PERFORMING OPEN SOURCE WIFI MESH SENSOR NETWORK USING ESP32

Mr.Jyoti Morai
PG Research Scholar
Department of Electronic and Communication
Ballarpur institute of Engineering,
Bamni,Ballarshah.

Prof.Sunil Vasant Kuntawar
Assistant Professor,
Department of Electronic and Communication Engineering
Ballarpur Institute of Engineering,
Bamni,Ballarshah

***Abstract*—ESP-MESH is a networking protocol built atopthe Wi-Fi protocol. ESP-MESH allows numerous devices (henceforth referred to as nodes) spread over a large physical area (both indoors and outdoors) to be interconnected under a single WLAN (Wireless Local-Area Network). ESP-MESH is self-organizing and self-healing meaning the network can be built and maintained autonomously.**

Keywords—IoT, Wi-Fi ESP32-Mesh network routing, mesh, auto-configuration

I. INTRODUCTION

Traditional infrastructure Wi-Fi network is a point-to-multipoint network where a single central node known as the access point (AP) is directly connected to all other nodes known as stations. The AP is responsible for arbitrating and forwarding transmissions between the stations. Some APs also relay transmissions to/from an external IP network via a router. Traditional infrastructure Wi-Fi networks suffer the disadvantage of limited coverage area due to the requirement that every station must be in range to directly connect with the AP. Furthermore, traditional Wi-Fi networks are susceptible to overloading as the maximum number of stations permitted in the network is limited by the capacity of the AP.

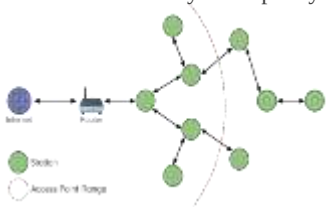


Figure 1-ESP-MESH Network Architecture

ESP-MESH differs from traditional infrastructure Wi-Fi networks in that nodes are not required to connect to a central node. Instead, nodes are permitted to connect with neighbouring nodes. Nodes are mutually responsible for relaying each others transmissions. This allows an ESP-MESH network to have much greater coverage area as nodes can still achieve interconnectivity without needing to be in range of the central node. Likewise, ESP-MESH is also less susceptible to overloading as the number of nodes permitted on the network is no longer limited by single central node

II. LITERATURE REVIEW

These communication systems, involved in the so called “network of networks” Internet of Things (IoT), will eventually allow humans to interact with billions of devices, including sensors, actuators, services, and other connected objects, in an Internet-like way, with a forecast of more than 40 billion connected (with short-range radio communication technologies) “things” by 2020 .In this context, connected things are generally defined as Smart Objects (SOs) and, thanks to the IoT, dynamically integrated in several scenarios, such as: smart industries applications, smart cities, smart agriculture, smart health, etc. Each application area has specific requirements to be taken into account, thus having implications for the communication technologies to be considered and, possibly, adopted. In particular, IoT-related wireless technologies developed in recent years are extremely heterogeneous in terms of protocols, performance, reliability, latency, cost effectiveness, and coverage. For instance, some of them are designed for short-range radio communications (e.g., Bluetooth and ZigBee), others are more suitable to cover wide areas with very small bandwidth (e.g., Sub-GHz), while others are designed for middle-range communications and high transmission rate (e.g., IEEE 802.11). Moreover, IoT network topologies are generally star- or tree-based, with data collected by groups of sensors and sent to a central collector or border router, in order to guarantee centralized processing. The emerging and constantly evolving IoT applications require more complex network topologies, without a predefined hierarchy but able to dynamically adapt themselves to changing conditions. For this reason, there is a strong academic and industrial interest on the development of hardware and protocols able to support Wireless Mesh Networks (WMNs).

