Secure Home Automation System based on ESP-NOW Mesh Network, MQTT and Home Assistant Platform

J. Cujilema, G. Hidalgo, D. Hernández-Rojas, and J. Cartuche

***Abstract*—Domotics allows for the control and monitoring of interconnected electronic devices in a home through effective communication protocols within an Internet of Things (IoT) architecture. To achieve proper functionality, it is necessary to ensure the communication and security of the IoT devices that make up the network, which are the main challenges faced by domotic systems. In this context, we have developed an algorithm that allows for the creation of a secure mesh network using the ESP-NOW communication protocol, with data encryption on the mesh network. The algorithm has been implemented and evaluated using low-cost commercial IoT devices and integrated through a secure MQTT Broker with the Home Assistant domotic platform. Tests carried out in a real environment on the proposed domotic system indicate that the network is fast and stable, with an overall average latency of 75 ms and an overall average packet loss rate of 9.25%.**

***Index Terms*—ESP-NOW, Smart Home, Mesh Network, Domotic System, MQTT, Home Assistant, IoT.**

I. Introduction

The Internet of Things (IoT) has made significant progress due to the impact it has had on the multiple areas in which it has been implemented [1], [2]. One of these areas or domains of IoT application is home automation [3], allowing users to monitor variables such as temperature and humidity, energy management [4], [5] and even remotely control household appliances [6].

However, each of these protocols presents some disadvantages that have motivated the development of our work. For example, intensive use of Wi-Fi [7] can cause interference and congestion in the network, which can affect the speed and quality of communication. Moreover, the use of Wi-Fi can increase the energy consumption of devices, which can reduce the battery life of portable devices. BLE [8] may present problems of limited range, meaning that devices must be very close to each other for effective communication. Zigbee [9] requires a network controller, which can add to the loss of home automation systematization. For communication to be effective, devices must be close to each other. Although Z-Wave [10] is a low-energy communication protocol, its range can also be limited, which can affect the quality of communication. The cost and complexity of home automation systems may increase with the requirement of a network controller. The biggest problem with KNX [11] is that it is a wired protocol, meaning that wired installation is required, which can increase the cost and complexity of the home automation system. In older buildings, it may be difficult to install wiring. While Modbus [12] is a reliable and robust communication protocol, its use can pose security problems. The data transmitted through the network can be vulnerable to attacks because they do not have implicit security functions. Another important issue is data security in home automation, where several authors reflect the known vulnerabilities of the mentioned protocols, susceptible to external attacks [13]. Even modern technologies such as ESP-Mesh, which have also entered home applications, present significant limitations in terms of security [14].

All of this has motivated us to offer a secure mesh network solution for home automation systems that users can easily use without compromising their security. In this article, we present a secure home automation system based on the ESP-NOW mesh network technology, MQTT protocol, and Home Assistant platform. We have chosen the ESP-NOW technology because it offers low power consumption, necessary for the constrained devices used in home automation, security with 128-bit encryption for data transmission, reliability, thanks to inherent packet retransmission mechanisms, and its ease of implementation on very cheap devices such as Espressif Systems' ESP32 and ES8266. The ESP-NOW protocol was conceived for use in point-to-point communication, but when used in a mesh, we add the possibility of having greater reach and network coverage, as devices communicate with each other, having several routes available to exchange messages [15].

Our proposal offers a secure and efficient mesh network solution for home automation systems as a solution to the problems posed, which include the following main contributions:

- An algorithm (called BRAM-Now, Broadcast Route Algorithm Mesh-Now) is defined for use in constrained devices that allows for the creation of a mesh network over the native ESP-NOW protocol. (See section: IV-B-1)
- A home automation system based on the use of open-

J. Cujilema Paguay. Research group AutoMathTIC, Faculty of civil engineering, Universidad Técnica de Machala. Machala, El Oro, Ecuador. jcujilema3@utmachala.edu.ec

G. Hidalgo Brito. Research group AutoMathTIC, Faculty of civil engineering, Universidad Técnica de Machala. Machala, El Oro, Ecuador. ghidalgo@utmachala.edu.ec

D. Hernandez Rojas. Research group AutoMathTIC, Faculty of civil engineering, Universidad Técnica de Machala. Machala, El Oro, Ecuador. dhernandez@utmachala.edu.ec

J. Cartuche Calva. Research group AutoMathTIC, Faculty of civil engineering, Universidad Técnica de Machala. Machala, El Oro, Ecuador. jcartuche@utmachala.edu.ec

source standards and platforms is presented and evaluated. The solution uses the BRAM-Now algorithm, which allows for the creation of a secure mesh WSN over ESP-Now. (See section: IV-A)
- A special IoT Gateway is presented for bidirectional communication between the open-source Hassio (Home Assistant) platform and the mesh network, allowing mesh nodes to use the MQTT standard for data messaging. (See section: IV-C)
- A real-time monitoring system based on Node-RED, implemented within the gateway, is presented and evaluated. This system allows for the automatic listing of all active nodes on the mesh in a Hassio dashboard. Additionally, it is possible to customize the watchdog timer for each node from the same dashboard. (See section: IV-D-1)
- New methods are added to the ESP-Now protocol for encrypting data using the AES encryption scheme. (See section: IV-B-2)

The document is organized as follows: Section II presents the work related to this paper, Section III presents the ESP-NOW communication protocol, Section IV describes the IOT ESP-NOW mesh architecture proposed, Section V demonstrates the performance of the proposed network through experimental tests; Section VI presents the analysis of the results obtained in the performance tests; and finally, Section VII with the conclusions of the work carried out and future works.

II. Related Literature

One of the most important applications of the Internet of Things (IoT) is in home automation systems that enable the automation and control of household devices to improve comfort, safety, and energy efficiency. Home automation has evolved from simple task automation to the creation of fully integrated and personalized smart homes [19]. To achieve this, integration and communication among all elements of an IoT architecture is necessary. Various protocols have been used in home automation, as mentioned in the previous section, with mesh networking protocols being particularly notable, as they allow for network extension and communication stability, enabling different routes when nearby devices are turned off or in power-saving mode. Table I summarizes previous work related to ours, indicating the main characteristics of these works in terms of available topologies and libraries, security, and the reach of these networks.

Additionally, we can mention other interesting works that can contribute to the integration of services in home automation, such as [20], which provides a solution for controlling IoT devices via Twitter, and [21], which uses event-based mesh models for the same purpose. Regarding security, there is still a long way to go in IoT applications, where works such as [22] on blockchain, [23] on Markov model-based security, [24] on IoT device authentication, and [25] on the use of AI techniques such as deep learning, among others, contribute to addressing this challenge.

