Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v10. n.50, p. 238-245. November/ December., 2024.
DOI: https://doi.org/10.5935/jetia.v10i50.1452

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

EVALUATION OF WI-FI MESH NETWORKS FOR READING ELECTRICITY CONSUMPTION

Carlos Bazán Prieto[1] and Alberto Bazán Guillén[2]

1 Universidad Central "Marta Abreu" de Las Villas – UCLV. Santa Clara - Villa Clara, Cuba.
2 Universitat Politècnica de Catalunya – UPC. Barcelona, España.

[1] http://orcid.org/0000-0002-2810-9599, [2] http://orcid.org/0000-0001-8634-6907

Email: cabazan@uclv.edu.cu, alberto.bazan@upc.edu

ARTICLE INFO

Article History
Received: November 28, 2024
Revised: November 30, 2024
Accepted: December 10, 2024
Published: December 31, 2024

Keywords:
Energy meters,
Wi-Fi mesh networks,
ESP-12.

ABSTRACT

In Smart electrical networks, the Advanced Metering Infrastructure allows bidirectional communication between the service company and the customers. It includes smart meters and a communication infrastructure, which among other functions, is responsible for reading electricity consumption and billing. Cable or wireless technologies are used for the exchange of information between the networks that comprise it. In Cuba there is no such Advanced Metering Infrastructure, the reading of electricity consumption is carried out manually by an operator, reader-collector. That is why we are working on a project where traditional meters are modified by adding a Wi-Fi communications module that allows wireless access. This work evaluates the possibility of interconnecting several modified meters with wireless modules to form a mesh network. Protocols, programming tools, metrics to evaluate performance, scenarios, and experiments are described. As a result, the possibility of forming a mesh network is verified, with the exchange of information, with the requirements of self-configuration and self-repair of the network. This allows the reader-collector to read consumption from any point on the network through a smart device.

I. INTRODUCTION

Smart grids (SG) are a solution to supply electricity from generation points to consumers, which adapts to the current and future energy expectations of humanity. Different definitions of SG are presented in [1–3], from which it is described as an advanced electrical network with a bidirectional energy flow that spans from the point of generation to consumption. It uses Information and Communications Technologies (ICT) to improve the reliability, security and efficiency of the electrical system. It has the ability to self-repair, adapt, be resistant and sustainable, as well as open to future modifications, as it is equipped with present and future standards for components, devices and systems in general. A series of technologies, operations and various smart devices are involved here, such as Smart Meters (SM), smart appliances, renewable energy resources, electric vehicles, flexible charging, smart dialers, various energy efficiency programs and smart end users [1]. The SG architecture consists of three main systems: energy, communication and information. These require the use of ICT for the exchange of data at high speeds while maintaining reliability and security. The Advanced Metering Infrastructure (AMI) is what makes these operations possible, allowing the management and regulation of energy generation and consumption. AMI also allows bidirectional communication between the service provider company and customers, which includes consumption billing data. In Cuba there is no AMI, its implementation would require changes to equipment, electrical and communications networks at a very high cost. The Cuban residential sector represents more than 60% of total consumption, has approximately 4,056,865 users with an average monthly consumption of 185kWh [4], with traditional meters. An operator, called a reader-collector, performs the tasks of reading electricity consumption and collecting the bill; both processes are carried out manually and monthly. The operator must move on foot through the area assigned to him. For reading, a small terminal (PDA) or more recently a mobile device with the appropriate application and in some cases infrared reading equipment is used. The limitation in both cases is that the operator has to have access to the meter or at least a very close line of sight to it. This makes reading impossible in some scenarios, for example, a closed house. In addition, there may be errors in manual reading, anomalies in consumption due to lack of appropriate measurements, readings outside established dates, cases of fraud, and others. On the other hand, savings are not encouraged or consumption habits are influenced. The amount to be charged each month is derived only from the notation of the previous and current month's reading.

Based on the above, a low-cost alternative solution was proposed, with free software and hardware. This alternative reuses traditional residential meters and adds reading facilities to users and the reader-collector [5]. The alternative proposes adding an electronic device to existing residential energy meters, capable of reading consumption remotely, turning them into advanced energy meters. The electronic device contains, among other elements, an ESP-12 module (ESP8266 Ai-Thinker) [6] that allows the exchange of consumption data via Wi-Fi. The communication capabilities as a base radio, as well as a Wi-Fi access point, give this advanced meter the facilities for local communication with the user [5]. This technology could also be used for the interconnection between modified meters, through Wi-Fi mesh networks, more specifically the ESP-WIFI-MESH networks, derived from them. This operation represents a saving of resources, expanding the network coverage area, which would allow the reader-collector to read the electricity consumption in an easier and faster way.

ESP-WIFI-MESH is a network protocol built on the Wi-Fi protocol that allows numerous devices (nodes) spread over a large physical area to be interconnected under a single network. The network is capable of self-organizing and self-repair. In [7] a wireless mesh network based on ESP8266 devices is evaluated using the PainlessMesh library, to collect data such as the temperature and humidity of fruit boxes or containers when they are stored or transported in refrigerated chambers. The performance of the network is evaluated in terms of delivery ratio and delivery delay, which consequently affects the energy consumption and therefore the lifetime of the network.

