electronics

MDPI

Article

Worker Presence Monitoring in Complex Workplaces Using BLE Beacon-Assisted Multi-Hop IoT Networks Powered by ESP-NOW

Raihan Uddin, Taewoong Hwang and Insoo Koo *

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, Republic of Korea; raihan80@mail.ulsan.ac.kr (R.U.); taeung0210@mail.ulsan.ac.kr (T.H.)
* Correspondence: iskoo@ulsan.ac.kr

Abstract: The increasing adoption of Internet of Things (IoT) technologies has facilitated the creation of advanced applications in various industries, notably in complex workplaces where safety and efficiency are paramount. This paper addresses the challenge of monitoring worker presence in vast workplaces such as shipyards, large factories, warehouses, and other construction sites due to a lack of traditional network infrastructure. In this context, we developed a novel system integrating Bluetooth Low Energy (BLE) beacons with multi-hop IoT networks by using the ESP-NOW communications protocol, first introduced by Espressif Systems in 2017 as part of its ESP8266 and ESP32 platforms. ESP-NOW is designed for peer-to-peer communication between devices without the need for a WiFi router, making it ideal for environments where traditional network infrastructure is limited or nonexistent. By leveraging the BLE beacons, the system provides real-time presence data of workers to enhance safety protocols. ESP-NOW, a low-power communications protocol, enables efficient, low-latency communication across extended ranges, making it suitable for complex environments. Utilizing ESP-NOW, the multi-hop IoT network architecture ensures extensive coverage by deploying multiple relay nodes to transmit data across large areas without Internet connectivity, effectively overcoming the spatial challenges of complex workplaces. In addition, the Message Queuing Telemetry Transport (MQTT) protocol is used for robust and efficient data transmission, connecting edge devices to a central Node-RED server for real-time remote monitoring. Moreover, experimental results demonstrate the system's ability to maintain robust communication with minimal latency and zero packet loss, enhancing worker safety and operational efficiency in large, complex environments. Furthermore, the developed system enhances worker safety by enabling immediate identification during emergencies and by proactively identifying hazardous situations to prevent accidents.

Keywords: IoT; BLE beacon; ESP-NOW; multi-hop networks; worker presence monitoring; worker safety; remote monitoring

check for updates

Citation: Uddin, R.; Hwang, T.; Koo, I. Worker Presence Monitoring in Complex Workplaces Using BLE Beacon-Assisted Multi-Hop IoT Networks Powered by ESP-NOW. *Electronics* **2024**, *13*, 4201. https://doi.org/10.3390/electronics13214201

Academic Editor: Antoni Morell

Received: 8 October 2024
Revised: 17 October 2024
Accepted: 24 October 2024
Published: 26 October 2024

Copyright: © 2024 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 widespread adoption of Internet of Things (IoT) technologies has enabled the development of innovative applications in various industries [1], including smart workplaces where safety and efficiency are paramount. One of the critical challenges in such environments is monitoring worker presence, especially in complex workplaces such as large factories, warehouses, and construction sites, due to falling under traditional network infrastructure. The need to estimate the workers and monitor them accurately is not only for productivity and workflow optimization but also for safety and emergency responses. Keeping people safe requires innovative solutions, and the IoT offers a promising approach that has expanded across industries, transforming the way we perceive and interact with the world, and it has revolutionized work paradigms [2].

One of the most transformative aspects of the revolution is presence-monitoring-based services, where Bluetooth Low Energy (BLE) is emerging as a champion, enabling beacon technology to provide a real-time view of presence-monitoring-based information [3].

Central to this technological evolution is the synergy between BLE and beacon technology. Particularly in the context of wireless communications and presence-monitoring-based services, a beacon is typically a small device that broadcasts signals to nearby devices by utilizing technologies such as BLE [4]. Its energy-efficient communications protocol ensures that connectivity is always available without compromising essential power integrity [5]. As a complementary technology, beacons play a key role in presence monitoring-based services.

In the diverse world of the IoT, a communications protocol acts as a backbone enabling seamless data exchange between connected devices. Several well-established protocols meet the diverse requirements of IoT applications. Among them, WiFi stands out for its widespread use and high data rates, making it ideal for applications that require robust connectivity [6]. Zigbee, with its low-power design and mesh networking capabilities, occupies an important place in applications such as home automation and industrial control systems [7]. BLE Mesh extends the range of Bluetooth-enabled IoT networks suitable for scenarios requiring wide coverage, while Thread, based on IEEE 802.15.4, is ideal for secure and reliable communications in IoT environments, particularly in home automation and industrial applications [8,9]. These protocols, each tailored to specific IoT needs, form the backbone of connected systems. However, the foundation of the proposed integrated system is laid by ESP-NOW, a low-power communications protocol with low latency, minimal power consumption, and an extended range that is particularly suited to complex workplaces [10,11].

Our journey begins with exploring BLE beacon technology in the context of complex workplaces within the IoT domain. The technology is a beacon of innovation, providing services for monitoring worker presence that arre key to improving safety protocols in large and complex workplaces. Moving forward, we explore ESP-NOW, a low-power communications protocol tailored for ESP32 devices. Then, we focus on multi-hop IoT networks, a strategy that makes use of multiple nodes to provide extended coverage within the vast areas of workplaces such as shipyards, large factories, warehouses, and other construction sites. This approach addresses the spatial challenges inherent in large complex environments by ensuring that every corner of the overall workplace remains within the communications network. Finally, we explore the Message Queuing Telemetry Transport (MQTT) protocol, which provides a robust means of transmitting data in an efficient IoT environment [12]. This protocol acts as a reliable bridge between edge devices and a central Node-RED server, enabling real-time monitoring and streamlined communication. By unfolding these layers, the proposed integrated system synthesizing BLE beacons, ESP-NOW, a multi-hop IoT network, and MQTT emerges as a holistic solution tailored to the specific needs of safety, promising a safer and more efficient workplace environment.

However, our primary goal is to usher in a new era of safety and efficiency in complex workplaces through an innovative real-time worker presence monitoring system. By integrating cutting-edge technologies such as the BLE beacon, nRF Connect, ESP-NOW, multi-hop IoT networks, MQTT, and the Node-RED remote server, and with our main focus on improving safety, the system enables the immediate identification of trapped workers during emergencies, facilitating fast and targeted rescue operations that have the potential to save lives. Addressing the critical aspect of risk reduction, the system proactively identifies workers in hazardous situations posing the risk of fire, explosion, and/or suffocation, helping to prevent accidents and promote a safer working environment. The overall objective is to ensure continuous monitoring of worker presence in complex workplaces. This initiative not only improves situational awareness and emergency response but also contributes to a safer and more efficient environment in the workplace. By prioritizing enhanced safety, improved efficiency, and risk reduction, we aim to set a new standard for monitoring worker presence and transforming workplaces in the landscape of the IoT.

The novelty of our approach is summarized as follows.

- The integration of BLE beacons with a multi-hop network: We developed a worker monitoring system using BLE beacon technology via an ESP32 microcontroller as a

beacon tag attached to workers, which broadcasts worker presence signals. The system also integrates an Android application called nRF Connect, which acts as an alternative beacon. These signals are scanned and transmitted via a multi-hop network using ESP-NOW, enabling efficient worker presence monitoring in large, complex workplace environments.

- Reliable data transmission without internet connectivity: We created a robust multi-hop network using ESP-NOW to extend communication coverage across large, complex workplaces, effectively overcoming spatial challenges without relying on traditional network infrastructure. This approach utilizes a scanner node along with multiple relay nodes to transmit data to its destination without requiring internet connectivity, ensuring minimal latency and zero packet loss, thereby enabling reliable worker presence monitoring even in environments with significant obstacles.
- Real-time remote monitoring via MQTT and Node-RED: We implemented a three-layer IoT architecture using MQTT and Node-RED, enabling efficient data transmission from the gateway node to a central server for real-time remote visualization of worker presence. This solution enhances workplace safety and scalability by providing stakeholders, especially during emergencies, with immediate remote access to worker presence data from anywhere, ensuring efficient monitoring even in complex environments.

2. Related Literature

Multi-hop network systems have seen widespread adoption in various applications, ranging from smart metering in rural areas to industrial automation and home automation systems. These networks are designed to enhance communication range, reliability, and scalability, especially in environments where traditional infrastructure like wired connections or centralized wireless systems may be impractical. Over the years, several studies have proposed solutions leveraging multi-hop communication to address specific challenges such as latency, packet loss, and scalability.

One study presents a wireless mesh network (WMN) for agility training, using the ESP32 module to create a flexible, scalable, and cost-effective system [13]. The system employs multi-hop communication in a tree topology, with each node acting as a relay to extend the network's range, enabling real-time data transmission and immediate feedback for athletes. It performs well in small-scale setups, maintaining low latency (under 300 ms) and minimal packet loss, making it ideal for university-level training. However, as the network expands beyond three layers, performance degrades, with response times increasing to around 330 ms and packet loss rising to 6%. The study indicates that while effective for smaller environments, the system faces scalability challenges and may struggle in larger or more complex setups, particularly due to the lack of advanced network management features.

A further contribution to the field is the development of the MRT-BLE protocol, designed for industrial wireless sensor networks, enabling real-time communication over Bluetooth Low Energy (BLE) mesh networks [14]. The protocol adopts a connection-oriented approach with Time Division Multiple Access (TDMA) scheduling, ensuring time-bound data transmission and predictable delays, making it ideal for industrial environments where timing guarantees are crucial. The system demonstrates reliable communication over multiple hops, with acceptable end-to-end delays, but it is primarily designed for static topologies, limiting its flexibility in dynamic or mobile networks. Additionally, the study does not provide extensive evaluation of the protocol's scalability beyond small network sizes, and it lacks data on packet loss and reliability in more complex networks with higher node densities.

Another paper explores a LoRa-based mesh network for smart metering in rural microgrids, using multi-hop communication to overcome the range limitations of traditional star-based LoRa networks [15]. This approach allows flexible node placement in remote areas and supports low-power, long-range communication, offering a cost-effective alternative to GSM-based systems. Simulations using the NS2 simulator show the network

can handle up to 250 nodes with acceptable end-to-end delay and packet delivery ratio (PDR). However, as the number of nodes and spreading factor (SF) increase, bandwidth limitations affect system efficiency. The study also lacks a detailed evaluation of real-world factors like terrain, noise, and interference, which could impact performance. These factors, including packet loss and network reliability, are not deeply evaluated in the simulation.

In the realm of home automation, another study proposes a system built on the ESP-NOW communication protocol, combined with MQTT for real-time remote control and monitoring [16]. This low-power, cost-effective system enables secure communication within a mesh network through the use of the custom BRAM-Now routing algorithm and AES encryption for data security. The system achieves low latency (around 75 ms) in real-world tests, but it experiences a packet loss rate of 9.25%, particularly in areas with high interference. Additionally, the system's scalability is limited to 10 encrypted nodes, and node management requires manual configuration, which reduces its suitability for larger-scale or dynamic environments where automated node setup would be more efficient. Despite these limitations, the system performs well for typical home automation setups but may require optimization for better reliability in larger or more interference-prone environments.

While these studies have made important advances in their fields, a key challenge still exists: achieving consistent reliability and zero packet loss in larger, more complex multi-hop networks, while keeping latency low. Many of the systems face issues with packet loss and higher latency as the network expands and becomes more complex, especially in environments with obstacles or interference. Additionally, few have found ways to scale up without running into major performance issues.

However, this paper addresses the aforementioned gaps by developing a multi-hop IoT system that guarantees zero packet loss and low latency, even in complex environments. Unlike previous studies that have experienced packet loss or latency issues as their networks scale, the system presented here optimizes both reliability and performance through advanced multi-hop communication techniques and efficient routing algorithms. This research focuses on ensuring that data transmission remains uninterrupted and consistent, regardless of network size or environmental challenges such as interference or obstacles. The system is designed to be highly scalable while maintaining real-time performance, making it ideal for applications that require both high reliability and low-latency communication, particularly in environments where consistency and efficiency are crucial.

3. BLE Beacons

BLE beacon technology emerged with the development of the Bluetooth 4.0 specification, which introduced the low-energy variant of Bluetooth that was formally adopted by the Bluetooth Special Interest Group (SIG) [17,18]. BLE is specifically designed to enable low-power communication between devices [19], making it ideal for applications where energy efficiency is a concern, such as wearable devices, sensors, and beacons. The concept of using beacons for proximity-based interactions and location-aware services existed before the formal introduction of BLE beacon technology [20].

However, BLE beacon technology forms a core part of the worker presence monitoring system presented in this study. It leverages BLE to facilitate efficient, real-time worker presence monitoring in complex workplace environments. The functionality of BLE beacon technology can be divided into two key aspects: proximity detection and data transmission. These features are crucial for providing a worker presence monitoring system integrated with robust and scalable multi-hop IoT networks that enhance both operational efficiency and worker safety.

Proximity detection

- Broadcast signals: BLE technology is utilized through beacon tags carried by workers in complex workplace environments. Each beacon tag continuously broadcasts signals containing a unique identifier, which acts as a digital signature to distinctly identify the worker. This constant broadcasting ensures that the worker's presence is reliably

detected within the specified range, thereby verifying and monitoring the worker's presence through the beacon tag within the designated area.