TABLE II
Libraries to create a mesh on ESP-NOW

Library	Devices Supported	Problems	Reference
PainlesMesh	ESP32, ESP8266	May be incompatible with other libraries. The messages can be lost or deleted due to high network traffic	[26]
m2mMEsh	ESP32	Under development. No security	[27]
FloodingMesh	ESP32	Deprecated	[28]
ESP-NOW Wrapper	ESP32	No security. Requires wifi connection	[29]
ZHNetwork	ESP32, ESP8266	It uses XOR-type encryption vulnerable to attacks	[30]
NowMEsh	ESP8266	Deprecated. No security	[31]

It is worth noting that the ESP-NOW protocol, which we selected for our system and justified in the previous section, was designed for peer-to-peer (P2P) communication. However, several efforts have been made to create a mesh network on top of this protocol. Table II lists the references of some of these efforts, along with the name of the required library, supported devices, and main problems detected, so that the community can use, improve, and share them. It is important to highlight that most of these solutions lack official documentation, and only what is shared on GitHub repositories is available, except for PainlessMesh, which is the library used by the ESP-Mesh protocol and is maintained by Espressif.

The difficulties shown in Table II, together with the lack of documentation to implement secure, stable, scalable, low-cost, and easy-to-implement mesh networks for DIY communities on the ESP-NOW protocol without interfering with a home's conventional Wi-Fi network, motivated us to develop a simple, straightforward, and effective algorithm for creating a mesh network that can also be integrated into popular home automation platforms such as Home Assistant. Our proposal is explained in detail in Section IV.

III. ESP-NOW Protocol

ESP-NOW is a communication protocol developed by Espressif company. It allows to transmit short packets using the MAC address of ESP32 or ESP8266 devices. The ESP-NOW protocol uses the IEEE 802.11 standard [32]. This protocol reduces the five layers of the OSI model to a single layer as shown in [15].

This protocol offers several advantages, which are described below:

- **Fast response:** devices after power-up can transmit and control data from other paired devices directly, giving responses with millisecond speeds.
- **Low energy consumption:** by reducing the five layers of the OSI model to a single layer, power consumption is reduced.
- **Good compatibility:** a device that is connected to a faulty router or the network is unstable, can perform fast and stable communication using ESP-NOW.
- **Improved range and reception:** registers a callback for sending ESP-NOW data.
- **Multilayer control:** with ESP-NOW you can send unicast and broadcast messages to control certain devices.

TABLE I
PROTOCOLS USED IN HOME AUTOMATION SYSTEMS

Protocol	Topologies	Libraries	Security	Scope
Wi-fi [7]	P2P, mesh	WiFi101, WiFiNINA, ESP8266WiFi	WPA2, WPA3, 802.1X	50m
BLE [8]	P2P, mesh	BlueZ, CoreBluetooth, Android Bluetooth API	Bluetooth Low Energy Security, AES-CCM, ECDH	10m
Zigbee [9]	mesh	Zigbee2MQTT, Zigbee-Shepherd, ZBOSS Zigbee Stack	AES-128	75m
Z-Wave [10]	mesh	Z-Wave JS, OpenZWave, Home Assistant Z-Wave	AES-128	30m
Thread [16]	mesh	OpenThread, Nordic nRF5 SDK for Thread, ThreadX	AES-128	100m
KNX [11]	System bus, LAN	ETS Inside, OpenHAB KNX Binding, KNX Association	KNX Data Secure	350m
ESP-NOW [17]	P2P	ESPNow, ESP-IDF	AES-128	100m
ESP-MESh [18]	mesh	ESP-MESH SDK, ESP-IDF	AES-128	100m

A. ESP-NOW API

The functions required for the ESP-NOW implementation can be found in the ESP-IDF library API. Some of the most commonly used functions are listed below:

- **esp_now_ init:** to initialize the ESP-NOW protocol.
- **esp_now_set_pmk:** is used to set the PMK key.
- **esp_now_add_peer:** is used to add or pair the devices to which the data will be sent.
- **esp_now_send:** sends data to the paired devices. This function will return ESP_NOW_SEND_SUCCESS, if the data is successfully received by the paired target device. Otherwise, it will return ESP_NOW_SEND_FAIL.
- **esp_now_register_recv_cb:** registers a callback for receiving ESP-NOW data which is called when the device receives ESP-NOW messages.
- **esp_now_register_send_cb:** registers a callback for sending ESP-NOW data.

This section contains a description of how the ESP-NOW mesh network works in a home automation system.

IV. IoT ESP-NOW Mesh architecture for home automation

A. Global Overview

The proposed smart home system based on a three-layer IoT architecture, namely Perception, Network, and Application layers, is illustrated in Fig. 1.

The Perception layer or device layer in this IoT architecture is where IoT smart devices are located, typically connected with sensors and actuators. For instance, humidity and temperature sensors, presence sensors, light level sensors, and actuators such as relay, display, and motors, among others. In our case, these devices form a WSN and, using our algorithm, create a fast and secure mesh network over the ESP-NOW communication protocol. This protocol allows up to 20 unencrypted nodes and up to 10 nodes with data encryption [15]. Our prototype consists of five nodes as a proof of concept, but the network can be extended, as explained later.

The Network layer allows the Perception layer to access higher layers, such as the Application layer, through which web users can monitor sensors or send commands to the actuators. A classic element is the IoT gateway, which enables communication between different communication protocols and data messaging used by the upper and lower layers. In our proposal, we have developed a special gateway to communicate the ESP-NOW protocol of the Perception layer with the MQTT protocol of the Application layer. Our gateway makes use of MQTT and BLE protocols for its operation.

The MQTT (Message Queuing Telemetry Transport) protocol is an ISO standard, which is based on a publish/subscribe mechanism [33], [34]. Clients communicate through topics, which perform the function of private communication channels [35] and Bluetooth Low Energy (BLE) is a short-range wireless technology, its power consumption is minimal, it is robust and efficient, among others [36], [37].

In our Application layer, we have the popular home automation platform Home Assistant (Hassio), which allows the integration of an MQTT broker such as Mosquitto and the deployment of dashboards for monitoring and controlling the entire system.

B. Perception Layer

It can consist of microcontrollers based on ESP32, in our case we have used commercial hardware platforms such as Lopy 4 and ESP32 development kit itself. The sensors used are DHT11 to measure the temperature and humidity of the environment, and a PIR sensor to detect motion. The actuators are relays that control the switching on and off of light bulbs. Our prototype is composed of only 5 nodes due to budget constraints. However, the system can be safely extended to 10 nodes per gateway. In our case, nodes 1, 2, and 3 are connected to sensors and actuators, while nodes 4 and 5 were only used to extend the network.

Node 1: For this node, the ESP32 board was used, which has a 4 MB flash/SRAM memory. This node has connected a DHT11 sensor that measures the humidity and temperature of the environment and a relay which turns on a light bulb. Finally, as shown in Fig. 2 and 13 an OLED Display that shows the data from the DHT11 sensor is also connected.