The performance results were affected by the increase in the number of sensor nodes, message sending rate, and message payload size, as expected. In [8], a mesh network is built using NodeMCU ESP 8266 and NodeMCU ESP 32 with two types of sensors, DHT 11 and DHT 22. The communication delay with line of sight and without line of sight is evaluated. The results give a shorter delay time in the line-of-sight condition for all connected nodes. In [9] the performance of ESP8266 mesh network has been tested and evaluated. A network consisting of two nodes has a single-hop delay of 2.49 ms.

As expected, a larger number of nodes tends to increase the network delay even for the same hop distance. An evaluation of data rate performance is also performed. It is found that a node can receive up to 461 messages/sec and 28 messages/sec for a payload of 10 bytes and 4400 bytes, respectively. It is also observed that sending messages with a payload greater than 4400 bytes results in broken and incomplete messages. With ESP 32, in [10] a MESH network of sensors is established using an ESP-MESH network protocol for real-time indoor and outdoor air quality monitoring. The performance of the MESH network is estimated in terms of packet loss rate (PLR), packet failure rate (PFR), and packet delivery rate (RPD). The RPD value is greater than 97% and the PMR and PER value for each active node is less than 1.8%, which is below the limit. The results show that the ESP-MESH network protocol offers considerably good quality of service, mainly for medium area networks.

This work aims to evaluate the possibility of interconnecting several modified energy meters with the ESP-12 modules, to form a mesh network, to increase the coverage area and facilitate the reading process.

II. COMMUNICATIONS NETWORK

By adding the ESP-12 module to traditional residential energy meters, one or more users can interact with the device via Wi-Fi. However, since there is no form of communication between the modified energy meters, the collection of information from the reader-collector is difficult. Since to read it, it must access each energy meter, extract its data and move on to the next one. The protocol and network functions related to the interconnection facilities of a mesh network are described below. As well as, the programming tools available for the mesh network, which allow better interaction between its nodes.

II.1 ESP-WIFI-MESH NETWORK PROTOCOL

ESP-WIFI-MESH [11] is a network protocol built on the Wi-Fi protocol that allows numerous devices (ESP-12 nodes) spread over a large physical area to be interconnected under a single network. The network is capable of self-organizing after the selection of a root node from which downward connections are formed layer by layer until all nodes have joined the network. It is also capable of self-repair by detecting and correcting failures that occur when the connection between any of the nodes is interrupted, or it becomes unstable.

With ESP-WIFI-MESH the nodes are not required to connect to a single central node (unlike Wi-Fi networks that have an AP and each node establishes a connection with it). Instead, connections are established between neighboring nodes, and each of them is mutually responsible for relaying each other's transmissions. Therefore, this type of networks has a much larger coverage and is less susceptible to overload, since the number of nodes allowed in the network is not limited by a single central node [11]. ESP-WIFI-MESH allows nodes to simultaneously act as a station and an AP. Therefore, a node can have multiple downstream connections using its softAP (software-defined access point) interface and a single upstream connection using its station interface. Therefore, a tree-shaped network topology is obtained with a parent-child hierarchy formed by multiple layers as shown in Figure 1. On the other hand, ESP-WIFI-MESH is a multi-hop network. Therefore, nodes transmit their own packets and simultaneously serve as repeaters for other nodes. This allows any pair of nodes within the network to communicate as long as any path exists at the physical layer (through one or more wireless hops). The total number of nodes that the ESP-WIFI-MESH network can support depends on the maximum number of layers allowed in the network and the maximum number of downstream connections that each node can have, both variables can be configured to limit the size of the network [11].

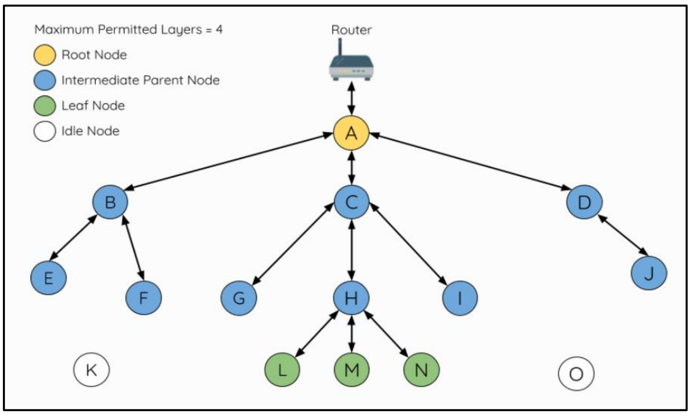

Figure 1: Types of nodes in ESP-WIFI-MESH.
Source: [11], (2024).