- Received signals: The signals emitted by the beacon tags are received by a scanner node specifically configured to receive BLE signals. The scanner node confirms the worker's presence by capturing these signals whenever they are within the defined range. By placing the scanner node in a designated area, the worker's presence is reliably verified in a complex workplace whenever the beacon signals are received. Unlike tracking detailed movements, the primary objective here is to verify whether a worker is present in designated zones based on the received BLE signals.

Data transmission

- Universally Unique Identifier (UUID): Each beacon tag broadcasts a UUID, which acts as the beacon's digital fingerprint. This UUID ensures that receiving devices can uniquely identify each beacon, distinguishing it from other beacons operating in the same area. This level of individualization is crucial in workplaces where numerous beacons may be deployed simultaneously, as it allows the system to accurately track each worker without any overlap or confusion.
- Transmission power: Depending on the configuration, some BLE beacon tags can broadcast their transmission power level. By combining this information with the received signal strength indicator (RSSI), the system can estimate the distance between the beacon and the receiver. However, this aspect is not a primary focus of our current study. Instead, we have included this information to highlight its potential significance, as it will form a crucial part of our future research, particularly for enhancing proximity estimation and improving worker safety in hazardous areas.
- Customizable data: BLE beacons are capable of transmitting additional information, which can be customized based on the specific requirements of the application. In our configuration, the beacon is programmed to broadcast the worker's name for accurate identification. This makes the system adaptable to various scenarios, such as monitoring worker presence in restricted areas or issuing alerts during emergencies. By transmitting specific names, the system improves real-time accuracy and reliability, especially in complex workplace environments.

However, configuring the beacon tags with these BLE features enables the system to effectively detect the presence of workers as they carry beacon tags throughout the workplace. As shown in Figure 1, a BLE beacon communicates its presence through an advertisement packet, which includes a preamble for synchronization, an access address for channel access, and core data within the protocol data unit (PDU). The preamble refers to a series of bits used for synchronization between the beacon and the scanner, allowing the scanner to detect and lock onto the start of the incoming packet, ensuring that it can properly interpret the incoming data. Additionally, the access address is vital for channel access, helping the system identify the start of the packet. These elements are essential for accurate communication and synchronization between the beacon and the scanner. The payload contains key elements, such as a UUID for uniquely identifying workers, a unique device address, transmission power (Tx), and the RSSI value. It also includes customizable data such as the worker's name, which helps ensure accurate identification. Scanner nodes, configured to receive these signals, capture the broadcast data when the beacon tags are within range, ensuring the worker's presence is verified in designated areas. The inclusion of an error-detecting cyclic redundancy check (CRC) helps maintain data integrity during transmission. In the BLE advertising packet, the CRC is used to detect any errors that may have occurred during the transmission of the packet. This mechanism contains a checksum, which is calculated from the preceding data in the packet. Upon receiving the packet, the scanner node recalculates the CRC and compares it with the transmitted CRC. If the values match, the packet is considered valid. If they do not match, the packet is discarded, ensuring that only error-free data is processed by the system. This simple but effective error-checking process enhances the reliability of beacon communication.

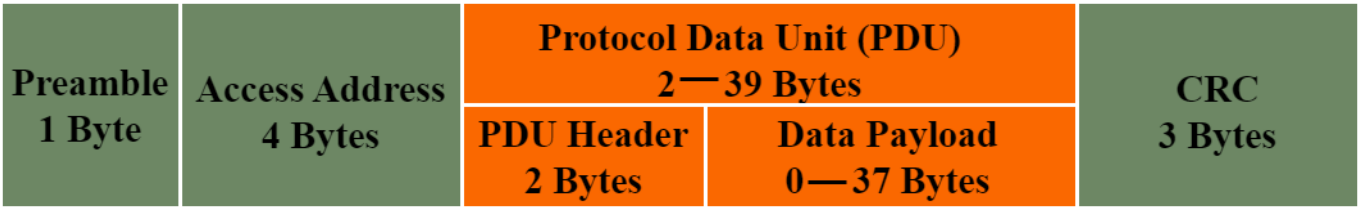


Figure 1. The structure of an advertising packet from a BLE beacon.

4. The ESP-NOW Protocol

ESP-NOW, developed by Espressif Systems, acts as a communication protocol that is specifically tailored for the ESP8266 and ESP32 WiFi modules [21]. The primary purpose of the protocol is to provide a smart and efficient solution for IoT networks. ESP-NOW, designed especially for resource-constrained devices, emphasizes simplicity, speed, and power efficiency, which makes it an ideal choice for a wide range of IoT applications. It eliminates the need for a central access point (router) by enabling direct, fast, and reliable communication between devices. This is achieved by focusing solely on the data link layer (Layer 2) of the OSI model [22]. This streamlined approach removes the complexities associated with higher layers, resulting in faster data transmission with minimal processing overhead. ESP-NOW is specifically intended to facilitate peer-to-peer communications, enabling direct data exchanges between devices without the need for a traditional WiFi network infrastructure. This feature is particularly beneficial for low-power devices that require efficient communication while conserving energy.

ESP-NOW supports both encrypted and unencrypted unicast communications. This allows the use of a mixed configuration of encrypted and unencrypted peer devices. It provides a send callback function to inform the application layer of the success or failure of the transmission. Nevertheless, the technology has limitations. In station mode, it supports a maximum of 10 encrypted peers, while in software access point (SoftAP) mode, or SoftAP combined with station mode, a maximum of six peers is supported. Although multiple unencrypted peers are allowed, in this case, the total number of unencrypted peers does not exceed 20, including encrypted peers [23].

However, the ESP-NOW protocol utilizes a unique action frame to encapsulate the data to be transmitted. This frame acts as a container for essential elements required for routing and delivery of the data to the intended recipient. The main components of an ESP-NOW packet include the following:

- Header: This section carries crucial information for routing the packet to the intended receiver. The most significant element within the header is typically the MAC address of the receiving device. This address ensures the data reach the correct destination.
- Payload: This is the core of the packet and contains the actual data to be transmitted. The payload size in ESP-NOW is limited to a maximum of 250 bytes. This limitation makes ESP-NOW ideal for transmitting small data packets commonly required in low-power applications, such as sensor readings, control commands, or configuration settings.
- Checksum: This value plays an important role in ensuring data integrity during transmission. The sender calculates a checksum based on the payload. After receiving the packet, the receiver recalculates the checksum based on the received data. If the checksums match, the data have arrived without errors. Conversely, a mismatch indicates the data were corrupted during transmission, which prompts a retransmission attempt.

By employing the above-structured approach, ESP-NOW ensures efficient data routing, transmission, and integrity verification, making it suitable for reliable peer-to-peer communications in IoT scenarios.

4.1. Pairing Mechanisms and Synchronization

Pairing establishes communication channels between devices, allowing them to recognize each other and participate in data forwarding within the multi-hop network. The two primary pairing mechanisms (static and dynamic) can be used for multi-hop communications in ESP-NOW implementations that support them. However, the specific approach implemented can determine how communication channels are established within the multi-hop network.

- Static pairing: In this approach to static multi-hop pairing, each device is pre-configured with the MAC addresses of its communication partners on the network. In this configuration, the sending device can communicate directly with the designated neighboring device, which acts as the first relay node in the data forwarding process. Thus, it essentially defines the forwarding paths that data packets will take to reach their destination. This improves the reliability of data delivery because the forwarding paths are predefined, and data packets follow a specific route.
- Dynamic pairing: This method enables devices to identify each other during runtime through a discovery process. In the discovery process, devices broadcast packets advertising their availability, possibly containing information about neighboring nodes to which they can connect. Based on the received discovery packets, devices can dynamically build a routing table containing information about potential relay nodes along the multi-hop path to the destination.

In our system, static pairing is implemented to manage routing protocols in the fixed multi-hop topology because it is less complex than dynamic pairing. In addition, static pairing provides a more deterministic approach in terms of flexibility and performance.

In the context of synchronization, handshakes in ESP-NOW are a crucial part of establishing communication channels in multi-hop networks and are similar to a pre-conversation handshake to ensure the availability of data exchange [24]. These signals are initiated by the sender via specially configured control packets sent to the relay nodes. These packets contain the MAC addresses of the sender and receiver, along with optional sequence numbers for synchronization. Upon receiving the control packet, relay nodes acknowledge their availability, typically by sending specially configured control packets back to the sender, indicating their availability to receive and potentially forward data packets. Once acknowledged, a temporary communications channel is established between the sender and the first relay node, allowing efficient data transmission along the designated route. Handshakes not only prevent wasted transmissions by confirming relay node availability but also act as a troubleshooting tool, identifying potential communication path issues when acknowledgments are not received and prompting appropriate recovery actions to maintain network reliability and integrity.

4.2. The ESP-NOW API

A wireless communication protocol for low-power systems, ESP-NOW has been established for IoT implementation, and provides a streamlined API for establishing communication channels and transmitting data between devices. Table 1 shows the ESP-NOW functions commonly used for data transmission, which are found in the Espressif IoT Development Framework library API [23]. This API was utilized in our system to efficiently transmit worker presence data from a scanner node to a gateway node via multiple relay nodes in a multi-hop communication network.

Table 1. API functions for ESP-NOW.

Function	Description	Note
esp_now_init()	Initializes the ESP-NOW protocol.	Allocates resources and prepares the device for ESP-NOW communications.
getWiFiChannel()	Obtains the WiFi channel of a specific SSID.	Sets the channel before using ESP-NOW for multi-hop communications.
esp_now_add_peer()	Adds a peer device to the local list.	Establishes a communication channel.
esp_now_register_send_cb()	Registers a callback for data sent.	Invoked upon successful transfer or transmission error.
esp_now_register_recv_cb()	Registers a callback for data received.	Triggered when data are received from a peer.
esp_now_send()	Sends data to a peer device.	Sets blocking or non-blocking send mode.

5. Worker Presence Monitoring System Architecture

The proposed worker presence monitoring system in a complex workplace is based on a three-layer IoT architecture [16], which uses perception, network, and application layers, as illustrated in Figure 2. These layers integrate the BLE beacon with multi-hop IoT networks to display the presence of the worker information via the Node-RED server, which can be accessed remotely from anywhere across the world.

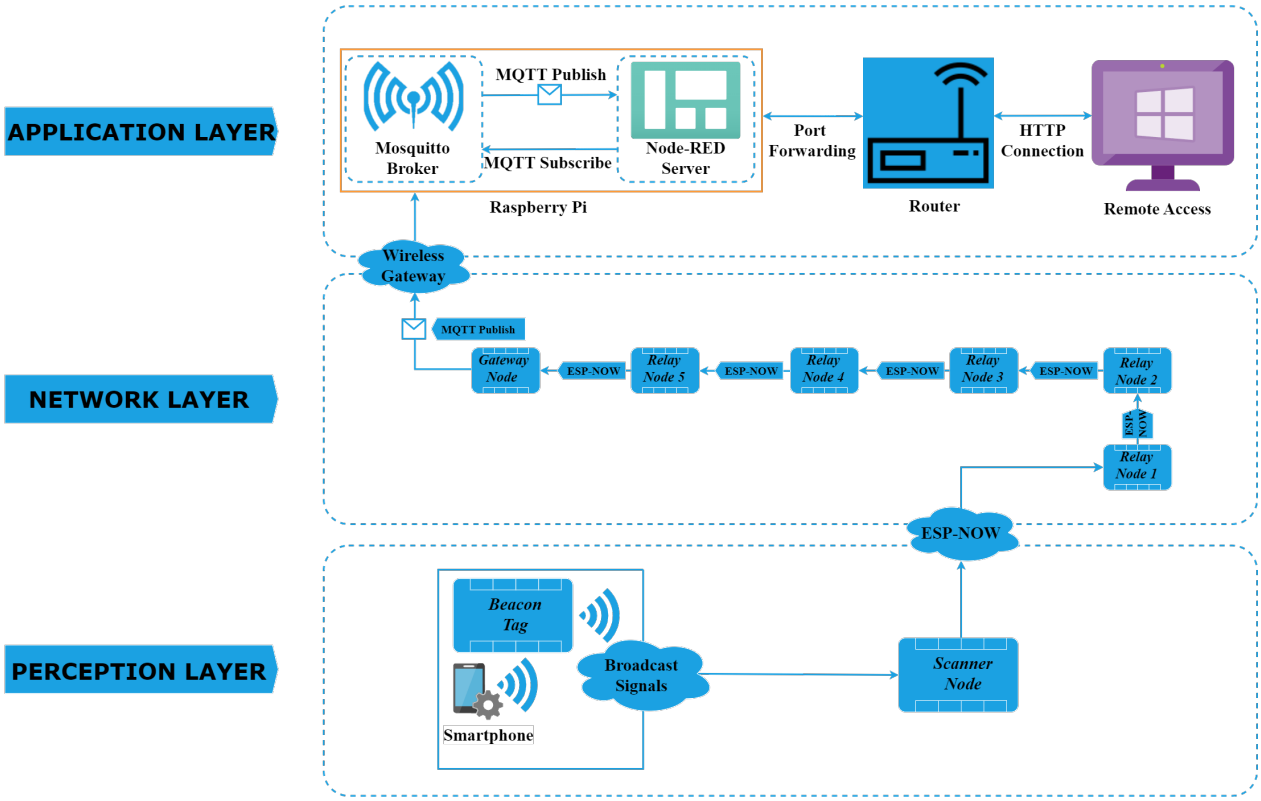


Figure 2. The system is a multi-hop IoT network integrating BLE beacon technology.