Node 2: Node 2 consists of a Lopy 4 that runs on the Espressif ESP32 chipset. This node has a PIR sensor that can be calibrated by distance and reading time and a buzzer

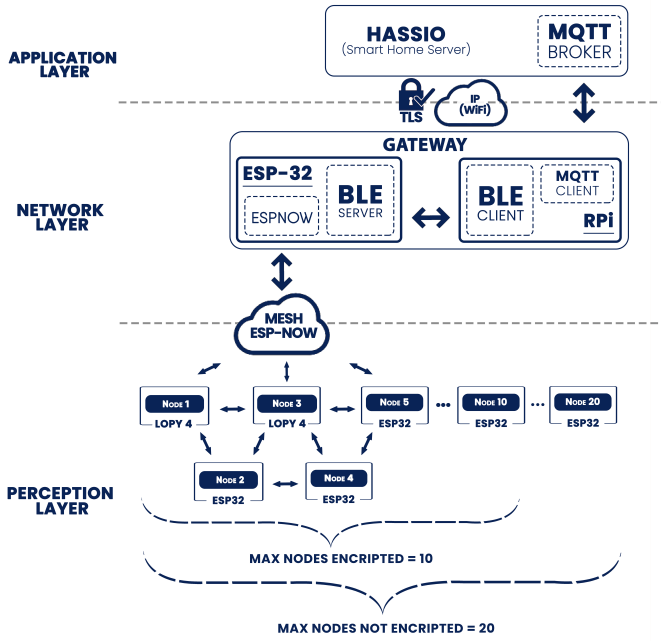

Fig. 1. IoT ESP-NOW Mesh architecture for home automation.

actuator, which is activated when the digital output of the PIR detects motion. These events are sent to the MQTT broker using ESP-NOW through the gateway. The assembly of this node is shown in Fig. 2 and 13.

Node 3 The operation of this node is similar to node 1, with the difference that the Lopy 4 board is used, as shown in Fig. 2 and 13.

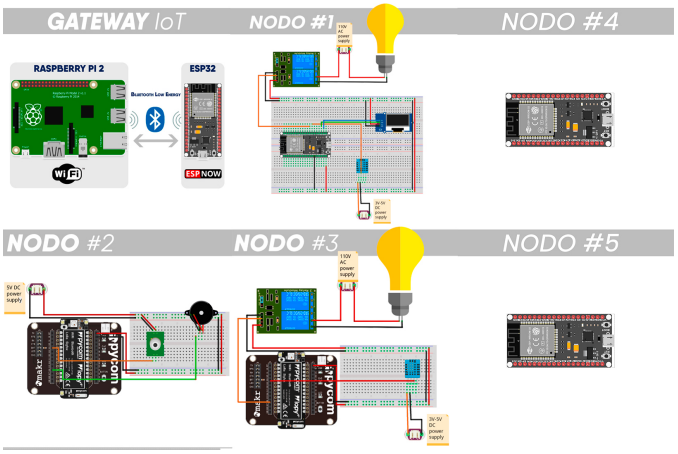

Fig. 2. Prototype connection diagram (nodes 1 to 5 and gateway).

1) BRAM-NOW algorithm:: The nodes that make up this layer use the BRAM-Now algorithm, in which each node verifies if a received message is addressed to it. If it is, the node consumes the message and reacts according to its own functioning. Otherwise, the node checks if its ID is already on the message's routing list. If it is, the node does not retransmit the message, but if it is not, it adds its ID and forwards the message to neighboring nodes. This simple algorithm, allows you to efficiently create a mesh network over ESP-NOW, and can be seen in more detail in Fig. 3 and the structure of the message in Table III. All these algorithms are available in our Github repository [38].

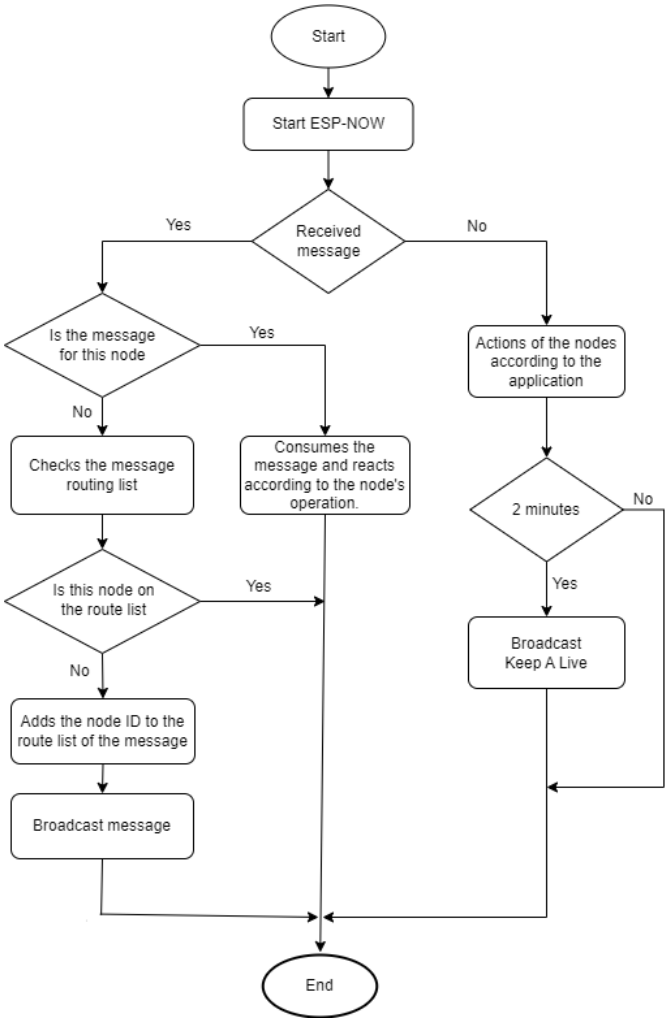

Fig. 3. BRAM-NOW Algorithm flowchart.

TABLE III
STRUCTURE OF THE ESP-NOW MESSAGE.

Destiny Id (2 bytes)	Routes (Array 10 bytes)				Data (10 bytes max)				
1	Node 1	Node 2	...	Id node	Var 1	Var 2	...	Var N	

2) AES encryption scheme:: Messages sent between nodes are encrypted using AES technology [39], [40].The entities involved in the generation of AES keys in the mesh are the following:

- **Entity generating the key:** In this work, it is the user who generates the 16-byte encryption keys following some recommended complexity guidelines such as special characters, uppercase and lowercase letters, and numbers from 0 to 9. This generated key must be protected as it is essential to ensure the security of the mesh data. The user may also make use

of cryptographically secure random number generation libraries to generate the key.

- **Entity that encrypts the data:** Each node is responsible for encrypting the data using the AES encryption key generated by the user. For this purpose, each node must have access to the same encryption key so that the data can be encrypted and decrypted correctly.
- **Entity decrypting the data:** Similarly to the previous point, each node is responsible for decrypting the data it receives from the mesh using the AES encryption key.

In Fig. 4, the data received, encrypted and decrypted with AES from a node is shown. This is done in order to protect the privacy and integrity of the transmitted data. It is important to emphasize that all nodes must have the key to be able to perform this procedure.