ESP-WIFI-MESH recognizes four types of nodes shown in Figure 1. The top node of the network is called the root node (node A). It serves as the only interface between the ESP-WIFI-MESH network and an external IP network, only one can exist on the network and its upstream connection is a conventional router. The nodes located in the maximum allowed layer of the network are assigned as leaf nodes (L, M and N nodes). They are not allowed to have downstream connections (child nodes), therefore they can only transmit or receive their own packets, but not forward packets from other nodes. These nodes ensure that no more layers are added to the network. Some nodes without a softAP interface are also designated in this category since they do not meet the requirements to establish downward connections. The intermediate nodes to the previous classifications are the so-called intermediate parent nodes (nodes B, C, D, E, F, G, H, I and J). These nodes must have a single upstream connection (a single parent node), and can have zero to multiple downstream connections (child nodes). They are capable of transmitting and receiving packets, but also forwarding packets sent from their upstream and downstream connections. Nodes of this type that do not have child nodes (nodes E, F and J) are not classified as leaf nodes, as they can form descending connections in the future. Finally, inactive nodes (K and O nodes) are those that have not yet joined the network, but may form an upstream connection with an intermediate parent node or will try to become the root node under the right circumstances [11].

Nodes that can form downstream connections transmit signaling frames periodically, allowing other nodes to detect their presence and know their status. Inactive nodes receive the frames to generate a list of possible parent nodes, with one of which they will form an upstream connection. To prevent this connection from being weak ESP-WIFI-MESH implements a received signal strength indication (RSSI) threshold mechanism for signaling frames. If a node detects a signaling frame with an RSSI lower than a pre-configured threshold, the transmitting node will not be considered to form an uplink. To choose the parent node among the possible candidates, the inactive node takes into account the layer where the candidate is located and the one located in the shallowest layer of the network (including the root node) has priority, in order to minimize the number total network layers. If there are several parent node candidates within the same layer, it will be preferred to have fewer child nodes, in order to balance the number of downward connections between nodes in the same layer. On the other hand, ESP-WIFI-MESH allows the user to define their own algorithm to select a predefined parent node, or force a node to only connect to a specific parent node [11].

II.2 NETWORK CONSTRUCTION

Building an ESP-WIFI-MESH network involves selecting a root node from which downward connections are formed layer by layer until all nodes have joined the network. Before starting the network formation, each node must be configured with the same: mesh network identifier, router configuration and softAP configuration. The root node can be designated during configuration or chosen dynamically based on the signal strength between each node and the router. Once selected, it will create an upstream connection with the router and from there the second layer will be formed with the inactive nodes in the range of the root node, becoming intermediate parent nodes. The remaining inactive nodes will connect to the intermediate parent nodes within their range, becoming intermediate parent nodes or leaf nodes, depending on the number of layers allowed in the network. The process is repeated until there are no idle nodes left in the network or until the maximum layer allowed in the network has been reached, therefore the remaining nodes will remain idle indefinitely.

In automatic root node selection, each idle node will transmit its MAC address (to identify each node in the network) and its RSSI value with the router through Wi-Fi signaling frames. Each node will then simultaneously search for signaling frames from other idle nodes, and if a node detects a frame with a stronger RSSI it transmits the contents of the frame. The process is repeated for a pre-configured minimum number of iterations and will result in the signaling frame with the strongest RSSI being propagated throughout the network. Finally, each node individually checks its vote percentage (number of votes/number of nodes participating in the election) and the one that contains a value greater than a pre-configured threshold will become the root node of the network. If more than one node meets the threshold requirements, there will be several root nodes within the same ESP-WIFI-MESH network, resulting in the creation of multiple networks. This conflict can be resolved automatically by internal ESP-WIFI-MESH mechanisms. However, conflicts where two or more root nodes have the same SSID (public name of a WLAN that serves to differentiate it from other wireless networks in the area) are not handled. Once a root node is chosen, ESP-WIFI-MESH does not automatically change it unless it is disconnected from the network.

If some nodes in the network are powered on asynchronously (separated by several minutes) the final structure of the ESP-WIFI-MESH network may differ from the ideal case where all nodes are powered on at the same time. The lagging node should join the network like any other idle node, even though it may have a stronger RSSI with the router. If this were the designated root node, all other nodes in the network will remain idle until the lagging node is powered on. Likewise, if it is a designated parent node, its child will remain inactive until it powers on. Finally, child nodes can change their upstream connections to another parent node in a lower layer, this process occurs autonomously and following the same threshold mechanism explained [11].

II.3 NETWORK SELF-REPAIR

ESP-WIFI-MESH can detect and correct failures that occur when the connection between any of the nodes is interrupted, or becomes unstable. Child nodes will autonomously select a new parent node and form an upstream connection to it to maintain network interconnectivity. If the root node fails, it will be detected by the nodes connected to it (second layer), which will initially try to reconnect to the root node. After several failed attempts, the second layer nodes will start a new round of electing a new root node following the process described above, and the remaining nodes will form upstream connections with it or a neighboring parent node if they are not in range. If the root node and several nearby layers fail, the top layer that is still functioning will begin the root node election process. When an intermediate parent node fails, a similar procedure is followed since the disconnected child nodes will search for possible parent nodes. Then each child node will individually select a new parent node and form an upstream connection with it. If there are no potential parent nodes for any child node, it remains inactive indefinitely [11].