In mesh topologies, network nodes are directly and dynamically connected in a non-hierarchical way, thus allowing many-to-many communications (among nodes cooperating with each other) to efficiently route data from a generic source to a generic destination. In fact, in a WMN each node composing the network can operate both as a host and as a router, relaying packets sent by other nodes when the destination is not in the visibility range of the source. Moreover, mesh networks do not require an infrastructure, since they dynamically self-organize and configure themselves, with consequent relevant advantages, in terms of: (i) deployment, installation and maintenance’s overhead and cost reduction; (ii) dynamic workload distribution; (iii) better reaction to node failures; and (iv) easy network topology modification. The organization of a WMN is generally handled through the definition of a routing policy shared among all nodes, aiming at discovering and determining the best routes, on the base of different metrics (e.g., throughput, link quality, hops number, etc.) measured on data streams. Therefore, streams of data in WMNs cross all nodes connecting the source and the destination.

Mesh topologies are thus the most attractive alternative to traditional centralized or tree-based network topologies, where nodes are directly linked to small subset of other nodes and the links between these infrastructure neighbours are hierarchically organized. While star- and tree-oriented topologies are very well established, highly standardized and vendor-neutral, in the case of mesh networks the research community and vendors have not yet all agreed on common standards, with the interoperability among devices from different vendors seldom assured. Moreover, comprehensive surveys on available options in the field of mesh networks for IoT are lacking in the literature.

In this paper, we aim at highlighting how a mesh network can be built under heterogeneous communication technologies, thus providing a comprehensive survey of relevant wireless communication technologies which can be employed in different IoT scenarios, namely: IEEE 802.11-based, Bluetooth, IEEE 802.15.4-based, and LoRa. We analyze these protocols from the point of view of their support to mesh networking, either as native applications or by proper adaptation. To provide a global vision of the state of the art

on WMNs, our survey includes both standard and academic/industrial solutions.

2. *Preliminaries on Routing Protocols for Ad-hoc Networks*

Key features of a mesh network, both wired and wireless, are flexibility and self-organizing capability. To build multi-hop routes and to define the network topology, a mesh network needs a routing protocol to rely on. Such protocol can be derived from routing protocols for ad hoc networks, that can be in turn classified as *proactive* and *reactive* protocols. *Proactive* routing protocols are characterized by the fact that each node is in charge of maintaining single or multiple routing tables to represent the entire network topology (or a portion of it). Therefore, proactive routing protocols are known as "table-driven", meaning that routing information for each tuple of nodes are continously updated, thus maintaining the routing table updated. Generally, proactive routing protocols are used for networks with a small number of nodes. One of the most known proactive routing protocol examples is represented by the Destination Sequence Distance Vector (DSDV) protocol .At the opposite, *reactive* routing protocols build multi-hop routes only upon specific requests and are thus characterized by a reduced overhead. They are based on the concept of flooding and involve three main phases: (i) route discovery, needed to find a possible existing route toward an unknown destination; (ii) route maintenance, used to detect link breaks and to find alternative routes; and (iii) an incremental search method to limit the number of links traversed when routing discovery is enabled. In this kind of networks, nodes that are not actively involved in communication flows do not generate any control or routing information traffic. Moreover, a route is maintained alike as long as it is needed by the source node. Among all available reactive routing protocols, the well-known ones are represented by the

2.1. Ad hoc On-Demand Distance Vector Routing Protocol

The AODV protocol is a reactive routing protocol developed for ad hoc networks, which supports unicast, multicast, and broadcast communications. The first phase in the AODV protocol is the discovery: a source node *S*, which has no entry, in its routing table, corresponding to the address of the destination node *D*, generates control traffic according to the following steps.
It follows that before starting the communication with the destination node *D*, the source node *S* has to wait for the completion of the ongoing route discovery process.One of the key features of the AODV protocol is represented by the *destination sequence number*, used also to have loop-free routes. This number is generated by each node in order to maintain the entries of the routing tables updated and is useful if there are two different routes from a source *S* to a destination *D*: in this case, *S* selects the route associated with the largest destination sequence number. Another interesting feature of the AODV protocol is the management of the local connectivity, in order to detect a route break, since nodes' mobility has to be taken into account as well. This mechanism works in the following way: in case a node, within a specific interval, does not highlight its presence to its neighbors through a specific *hello interval*, then, through a RREP message, it broadcasts a special request containing its identity, thus *forcing* a discovery phase. In detail, if a relay node between *S* and *D* fails to receive a minimum amount of *hello messages*—denoted as *allowed hello loss*, then a list of all the destinations which are unreachable due to the link loss will be forwarded by a broadcast Route Error (RERR) message by the node upstream of the break (propagating back to *S*, the latter has the possibility to perform again the route discovery process if it still needs the route). Thanks to these mechanisms, the AODV protocol has the capability of quickly adapting to dynamic link conditions, and can be employed with low processing power devices, as it has a reduced network utilization and memory overhead.