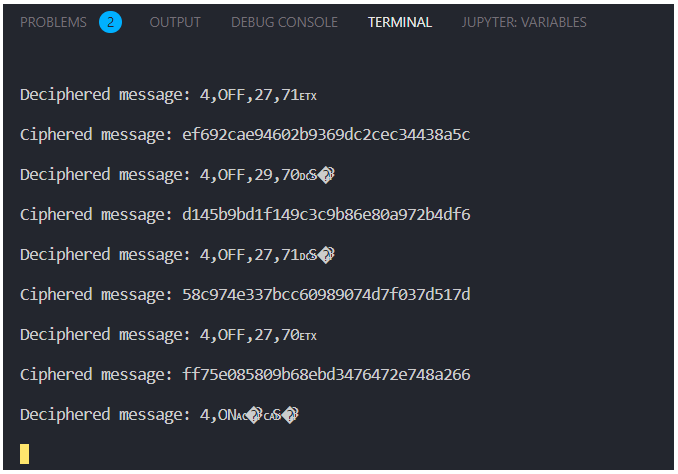

Fig. 4. ESP-NOW mesh data encrypted with AES.

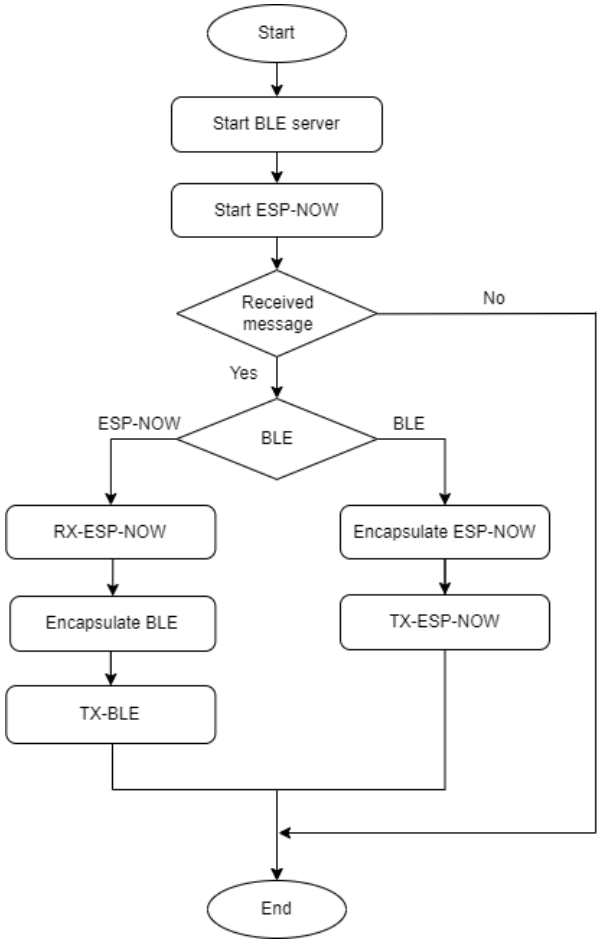

Fig. 5. Gateway flowchart (ESP32 side)

C. Network Layer (IoT Gateway)

For the development of this gateway, a Raspberry Pi and an ESP32 that communicate using BLE technology were used, as shown in Fig. 2 y la Fig. 13.

The reason for the realization of this gateway is due to the transmission channel conflict between wi-fi and ESP-NOW, forcing the nodes to use the same transmission channel of the wi-fi network, which limits adding nodes outside the coverage of the wi-fi router or AP (Access Point).

The ESP32 is responsible for receiving data from the ESP-NOW mesh to send to the Raspberry Pi via BLE, where the Raspberry acts as the client and the ESP32 as the server, as shown in the Fig. 5. The Raspberry uses the Node-RED visual programming tool [41], to connect as a client to the BLE server.

The communication between the MQTT broker and the Raspberry is encrypted with TLS to add further security to the network. The implementation of TLS in the MQTT broker has led to enhanced privacy and integrity protection for the data transmitted within Home Assistant's network as shown in the Fig. 6. This security measure is especially important for home networks, where the protection of users' personal data and privacy is critical. Fig. 7 shows the flow used to receive data from the MQTT broker and send it through BLE to the ESP32.

The data received by the BLE server is sent to the MQTT Broker using a switch type node that is responsible for parsing the data to send it to its respective topics, as shown in Fig. 8. This flow must first connect to the BLE server and then activate the notify node to receive the data.

For the local administration and monitoring of the nodes connected to the network, a dashboard was created in Node-RED with a list where the node number and its on or off status are displayed, see Fig. 9. It runs on the Raspberry Pi integrated in Home Assistant (Hassio).

D. Application Layer

For the home automation system server, the Home Assistant (Hassio) platform was used because it is free software and allows home automation [42]. Hassio has an extensive list of add-ons, including Mosquitto which was used as an MQTT broker to receive and send data in json format such as sensor data or activate actuators. This software has its own dashboard where the received data such as temperature, humidity and motion detection are displayed, as shown in Fig. 10. In addition, the user can control the switching on and off of lights or other devices that have been configured.

Fig. 6. MQTT data encrypted with TLS.

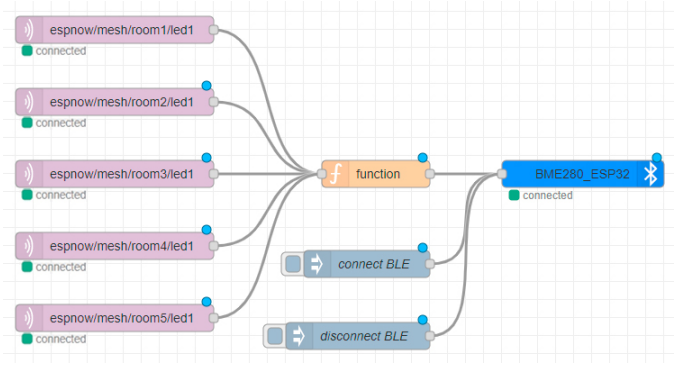

Fig. 7. Node-RED flow to receive data from MQTT broker and send it to BLE server. (Gateway - Rpi side)

The user can configure triggers that are activated when an event occurs or the value of a sensor changes, for example, if the temperature exceeds a set range a notification is sent to the cell phone.

The configurations in Hassio are done through a configuration.yaml file in an easy and fast way, since it does not require complex commands, has good documentation and an active community. In our repository [38] you can find the configurations for the on buttons of the rooms, in this case each switch has a unique ID, a name that will be used in the dashboard, the state_topic that indicates the topic by which it will receive and send the data, the value_template that indicates the value of the data in JSON format to be taken and finally the on and off states.