The perception layer, which can be considered the foundation layer in terms of our system, plays a complex role by utilizing a microcontroller referred to as a node. For example, during the construction of a ship, workers carry a node configured with BLE beacon

technology attached to their helmets, which can be considered a beacon tag, and an additional option is to use the nRF Connect application on their smartphone. Both continuously broadcast worker presence-based signals to their surroundings, which will be detected by a scanner node placed in a fixed position around the workers. This scanner node is connected to a relay node.

The network layer enables communication between the perception layer and the application layer. By adding multiple relay nodes strategically placed in different locations, the relays extend coverage across vastly larger areas of a workplace to reach the final destination without relying on an Internet connection. In addition, the last node of the relays is connected to a gateway node. This gateway node is in a fixed position at the final destination, called the server room. However, the scanner, multiple relays, and the gateway node collectively form a multi-hop IoT network using the ESP-NOW communication protocol. Moreover, the network layer utilizes the MQTT communication protocol, a lightweight messaging protocol ideal for resource-constrained IoT devices [25,26]. MQTT works as a publish-and-subscribe system [27], bridging the network and application layers on a given topic. Accordingly, the gateway node uses a wireless gateway through a dedicated WiFi router to publish the workers' presence data (processed by the BLE beacon and the nRF Connect application via smartphone) using MQTT connecting to a central platform called Mosquitto Broker.

The application layer includes Mosquitto Broker and the Node-RED server, both installed on top of Raspberry Pi. Node-RED is a visual programming system platform that facilitates the development of data processing and visualization operations [28,29]. Mosquitto Broker is a platform that allows the MQTT protocol to be implemented. It also acts as a central message hub, often referred to as a Mosquitto MQTT message broker [30]. It does not process or interpret data but efficiently forwards messages to subscribers based on the topic. Accordingly, the Node-RED server subscribes to a specific topic on Mosquitto Broker, and uses MQTT to receive worker estimation data. The Node-RED server interface is then used to visualize the data on a dashboard. In addition, to allow remote access from anywhere in the world via the Internet, a port-forwarding method is configured on the WiFi router located in the server room. This redirects incoming traffic on a specific port to the server's internal IP address, allowing users to remotely view the worker presence data on the Node-RED server's dashboard. This three-layer architecture facilitates a real-time worker presence monitoring system in complex workplaces with the flexibility of remote accessibility.

6. System Implementation

6.1. The Perception Layer

6.1.1. Beacon Tags

In the first phase of the perception layer, a microcontroller is used as a BLE beacon tag. This is performed by pre-programming ESP32 devices. This was accomplished by writing code in C++ on the Arduino IDE platform (version 1.8.19) and then uploading it to the ESP32 microcontroller. The device was manufactured by Espressif Systems, a semiconductor-based company in Shanghai, China, known for developing advanced wireless communication and microcontroller solutions. This microcontroller was chosen because it integrates BLE, allowing the user to transmit a low-power beacon to be detected. The ESP32 chip with the ESP32-WROOM-32 module, shown in Figure 3, has 4 MB of on-board SPI flash memory and is powered by the Xtensa dual-core 32-bit LX6 microprocessor that operates from 80 MHz to 240 MHz [31].

Part of configuring the BLE beacon technology is defining the identity of our beacon as Raihan_ESP32, establishing its unique presence in the digital realm. This identity is represented in the code by the string DEVICE_NAME. In addition, we assign a 16-byte UUID (4fafc201-1fb5-459e-8fcc-c5c9c331914b) [32] that allows a BLE-compatible receiver to identify itself to the beacons. This UUID is predefined and generated by the Bluetooth SIG. There is also CHARACTERISTIC_UUID in the functions, which acts as a unique fingerprint

for a specific piece of data advertised by the BLE beacon. It is a 16-byte value that allows connected devices to identify the exact data point they want to interact with. Once these key elements are in place, BLE initializes the beacon, essentially powering it up for operation. After initialization, the beacon is ready to broadcast its presence to other nearby devices. This broadcasting can be compared to a lighthouse that announces its existence. Within the operational loop, the beacon continuously broadcasts relevant data. This loop ensures the beacon's continued functionality and enables it to fulfill its purpose effectively. Ultimately, the BLE beacon acts as a silent messenger, seamlessly communicating its presence and relevant information to nearby devices.

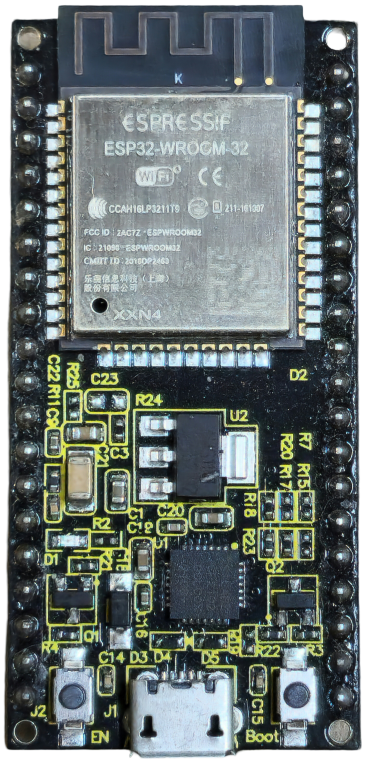


Figure 3. The ESP32 chip with the ESP32-WROOM-32 module configured as a beacon tag.

Furthermore, we incorporate nRF Connect, which a smartphone uses to broadcast a presence-based signal similar to a beacon tag [33]. This powerful tool, developed by Nordic Semiconductor, leverages BLE to interact with other BLE-enabled devices [34]. Through its simple interface, nRF Connect allows users to scan for nearby BLE devices, connect to them, and interact with their services and features. How nRF Connect advertises its presence to nearby devices is typically conducted through advertisement packets. When a BLE-powered device, such as a smartphone using nRF Connect, is in advertising mode, it periodically broadcasts small packets of data (the advertisement packets) to nearby devices. These advertisement packets contain information about the device, such as its name (e.g., Raihan_Smartphone) and other relevant data. The content of these packets is defined by the smartphone's configuration, shown in Figure 4. Advertisement packets are transmitted on specific advertising channels defined by the BLE standard. Nearby BLE-enabled devices can detect these advertisement packets during scanning operations. Once detected, and based on the information contained in the advertisement packets, these devices can initiate a connection to perform other actions.

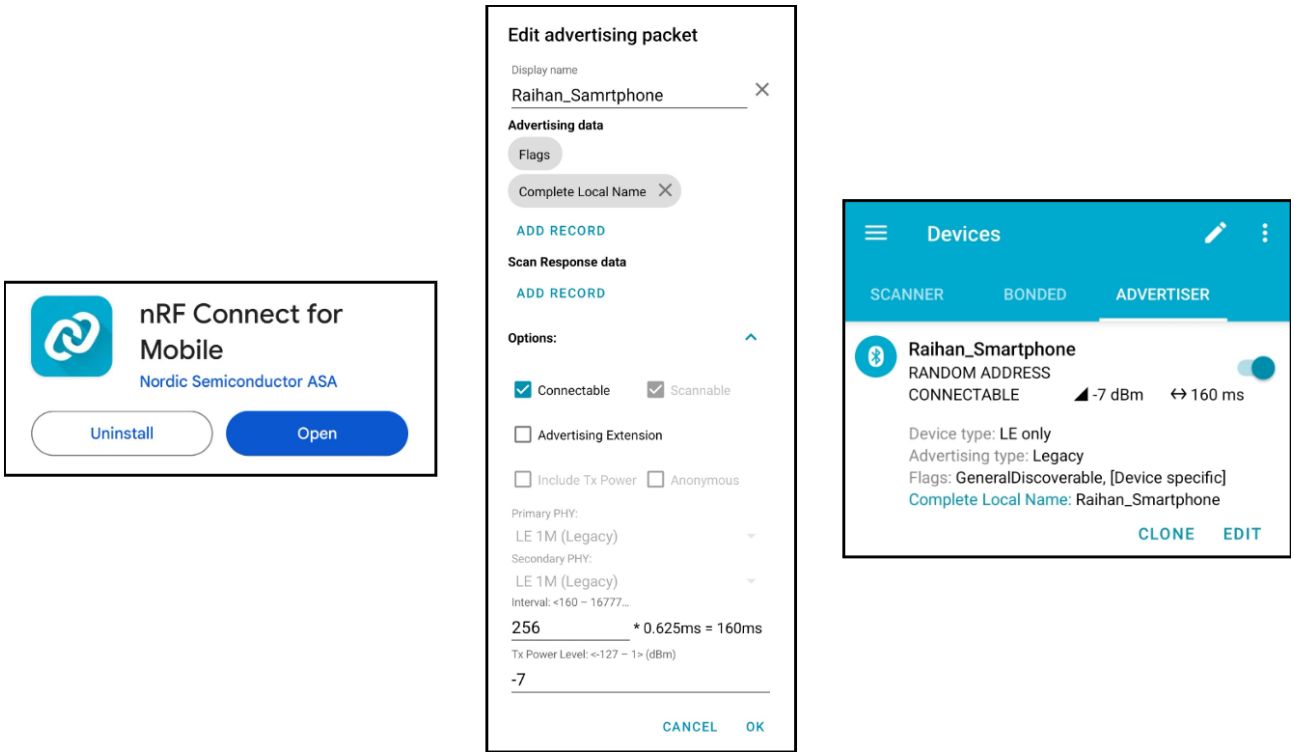


Figure 4. Set up a configuration on nRF Connect to broadcast BLE signals via smartphone.

6.1.2. Scanner Node

In the second phase of the perception layer, a microcontroller is utilized to simultaneously communicate over long distances and through complex obstacles. For this purpose, we used another module of the ESP32 chip, ESP32-WROOM-32UE [35], shown in Figure 5, owing to its external antenna connector. The external antenna is a 5 dBi dipole antenna using the U.FL connector attached to the module, which operates in the 2.4 GHz frequency band. Compared to the built-in chip antenna on the regular module of the ESP32 chip, this external antenna can provide a more extended range. The ESP32 is designed for low-power IoT applications and a wide range of other applications.

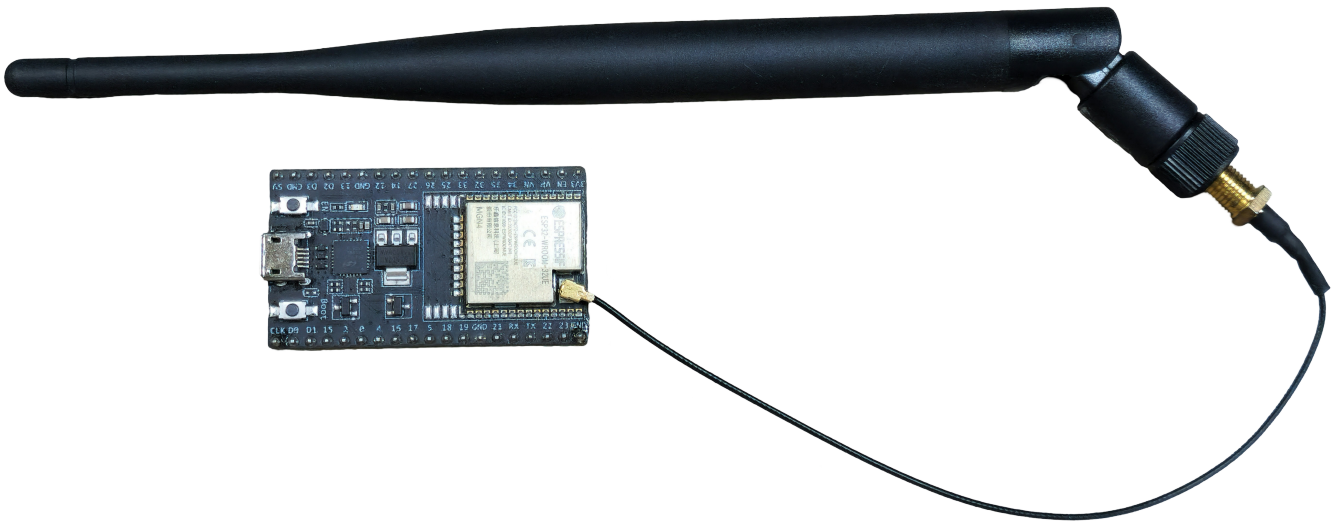


Figure 5. The ESP32 chip with the ESP32-WROOM-32UE module configured as a scanner node.

In the first step, based on its function as a scanner node, the module scans for specific BLE beacon tags by matching predefined target names in the broadcast signals. Upon detection of these devices, it constructs a message containing relevant information based on the pre-configured names of beacon tags, the MAC addresses of the BLE devices, the RSSI values, and the UUIDs. In the second step, it transmits the constructed data to a relay node utilizing the ESP-NOW protocol, as shown in Figure 6.