2.2. Optimized Link State Routing (OLSR) Protocol

The Optimized Link State Routing Protocol (OLSR) is a proactive routing protocol designed to work with large and dense Mobile Ad-Hoc Networks (MANETs) and using a hop-by-hop strategy to achieve a performance optimization higher than other classic link state routing algorithms. OLSR has been designed to work in dynamic scenarios where the communicating peers change over time and, since routes are maintained for all known destinations at all times , the control packets in the network are very limited and are usually handled only by selected nodes, denoted as Multipoint Relays (MPRs)—key element of the OLSR protocol—whose tasks are: (i) route creation and selection, and (ii) relay of messages between nodes.
In detail, a MPR reduces redundant retransmissions in the network region it is handling, thus minimizing the overhead of flooding messages. In general, a node obtains the information about its neighbors through periodic *hello* messages received from the neighbors themselves. A generic intermediate node Nx selects a set of nodes at 1-hop distance as its MPRs. Then the MPRs advertise link-state information for their "child" nodes periodically in their control messages. OLSR uses packets encapsulated in UDP datagrams transmitted on the reserved port 698. Moreover, the packet format is unified for all data (control messages, actual data messages) in order to have a fast and easy extensibility of the protocol maintaining compatibility with older versions of the protocol itself.

2.3. Dynamic Source Routing (DSR) Protocol

The Dynamic Source Routing (DSR) protocol is a reactive routing protocol, similar to AODV, specifically designed for use in multi-hop wireless ad hoc networks composed of mobile nodes. Since DSR adheres to the *source routing* philosophy, it does not use any periodic routing advertisement. This feature leads to a reduction of the control messages in the network—ideally to zero: when all nodes are stationary with respect to each other and all routes for current communication have already been discovered, there is no need for any route discovery process. A DSR-based network is completely self-organizing and self-configuring, thus requiring no existing network infrastructure, since nodes cooperate to forward packets over multiple hops between nodes not placed in visibility. The DSR protocol is based on: (i) route discovery—when a source node *S* wants to send a packet to a destination node *D* and does not know a route to *D*, it sends a route request that will be filled by each intermediate node with the hops the packets have to follow to go from *S* to *D*; and (ii) route maintenance—used when the data are actually transmitted and to detect network topology changes (e.g., a link along a route no longer works) using an acknowledgement-based system. In particular, when a source route is indicated as broken, the node *S* can use any other known route (which are cached in the node) to *D*, or it can start a route discovery to find a new route to *D*. Focusing on the route maintenance, thanks to the caching mechanism of the routes, DSR-based nodes have a rapid reaction to topology changes. In fact, a node with multiple routes to a destination can try another cached route if the used one fails. In this way, there is no need for a new route discovery procedure, leading to a significant reduction of control messages across the network.

3. *Mesh in IEEE 802.11 Networks*

In the last decade, a growing interest in mesh networking, from both academic and industrial entities, brought to the definition of an IEEE 802.11 standard amendment specifically addressed to IEEE 802.11 mesh networks, denoted as IEEE 802.11s. The physical layers of IEEE 802.11s and IEEE 802.11 standards are the same: it introduces new routing procedures that are performed at the Medium Access Control (MAC) layer, rather than at the network layer. To have an efficient routing, the nodes must have an accurate knowledge of the wireless links connecting them to their 1-hop neighbors. This leads to a seamless routing for protocols of the higher layer. In an IEEE 802.11s mesh network, also named as Mesh Basic Service Set (MBSS), there are different logical components,The main ones are the mesh stations (mesh STAs) that can participate to the formation of the MBSS, in which each node has the same level of complexity and there is no hierarchical structure. The mesh STAs, moreover, participate to the path selection and forwarding, leading to a very simple self-organizing network. In case of integration with other type of networks, such as the "traditional" IEEE 802.11 infrastructure BSS, or if the MBSS has to access external networks, other logical components are needed; the one that guarantees the access to the mesh network for "traditional" IEEE 802.11 stations is named as Mesh APs (MAPs). A MAP, however, does not enable the