1) Mesh Monitoring based on Watchdog mechanism: On this platform, a watchdog mechanism has been implemented to list and observe the status of the active nodes in the ESP-NOW network. In Hassio each node has a card associated with the status and a timer with a duration of three minutes that is reset every time a node sends a message, if a node does not send a message within that time the card is deactivated indicating that the node is inactive, as shown in Fig. 11. For this, the nodes are scheduled to send Keep Alive messages every 2 minutes.

The cards must be manually configured by the network administrator, which is why a list was added to the dashboard which is updated in real time with all the nodes that are active in the system.

V. Experiments

In this section, the proposed home automation system is tested to verify the performance of the BRAM-NOW algorithm and the IoT architecture in a real environment. Latency and packet loss tests, which are the most commonly used metrics in domotics and WSN experiments, were conducted in five different scenarios. The majority of the experiments found in the referenced literature on home automation utilize simulations, testbed, open spaces without obstacles, or use few nodes within the same room. Figure 12 shows the floor plan of the house where the tests were conducted, as well as the furniture layout and the exact location of each node and the gateway.

The gateway was placed in the same room as the home's internet router, and the rest of the 5 nodes of our prototype were deployed in each room of the house, covering from the best to the worst scenario and also forcing the use of mesh, given the distances and obstacles inherent in the house. Table IV shows the linear distances between the gateway and each node and the obstacles involved.

TABLE IV
Test Scenarios Specifications

Scenario	Distance (m)	Obstacles
1	5	Walls, metal bed, bathroom, wooden door
2	7	Walls, wooden goal bed, closet, refrigerator, wooden door, television, furniture
3	10	Walls, metal bed, bathroom, wooden door
4	5	walls, TV, furniture, wooden door, bed
5	12	Walls, wooden goal bed, closet, refrigerator, wooden, TV, wooden door, aluminum doors

For the latency tests, 100 measurements were taken in each scenario, and for the packet loss tests, messages were sent every 30 seconds for a total of 200 measurements in each scenario. In both cases, the tests were bidirectional, that is, a set of tests were carried out by sending commands from the gateway to the nodes (G-N) through the mesh and another set in the opposite direction (N-G) by sending sensor data.

A. Experimental Environments

Figure 13 shows the elements used in the experiments: nodes (1 to 5) and IoT gateway described in section IV, other RPi running node-red flows for the tests, and a laptop for managing messaging in the debug of these node-red flows and also for graphing the results of the experiments in the RStudio statistical analysis environment. The specifications of these elements are described in Table V.

B. Latency and Packet Loss Testing

Both measurements were obtained at the same time in each experiment using the following methodology: Each

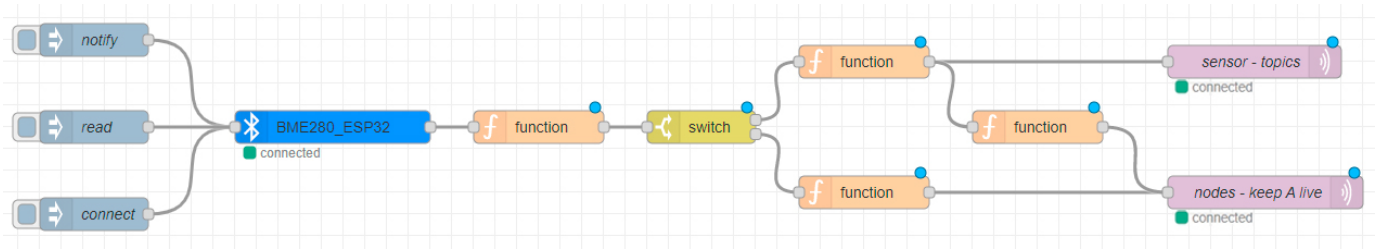

Fig. 8. Node-RED flow to receive messages from the BLE and send them to the MQTT Broker. (Gateway - Rpi side)

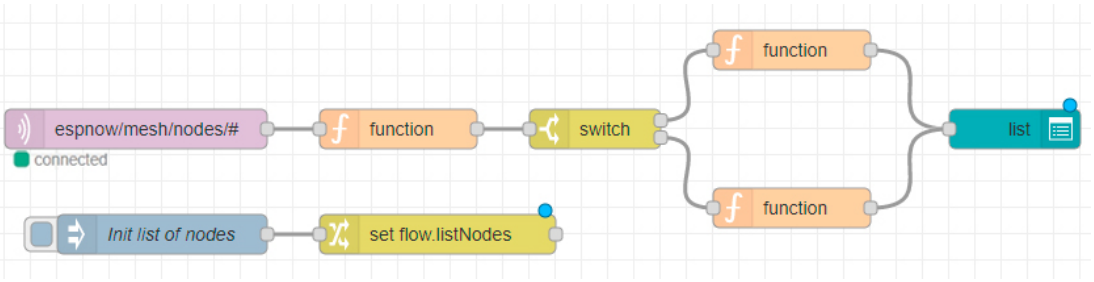

Fig. 9. Flow in Node-RED that lists the nodes connected to the network.

Fig. 10. Hassio dashboard for monitoring and control of home automation system.

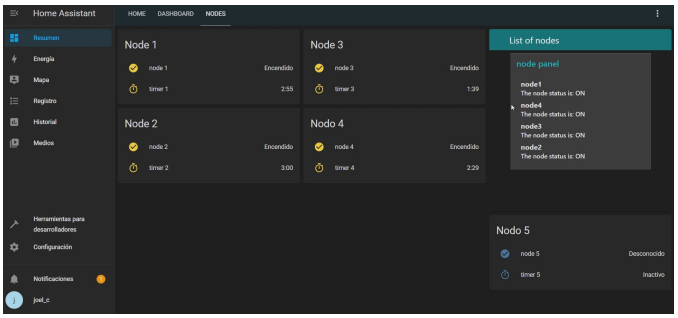

Fig. 11. Hassio dashboard for monitoring the status of ESP-Now mesh network nodes.

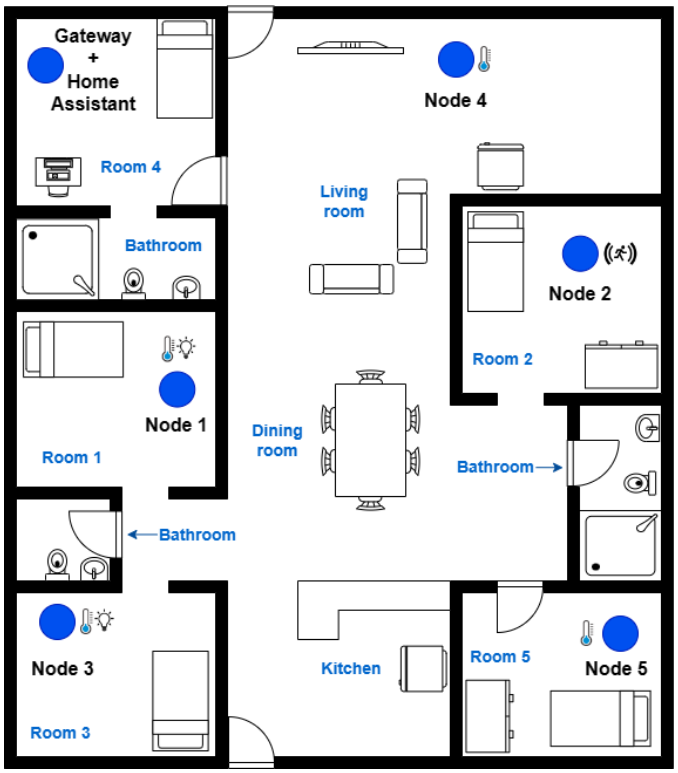

Fig. 12. Location of nodes in a typical house.