II.4 PROGRAMMING TOOLS

The painlessMesh library [12] allows the implementation of the ESP-WIFI-MESH protocol and creating mesh networks using ESP-12 modules. Below are the functions that are of interest for this job:

- *void receivedCallback(uint32_t from, String &msg):* every time this node receives a message, this routine will be called, where *from* is a variable of type integer and size 32 bits corresponding to the identification of the sender of the message and *msg* is the message sent (in this case a string, but it can be any type of data).
- *void newConnectionCallback(uint32_t nodeId):* is triggered every time a node establishes a new connection, where *nodeId* is the ID of the new node connected to the mesh.
- *void changedConnectionCallback():* runs every time a connection on the network changes (a node leaves or joins the network).
- *void nodeTimeAdjustedCallback (int32_t offset):* it is executed when the network adjusts the time, so that all nodes are synchronized, offset is the offset.

III. MATERIALS AND METHODS

To evaluate the possibility of interconnecting several modified energy meters with the ESP-12 modules, the parameters to be evaluated are defined and three scenarios are formed with characteristics of mesh networks. With these conditions, the verification experiments are carried out [13].

III.1 PARAMETERS FOR NETWORK EVALUATION

It is necessary to define certain parameters to measure the quality of the ESP-WIFI-MESH network, as well as evaluate its behavior in different scenarios. In this work, the following are used: the power of the received signal as a measure of quality, as well as the network discovery time and the network recovery time as distinctive measures of this type of networks.

The signal power level (in dBm) perceived by a device in wireless networks (RSSI) [14] is the parameter used by Wi-Fi transceivers to adapt their modulation and coding scheme so that the optimal data rate. The modulation scheme that supports high-speed transmission requires a high signal-to-noise ratio, which is correlated with a strong RSSI.

The network discovery time or convergence time is defined in [15] as the time interval that elapses from the start-up of the devices until the complete formation of the network. The latter refers to the state in which each node is able to communicate with any other node through a route defined in its routing table. To carry out the measurement, each node was configured to periodically send a "measurement update" packet. To quantify the time it takes a node to discover the network, each node records a timestamp when it starts up (*t_start*). Every time a node discovers a network and decides to join it, it records another timestamp (*t_join*), then the difference between both timestamps (*t_join-t_start*) is the time it took the node to discover the network [16]. The value of this difference corresponding to the last node to join the network is the network discovery or convergence time.

The recovery or self-repair time of ESP-WIFI-MESH networks is defined in [16] as the recovery latency parameter. Which is the amount of time it takes the network to detect a node breakdown and take the appropriate actions to repair the network. To measure the same, whenever a node detects that it may have lost connectivity, it records a timestamp (*t_(connection lost)*) and initiates a connectivity test. If the result confirms the loss of connectivity, the node is released from said link and begins a channel exploration procedure (other possible connections). When it discovers a hub in the vicinity and decides to join it, it records another timestamp (*t_join*). The difference between both brands (*t_join-t_(connection lost)*) is the time it took for the node to recover connectivity and therefore, the time it took for the network to recover.

III.2 DESCRIPTION OF THE SCENARIOS AND EXPERIMENTS

The experiments are carried out from three ESP-12 modules (nodes) whose Id are shown in Figure 2, which illustrates a possible network topology with two levels. There is one ESP-12E nodeMCU and two ESP-12E SoM. The dashed lines represent the Wi-Fi connections between them.

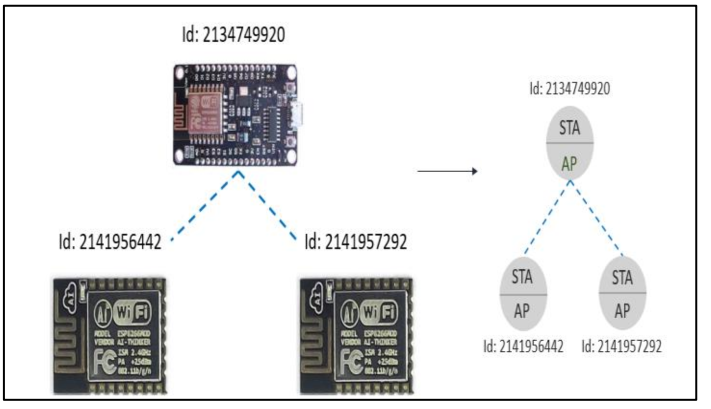

Figure 2: ESP-WIFI-MESH Network.
Source: Authors, (2024).

In addition, a possible network diagram is shown, where the stations (STA) of the modules with Id 2141956442 and 2141957292 respectively, connect to the access point (AP) of the remaining module (Id 2134749920). Therefore, the node with Id 2134749920 acts as a bridge (root), so that the two remaining nodes (intermediate parents) can establish communication.

Three scenarios were implemented in adjoining rooms and exterior hallways, separated by reinforced concrete walls, the details of which are shown in Figure 3. The legend specifies the meaning of each shape belonging to Figure 3 and the black arrows are the dimensions of distance between the nodes. Finally, the blank spaces on the walls in Scenario 3 represent the doors to the rooms.