communication between a mesh STA and a non-mesh STA. In fact, the logical component that enables the integration between mesh BSS and infrastructure BSS—thus enabling the communication between mesh STAs and non-mesh STAs—is the mesh gate. Furthermore, in order to enable also the communication between the mesh BSS and non-IEEE 802.11 Local Area Networks (LANs), such as wired LAN, other logical components are used, namely the Mesh Portal Points (MPPs), which enable the communication with an external entities.

A. ESP-MESH Terminology Concept

Term	Description
Node	Any device that is or can be part of an ESP-MESH network
Root Node	The top node in the network
Child Node	A node X is a child node when it is f
Parent Node	The converse notion of a child node
Descendant Node	Any node reachable by repeated proceeding from parent to child
Sibling Nodes	Nodes that share the same parent node
Connection	A traditional Wi-Fi association between an AP and a station. A node in ESP-MESH will use its station interface to associate with the softAP interface of another node, thus forming a connection. The connection process includes the authentication and association processes in Wi-Fi.
Upstream Connection	The connection from a node to its parent node
Downstream Connection	The connection from a node to one of its child nodes
Wireless Hop	The portion of the path between source and destination nodes that corresponds to a single wireless connection. A data packet that traverses a single connection is known as single-hop whereas traversing multiple connections is known as multi-hop.
Subnet work	A subnetwork is subdivision of an ESP-MESH network which consists of a node and all of its descendant nodes. Therefore the subnetwork of the root node consists of all nodes in an ESP-MESH network.
MAC Address	Media Access Control Address used to uniquely identify each node or router within an ESP-MESH network.
DS	Distribution System (External IP Network)

B.ESP-MESH Tree Topology

ESP-MESH is built atop the infrastructure Wi-Fi protocol and can be thought of as a networking protocol that combines many individual Wi-Fi networks into a single WLAN. In Wi-Fi, stations are limited to a single connection with an AP (upstream connection) at any time, whilst an AP can be simultaneously connected to multiple stations (downstream connections). However ESP-MESH allows nodes to simultaneously act as a station and an AP. Therefore a node in ESP-MESH can have multiple downstream connections using its softAP interface, whilst simultaneously having a single upstream connection using its station interface. This naturally results in a tree network topology with a parent-child hierarchy consisting of multiple layers.

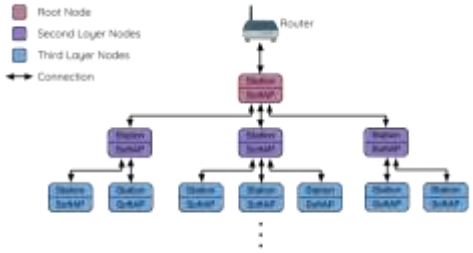


Figure 2.ESP-MESH Tree Topology

ESP-MESH is a multiple hop (multi-hop) network meaning nodes can transmit packets to other nodes in the network through one or more wireless hops. Therefore, nodes in ESP-MESH not only transmit their own packets, but simultaneously serve as relays for other nodes. Provided that a path exists between any two nodes on the physical layer (via one or more wireless hops), any pair of nodes within an ESP-MESH network can communicate.