ESP32 (nodes and gateway-side) prints the received message to the console (USB port), to which we have temporarily connected an RPI running a new node-red flow developed for these tests, which allows adding the datetime field to the message, necessary to determine the latency. On a laptop, the debugs of each node-red flow are managed to calculate the latency and to mark the received messages and thus obtain the lost messages. For example, when the gateway sends a command to a node x, the ESP32 of the gateway prints the message to its USB port, then the test RPI connected to that port adds the datetime and prints it in the Node-red debug. In the test laptop, this message is labeled as ”sent” and as ”initial time”. When the message reaches node x (through the mesh), it is printed to its USB port, which has another test RPI connected to send the message with the current datetime to the laptop, where this message is marked as ”received” and as ”final time”. Finally, the latency

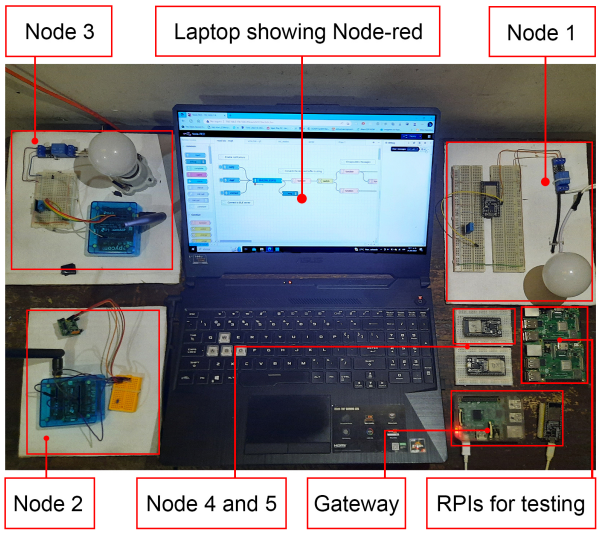

Fig. 13. Experimental environments.

TABLE V
MAIN SPECIFICATIONS OF THE EXPERIMENTAL ENVIRONMENTS ELEMENTS

Element	Main Specifications
Laptop	ASUS TUF Gaming A15, AMD Ryzen 5 4600H with Radeon, 16 GB RAM
Raspberry Pi 2B	Broadcom BCM2836 de cuatro núcleos ARM Cortex-A7 a 900 MHz, 1 GB RAM, Ethernet (10/100), Wi-Fi 802.11n y Bluetooth 4.1.
ESP32	Dual-core Tensilica LX6 de 32 bits, 520 KB de RAM SRAM interna + 8 MB de memoria PSRAM externa, Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR y BLE.
Node-RED	graphical programming tool V3.0
RStudio	IDE for statistics and graphics with R V2023.03.0+386

is calculated as the difference between the final and initial times obtained from that experiment, in the G-N direction. Since each message has a unique ID, it is easy to determine how many messages sent from the gateway did not reach node x, thus obtaining the number of lost packets. The same methodology is applied when a node x sends the value of a sensor to the gateway (N-G). All node-red flow tests are also available in our shared repository at [38].

The results of the average of the set of measurements carried out in each scenario to obtain the latencies in the gateway to node (G-N) and node to gateway (N-G) directions are shown in Table VI, and the lost packets in Fig. 14.

TABLE VI
LATENCY RESULTS FOR EACH OF THE SCENARIOS

	Minimum (ms)		Maximum (ms)		Average (ms)	
	N-G	G-N	N-G	G-N	N-G	G-N
Scenario 1	5	0	725	350	67	80
Scenario 2	12	0	1.961	536	150	35
Scenario 3	1	0	636	65	67	18
Scenario 4	12	0	924	248	91	32
Scenario 5	1	0	1.270	1011	152	58

In summary, our solution presents a overall average latency of 75 ms and a overall average of lost packets of 18.5 out of 200 packets sent, resulting in a 9.25% loss rate.

VI. RESULTS ANALYSIS

In the previous section, we presented the latency measurement results in Table VI. However, we have also included these values in the bar chart of Fig. 15 to facilitate comparison of the scenarios. From the chart, we can observe that almost all scenarios in the G-N direction (command sending) have a latency lower than the overall average of 75 ms.

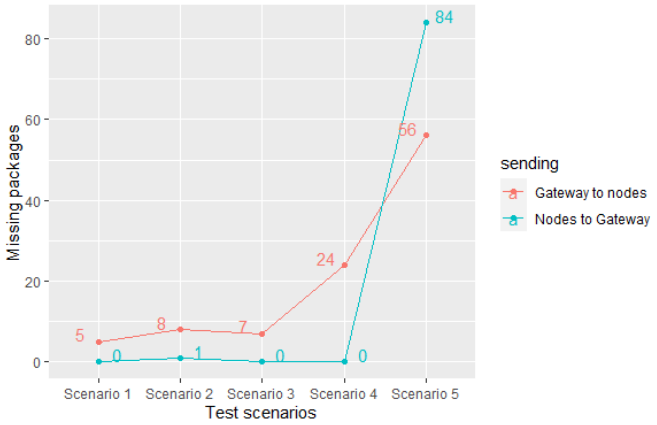

Fig. 14. Lost packet rate.

Additionally, we can see the expected results for scenario 5, which is the most critical due to its distance from the gateway and therefore requiring more hops in the mesh network, as well as the large number of obstacles involved. This explains the latencies and packet losses obtained. One possible solution for this node would be to add another node as a network extender and place it in an intermediate point with the gateway, ensuring the fewest possible obstacles such as concrete, metal and furniture.

It is important to note that the tests were performed during a time when all household members were present, making extensive use of internet-connected devices such as TVs, smartphones, and PCs, generating possible interferences. This may explain the high packet losses only in the G-N direction of scenario 4, located next to the TV and typically where more people gather in the home. Nevertheless, the results shown in Fig. 14 indicate that our solution is excellent for a sensor network (N-G) with practically no packet loss.