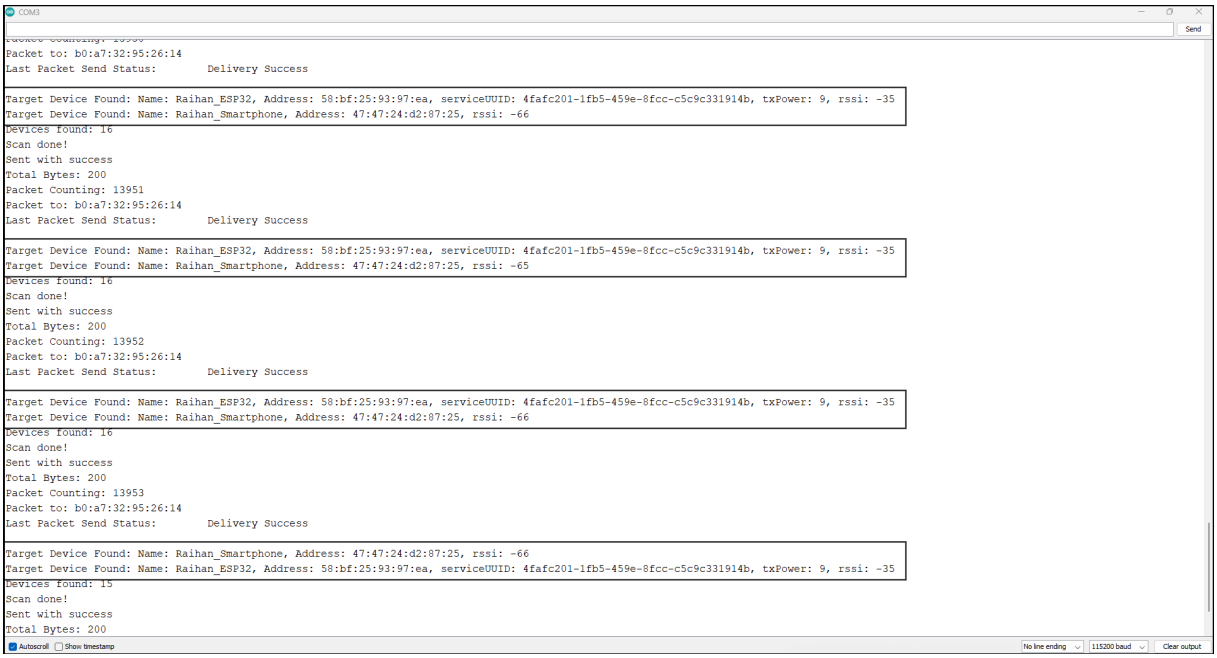


Figure 6. Scanning for broadcast signals from beacon tags and transmitting from the scanner node to the relay node, which is captured by the serial monitor of the Arduino IDE.

The scanner node initializes the necessary libraries for ESP-NOW with BLE, defines a structure for storing the received data from the beacon tags, and sets up WiFi in station mode to adjust the WiFi channel settings accordingly. It then creates a function to scan for BLE signals, and sets up another function to be called when a signal is discovered based on the target list of predefined beacon tags. This function checks whether the name of a detected beacon matches one of the names in the target list. If a matching name is found, the scanner node retrieves the beacon's name and assigns it a unique number. This number is used to keep track of the number of packets successfully sent to the relay node in the network. At the same time, the OnDataSent function is used as a callback to acknowledge the data sent via the ESP-NOW protocol. It is crucial to know the status of data transmission in a network using communication protocols via ESP-NOW to ensure reliability. In this case, the callback function can respond to data transmissions in real-time. Afterward, the setup function initializes ESP-NOW, registers the callback function to handle the data transmission status, and registers a peer in the network to transmit data according to the predefined MAC address of the relay node to establish a reliable network.

Finally, in a loop operation, the scanner node repeatedly scans for beacon tags specified as targets, as seen in more detail in Figure 7. It continuously clears the memory for received data and then starts a BLE scan for each targeted beacon tag. Upon detecting each target, it prepares and sends a message via ESP-NOW to a relay node. If successful, it records details such as data size and packet count; otherwise, it logs an error.

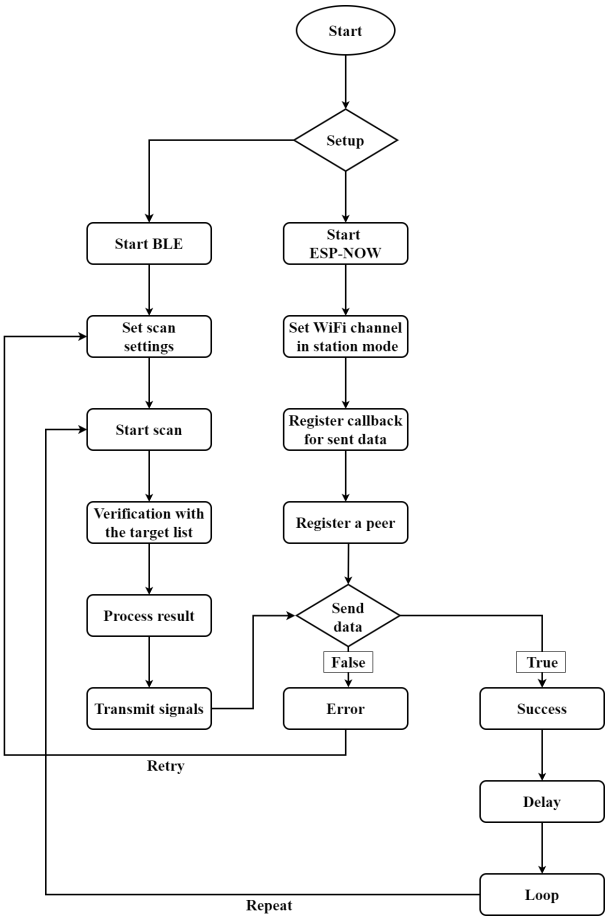


Figure 7. A flowchart for scanning broadcast signals and transmitting data to a relay node.

6.2. The Network Layer

6.2.1. Relay Nodes

In the first step of the network layer, a relay node utilizes ESP-NOW to facilitate communication between scanner and gateway node to establish a robust multi-hop IoT network. In this phase, we also use the ESP32-WROOM-32UE module of the ESP32 chip for both relay and gateway nodes. At initialization, the necessary libraries are included along with a defined structure for storing data from the scanner node, and this structure must be the same as that of the scanner node. In ESP-NOW communications, both the scanner and the relay or gateway node must agree on the structure of the data to be transmitted. This ensures the data are correctly interpreted at both ends of the communications link. The WiFi setup next configures the node into station mode to set the WiFi channel like the scanner node. Subsequently, ESP-NOW is initialized, and callbacks are defined to handle data transmission events. In particular, the OnDataRecv function processes incoming data and extracts information such as the MAC address of the scanner node as well as the details of the received data. On the other hand, the OnDataSent function is a callback to the scanner node in response to the data sent via the ESP-NOW mechanism. In the setup configuration, the relay node prepares for ESP-NOW communication by registering both data-received and data-sent callback functions to handle the transmission status, as well as registering a peer in the network to transmit data according to the predefined MAC address of the next relay node or the destination node.

In the main loop, the relay node waits for data from the scanner node using ESP-NOW. Once received, the data are copied into a structure and are logged for debugging. The received data are then forwarded to the designated relay or gateway node, as shown in Figure 8. Finally, the data structure is cleared to prevent the same information from being sent again, and the loop repeats, ready to receive and forward new data packets.

However, with this process for relay nodes, we have designed a multi-hop IoT network by adding multiple relay nodes that can be placed as needed to reach the destination based on the environment. In a nutshell, a relay node plays a vital role in a multi-hop scenario by extending its range to transmit data over long distances.

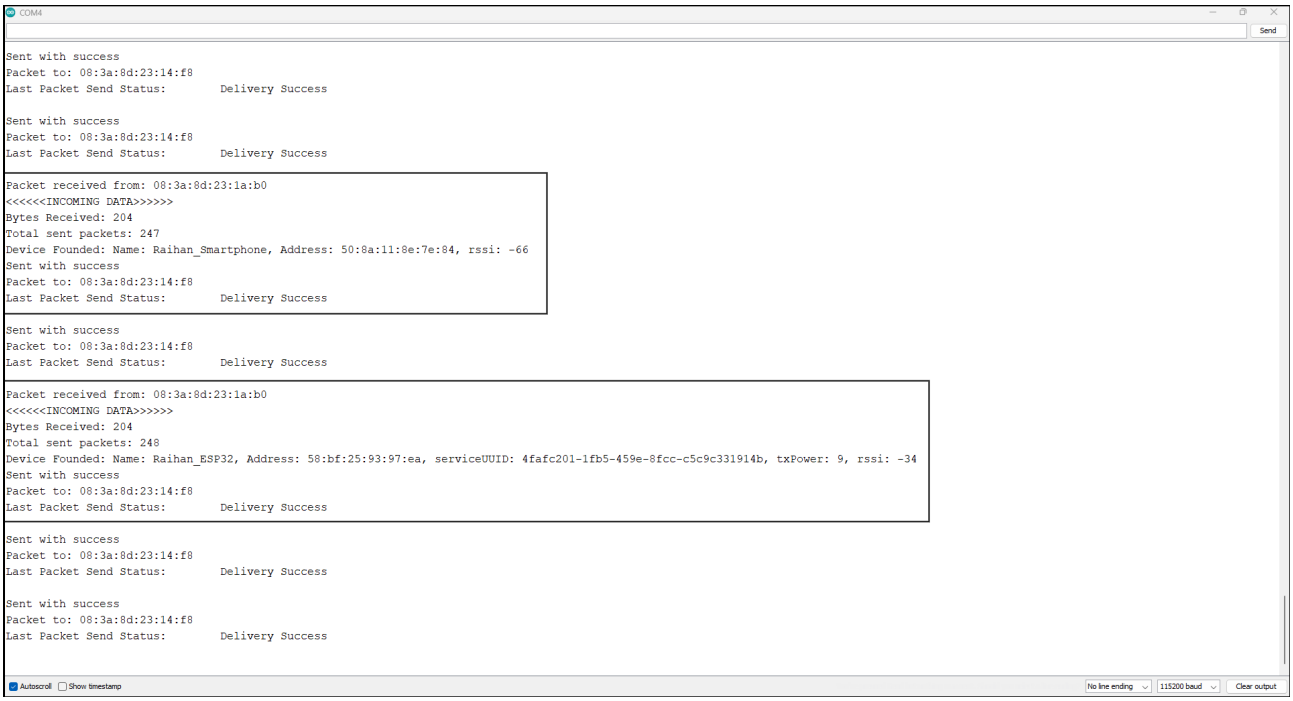


Figure 8. The relay node forwards data to the gateway node upon receiving them from the sender node.

6.2.2. Gateway Node

In the second step of the network layer, the gateway node is designed to receive data from a relay node where the data travel from the scanner node through multiple relay nodes to reach the destination. This gateway node multitasks by utilizing ESP-NOW to receive data and by connecting to a dedicated WiFi network to communicate with Mosquitto Broker to forward the data using the MQTT communication protocol, as shown in Figure 9. In this case, the gateway node plays a significant role in setting up the multi-hop IoT network.

The MQTT protocol facilitates data transmission between the gateway, which acts as an edge node, and the central server, Node-RED, via the Mosquitto Broker using a publish-subscribe architecture. In this system, the gateway node serves as the sole publisher, sending worker presence data to the Mosquitto Broker on a specific MQTT topic (beacon/signal/data). The Mosquitto Broker then forwards this data to the Node-RED server, acting as a client and the sole subscriber to that topic. The Node-RED server retrieves and visualizes the data, enabling real-time monitoring of worker presence.

Based on the algorithm, the gateway node first defines constant variables for the WiFi network credentials, along with the IP address and port of the Mosquitto Broker for communication via MQTT. It then creates a class called PubSubClient to handle MQTT communications, along with a long variable to track the time of the last message sent. To publish beacon signal data on a topic, a constant is defined for the MQTT topic (beacon/signal/data), and a structure is declared to store the data, as was performed for the relay and scanner nodes. The OnDataRecv function is the callback to respond to incoming data received via ESP-NOW, just like the relay node. In the setup function, the gateway node uses the WIFI_AP_STA function to set the WiFi to both station and access point mode. This function plays a crucial role in receiving the data from the relay node and simultaneously forwarding it to Mosquitto Broker over the wireless network. Using this function,

the gateway node automatically assigns a WiFi channel by connecting the WiFi router in access point mode. On the other hand, the scanner and all the relay nodes must use the same WiFi channel as the gateway node, which is performed by using the getWiFiChannel function to keep them only in station mode (WIFI_STA). In order to connect to a wireless network, the setup_wifi function connects to the WiFi network specified by WIFI_SSID and WIFI_PASSWORD, which repeatedly tries to connect until successful.

Although there is no multi-client scenario in this setup, MQTT ensures efficient data transmission between the gateway node and the Node-RED server, even in high-traffic environments. The lightweight nature of MQTT, along with its quality of service (QoS) levels, guarantees reliable message delivery. In heavy traffic or complex environments, QoS 1 or 2 can be used to ensure messages are delivered reliably to the server, even in the event of network interruptions. Topic-based filtering ensures that only subscribed data is delivered to the Node-RED server, maintaining smooth performance. Additionally, features like retained messages ensure that data is not lost, even if the connection between the Node-RED server and Mosquitto Broker is temporarily interrupted.