1.Node Types

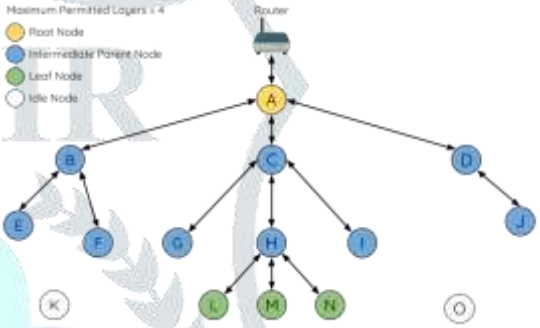


Figure 3.ESP-MESH Node Types

a.Root Node: The root node is the top node in the network and serves as the only interface between the ESP-MESH network and an external IP network. The root node is connected to a conventional Wi-Fi router and relays packets to/from the external IP network to nodes within the ESP-MESH network. There can only be one root node within an ESP-MESH network and the root node's upstream connection may only be with the router. Referring to the diagram above, node A is the root node of the network.

b.Leaf Nodes: A leaf node is a node that is not permitted to have any child nodes (no downstream connections). Therefore a leaf node can only transmit or receive its own packets, but cannot forward the packets of other nodes. If a node is situated on the network's maximum permitted layer, it will be assigned as a leaf node. This prevents the node from forming any downstream connections thus ensuring the network does not add an extra layer. Some nodes without a softAP interface (station only) will also be assigned as leaf nodes due to the requirement of a softAP interface for any downstream connections. Referring to the diagram above, nodes L/M/N are situated on the networks maximum permitted layer hence have been assigned as leaf nodes .

c.Intermediate Parent Nodes: Connected nodes that are neither the root node or a leaf node are intermediate parent nodes. An intermediate parent node must have a single upstream connection (a single parent node), but can have zero to multiple downstream connections (zero to multiple child nodes). Therefore an intermediate parent node can transmit and receive packets, but also forward packets sent from its upstream and downstream connections. Referring to the diagram above, nodes B to J are intermediate parent nodes. Intermediate parent nodes without downstream connections such as nodes E/F/G/I/J are not equivalent to leaf nodes as they are still permitted to form downstream connections in the future.

e.Idle Nodes: Nodes that have yet to join the network are assigned as idle nodes. Idle nodes will attempt to form an upstream connection with an intermediate parent node or attempt to become the root node under the correct circumstances. Referring to the diagram above, nodes K and O are idle nodes.

C. Beacon Frames & RSSI Thresholding

Every node in ESP-MESH that is able to form downstream connections (i.e. has a softAP interface) will periodically transmit Wi-Fi beacon frames. A node uses beacon frames to allow other nodes to detect its presence and know of its status. Idle nodes will listen for beacon frames to generate a list of potential parent nodes, one of which the idle node will form an upstream connection with. ESP-MESH uses the Vendor Information Element to store metadata such as:
Node Type (Root, Intermediate Parent, Leaf, Idle)
Current layer of Node
Maximum number of layers permitted in the network
Current number of child nodes
Maximum number of downstream connections to accept
The signal strength of a potential upstream connection is represented by RSSI (Received Signal Strength Indication) of the beacon frames of the potential parent node. To prevent nodes from forming a weak upstream connection, ESP-MESH implements an RSSI threshold mechanism for beacon frames. If a node detects a beacon frame with an RSSI below a preconfigured threshold, the transmitting node will be disregarded when forming an upstream connection.

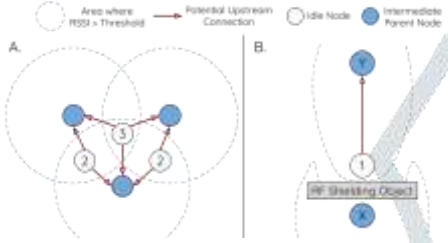
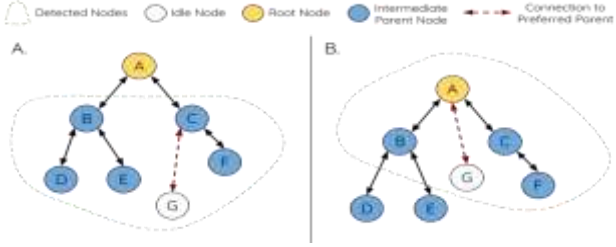