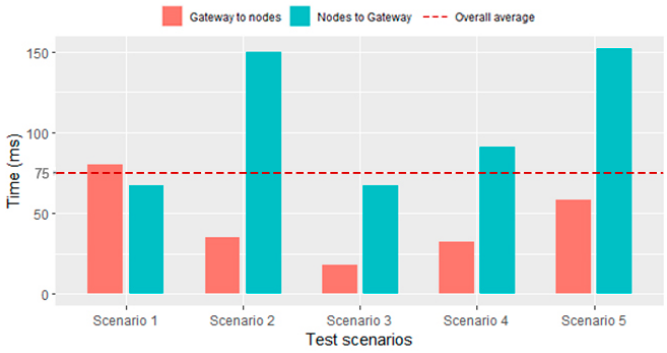

Fig. 15. Total average latencies for each scenario.

The table VII shows a comparison of the results obtained in our experiments with other works that use mesh protocols in home automation applications. We can highlight that our latency (75ms) is comparable with BLE (60ms) and Zigbee (30ms) solutions, despite the fact that these two were tested in obstacle-free environments. The rest of the compared solutions present higher latencies than ours. Regarding packet loss, our solution (9.25%) is similar to ESP-Mesh (6%), although it was also tested without obstacles. The Zigbee solution (35%) is much higher than the mentioned ones. The rest of the compared works do not provide information on this metric.

TABLE VII
COMPARISON OF RESULTS WITH OTHER WORKS

	Our solution	ESP mesh	BLE mesh	BLE mesh	Lora mesh	Lora mesh	ESP-NOW	Zigbee mesh
Latency (ms)	75	300	240	60	300	2000	30	NI
Lost packages(%)	9.25	6	NI	NI	NI	15	NI	35
Obstacles	yes	no	no	no	no	no	NI	no
In/Out	In	Out	Sim	In	Sim	Sim	In	NI
Security	yes	no	no	no	no	no	no	no
Reference	-	[43]	[8]	[44]	[45]	[46]	[47]	[9]

Caption: In-Indoor, Out-Outdoor, Sim-Simulator, NI-not indicate

VII. CONCLUSION AND FUTURE WORKS

A fast and secure mesh network was implemented for home automation systems, independent of the home wi-fi network, therefore, there is no interference between the rest of the home appliances, multimedia devices available in common homes. The ESP-NOW mesh network was integrated into the Hassio home automation platform through a special IoT gateway. A system was implemented to monitor the status of the nodes in the ESP-NOW network in real time by means of a proprietary watchdog system.

Data security is guaranteed throughout the system, through the security protocols implemented in both the ESP-Now mesh and MQTT. However, in future projects, we plan to extend the size of the secure network by adding multiple gateways and design and deploy a set of tests to detect vulnerabilities to possible attacks, with their respective solutions.

REFERENCES

[1] J. Berrú-Ayala, D. Hernandez-Rojas, P. Morocho-Díaz, J. Novillo-Vicuña, B. Mazon-Olivo, and A. Pan, "Scada system based on iot for intelligent control of banana crop irrigation," in *International Conference on Applied Technologies*. Springer, 2019, pp. 243–256.

[2] D. L. Hernández-Rojas, T. M. Fernández-Caramés, P. Fraga-Lamas, and C. J. Escudero, "A plug-and-play human-centered virtual teds architecture for the web of things," *Sensors*, vol. 18, no. 7, p. 2052, 2018.

[3] T. N. Hoang, S.-T. Van, and B. D. Nguyen, "Esp-now based decentralized low cost voice communication systems for buildings," in *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, 2019, pp. 108–112.

[4] J. N. Vicuña, D. L. H. Rojas, B. M. Olivo, and K. D. C. Elizaldes, "Monitoreo inalámbrico de señales eléctricas de voltaje 110/220v a través de arduino," *Alternativas*, vol. 19, no. 1, pp. 55–62, 2018.

[5] D. Eridani, A. F. Rochim, and F. N. Cesara, "Comparative performance study of esp-now, wi-fi, bluetooth protocols based on range, transmission speed, latency, energy usage and barrier resistance," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2021, pp. 322–328.

[6] R. Rizal Isnanto, Y. Eko Windarto, J. Imago Dei Gloriawan, and F. Noerdiyan Cesara, "Design of a robot to control agricultural soil conditions using esp-now protocol," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–6.

[7] X. Wen and Y. Wang, "Design of smart home environment monitoring system based on raspberry pi," in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 4259–4263.

[8] L. Leonardi, G. Patti, and L. Lo Bello, "Multi-hop real-time communications over bluetooth low energy industrial wireless mesh networks," *IEEE Access*, vol. 6, pp. 26 505–26 519, 2018.

[9] H. Yuliandoko and A. Rohman, "Flooding detection system based on water monitoring and zigbee mesh protocol," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2019, pp. 385–390.

[10] M. S. Amjad, J. Wan, H. Li, X. Chen, and J. Yao, "Latency measurement and analysis of z-wave network in smart home environment," in *Advances in Information and Communication*, A. Kiv, T. D'Hondt, and J. Steinberger, Eds. Cham: Springer International Publishing, 2020, pp. 330–340.

[11] E. Feki, K. Kassab, and A. Mami, "Integration of the small board computers rasp berry pi in home automation based on knx protocol," in *2019 IEEE 19th Mediterranean Microwave Symposium (MMS)*, 2019, pp. 1–4.

[12] W. You and H. Ge, "Design and implementation of modbus protocol for intelligent building security," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, 2019, pp. 420–423.

[13] J. J. C. Calva, D. L. H. Rojas, R. F. M. Román, and C. D. R. García, "Seguridad iot: Principales amenazas en una taxonomía de activos," *HAMUT'AY*, vol. 7, no. 3, pp. 51–59, 2021.

[14] W. Liu, Y. Li, J. Li, Y. Xue, and B. Li, "Vulnerability analysis of esp-mesh protocol in smart home," *Security and Communication Networks*, vol. 2021, pp. 1–11, 2021.

[15] espressif. (2022) Esp-now. [Online]. Available: https://github.com/espressif/esp-now

[16] C. M. Tupas Castro, A. Sharma, D. Sampath Kumar, K. Abidi, and N. Kim, "The implementation of thread network for a smart factory," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 253–260.

[17] M. G. Gómez, "Sistema de control de casa inteligente utilizando el protocolo esp-now: Smart-home control system using the esp-now protocol," *Investigación y Ciencia Aplicada a la Ingeniería*, vol. 4, no. 24, pp. 1–6, 2021.