These scenarios are equivalent to the situations of several energy meters in the same room, in the case of an apartment building. As well as several energy meters that are in separate houses a short distance away, located inside and outside the houses. In scenarios 1 and 2, the nodes have a direct line of sight, only the distances vary, while in scenario 3, two of the modules are inside the rooms and a third in the hallway, therefore, there is no line of sight between any of them.

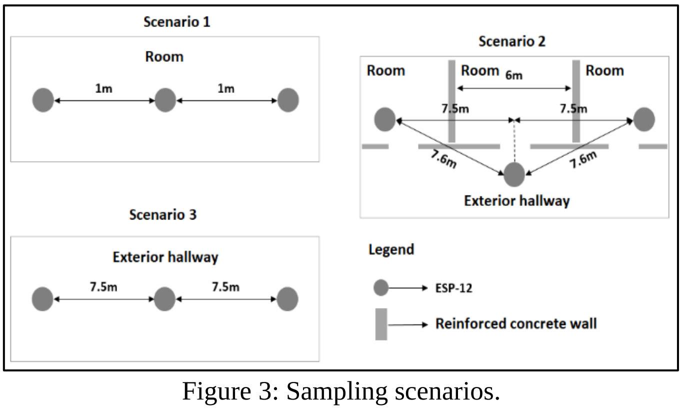

Figure 3: Sampling scenarios.
Source: Authors, (2024).

The first experiment is to verify that the three nodes can connect to each other forming the ESP-WIFI-MESH network. To verify that the connection is established, all nodes send a broadcast message, and the recipient prints the messages received through its serial port. A web server is used with the objective of simulating a possible interface between the reader-collector and the network of modified energy meters, where information on the connection states is also displayed. Actions were carried out on the network to evaluate its response to situations that correspond to real problems. For example, modules were intentionally turned off (simulating the possible failure of a modified energy meter) and a module that was turned off during network establishment was turned on (simulating a new modified energy meter in the area), among others. The network should not be affected in any case due to its self-healing and self-configuring properties.

The second experiment is aimed at evaluating the parameters in different scenarios, with the greatest amount of data possible, and presenting the result of the average values in each case. By having three modules, disconnecting an extreme node from the network (a node that only has a connection with another node) does not provide information. In this case, the network loses a node. However, since there is no bridge connection between the two remaining nodes, an interruption does not occur, only the topology changes.

IV. RESULTS AND DISCUSSIONS

The first connectivity experiment is performed in a local scenario like scenario 1 in Figure 3, using broadcast messages. By exchanging information between the three nodes, the self-configuration of the ESP-WIFI-MESH networks was verified, since the network was created automatically and all the nodes were able to send messages between them.

The reader-collector must be able to collect data from the modified energy meters by connecting with their smart device (phone, tablet, laptop, among others) to the network formed by them. To test this function, we create a web server and a function that returns the connection status in the form of a table. The columns **Id Node**, **Value** (in this case a random value from 1000 to 5000 will be displayed that simulates the current electricity consumption), the **Connection status** and the **Id** of the nodes that are connected to the network. In addition, a "*****" is used next to the Id of the node that serves as the AP for the intelligent device that simulates the reader-collector interface. The result of this operation is shown in Figure 4, where a diagram of the possible topology of said network is also shown.

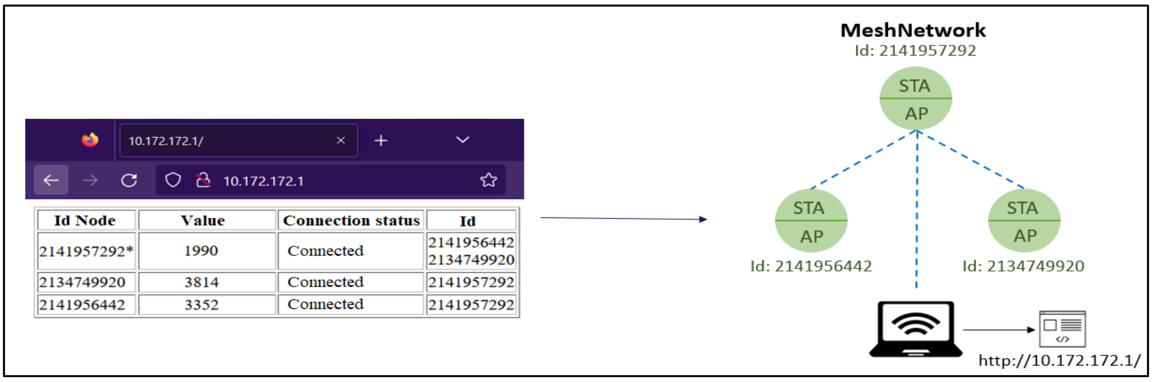

Figure 4: Network with all nodes connected.
Source: Authors, (2024).