Figure 4.Effects of RSSI Thresholding

Panel A of the illustration above demonstrates how the RSSI threshold affects the number of parent node candidates an idle node has.
Panel B of the illustration above demonstrates how an RF shielding object can lower the RSSI of a potential parent node. Due to the RF shielding object, the area in which the RSSI of node X is above the threshold is significantly reduced. This causes the idle node to disregard node X even though node X is physically adjacent. The idle node will instead form an upstream connection with the physically distant node Y due to a stronger RSSI.When an idle node has multiple parent nodes candidates (potential parent nodes), the idle node will form an upstream connection with the preferred parent node. The preferred parent node is determined based on the following criteria:
Which layer the parent node candidate is situated on
The number of downstream connections (child nodes) the parent node candidate currently has he selection of the preferred parent t node will always prioritize the parent node candidate on the shallowest layer of the network (including the root node). This helps minimize the total number of layers in an ESP-MESH network when upstream connections are formed. For example, given a second layer node and a third layer node, the second layer node will always be preferred.
If there are multiple parent node candidates within the same layer, the parent node candidate with the least child nodes will be preferred. This criteria has the effect of balancing the number of downstream connections amongst nodes of the same layer.



Preferred Figure 5.Parent Node Selection

Panel A of the illustration above demonstrates an example of how the idle node G selects a preferred parent node given the five parent node candidates B/C/D/E/F. Nodes on the shallowest layer are preferred, hence nodes B/C are prioritized since they are second layer nodes whereas nodes D/E/F are on the third layer. Node C is selected as the preferred parent node due it having fewer downstream connections (fewer child nodes) compared to node B.
Panel B of the illustration above demonstrates the case where the root node is within range of the idle node G. In other words, the root node's beacon frames are above the RSSI threshold when received by node G. The root node is always the shallowest node in an ESP-MESH network hence is always the preferred parent node given multiple parent node candidates.
Routing Tables
Each node within an ESP-MESH network will maintain its individual routing table used to correctly route ESP-MESH packets to the correct destination node. The routing table of a particular node will **consist of the MAC addresses of all nodes within the particular node's subnetwork** (including the MAC address of the particular node itself). Each routing table is internally partitioned into multiple subtables with each subtable corresponding to the subnetwork of each child node.

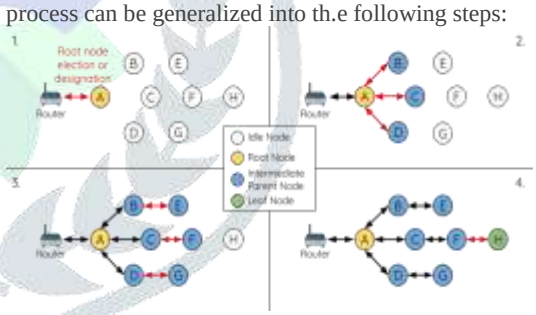
D. ESP-MESH Routing

Using the diagram above as an example, the routing table of node B would consist of the MAC addresses of nodes B to I (i.e. equivalent to the subnetwork of node B). Node B's routing table is internally partitioned into two subtables containing of nodes C to F and nodes G to I (i.e. equivalent to the subnetworks of nodes C and G respectively).

III.PROJECT WORK

1. If the packet's destination MAC address is within the current node's routing table and is not the current node, select the subtable that contains the destination MAC address and forward the data packet downstream to the child node corresponding to the subtable.

2. If the destination MAC address is not within the current node's routing table, forward the data packet upstream to the current node's parent node. Doing so repeatedly will result in the packet arriving at the root node where the routing table should contain all nodes within the network.reset. However, the ESP-MESH network building process can be generalized into th.e following steps:



Figur6.ESP-MESH Network Building Process

1. Root Node Selection
The root node can be designated during configuration on or dynamically elected based on the signal strength between each node and the router. Once selected, the root node will connect with the router and begin allowing downstream connections to form. Referring to the figure above, node A is selected to be the root node hence node A forms an upstream connection with the router.

