and SNR results for scenario 3 are illustrated in Figures 18 and 19, respectively. The RSSI values for device 5, which changes to mesh topology mode, remain fairly stable while transmitting through device 2, but the RSSI is substantially improved when the transmission is performed through device 4. Devices 2 and 4, in hybrid topology mode, do not present significant changes. Lastly, the devices in star topology mode, corresponding to device numbers 1, 3, and 6, remain stable or worsen their performance as the test evolves. On the other hand, the SNR remains stable for the complete test, with the exception of some peaks with higher noise for devices 1 and 3.

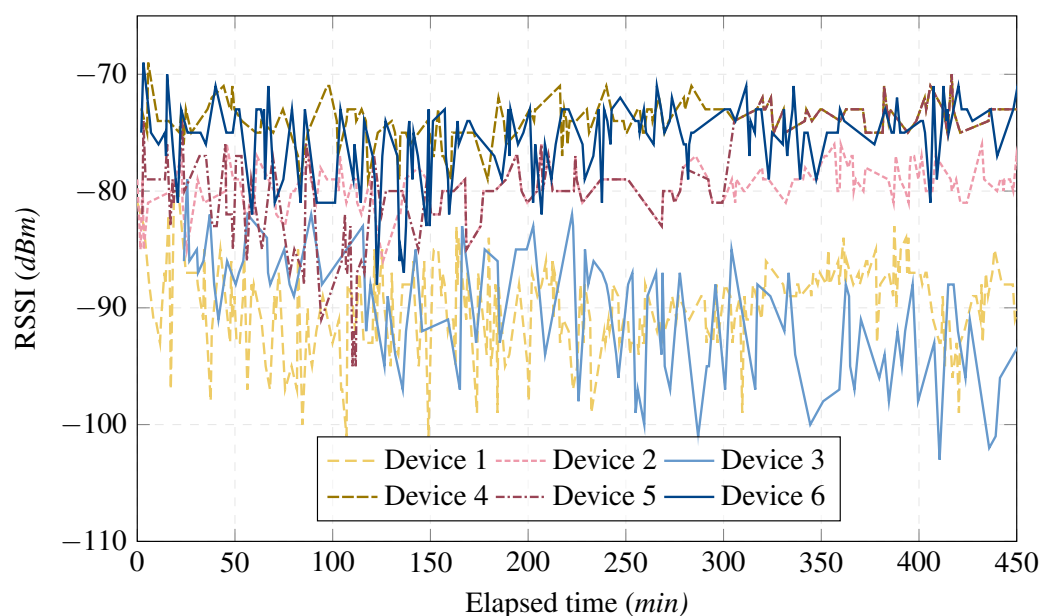


Figure 18. RSSI of the devices in scenario 3.

The compilation of minimum, maximum, average, and variance of the metrics for each device in scenario 3 is provided in Table 5.

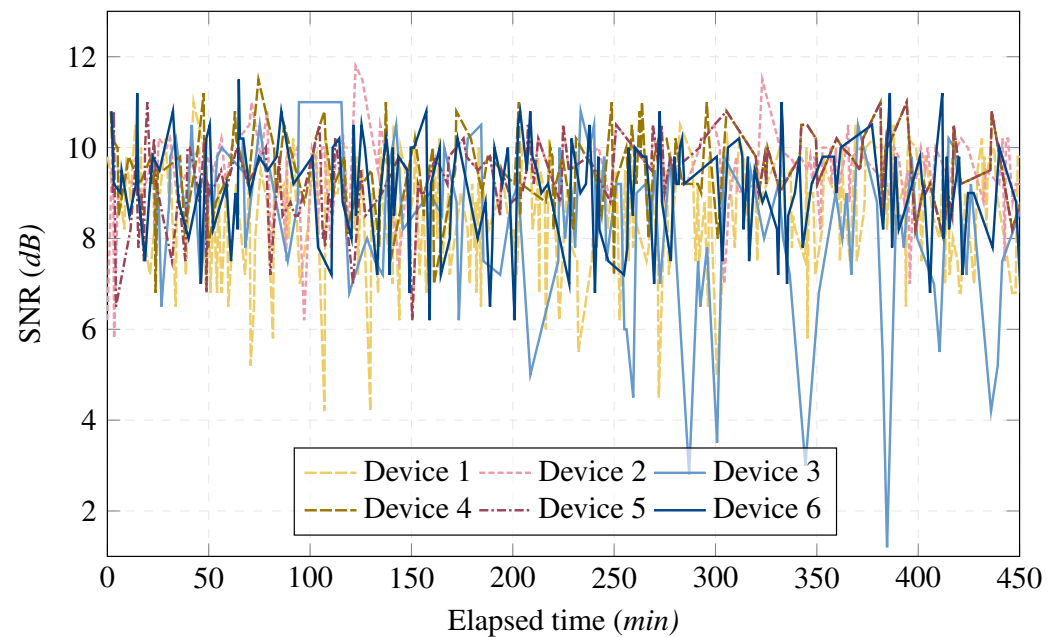


Figure 19. SNR of the devices in scenario 3.