The green circles in Figure 4 represent the turned on nodes, the blue dashed lines represent the Wi-Fi connections between all devices, and the web page that the laptop accesses is illustrated by specifying its domain. In this case, the network is accessed through the node with Id 2141957292, this means that the laptop is connected to the AP of that node. If a node leaves the network, it reorganizes itself due to its self-healing and self-configuring features. To verify this, the module that acted as a bridge node between the rest (Id: 2141957292) was intentionally turned off. In Figure 5, the turned off module is represented in gray, and it is shown how the remaining nodes connect to each other and the laptop accesses the network through another node. The Id of the nodes with which the connection is established is displayed, "-" indicates that the node has no connections because it is not active on the network. This process was done automatically and takes a few seconds. Then the module was turned on again, and in seconds, it was integrated back into the network forming a different topology.

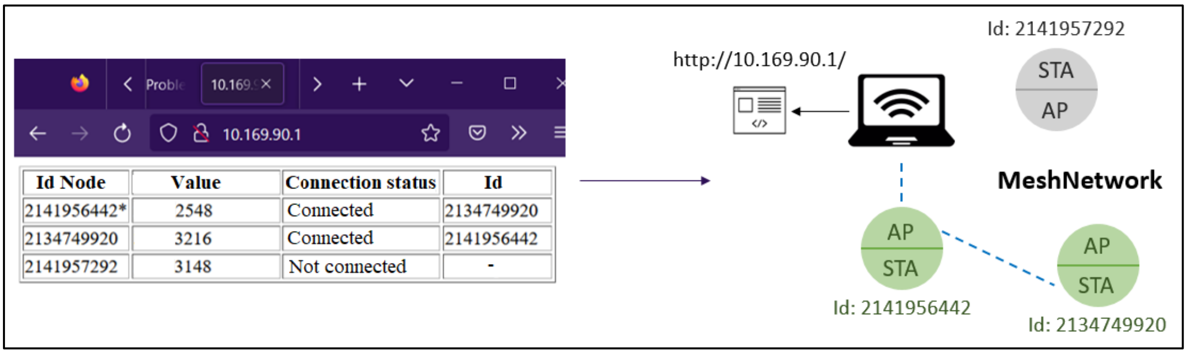

Figure 5: Network with two connected nodes.
Source: Authors, (2024)

After establishing the network and checking its correct operation, the code was modified with the objective of evaluating the previously defined parameters, with the second experiment. Network discovery and recovery times were calculated from a list of network events, which are broadcast messages that are modified each time a change occurs in the network topology. The values are sent through the serial port and are managed through another web page. The parameters are evaluated in the three scenarios. The results of the evaluation of scenario 1 are presented below, as an example.

The results of each of the five tests in scenario 1 are shown in Table 1. For each test, several measurements are made and the average values are presented. The RSSI values are in the range of -30dBm to -60dBm, which are associated with an ideal signal, which was expected, since there are no obstacles and the distance between the ESP-12 is small. From the average times in each test, it is obtained that in scenario 1 the network discovery time is approximately 2.6122 seconds.

Table 1: Network and RSSI discovery times for scenario 1.

Test	Parameter	Id: 2134749920	Id: 2141956442	Id: 2141957292	Average
1	Time (ms)	2951	2224	2874	2683
	RSSI (dBm)	-	-55	-48	-
2	Time (ms)	2708	2287	2364	2453
	RSSI (dBm)	-51	-50	-	-
3	Time (ms)	2610	2209	2745	2521.33
	RSSI (dBm)	-57	-47	-	-
4	Time (ms)	2910	2206	3053	2723
	RSSI (dBm)	-51	-48	-	-
5	Time (ms)	2813	2280	2949	2680.67
	RSSI (dBm)	-45	-54	-	-

Source: Authors, (2024).

In some of the tests, the node that functioned as a bridge for the remaining nodes was intentionally disconnected, to measure the network's recovery time in the event of its loss. Figure 6 shows the network topologies before and after said action, where the green color indicates that the node is on and the gray color is off, the black arrows correspond to distance levels and the blue dashed lines represent the connections through Wi-Fi of the nodes. Where Id 2141957292 identifies the bridge node. By disconnecting the bridge node, the remaining nodes lose connection between them and are unable to exchange information. Figure 7 shows the events corresponding to the node with Id 2141956442. After approximately 14,854 seconds (in Figure 7, t = 361053 - 346199) the network is reestablished, with a new topology. This time is calculated taking into account the times of the node with Id 2141956442, the same procedure is followed for the network discovery time, calculating the recovery times seen from each node and averaging them. In Figure 6, the RSSI value in the node with Id 2141956442 also increases as the distance with its new AP decreases.