[18] A. U. Khan, M. E. Khan, M. Hasan, W. Zakri, W. Alhazmi, and T. Islam, "An efficient wireless sensor network based on the esp-mesh protocol for indoor and outdoor air quality monitoring," *Sustainability*, vol. 14, no. 24, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/24/16630

[19] G. Facchinetti, G. Petrucci, B. Albanesi, M. G. De Marinis, and M. Piredda, "Can smart home technologies help older adults manage their chronic condition? a systematic literature review," *International Journal of Environmental Research and Public Health*, vol. 20, no. 2, 2023. [Online]. Available: https://www.mdpi.com/1660-4601/20/2/1205

[20] M. L. Daniel, G. G. Cristian, P. G.-B. Cristina, and M. C.-L. Juan, "Bilrost: Handling actuators of the internet of things through tweets on twitter using a domain- specific language," *International Journal of*

Interactive Multimedia and Artificial Intelligence, vol. 6, no. 6, pp. 133–144, 2021.
[21] R. Berjón, M. Mateos, M. E. Beato, and A. Fermoso García, "An event mesh for event driven iot applications," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 6, pp. 54–59, 2022.
[22] P. Sharma, S. Namasudra, R. Gonzalez Crespo, J. Parra-Fuente, and M. Chandra Trivedi, "Ehdhe: Enhancing security of healthcare documents in iot-enabled digital healthcare ecosystems using blockchain," *Information Sciences*, vol. 629, pp. 703–718, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025523001639
[23] X. Zhu and H. Deng, "A security situation awareness approach for iot software chain based on markov game model," *Special Issue on Multimedia Streaming and Processing in Internet of Things with Edge Intelligence*, vol. 7, no. 5, pp. 59–65, 2022.
[24] S. Das and S. Namasudra, "A lightweight and anonymous mutual authentication scheme for medical big data in distributed smart healthcare systems," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–12, 2022.
[25] C. Zhang, B. Vinodhini, and B. A. Muthu, "Deep learning assisted medical insurance data analytics with multimedia system," *ijimai journal*, 2023.
[26] painlessMesh. (2016) Painlessmesh. [Online]. Available: https://gitlab.com/painlessMesh/painlessMesh
[27] N. Reynolds. (2021) m2mmesh. [Online]. Available: https://github.com/ncmreynolds/m2mMesh
[28] arttupii. (2022) Espnow flooding mesh library. [Online]. Available: https://github.com/arttupii/espNowFloodingMeshLibrary
[29] ESP32Home. (2020) Espnowwrapper. [Online]. Available: https://github.com/ESP32Home/ESPNowWrapper
[30] A. Zholtikov. (2022) Zhnetwork. [Online]. Available: https://github.com/aZholtikov/ZHNetwork#esp-now-based-mesh-network-for-esp8266esp32
[31] D. Holdeman. (2017) Nowmesh. [Online]. Available: https://github.com/chuckwagoncomputing/NowMesh
[32] M. F. Wicaksono and M. D. Rahmatya, "Iot for residential monitoring using esp8266 and esp-now protocol," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 8, no. 1, pp. 93–106, 2022.
[33] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, and A. Pan, "Rules engine and complex event processor in the context of internet of things for precision agriculture," *Computers and Electronics in Agriculture*, vol. 154, pp. 347–360, 2018.
[34] A. Alzahrani and T. H. H. Aldhyani, "Artificial intelligence algorithms for detecting and classifying mqtt protocol internet of things attacks," *Electronics*, vol. 11, no. 22, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/22/3837
[35] A. M. Campoverde, D. L. Hernández, and B. E. Mazón, "Cloud computing con herramientas open-source para internet de las cosas," *Maskana*, vol. 6, pp. 173–182, 2015.
[36] D. L. Hernández-Rojas, T. M. Fernández-Caramés, P. Fraga-Lamas, and C. J. Escudero, "Design and practical evaluation of a family of lightweight protocols for heterogeneous sensing through ble beacons in iot telemetry applications," *Sensors*, vol. 18, no. 1, p. 57, 2017.
[37] D. Hernandez-Rojas, B. Mazon-Olivo, J. Novillo-Vicuña, C. Escudero-Cascon, A. Pan-Bermudez, and G. Belduma-Vacacela, "Iot android gateway for monitoring and control a wsn," in *International Conference on Technology Trends*. Springer, 2017, pp. 18–32.
[38] JoelDR. (2023) Bram-now. [Online]. Available: https://github.com/JoelDR/BRAM-NOW-mesh
[39] M. Al-Mashhadani and M. Shujaa, "Iot security using aes encryption technology based esp32 platform." *Int. Arab J. Inf. Technol.*, vol. 19, no. 2, pp. 214–223, 2022.
[40] F. E. Potestad Ordóñez, E. Tena Sánchez, M. d. P. Parra Fernández, M. d. C. Baena Oliva, A. J. Acosta Jiménez, M. Valencia Barrero, and C. J. Jiménez Fernández, "Metodología de diseño para la detección de fallos en cifradores de bloques basada en códigos de hamming," *Sinergías en la investigación en STEM*, 2022.
[41] Node-RED. (2022) Node-red guide. [Online]. Available: http://noderedguide.com
[42] J.-H. Park, H.-S. Kim, and W.-T. Kim, "Dm-mqtt: An efficient mqtt based on sdn multicast for massive iot communications," *SENSORS*, vol. 18, no. 9, SEP 2018.
[43] E. Wang, L. Xiao, X. Han, B. Tan, and L. Luo, "Design of an agile training system based on wireless mesh network," *IEEE Access*, vol. 10, pp. 84 302–84 316, 2022.
[44] M. Wang, Y. Li, J. Lv, Y. Gao, C. Qiao, B. Liu, and W. Dong, "Ace: A routing algorithm based on autonomous channel scheduling for bluetooth mesh network," *Electronics*, vol. 11, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/1/113
[45] A. Marahatta, Y. Rajbhandari, A. Shrestha, A. Singh, A. Thapa, F. Gonzalez-Longatt, P. Korba, and S. Shin, "Evaluation of a lora mesh network for smart metering in rural locations," *Electronics*, vol. 10, no. 6, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/6/751
[46] V. D. Pham, D. T. Le, R. Kirichek, and A. Shestakov, "Research on using the aodv protocol for a lora mesh network," in *Distributed Computer and Communication Networks*, V. M. Vishnevskiy, K. E. Samouylov, and D. V. Kozyrev, Eds. Cham: Springer International Publishing, 2020, pp. 149–160.
[47] K. Khanchuea and R. Siripokarpirom, "A multi-protocol iot gateway and wifi/ble sensor nodes for smart home and building automation: Design and implementation," in *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2019, pp. 1–6.

Joel A. Cujilema Paguay IT engineering student at UTMACH. Current research in IoT and embedded systems. Active member of the AutoMathTIC research group.

Gustavo A. Hidalgo Brito IT engineering student at UTMACH. Current research in IoT and embedded systems. Active member of the AutoMathTIC research group.

Dixys L Hernandez Rojas Electronic Engineer and Master in Electronics from UCLV - Cuba. PhD in ICT-Mobile Networks from UDC - Spain. Expert in full-stack embedded systems. Current research line in IoT, Augmented Reality, Blockchain and IoT Security. Full professor and researcher at UTMACH where he leads the AutoMathTIC research group.

Joffre J Cartuche Calva Computer Systems Engineer from ESPOCH. Master's degree in software engineering from ESPE. PhD candidate in ICT-Mobile Networks at UDC, Spain. Lines of research in software engineering, project management and IoT security. He has worked in public and private companies and currently teaches at UTMACH.