Table 5. Link quality metrics collected in scenario 3.

Metric	Device	Minimum	Maximum	Average	Variance
Latency (s)	1	1.4	2.3	2.2	0.02
	2	0.8	2.2	1.7	0.05
	3	1.5	2.5	2.3	0.03
	4	1.2	2.3	1.9	0.03
	5	0.7	2.3	1.7	0.12
	6	1.3	2.4	2.1	0.05
RSSI (dBm)	1	−99	−83	−89.4	10.51
	2	−86	−76	−89.4	10.51
	3	−103	−79	−90.7	27.09
	4	−79	−69	−73.81	3.09
	5	−95	−70	−78.53	26.70
	6	−88	−69	−75.41	10.62
SNR (dB)	1	4.5	11.2	8.6	1.44
	2	5.8	11.8	9.4	0.87
	3	1.2	11	8.22	3.19
	4	6.8	11.5	9.5	0.78
	5	6.2	11	9.4	0.85
	6	6.2	11.5	9.1	1.38

6. Conclusions

LoRaWAN networks allow data transmission at long distances for countless added-value IoT applications. However, factors such as coverage in inaccessible areas, energy consumption limitations, and physical link quality may make the conventional star topology of LoRaWAN networks less than an ideal option for wireless IoT deployments. In this paper, a specification-compliant solution for star and mesh adaptable topology coexistence in LoRaWAN networks that introduces multi-hop transmissions while remaining completely transparent to the gateway and LoRaWAN server has been presented. The proposed solution has been implemented with the use of open-source software for the creation of LoRaWAN networks and the development of a set of software modules for topology management, network control through a user interface, and operation in star, mesh, and hybrid

topology modes at the end devices. This implementation has allowed for a testbed for experimentation on a university campus. Three scenarios for proof-of-concept experiments were designed to evaluate the feasibility and performance of coexisting star and mesh topology end devices under different configurations. The results obtained from these tests corroborate the operability of our proposed solution where, under the conditions of the experiments, the latency does not exhibit a significant increase and can even be improved in the case of multi-hop topologies. Therefore, LoRaWAN networks can benefit from mesh topologies while maintaining the performance expected for conventional LoRaWAN IoT telemetry applications characterized by not operating in strict real time.

As this paper has demonstrated the technical feasibility of star and mesh topology coexistence in LoRaWAN networks, for future work, we are planning to equip the topology manager with reinforcement learning techniques. This will allow for exploring the optimization of selecting the network topology mode according to multiple factors, such as link quality, energy consumption, and carbon footprint, among many others.