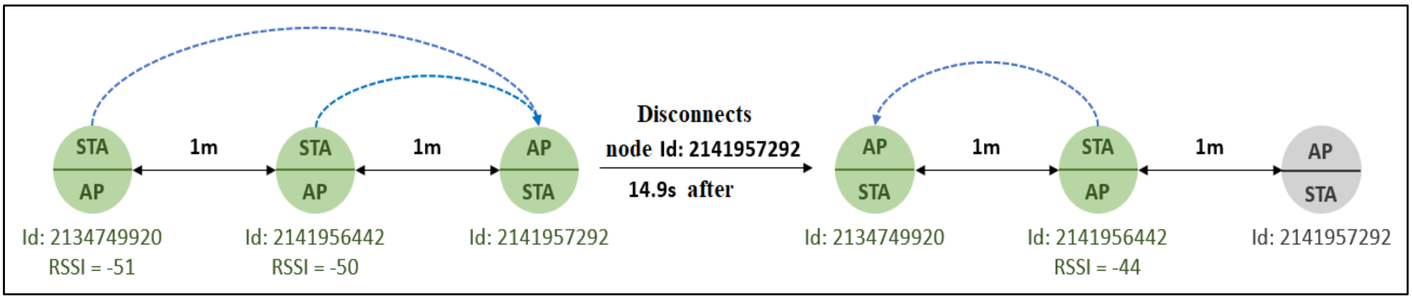

Figure 6: Topology changes in test 2.
Source: Authors, (2024).

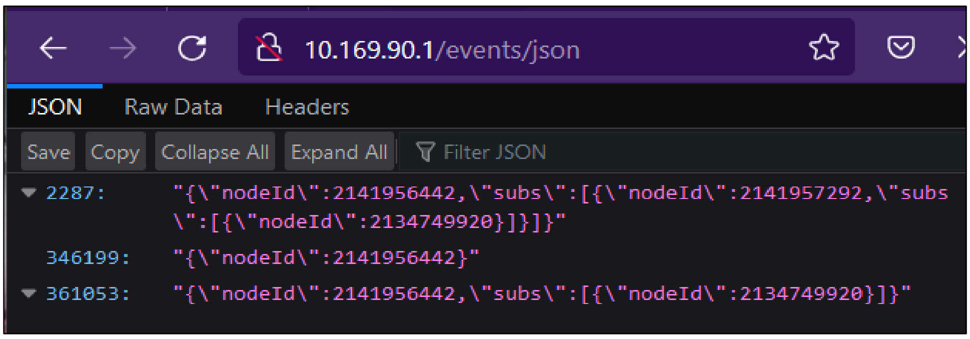

Figure 7: Events of the node with Id 2141956442 in Test 2.
Source: Authors, (2024).

Similar treatment was carried out with each of the scenarios. Table 2 shows a comparison between the results for each scenario, with the average values of several trials. Network discovery times remain almost constant when only the distance between modules increases (scenarios 1 and 2, in Figure 3), even though the signal strength decreases considerably. Furthermore, when the bridge node was intentionally disconnected from the network, and the distance increased further in some cases, the RSSI decreased as expected. However, when the modules have no line of sight between them by walls (scenario 3 in Figure 3), the network formation time increases by approximately 0.368 seconds (compared to scenario 2 in Figure 3). Obstacles also affected the RSSI, which decreases reaching unacceptable values (less than -90dBm).

Table 2: Comparison of the results of each scenario.

Scenerios	Network discovery time (s)	Network recovery time (s)	RSSI Range
1	2.6122	20.78675	(-45) – (-
2	2.797732	28.126	(-60) – (-
3	3.166268	72.043	(-70) – (-

Source: Authors, (2024).

Network recovery times vary in each of the scenarios. Figure 3 shows that the maximum distance between two nodes is 2m in scenario 1 and 15m for scenarios 2 and 3 respectively. Therefore, when the intentionally disconnected node was the bridge (scenarios 2 and 3 in Figure 3), the distance between the remaining nodes increased. The recovery times for scenarios 1 and 2 differ by 7.34s, only due to the increase in said distance, which brings with it a decrease in the RSSI value. In this case not very considerable, since it continued in the same range as for its initial configuration. However, in scenario 3 the value of this parameter increases considerably (43.917s compared to scenario 2 where the distance remains constant), in addition the RSSI decreases, reaching values below the minimum acceptable signal state. Therefore, the existence of obstacles that reduce the intensity of the received signal (in this case reinforced concrete walls) affects network recovery times more.

V. CONCLUSIONS

It is possible to form an ESP-WIFI-MESH network with the ESP-12 modules using the painlessMesh library, which meets the self-configuration and self-healing requirements. By accessing one of its nodes, the data stored in the remaining nodes can be obtained. Therefore, it is possible to form a mesh network between the modified energy meters, allowing the exchange of information between them and its collection, so that the coverage area is increased. This allows the reader-collector to read consumption from any point on the network through a smart device. The evaluation of the network parameters in the different scenarios also demonstrated its viability for this purpose. However, when deploying the new energy meters it is important to take into account the obstacles and distances between them in each of the areas. The distance between the modules has a direct impact, decreasing the RSSI. This decrease can cause loss of connection stability or the most critical result, a complete loss of connection. On the other hand, some materials (such as reinforced concrete walls) can cause signal attenuations, this results in delays in the discovery time and fundamentally in the network recovery time. Tests must be carried out with the modified energy meters in real environments and the type of antenna to be used must be evaluated, so that the greatest possible coverage is guaranteed.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Carlos Bazán Prieto and Alberto Bazán Guillén.
Methodology: Carlos Bazán Prieto and Alberto Bazán Guillén.
Investigation: Carlos Bazán Prieto and Alberto Bazán Guillén.
Discussion of results: Carlos Bazán Prieto and Alberto Bazán Guillén.
Writing – Original Draft: Carlos Bazán Prieto and Alberto Bazán Guillén.
Writing – Review and Editing: Carlos Bazán Prieto and Alberto Bazán Guillén**.**
Resources: Carlos Bazán Prieto and Alberto Bazán Guillén.
Supervision: Carlos Bazán Prieto and Alberto Bazán Guillén.
Approval of the final text: Carlos Bazán Prieto and Alberto Bazán Guillén.