2. Second Layer Formation
Once the root node has connected to the router, idle nodes in range of the root node will begin connecting with the root node thereby forming the second layer of the network. Once connected, the second layer nodes become intermediate parent nodes (assuming maximum permitted layers > 2) hence the next layer to form. Referring to the figure above, nodes B to D are in range of the root node. Therefore nodes B to D form upstream connections with the root node and become intermediate parent nodes.

3. Formation of remaining layers
The remaining idle nodes will connect with intermediate parent nodes within range thereby forming a new layer in the network. Once connected, the idles nodes become intermediate parent node or leaf nodes depending on the networks maximum permitted layers. This

step is repeated until there are no more idle nodes within the network or until the maximum permitted layer of the network has been reached. Referring to the figure above, nodes E/F/G connect with nodes B/C/D respectively and become intermediate parent nodes themselves.

4. Limiting Tree Depth
To prevent the network from exceeding the maximum permitted number of layers, nodes on the maximum layer will automatically become leaf nodes once connected. This prevents any other idle node from connecting with the leaf node thereby prevent a new layer form forming. However if an idle node has no other potential parent node, it will remain idle indefinitely. Referring to the figure above, the network's number of maximum permitted layers is set to four. Therefore when node H connects, it becomes a leaf node to prevent any downstream connections from forming.

RELATING ESP-NOW WITH ESP-32

The ESP-Now is a very special, high-speed network, making it perfect for residential and industrial automation. It is another protocol developed by Espressif. We'll be talking about this network today, which allows several devices to communicate without using a WiFi network made by a router. I'll show you an introduction to the subject and make several ESPs32 communicate through this scheme. Therefore, an ESP32 will read the pins and transmit their values, while the other devices will receive these values and change the output of the pins according to those numbers.This network is reliable and is 2.4GHz. This network also is with the same frequency and channels as your WiFi router. The highlight, however, is that it goes away from WIFI,as it is instant.In our assembly, we have an ESP32 isolated, which is configured as Master. It is important to remember that there is no Master device, and that they all function as Stations. However, to facilitate the identification, I point to this first ESP as Master, in which I set up a button in the GPIO02.When this button is pressed, a LED lights up in this microcontroller, and all four other ESPs32 instantly restart the action. Why does this occur? Because the moment the Master sends the information to the station, it is sending a MAC address of Broadcast, which means that everyone on the network receives the data at the same time.In this example, I compiled the same receiver for all microcontrollers. I copied the code that sends to the Master, and this one also sends the Broadcast to the others. In this scheme, it is still possible to see that we practically don't have Boot time, because when the ESP is switched off and reconnected, the operation resumes immediately.In the serial print of Setup, both the s ending and the receiving code contain the MAC address values of each of the chips involved.

IV AUTOMATIC ROOT NODE SELECTION

The automatic selection of a root node involves an election process amongst all idle nodes based on their signal strengths with the router. Each idle node will transmit their MAC addresses and router RSSI values via Wi-Fi beacon frames**.** The MAC address is used to uniquely identify each node in the network whilst the router RSSI is used to indicate a node's signal strength with reference to the router.
Each node will then simultaneously scan for the beacon frames from other idle nodes. If a node detects a beacon frame with a stronger router RSSI, the node will begin transmitting the contents of that beacon frame (i.e. voting for the node with the stronger router RSSI). The process of transmission and scanning will repeat for a preconfigured minimum number of iterations (10 iterations by default) and result in the beacon frame with the strongest router RSSI being propagated throughout the network.
The setting the Channel Switch Count value to a custom value is due to the fact that the ESP-MESH network and its router may have a different and varying beacon intervals. Therefore, the Channel Switch Count value provided by the router is irrelevant to an ESP-MESH network. By using a custom value, nodes within the ESP-MESH network are able to switch channels synchronously relative to the ESP-MESH network's beacon interval. However, this will also result in the ESP-MESH network's channel switch being unsynchronized with the channel switch of the router and its connected stations.

The following diagram demonstrates how an ESP-MESH network is built when the root node is automatically selected.

Figure7.Root Node Election Example

1. On power-on reset, each node begins transmitting beacon frames consisting of their own MAC addresses and their router RSSIs.