Author Contributions: Conceptualization, R.A.-C.; methodology, R.A.-C.; software, C.C.; validation, J.G.-H.; formal analysis, C.C., L.G. and J.G.-H.; investigation, C.C.; resources, C.-L.Z.-C.; data curation, C.C. and L.G.; writing—original draft preparation, L.G.; writing—review and editing, L.G. and J.G.-H.; visualization, L.G.; supervision, L.G., R.A.-C., C.-L.Z.-C. and A.-J.G.-S.; project administration, C.-L.Z.-C.; funding acquisition, L.G., R.A.-C., C.-L.Z.-C., A.-J.G.-S. and J.G.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the grants TED2021-129336B-I00, and FJC2021-047073-I, funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. This work is also supported by PID2023-148214OB-C21 funded by MICIU/AEI/10.13039/501100011033/ and by FEDER, EU. This work was also supported by the grant PCI2024-153485 funded by MICIU/AEI/10.13039/501100011033 and by the European Union. This work was also funded by Fundación Séneca (22236/PDC/23). This research was also supported by the project “Crowdsourcing Optimized Wireless Sensor Network Deployment (CRoWD)”, grant No.613-621119-852, funded by Dirección General de Investigaciones of Universidad Santiago de Cali. This work is also the result of the young researchers program through the project “Incorporación de jóvenes investigadores e innovadores en las regiones para atención de demandas definidas por los CODECTI de los departamentos de Chocó, Valle del Cauca, Cauca, Nariño”, BPIN 2022000100068 funded by Fondo de Ciencia, Tecnología e Innovación (FCTeI) of Sistema General de Regalías (SGR)—MINCIENCIAS.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. LoRa Alliance. Available online: <https://lora-alliance.org/> (accessed on 24 February 2025).
2. The Things Network. Available online: <https://www.thethingsnetwork.org/> (accessed on 24 February 2025).
3. Basford, P.J.; Bulot, F.M.; Apetroaie-Cristea, M.; Cox, S.J.; Ossont, S.J. LoRaWAN for smart city IoT deployments: A long term evaluation. *Sensors* **2020**, *20*, 648. [CrossRef] [PubMed]
4. Sisinni, E.; Carvalho, D.F.; Ferrari, P. Emergency communication in IoT scenarios by means of a transparent LoRaWAN enhancement. *IEEE Internet Things J.* **2020**, *7*, 10684–10694.
5. LoRa Alliance Technical Committee Regional Parameters Workgroup. *LoRaWAN 1.0.3 Regional Parameters*; LoRa Alliance: Fremont, CA, USA, 2018.
6. Todoli-Ferrandis, D.; Silvestre-Blanes, J.; Sempere-Payá, V.; Planes-Martinez, A. Analysis of bidirectional ADR-enabled class B LoRaWAN networks in industrial scenarios. *Appl. Sci.* **2020**, *10*, 7964. [CrossRef]
7. Luvisotto, M.; Tramarin, F.; Vangelista, L.; Vitturi, S. On the use of LoRaWAN for indoor industrial IoT applications. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 3982646.