VII. REFERENCES

[1] S. S. Refaat, O. Ellabban, S. Bayhan, H. Abu-Rub, F. Blaabjerg, and M. Begovic, Smart Grid and Enabling Technologies. 2021. doi: 10.1002/9781119422464.

[2] M. R. Hossain, A. M. T. Oo, and A. B. M. S. Ali, "Smart Grid", in Smart Grids: Opportunities, Developments, and Trends, A. B. M. S. Ali, Ed. London: Springer, 2013, pp. 23–44. doi: 10.1007/978-1-4471-5210-1_2.

[3] G. Dileep, "A survey on smart grid technologies and applications", Renew. Energy, vol. 146, pp. 2589–2625, Feb. 2020, doi: 10.1016/j.renene.2019.08.092.

[4] "Electricidad en Cuba: entre el costo y la tarifa", https://www.juventudrebelde.cu/cuba/2020-12-13/electricidad-en-cuba-entre-el-costo-y-la-tarifa (accessed Oct. 12, 2022).

[5] A. Bazán Guillén, "Infraestructura escalable para lectura remota de contadores de energia de energía eléctrica", Tesis de Maestría en Telemática, Universidad Central "Marta Abreu" de Las Villas, 2022.

[6] "ESP-12E Datasheet", https://datasheet4u.com/datasheet-pdf/AI-Thinker/ESP-12E/pdf.php?id=1258546 (accessed Nov. 03, 2022).

[7] L. Santos, T. Costa, JMLP. Caldeira, VNGJ. Soares. "Performance Assessment of ESP8266 Wireless Mesh Networks", Information. 2022; 13(5):210. https://doi.org/10.3390/info13050210

[8] N.A. Afifie, A.W.Y. Khang, A.S. Bin Ja'afar, A.F.B.M. Amin, W.A. Indra, S.G. Herawan and A.B. Ramli, 2021. "Evaluation Method of Alsayaydeh, J.A.J Mesh Protocol over ESP32 and ESP8266", Baghdad Science Journal. 18, 4(Suppl.) (Dec. 2021), 1397. https://doi.org/10.21123/bsj.2021.18.4(Suppl.).1397

[9] Yoppy, R. H. Arjadi, E. Setyaningsih, P. Wibowo, M. I. Sudrajat, IOP Conf. Series: Journal of Physics: Conf. Series 1230 (2019) 012023, doi:10.1088/1742-6596/1230/1/012023

[10] A.U. Khan, M.E. Khan, M. Hasan, W. Zakri, W. Alhazmi, T. Islam, "An Efficient Wireless Sensor Network Based on the ESP-MESH Protocol for Indoor and Outdoor Air Quality Monitoring", Sustainability. 2022; 14(24):16630. https://doi.org/10.3390/su142416630

[11] "ESP-WIFI-MESH - ESP32 - ESP-IDF Programming Guide v4.4.2 documentation", https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/esp-wifi-mesh.html (accessed Nov. 03, 2022).

[12] G. Martín, "Intro to painlessMesh", Oct. 18, 2022. Accessed: Nov. 05, 2022. [Online]. Available: https://github.com/gmag11/painlessMesh

[13] A. Arkhipenko, "Task Scheduler", Nov. 05, 2022. Accessed: Nov. 05, 2022. [Online]. Available: https://github.com/arkhipenko/TaskScheduler

[14] Yoppy, R. H. Arjadi, H. Candra, H. D. Prananto, and T. A. W. Wijanarko, "RSSI Comparison of ESP8266 Modules", in 2018 Electrical Power, Electronics,

Communications, Controls and Informatics Seminar (EECCIS), Oct. 2018, pp. 150–153. doi: 10.1109/EECCIS.2018.8692892.

[15] E. Giacomini, F. D'Alterio, A. Lacava, and F. Cuomo, "BLUES: A Self-organizing BLE Mesh-network Paradigm for IoT Environments," in 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Aug. 2020, pp. 409–414. doi: 10.1109/WoWMoM49955.2020.00075.

[16] P. Kulkarni, S. Gormus, Z. Fan, and F. Ramos, "AMI Mesh Networks—A Practical Solution and Its Performance Evaluation", IEEE Trans. Smart Grid, vol. 3, no. 3, pp. 1469–1481, Sep. 2012, doi: 10.1109/TSG.2012.2205023.