2. Over multiple iterations of transmission and scanning, the beacon frame with the strongest router RSSI is propagated throughout the network. Node C has the strongest router RSSI (-10 dB) hence its beacon frame is propagated throughout the network. All nodes participating in the election vote for node C thus giving node C a vote percentage of 100%. Therefore node C becomes a root node and connects with the router.

3. Once Node C has connected with the router, nodes A/B/D/E connect with node C as it is the preferred parent node (i.e. the shallowest node). Nodes A/B/D/E form the second layer of the network.

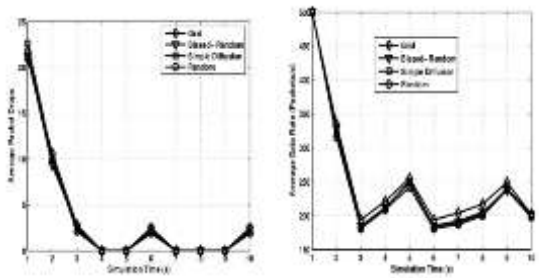
4. Node F and G connect with nodes D and E respectively and the network building process is complete

(a) TCP:Average Data Rate (packets/s)(b)TCP:Average Packet Drops
DATA TRANSMISSION ESP-MESH PACKET

ESP-MESH network data transmissions use ESP-MESH packets. ESP-MESH packets are entirely contained within the frame body of a Wi-Fi data frame. A multi-hop data transmission in an ESP-MESH network will involve a single ESP-MESH packet being carried over each wireless hop by a different Wi-Fi data frame.
The following diagram shows the structure of an ESP-MESH packet and its relation with a Wi-Fi data frame.

Figure 8.ESP-MESH Packet

The header of an ESP-MESH packet contains the MAC addresses of the source and destination nodes. The options field contains information pertaining to the special types of ESP-MESH packets such as a group transmission or a packet originating from the external IP network (see **MESH_OPT_SEND_GROUP**)



V.CONCULSION

Wireless Sensor Nodes(WSNs) provide the communications required to deploy Intelligent Transportation Systems (ITS). In the current state of the art in this field there need studies on real outdoor experiments to validate the new WSNs optimizing protocols and applications proposed by designers. In this work we have addressed the definition of a WSN handovers in order to study the performance of the Data Transfer Protocol (DTP) in a real WSN. Implementation of Sensor Network ESP-MESH is a multiple hop (multi-hop) network meaning nodes can transmit packets to other nodes in the network through one or more wireless hops. Therefore, nodes in ESP-MESH not only transmit their own packets, but simultaneously serve as relays for other nodes. Provided that a path exists between any two nodes on the physical layer (via one or more wireless hops), any pair of nodes within an ESP-MESH network can communicate.

REFERENCES

[1] Espressif Systems, "ESP8266EX Datasheet", Version5.8, 2018, https://www.espressif.com/sites/default/files/documentation/0 a-esp8266ex_datasheet_en.pdf.

[2] Hiertz, G. R. et al., "IEEE 802.11s: the WLAN mesh standard", IEEE Wireless Communications, Vol. 17, No. 1, pp. 104-111, 2010.

[3] Ivan Grokhotkov, "ESP8266 Arduino Core Documentation, Release 2.4.0", May 14, 2017.

[4] Perkins, C., Belding-Royer, E., Das, S., "Ad Hoc On-Demand Distance Vector (AODV) Routing", Jul. 2003.

[5} Spranger, M., "ESP8266 WLAN-Mesh über 4 Nodes", arduino-und-raspberry-pi/esp-als-w-lan-router

[6]Srisuresh, P., and Egevang, K., "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, RFC Editor, 2001.

[7]Chaudet,C.,Dhoutaut D. and Guerin Lassous,I.",Performing issue with IEEE 802.11 in ad hoc Networking."

[8]"The aiirmesh"Onlineavailable:http/www.aiirmesh.com/Cerritos/

[9].'BWN Lab Wireless Mesh Network Test-bed"-Online Available

http/www.ece.gatech.edu/research/labs/bwn/mesh/