Furthermore, a reconnect function establishes MQTT communications by connecting over the wireless network to the specified IP address and port of Mosquitto Broker. The function of MQTT communications is to continuously attempt to reconnect until a connection is established. Finally, the loop function continuously checks the MQTT connection and attempts to re-establish it if needed. It then checks a timer and publishes the data to a specific predefined MQTT topic every two seconds. Thus, this system establishes a robust multi-hop IoT network.

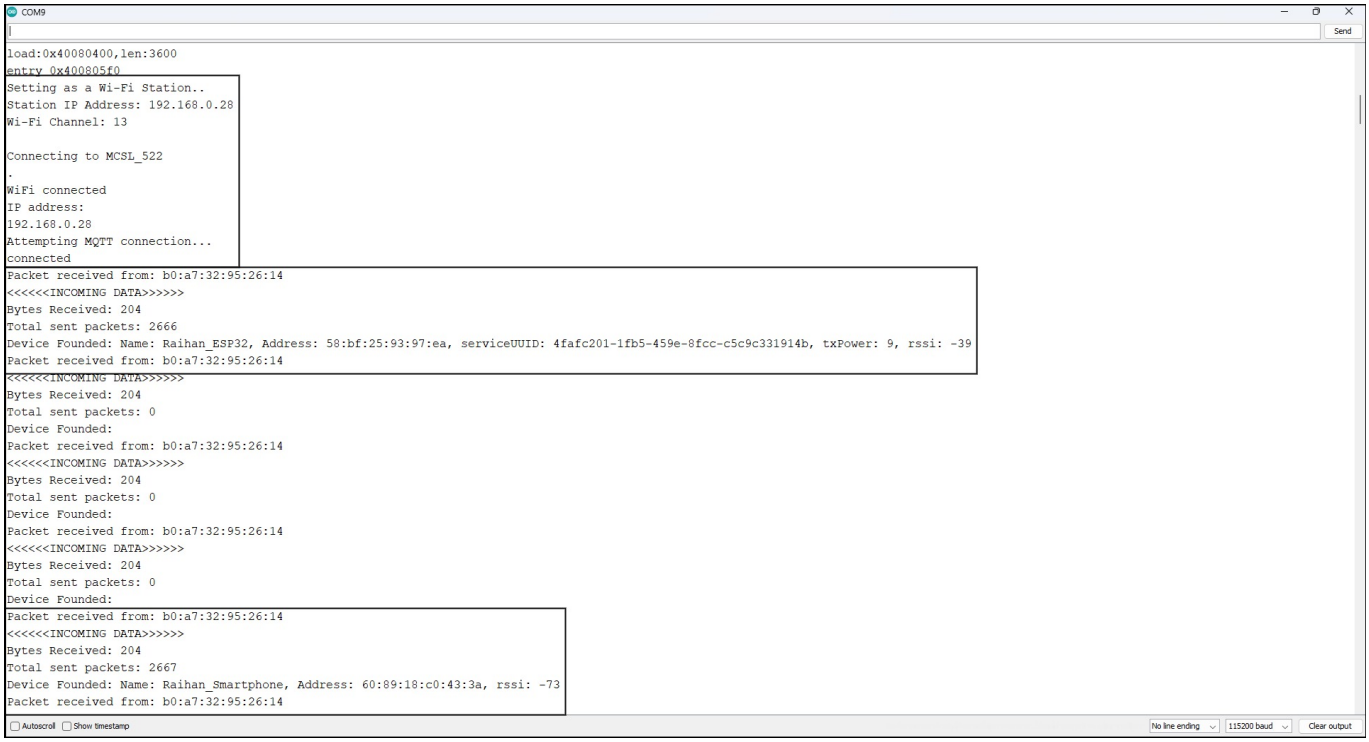


Figure 9. The gateway node forwards received data from the relay node to Mosquitto Broker by using MQTT communications via the wireless gateway.

6.3. The Application Layer

In order to leverage remote access to monitor worker presence in complex workplaces, a Raspberry Pi 4 (Model B) device is utilized by incorporating several processes in the application layer. The first step installs Mosquitto Broker on the Raspberry Pi to enable MQTT communications. The MQTT protocol initiates a connection to Mosquitto Broker on its default port (1883) and IP address, which are based on a predefined constant variable

in the gateway node's algorithm. The IP address refers to the actual IP address for the Raspberry Pi because Mosquitto Broker runs on it. Following this process, the gateway node publishes the data to the predefined MQTT topic (beacon/signal/data) on Mosquitto Broker by incorporating the MQTT protocol via the wireless gateway.

The second step installs the Node-RED tool on the Raspberry Pi, which provides a browser-based interface by setting it up on the server. Once the Node-RED server is successfully set up, it can be accessed under the local WiFi network using the browser at http://192.168.0.8:1880. The URL opens a canvas that allows users to design and manage flows using a palette of nodes that represent different functionalities. Nodes can be connected to construct flows that represent a series of processes by dragging and dropping them from the palette onto a canvas. After this, we drag a node called *mqtt in* from the palette onto the canvas to receive the data by subscribing to the MQTT topic on Mosquitto broker. Because Node-RED supports a wide range of protocols and technologies (MQTT, HTTP, and more), we configure the node using the MQTT protocol. In order to configure that node, we select port 1883, where Mosquitto Broker is running locally on the server, by subscribing to the same MQTT topic (beacon/signal/data) because the gateway node publishes the data on that specific topic. Next, we select the string format to receive the data, since the gateway node also publishes the data in that specific format. Next, we drag and drop another node called *text* from the palette to transform and visualize the data on the dashboard in real-time when the data are received from the gateway node via Mosquitto Broker using the MQTT protocol. We also drag and drop a node called *notification* to let the user know during the delay that data are being processed. The dashboard in the user interface is shown in Figure 10. Thus, we configure the Node-RED server by deploying the flow where the *text* and *notification* nodes are connected to the node *mqtt in* on the canvas. However, after deploying the server, the dashboard can be accessed through the local network at http://192.168.0.8:1880/ui.

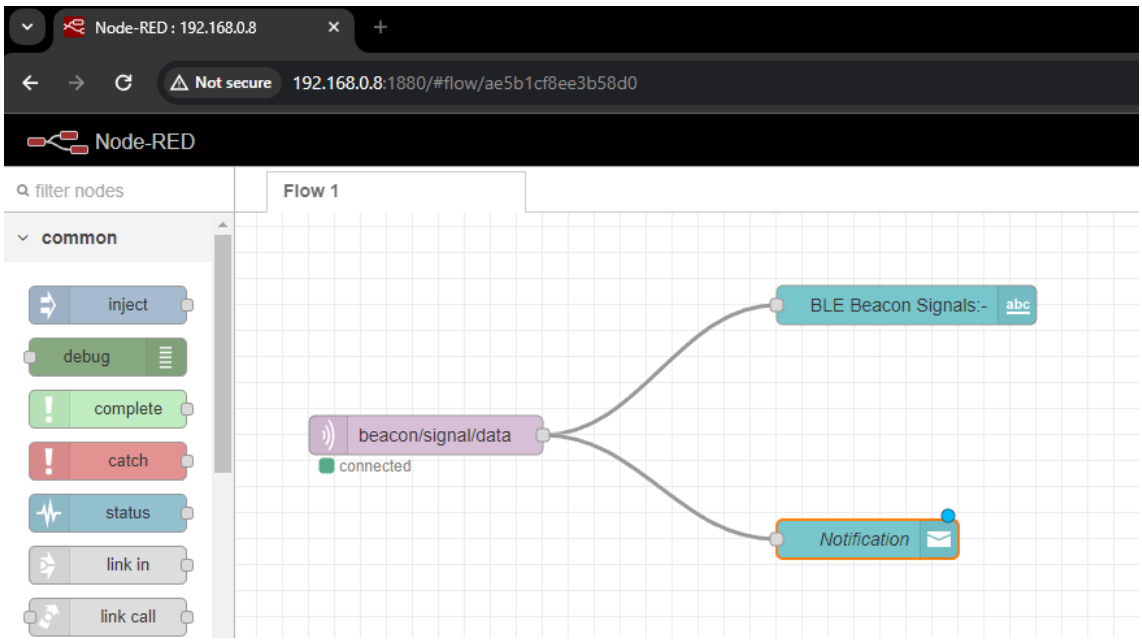


Figure 10. Configuration of the Node-RED server, where nodes are connected to each other on the canvas.

The third step is to access the server remotely so that the user can monitor the workplace through a different network from anywhere on the Internet. In order to do this, we first place the Raspberry Pi server in a specific location (the server room), connecting to the Internet via the WiFi router. To access the server remotely, we use a static public IP address to establish an HTTP connection, registering it in the Domain Name

System, as shown in Figure 11, through a reliable provider. When a user tries to access the server remotely from a different network by entering the URL in a browser, such as http://mcsllab.onthewifi.com:33891/ui (accessed on 18 July 2024), the browser first sends a DNS request to the DNS hosting provider for the domain name mcsllab.onthewifi.com to an IP address. The DNS provider responds to this browser by performing a DNS resolution, allowing the browser to initiate the HTTP connection. The browser sends an HTTP request to the server, where the router receives the incoming HTTP request on port 33891 by establishing a Transmission Control Protocol connection. The router forwards the HTTP request to the server's local addresses using the port forwarding method to the server port at 1880. The server then processes, prepares, and sends the HTTP response back to the user's browser via the established TCP connection through the router.

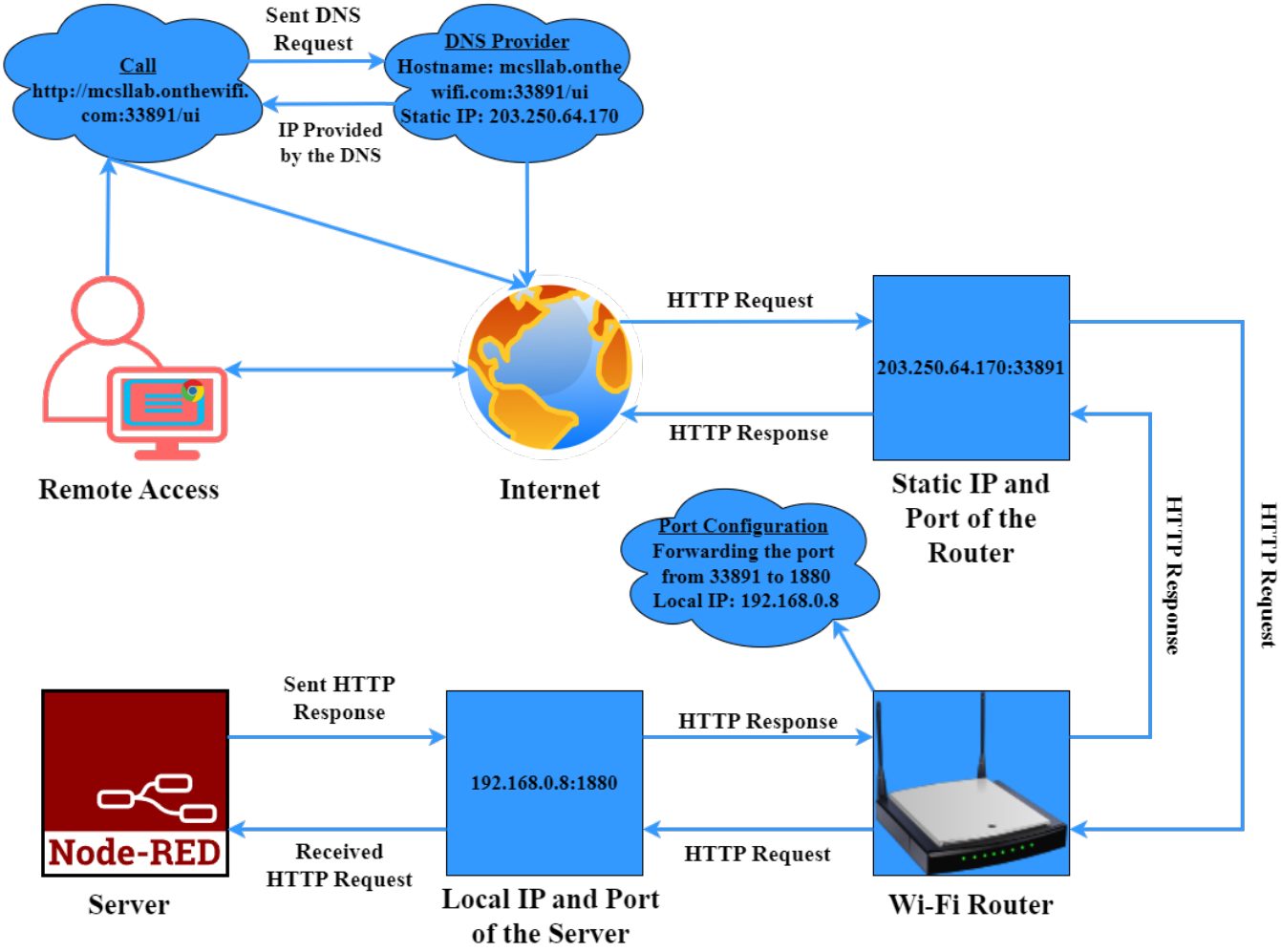


Figure 11. Accessing the server remotely from anywhere on the Internet.