8. Marquez, L.E.; Osorio, A.; Calle, M.; Velez, J.C.; Serrano, A.; Candelo-Becerra, J.E. On the use of LoRaWAN in smart cities: A study with blocking interference. *IEEE Internet Things J.* **2019**, *7*, 2806–2815.
9. Schroder Filho, H.; Pissolato Filho, J.; Moreli, V. The adequacy of LoRaWAN on smart grids: A comparison with RF mesh technology. In Proceedings of the 2016 IEEE International Smart Cities Conference (ISC2), Trento, Italy, 12–15 September 2016; pp. 1–6.
10. Singh, R.K.; Aernouts, M.; De Meyer, M.; Weyn, M.; Berkvens, R. Leveraging LoRaWAN technology for precision agriculture in greenhouses. *Sensors* **2020**, *20*, 1827. [\[CrossRef\]](#)
11. Liu, J.; Gao, J.; Jha, S.; Hu, W. Seirios: leveraging multiple channels for LoRaWAN indoor and outdoor localization. In Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, New Orleans, LA, USA, 25–29 October 2021; pp. 656–669.
12. Sherazi, H.H.R.; Grieco, L.A.; Imran, M.A.; Boggia, G. Energy-efficient LoRaWAN for industry 4.0 applications. *IEEE Trans. Ind. Inform.* **2020**, *17*, 891–902.
13. Gaitan, N.C. A long-distance communication architecture for medical devices based on LoRaWAN protocol. *Electronics* **2021**, *10*, 940. [\[CrossRef\]](#)
14. Haxhibeqiri, J.; De Poorter, E.; Moerman, I.; Hoebeke, J. A survey of LoRaWAN for IoT: From technology to application. *Sensors* **2018**, *18*, 3995. [\[CrossRef\]](#)
15. Adelantado, F.; Vilajosana, X.; Tuset-Peiro, P.; Martinez, B.; Melia-Segui, J.; Watteyne, T. Understanding the limits of LoRaWAN. *IEEE Commun. Mag.* **2017**, *55*, 34–40.
16. Soto-Vergel, A.; Arismendy, L.; Bornacelli-Durán, R.; Cardenas, C.; Montero-Arévalo, B.; Rivera, E.; Calle, M.; Candelo-Becerra, J.E. LoRa performance in industrial environments: Analysis of different ADR algorithms. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10501–10511.
17. Farooq, M.O. Clustering-based layering approach for uplink multi-hop communication in LoRa networks. *IEEE Netw. Lett.* **2020**, *2*, 132–135.
18. Lalle, Y.; Fourati, M.; Fourati, L.C.; Barraca, J.P. Routing strategies for LoRaWAN multi-hop networks: A survey and an SDN-based solution for smart water grid. *IEEE Access* **2021**, *9*, 168624–168647.
19. Alvarado-Alcon, F.J.; Asorey-Cacheda, R.; Garcia-Sanchez, A.J.; Garcia-Haro, J. Carbon Footprint vs Energy Optimization in IoT Network Deployments. *IEEE Access* **2022**, *10*, 111297–111309.
20. Aslam, M.S.; Khan, A.; Atif, A.; Hassan, S.A.; Mahmood, A.; Qureshi, H.K.; Gidlund, M. Exploring multi-hop LoRa for green smart cities. *IEEE Netw.* **2019**, *34*, 225–231.
21. Choi, R.; Lee, S.; Lee, S. Reliability improvement of lora with arq and relay node. *Symmetry* **2020**, *12*, 552. [\[CrossRef\]](#)
22. Liao, C.H.; Zhu, G.; Kuwabara, D.; Suzuki, M.; Morikawa, H. Multi-hop LoRa networks enabled by concurrent transmission. *IEEE Access* **2017**, *5*, 21430–21446.
23. Kim, H.; Kim, H.; Baek, S.; Melenchuk, R.; Soroka, J.; Smith, A. Hybrid LoRa Network Architecture: Automatic Switching between LoRaWAN and LoRa Mesh Network in Environments with Dynamic Obstacle Variations. In Proceedings of the 2024 33rd International Conference on Computer Communications and Networks (ICCCN), Big Island, HI, USA, 29–31 July 2024; pp. 1–6.
24. Resources LoRa Alliance. TS011-1.0.0 Relay. Available online: <https://resources.lora-alliance.org/technical-specifications/ts011-1-0-0-relay> (accessed on 24 February 2025).
25. 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. [\[CrossRef\]](#)
26. Pan, M.; Chen, C.; Yin, X.; Huang, Z. UAV-aided emergency environmental monitoring in infrastructure-less areas: LoRa mesh networking approach. *IEEE Internet Things J.* **2021**, *9*, 2918–2932.
27. Ebi, C.; Schaltegger, F.; Rüst, A.; Blumensaat, F. Synchronous LoRa mesh network to monitor processes in underground infrastructure. *IEEE Access* **2019**, *7*, 57663–57677.
28. Lloret, J.; García, L.; Jimenez, J.M.; Sendra, S.; Lorenz, P. Cluster-based communication protocol and architecture for a wastewater purification system intended for irrigation. *IEEE Access* **2021**, *9*, 142374–142389.
29. 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.
30. Solé, J.M.; Centelles, R.P.; Freitag, F.; Meseguer, R. Implementation of a LoRa mesh library. *IEEE Access* **2022**, *10*, 113158–113171.
31. Veloso, A.F.d.S.; Júnior, J.V.R.; Rabelo, R.d.A.L.; Silveira, J.D.f. HyDSMaaS: A Hybrid Communication Infrastructure with LoRaWAN and LoraMesh for the Demand Side Management as a Service. *Future Internet* **2021**, *13*, 271. [\[CrossRef\]](#)
32. Sisinni, E.; Ferrari, P.; Carvalho, D.F.; Rinaldi, S.; Marco, P.; Flammini, A.; Depari, A. LoRaWAN range extender for industrial IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5607–5616.
33. Zhou, W.; Tong, Z.; Dong, Z.Y.; Wang, Y. LoRa-Hybrid: A LoRaWAN Based multihop solution for regional microgrid. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019; pp. 650–654.

34. Almeida, N.C.; Rolle, R.P.; Godoy, E.P.; Ferrari, P.; Sisinni, E. Proposal of a hybrid LoRa Mesh/LoRaWAN network. In Proceedings of the 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Rome, Italy, 3–5 June 2020; pp. 702–707.
35. Code Visual Studio. Code Editing. Available online: <https://code.visualstudio.com/> (accessed on 24 February 2025).
36. Marketplace Visual Studio. Pymakr. Available online: <https://marketplace.visualstudio.com/items?itemName=pycom.Pymakr> (accessed on 24 February 2025).
37. Wifx IoT. LORIX One LoRaWAN Gateway. Available online: <https://iot.wifx.net/en/products/lorix-one/> (accessed on 24 February 2025).
38. ChirpStack Open-Source. LoRaWAN Network Server. Available online: <https://www.chirpstack.io/> (accessed on 24 February 2025).
39. Mosquitto. Eclipse. Available online: <https://mosquitto.org/> (accessed on 24 February 2025).
40. MongoDB: The Developer Data Platform. Available online: <https://www.mongodb.com/> (accessed on 24 February 2025).

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.