Users can access the server from anywhere via the Internet. The dashboard of the Node-RED server displays the visualized worker presence data at two-second intervals. Users can monitor the workplace by accessing the server through the interface shown in Figure 12.

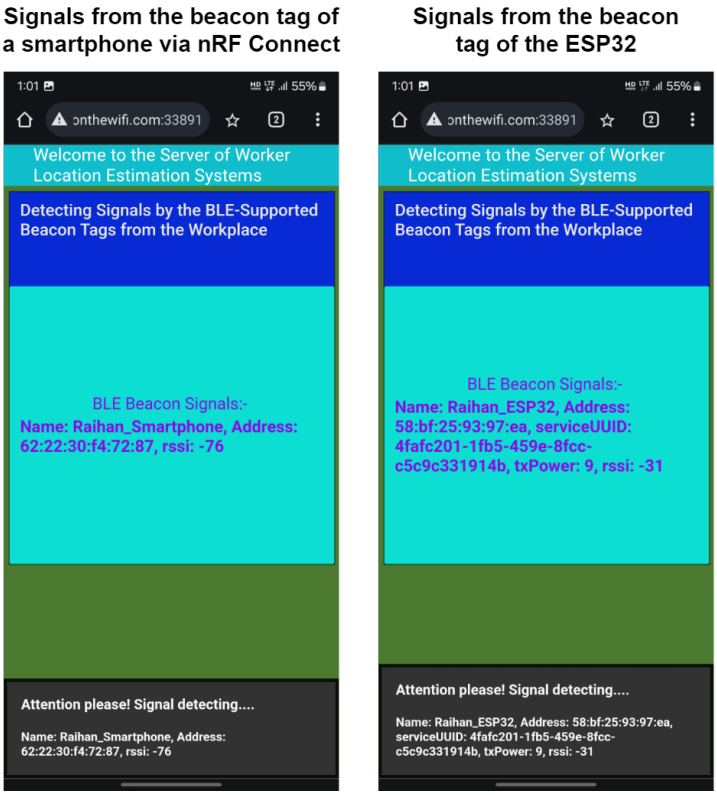


Figure 12. The user interface of the Node-RED server displays visualized worker presence data from beacon tags in a workplace.

7. Experiments

This section experiments with the proposed worker presence monitoring system in a real workplace environment. In order to verify network performance, we examined latency and lost data packets by deploying beacon tags, a scanner node, multiple relay nodes, and a gateway node in the Department of Electrical, Electronic and Computer Engineering at the University of Ulsan. The entire experimental environment was filled with several obstacles. We deployed the scanner node in a fixed position along with two beacons at an optimal distance around the scanner node to broadcast presence-based signals. We placed them in Room 302, which is part of the MCSL laboratory and a good example of a workplace. After that, we deployed five relay nodes to transmit data over long distances but without Internet connectivity. The first relay node was deployed outside Room 302, with the rest of the relay nodes deployed towards the last node, which was deployed outside Room 522. This last relay node was connected to the gateway node in Room 522, which was connected to the server.

In order to measure the distances for connection ranges among the nodes in our proposed multi-hop network, we utilized the Emesent Hovermap, which uses light detection and ranging (LiDAR) to create high-resolution 3D maps of the surroundings [36]. We used the Hovermap as a handheld device moving through the building; in this case, the LiDAR sensor emitted laser pulses and captured the return signals. The simultaneous localization and mapping algorithm employs these signals to create a detailed 3D point cloud map of the building. Each deployed node can be identified and marked on the 3D map during the mapping process. Once the 3D map is complete, specialized Emesent software called CloudCompare (version 2.6.3 beta) can calculate the distances among the nodes. The 3D point cloud data included precise spatial coordinates for each point, allowing the software to determine the distances shown in Figure 13. The 3D map shows the distances between two nodes, where each node is

connected to its neighbor node in sequence (scanner node to relay node, relay node to another relay, and so on).

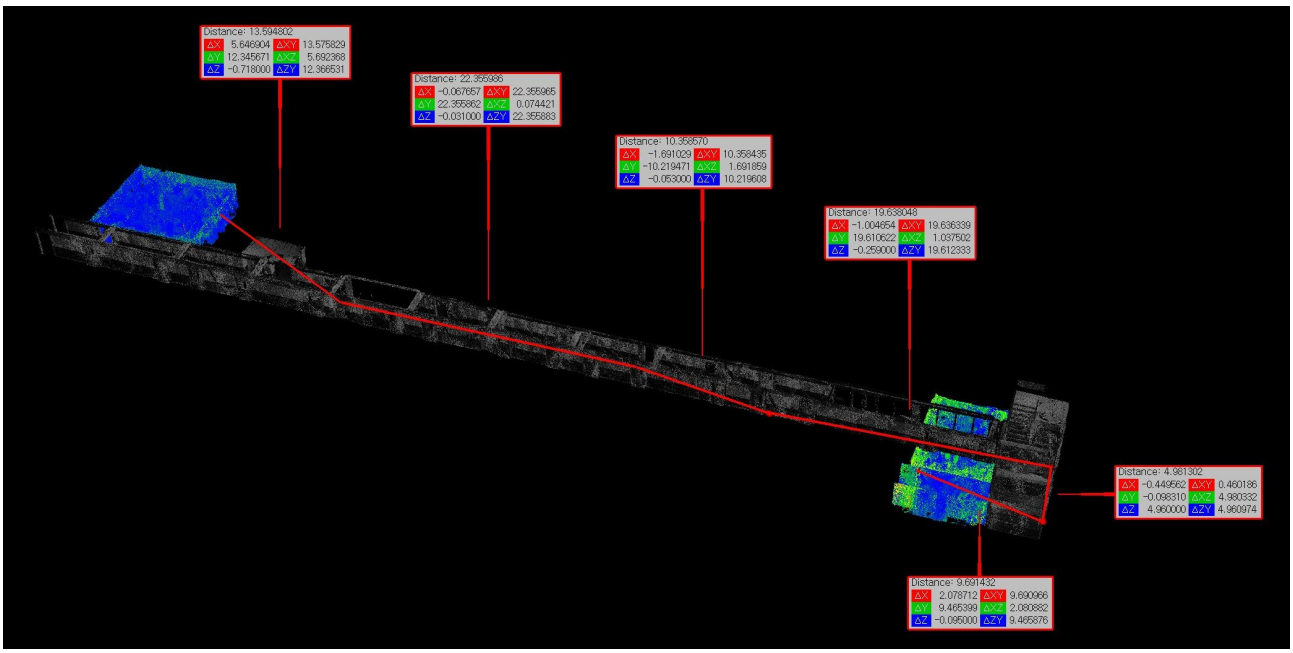


Figure 13. Distances generated by the Emesent Hovermap, in meters, among the deployed nodes of the multi-hop network.

The distances between nodes varied significantly due to obstacles such as walls, machinery, and other structural elements commonly found in complex workplace settings. These obstacles disrupt direct signal paths, causing multi-path propagation, where radio signals reflect off surfaces and take multiple routes to reach the receiver, leading to interference and signal fading. Materials like metal and dense concrete further worsen these issues by reflecting, absorbing, or blocking signals, which weakens the communication link and effectively increases the transmission distance needed for reliable connectivity. These factors cause significant fluctuations in signal strength and quality, contributing to higher packet loss rates, increased error rates, and overall unreliable communication, which are critical challenges in environments where timely and accurate data transmission is essential. To address these challenges, the system employs a structured and deterministic multi-hop communication approach with strategically placed relay nodes. Each relay node is carefully positioned to reduce the distance that each signal must travel, breaking the communication path into shorter, more manageable segments. This configuration optimizes signal propagation by ensuring signals take the most direct and least obstructed paths between nodes, enhancing signal strength and minimizing interference from environmental factors. By using predefined communication paths and fixed MAC address routing, the network avoids the complexity and variability of dynamic routing, which can introduce delays and inconsistencies. Fixed routing ensures that each node follows a clear, predetermined path, resulting in more predictable and stable performance. This strategic placement of relay nodes not only mitigates the adverse effects of environmental obstacles but also continuously boosts and forwards the signal at each hop, ensuring reliable data transmission with reduced packet loss. As a result, the network maintains high performance even in complex and obstacle-rich environments, significantly enhancing overall network efficiency, reliability, and effectiveness in demanding workplace settings.

Figure 14 illustrates the deployment of the scanner node and beacons in Room 302. In our complex indoor environment filled with obstacles, the effective reception range of the BLE signals was approximately 20-30 m. The scanner node was placed in a fixed position, while the beacon tags were optimally positioned to ensure accurate data reception based on

the workplace layout. It is important to note that, depending on the specific environment and the nature of the obstacles present, the ranges of reception distances can increase or decrease accordingly. This setup was crucial for maintaining reliable communication within the experimental environment, despite the challenges posed by physical obstructions.

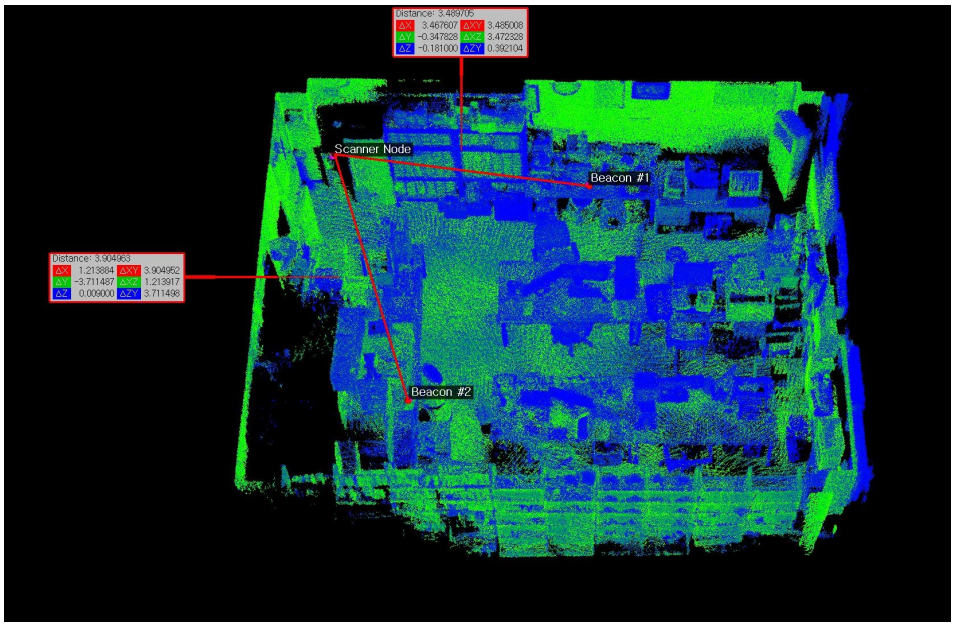


Figure 14. This 3D map shows the deployment of the scanner node and beacons in our complex workplace.

To ensure the accuracy of both latency and packet loss measurements, we performed experiments simultaneously while assessing the connection range among nodes in our proposed multi-hop IoT network. The precise measurement of latency and packet loss posed significant challenges; thus, we leveraged the Arduino IDE (version 1.8.19) as our primary software tool. In our algorithm for a multi-hop IoT network, the statement sentPacketCount++ is employed to increment the count of sent data packets each time a packet is delivered from the scanner node to the gateway node through the five relay nodes. This incrementation is crucial for tracking successful data transmissions monitored via the Arduino IDE serial monitor, which includes a timestamp feature that displays time in hours, minutes, seconds, and milliseconds, providing precise temporal data for each packet sent. When a data packet is dispatched from the scanner node, the serial monitor records and displays the packet's transmission time while incrementing the packet count. Upon receipt of this packet at the gateway node, which has traversed five relay nodes, the serial monitor at the gateway also logs the packet number and the exact time. For our experiments, we utilized two laptops: one associated with the scanner node and the other with the gateway node. Each laptop captured and stored the transmitted data. By comparing the timestamps between the gateway node and the scanner node, we calculated the latency for each data packet. Additionally, we monitored the sequence of packet numbers to determine packet loss. If the gateway node received the packet successfully, it logged the same packet number sent by the scanner node. However, if the sequence number was lost, or if the gateway node printed a different number, this indicated packet loss. By tracking these discrepancies, we were able to accurately quantify the packet loss.

8. Results Analysis

In this section, we present the latency and packet loss results that we obtained previously from the experimental methodology. Figure 15 shows the latency for each of 100 total data packets successfully transmitted from the scanner node and received by the gateway node via the relay nodes. The collected latency data revealed a minimum la-

tency of nine milliseconds and a maximum latency of 178 ms, with an average latency of 120 ms. To provide a clear visualization, the bar chart categorizes the latencies using three different colors: green for low latency, yellow for medium latency, and red for high latency. This color-coded representation helps in quickly identifying the distribution of latency values across the data set.

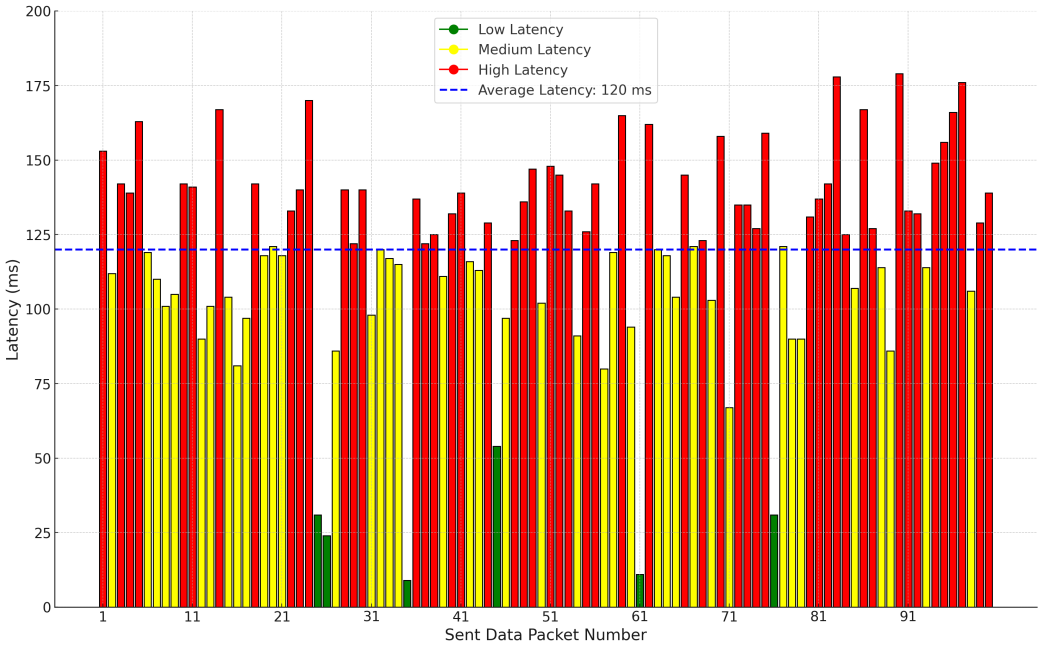


Figure 15. Latency measurements for 100 data packets sent in the multi-hop IoT network.

Moreover, we analyzed the latency for data transmissions in the multi-hop IoT system by using different relay node configurations. Figure 16 provides a detailed comparison of the minimum, average, and maximum latencies observed in the same environment but with data transmissions through five, four, and three relay nodes. The bars are again color-coded for clarity, with yellow representing minimum latency, green representing average latency, and orange representing maximum latency. The data reveal several important trends. For the configuration with five relay nodes, the system achieved the shortest latencies: a minimum of 0.009 s, an average of 0.120 s, and a maximum of 0.178 s. This demonstrates the efficiency of the multi-hop setup in reducing latency as the number of relay nodes increases. In contrast, when the system utilized four relay nodes, the minimum latency increased significantly to 2.209 s, with an average latency of 2.275 s and a maximum latency of 2.395 s. This substantial increase in latency indicates a noticeable degradation in performance with a reduction of one relay node. The trend continued with the three-node configuration, where the minimum latency further increased to 3.118 s, the average latency to 3.180 s, and the maximum latency to 3.379 s. This configuration underscores the impact of fewer relay nodes on the overall transmission delay. As a result, adding more relay nodes to a network significantly improves signal propagation and reduces interference by shortening the distance each signal must travel between nodes [37]. This shorter distance maintains stronger signal strength and quality, minimizing the chances of signal degradation and interference, which in turn ensures more reliable and faster data transmission. Additionally, shorter distances reduce the errors and the need for retransmissions, thereby enhancing overall latency. Furthermore, distribution of the load across more nodes prevents any single node from becoming a bottleneck, ensuring balanced traffic flow, and avoiding overloads. This balanced load distribution helps reduce delays and maintains efficient network performance. The graph clearly illustrates that the addition of relay nodes can significantly enhance the efficiency of data transmission in a multi-hop IoT system by minimizing latency.

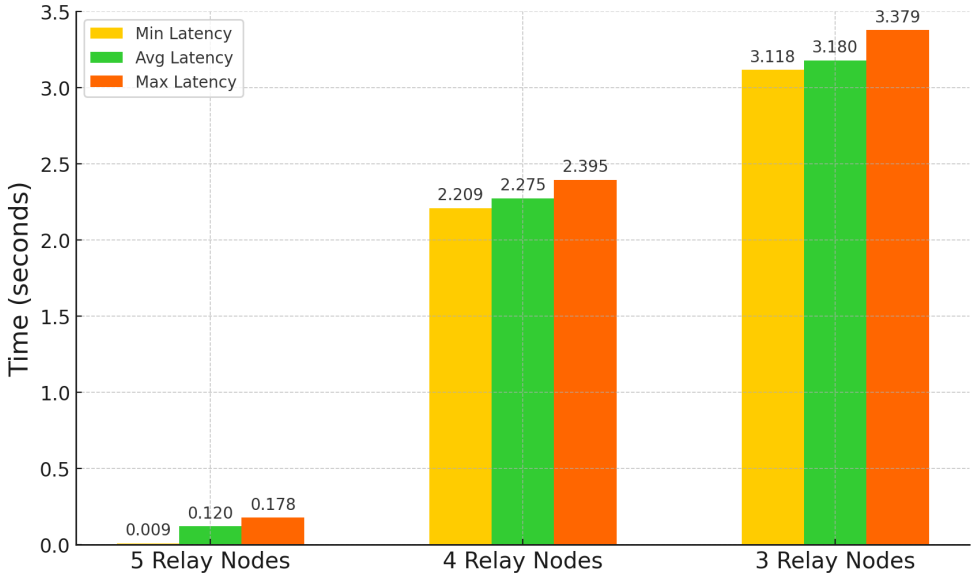


Figure 16. Latency in the multi-hop IoT system when varying the number of relay nodes.

For the packet loss test, we evaluated packet loss in the multi-hop IoT system by again using different relay node configurations, as shown in Figure 17. Specifically, we sent 500 data packets for each configuration of five, four, and three relay nodes and measured the packet loss. The line graph provides a clear comparison of packet loss across these configurations. The data show there was no packet loss when using five relay nodes, indicating optimal performance with this configuration. However, when the number of relay nodes was reduced to four, packet loss increased to 15 out of 500 packets. This trend continued with the three-node configuration, where packet loss increased further to 21 out of 500 packets. This significant increase in packet loss underscores the impact on data transmission efficiency from reducing the number of relay nodes. The graph effectively illustrates that packet loss rises as the number of relay nodes decreases, emphasizing the importance of more relay nodes to ensure robust and reliable data transmission in multi-hop IoT systems.

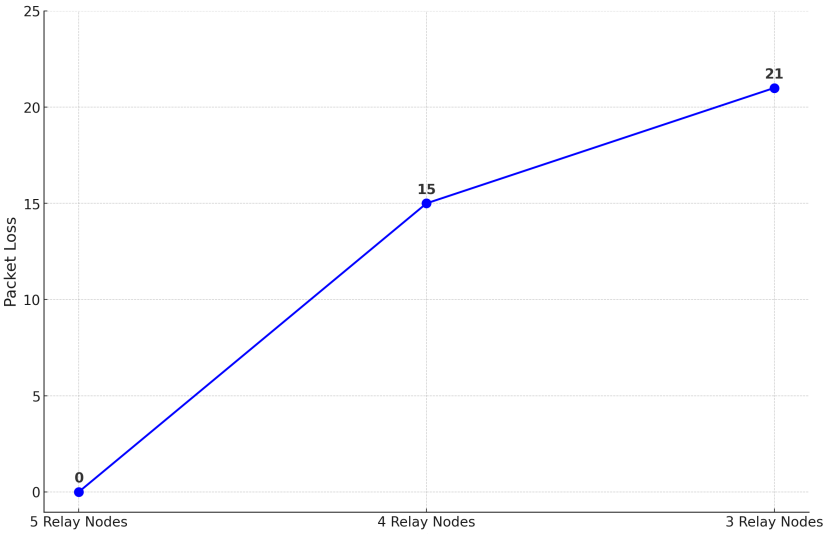


Figure 17. Packet loss in a multi-hop IoT system when varying the number of relay nodes.

Overall, our analysis revealed that increasing the number of relay nodes in a multi-hop IoT system with ESP-NOW significantly improves performance by reducing latency and minimizing packet loss. The configuration with five relay nodes achieved the lowest latency and zero packet loss, demonstrating optimal efficiency and reliability. In contrast, reducing the number of relay nodes to four and three resulted in higher latencies and increased packet loss, indicating a decline in performance. These findings highlight the importance of having more relay nodes to ensure robust and efficient data transmission in multi-hop IoT networks.

Table 2 shows a comparison that highlights our solution, with a latency of 178 ms and 0% packet loss, which excels in complex indoor environments with obstacles, outperforming other multi-hop protocols tested in more controlled or simulated conditions. ESP-MESH, with a latency of 330 ms and 6% packet loss, and LoRa-MESH, with a latency of 750 ms where packet loss was not measured, both fall short in obstacle-rich environments such as ours. MRT-BLE has the highest latency of 1400 ms, and packet loss was also not measured, suggesting limitations for time-sensitive applications, while ESP-NOW Mesh has the lowest latency of 75 ms, but 6.25% packet loss in typical home environments. Overall, our approach stands out for its superior balance of low latency and zero packet loss, making it ideal for IoT applications that require reliable performance in real-world, challenging conditions.

Table 2. Comparison of latency and packet loss with other multi-hop protocols.

Communication Protocol	Latency	Packet Loss	Experiment Environment	Data Measurement	Reference
Our solution	120 ms	0%	Complex indoor environment filled with obstacles.	Real environment in the Department of Electrical, Electronic and Computer Engineering at the University of Ulsan.	-
ESP-MESH	330 ms	6%	No specific obstacles mentioned, but uses random distributed agile training equipment in sports environments.	Real environment using sports students at Hubei University of Technology.	[13]
LoRa-Mesh	750 ms	Not indicated	Simulation-based, considering large-scale rural locations, no specific obstacle details mentioned.	Simulation-based analysis with NS2 simulator, focused on rural energy systems.	[15]
MRT-BLE	1400 ms	Not indicated	Realistic industrial environment with detailed experimental setup for Bluetooth Low Energy mesh networking.	Experimental measurements obtained through a real testbed in a realistic industrial scenario.	[14]
ESP-NOW Mesh	75 ms	6.25%	Real environment in a typical house with obstacles like walls, doors, and furniture.	Real environment with tests conducted in various scenarios in a typical home setup.	[16]

9. Discussion

In this study, the proposed system leverages multi-hop IoT networks using the ESP-NOW protocol to monitor worker presence in complex workplaces. However, several limitations and challenges were encountered during the development and deployment of the system. A significant limitation is the payload size restriction inherent in the ESP-NOW protocol, which allows a maximum payload size of 250 bytes. This constraint becomes critical when transmitting larger data sets across the network. To address this

issue, data compression is required to ensure that the data packet size remains below the maximum payload that can be transmitted effectively within the network. Integrating data compression techniques is essential for optimizing communication efficiency, particularly when scaling the system to larger or more data-intensive environments.

Another significant challenge was the need to synchronize the WiFi channels across the multi-hop network when connecting to a central server through a gateway node. Specifically, for this communication to work, all nodes in the multi-hop network had to operate on the same WiFi channel as the router connected to the server. To achieve this, the gateway node was configured in dual mode using the WIFI_AP_STA function, allowing it to act as both a station and an access point. Meanwhile, all other nodes, including the scanner and relay nodes, were set to station mode (WIFI_STA) only. This configuration ensured that the entire network stayed on the same WiFi channel, enabling seamless data transmission from the scanner and relay nodes to the gateway without the need for internet connectivity. Establishing stable communication in a multi-hop IoT network via ESP-NOW under these conditions was a complex task, requiring precise channel synchronization between all nodes. This issue could be mitigated by developing more flexible, dynamic solutions for channel management.

Additionally, ensuring reliable communication with zero packet loss proved to be a significant challenge during the system's experimental deployment. We tested multiple modules of the ESP32 chip to identify the most suitable one for configuring our multi-hop IoT network. Initially, we used the ESP32-WROOM-32, which features a built-in antenna, but its coverage was insufficient, leading to packet loss and unstable communication. To overcome this, we transitioned to the ESP32-WROOM-32UE module, which allows for the use of an external antenna. However, even after testing it with the Dual Band Antenna-U.FL (2.4 GHz) [WRL-18087], the coverage remained inadequate, and we could not achieve the expected performance. Finally, we opted for a 5 dBi dipole antenna operating at 2.4 GHz, which significantly improved both coverage and reliability. Paired with the ESP32-WROOM-32UE module, this setup enabled us to achieve zero packet loss and ensured stable communication across the entire multi-hop network during our experiments.

Overall, the development and deployment of the system involved addressing several technical challenges, including data payload limitations, network configuration complexities, and the selection of appropriate hardware and antennas. These challenges were successfully overcome through meticulous system design and thorough experimentation, ultimately resulting in a reliable and scalable solution for monitoring worker presence in complex environments.

10. Conclusions and Future Works

This research explored the development and implementation of a BLE beacon integrated with a multi-hop IoT network using the ESP-NOW protocol for monitoring worker presence in complex workplaces. The system utilizes BLE technology to leverage ESP32 microcontrollers as beacon tags attached to workers, along with the nRF Connect application on smartphones that also acts as beacon tags by broadcasting signals, enabling precise, real-time presence data. This dual approach enhances the system's ability to accurately estimate workers in complex workplaces, providing versatile and reliable monitoring capabilities that extend functionality without the need for additional hardware. ESP-NOW enables a multi-hop network with seamless, low-power communication across large, complex environments without relying on traditional network infrastructure. This approach supports the unlimited deployment of relay nodes to overcome spatial challenges by extending the communication range and ensuring reliable data transmission from the scanner node to the gateway node without internet connectivity, while maintaining minimal latency and zero packet loss. The system's efficiency is further enhanced by the MQTT protocol, which manages data transfer from the gateway node to a centralized Node-RED server, enabling real-time remote monitoring and visualization of worker data. Experimental results demonstrated the system's effectiveness in a real-world environment with complex

indoor obstacles, achieving a low average latency of 120 milliseconds and zero packet loss when configured with five relay nodes, confirming the system's capability to maintain robust communication. This three-layer IoT framework—comprising perception, network, and application layers—provides a scalable and comprehensive monitoring solution for workplaces such as shipyards, factories, and construction sites, enhancing worker safety by enabling immediate emergency responses and proactive risk management by identifying hazardous conditions early. This approach marks a significant advancement in multi-hop IoT network-based monitoring, setting a new benchmark for scalability, efficiency, and adaptability in complex workplace environments.

Future work will focus on integrating trilateration techniques using RSSI values to improve the accuracy of distance measurements between workers and scanner nodes. By collecting the RSSI values of the multiple beacons from the strategically placed scanner node and converting these values into distance estimates using a path loss model, trilateration will be used to determine the exact positions of workers. This advanced method will mitigate the inaccuracies associated with RSSI-based distance estimation, providing a more reliable solution for real-time location monitoring and improving worker safety and operational efficiency in complex work environments. Moreover, we aim to incorporate geofencing into the multi-hop IoT network. Geofencing technology has been widely recognized for its effectiveness in creating virtual boundaries to monitor and control movement in specific zones, particularly in hazardous or restricted areas [38]. By defining virtual boundaries, geofencing will enable automatic alerts when workers enter restricted or hazardous zones. BLE beacons will provide real-time location data, and the multi-hop network will relay this information to trigger safety protocols, such as sending alerts or deactivating equipment when boundaries are crossed. This integration will further enhance worker safety by enabling precise location tracking and proactive risk management in large, complex environments. Additionally, we plan to integrate GNSS tracking for workers in large, outdoor, or remote areas where BLE beacons may not be sufficient. By compressing GNSS data before transmission through the multi-hop IoT network, we can optimize bandwidth and ensure reliable data transmission even over large distances. Leveraging GNSS data compression techniques, as demonstrated in recent research [39], will allow the system to efficiently transmit location data while minimizing network load and ensuring real-time monitoring. This approach will further enhance the system's scalability and ensure accurate worker tracking in a variety of workplace environments, thereby extending the capabilities of our multi-hop IoT network to handle more demanding use cases.

Author Contributions: Conceptualization, R.U., T.H. and I.K.; methodology, R.U.; software, R.U.; validation, T.H. and I.K.; formal analysis, R.U.; investigation, R.U.; resources, T.H. and I.K.; data curation, R.U.; writing—original draft preparation, R.U.; writing—review and editing, I.K.; visualization, R.U.; supervision, I.K.; project administration, I.K.; funding acquisition, I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Hernández-Rojas, D.L.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Escudero, C.J. A plug-and-play human-centered virtual TEDS architecture for the web of things. *Sensors* **2018**, *18*, 2052. [CrossRef] [PubMed]
2. Márquez-Sánchez, S.; Campero-Jurado, I.; Herrera-Santos, J.; Rodríguez, S.; Corchado, J.M. Intelligent platform based on smart PPE for safety in workplaces. *Sensors* **2021**, *21*, 4652. [CrossRef]

3. De Blasio, G.S.; Quesada-Arencibia, A.; García, C.R.; Rodríguez-Rodríguez, J.C. Bluetooth low energy technology applied to indoor positioning systems: An overview. In Proceedings of the Computer Aided Systems Theory–EUROCAST 2019: 17th International Conference, Las Palmas de Gran Canaria, Spain, 17–22 February 2019; Revised Selected Papers, Part I 17; Springer: Berlin/Heidelberg, Germany, 2020; pp. 83–90.
4. Lin, Y.W.; Lin, C.Y. An interactive real-time locating system based on bluetooth low-energy beacon network. *Sensors* **2018**, *18*, 1637. [CrossRef]
5. Ke, C.K.; Ho, W.C.; Lu, K.C. Developing a beacon-based location system using Bluetooth low energy location fingerprinting for smart home device management. In Proceedings of the Wireless Internet: 11th EAI International Conference, WiCON 2018, Taipei, Taiwan, 15–16 October 2018; Proceedings 11; Springer: Berlin/Heidelberg, Germany, 2019; pp. 235–244.
6. Hlaing, W.; Thepphaeng, S.; Nontaboot, V.; Tangsunantham, N.; Sangsuwan, T.; Pira, C. Implementation of WiFi-based single phase smart meter for Internet of Things (IoT). In Proceedings of the 2017 International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 8–10 March 2017; pp. 1–4.
7. Adi, P.D.P.; Sihombing, V.; Siregar, V.M.M.; Yanris, G.J.; Sianturi, F.A.; Purba, W.; Tamba, S.P.; Simatupang, J.; Arifuddin, R.; Prasetya, D.A.; et al. A performance evaluation of ZigBee mesh communication on the Internet of Things (IoT). In Proceedings of the 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), Surabaya, Indonesia, 9–11 April 2021; pp. 7–13.
8. Darroudi, S.M.; Gomez, C. Bluetooth low energy mesh networks: A survey. *Sensors* **2017**, *17*, 1467. [CrossRef]
9. Unwala, I.; Taqvi, Z.; Lu, J. Thread: An iot protocol. In Proceedings of the 2018 IEEE Green Technologies Conference (GreenTech), Austin, TX, USA, 4–6 April 2018; pp. 161–167.
10. Yukhimets, D.; Sych, A.; Sakhnenko, A. Designing a method for constructing distributed open ACS based on the ESP-NOW wireless protocol. In Proceedings of the 2020 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 6–12 September 2020; pp. 642–647.
11. Hoang, T.N.; Van, S.T.; Nguyen, B. ESP-NOW based decentralized low cost voice communication systems for buildings. In Proceedings of the 2019 international symposium on electrical and electronics engineering (ISEE), Ho Chi Minh, Vietnam, 10–12 October 2019; pp. 108–112.
12. Sadeq, A.S.; Hassan, R.; Al-rawi, S.S.; Jubair, A.M.; Aman, A.H.M. A qos approach for Internet of Things (Iot) environment using mqtt protocol. In Proceedings of the 2019 International Conference on Cybersecurity (ICoCSec), Negeri Sembilan, Malaysia, 25–26 September 2019; pp. 59–63.
13. Wang, E.; Xiao, L.; Han, X.; Tan, B.; Luo, L. Design of an agile training system based on wireless mesh network. *IEEE Access* **2022**, *10*, 84302–84316. [CrossRef]
14. Leonardi, L.; Patti, G.; Bello, L.L. Multi-hop real-time communications over bluetooth low energy industrial wireless mesh networks. *IEEE Access* **2018**, *6*, 26505–26519. [CrossRef]
15. Marahatta, A.; Rajbhandari, Y.; Shrestha, A.; Singh, A.; Thapa, A.; Gonzalez-Longatt, F.; Korba, P.; Shin, S. Evaluation of a lora mesh network for smart metering in rural locations. *Electronics* **2021**, *10*, 751. [CrossRef]
16. Paguay, J.A.C.; Brito, G.A.H.; Rojas, D.L.H.; Calva, J.J.C. Secure home automation system based on ESP-NOW mesh network, MQTT and Home Assistant platform. *IEEE Lat. Am. Trans.* **2023**, *21*, 829–838. [CrossRef]
17. Jeon, K.E.; She, J.; Soonsawad, P.; Ng, P.C. Ble beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet Things J.* **2018**, *5*, 811–828. [CrossRef]
18. Chang, K.H. Bluetooth: A viable solution for IoT? [Industry Perspectives]. *IEEE Wirel. Commun.* **2014**, *21*, 6–7. [CrossRef]
19. Hernández-Rojas, D.L.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Escudero, C.J. Design and practical evaluation of a family of lightweight protocols for heterogeneous sensing through BLE beacons in IoT telemetry applications. *Sensors* **2017**, *18*, 57. [CrossRef] [PubMed]
20. Mangold, S.; Jarosch, A.; Monney, C. Operator assisted cognitive radio and dynamic spectrum assignment with dual beacons-detailed evaluation. In Proceedings of the 2006 1st International Conference on Communication Systems Software & Middleware, New Delhi, India, 8–12 January 2006; pp. 1–6.
21. Eridani, D.; Rochim, A.F.; Cesara, F.N. Comparative performance study of ESP-NOW, Wi-Fi, bluetooth protocols based on range, transmission speed, latency, energy usage and barrier resistance. In Proceedings of the 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), Virtual, 18–19 September 2021; pp. 322–328.
22. Amponis, G.; Lagkas, T.; Karampatzakis, D.; Radoglou-Grammatikis, P.; Argyriou, V.; Moscholios, I.; Goudos, S.; Sarigiannidis, P. Efficient peer-to-peer unicasting for VANET architectures via enhanced monolithic protocols. In Proceedings of the 2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Ioannina, Greece, 23–25 September 2022; pp. 1–8.
23. Espressif. Overview of the ESP-NOW. 2024. Available online: https://docs.espressif.com/projects/esp-idf/en/latest/esp32c3/api-reference/network/esp_now.html (accessed on 10 July 2024).
24. Kondareddy, Y.R.; Agrawal, P. Synchronized MAC protocol for multi-hop cognitive radio networks. In Proceedings of the 2008 IEEE International Conference on Communications, Beijing, China, 19–23 May 2008; pp. 3198–3202.
25. Mun, D.H.; Le Dinh, M.; Kwon, Y.W. An assessment of internet of things protocols for resource-constrained applications. In Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016; Volume 1, pp. 555–560.

26. Prada, M.A.; Reguera, P.; Alonso, S.; Morán, A.; Fuertes, J.J.; Domínguez, M. Communication with resource-constrained devices through MQTT for control education. *IFAC-PapersOnLine* **2016**, *49*, 150–155. [CrossRef]
27. Hunkeler, U.; Truong, H.L.; Stanford-Clark, A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In Proceedings of the 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), Bangalore, India, 6–10 January 2008; pp. 791–798.
28. Clerissi, D.; Leotta, M.; Reggio, G.; Ricca, F. Towards an approach for developing and testing Node-RED IoT systems. In Proceedings of the 1st ACM SIGSOFT International Workshop on Ensemble-Based Software Engineering, Lake Buena Vista, FL, USA, 4 November 2018; pp. 1–8.
29. Torres, D.; Dias, J.P.; Restivo, A.; Ferreira, H.S. Real-time feedback in node-red for iot development: An empirical study. In Proceedings of the 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Prague, Czech Republic, 14–16 September 2020; pp. 1–8.
30. Hwang, H.C.; Park, J.; Shon, J.G. Design and implementation of a reliable message transmission system based on MQTT protocol in IoT. *Wirel. Pers. Commun.* **2016**, *91*, 1765–1777.
31. Espressif. Datasheet of ESP32-WROOM-32. 2024. Available online: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (accessed on 11 July 2024).
32. UUID. Online UUID Generator By the Bluetooth SIG. 2024. Available online: https://www.uuidgenerator.net/#google_vignette (accessed on 13 July 2024).
33. Schütz, M. RF Harvesting at 2.4 GHz for Scattering between Battery-less Transponder and Mobile Telephones. In Proceedings of the 2021 IEEE International Conference on RFID Technology and Applications (RFID-TA), Delhi, India, 6–8 October 2021; pp. 93–96.
34. Semiconductor, N. nRF Connect for Mobile. 2024. Available online: https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-mobile (accessed on 11 July 2024).
35. Espressif. Datasheet of ESP32-WROOM-32UE. 2024. Available online: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf (accessed on 12 July 2024).
36. Lozano Bravo, H.; Lo, E.; Moyes, H.; Rissolo, D.; Montgomery, S.; Kuester, F. A Methodology for Cave Floor Basemap Synthesis from Point Cloud Data: A Case Study of Slam-Based LIDAR at Las Cuevas, Belize. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2023**, 179–186. [CrossRef]
37. Georgiou, O.; Nguyen, C. Multihop connectivity of ad hoc networks with randomly oriented directional antennas. *IEEE Wirel. Commun. Lett.* **2015**, *4*, 369–372. [CrossRef]
38. Nachiappan, B.; Rajkumar, N.; Viji, C.; Mohanraj, A. Ensuring Worker Safety at Construction Sites Using Geofence. *SSRG Int. J. Civ. Eng.* **2024**, *11*, 107–113. [CrossRef]
39. Rakhmanov, A.; Wiseman, Y. Compression of GNSS data with the aim of speeding up communication to autonomous vehicles. *Remote Sens.* **2023**, *15*, 2165. [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.