Dynamic Configuration of Wi-Fi Mesh Network

Anupama S, Bijwal B Jain, Ganavi H, Pavithra C, Shreshta

Abstract—Wireless device users expect to have a guaranteed quality of experience at all times, at any location, and across devices. Wi-Fi has become an access network of preference for service/network providers and users as well for public and private access. This sets a challenging requirement for next-generation Wi-Fi technology to provide seamless and uniform network quality of service. Wireless Mesh Network (WMN) is the network architecture where nodes can communicate with one another via multi-hop routing. Dynamic self-organization, self-configuration and self-correction tends to the possibility of the network architecture. In this project, a mesh network is designed by nodes made up of Atmega328p micro-controllers connected using ESP8266 Wi-Fi modules. The mesh network is able to fully connect itself through a variety of graph topology algorithms and is able to route packets through the network using a shortest path approach. The network system also exposes a very simple application layer that can be extended to a virtually unlimited set of networking ideas.

Index Terms— Wireless mesh network, self- configuration, multi-hop routing, topology algorithms, Wi-Fi, private access, self-organizing.

1. INTRODUCTION

A mesh network is a network topology in which the nodes (bridges, switches, and other devices) connected directly, dynamically, and non-hierarchically to as many other nodes as possible and cooperate with other nodes to efficiently route data from/to clients. The lack of dependency on a single node enables every node to participate in the forwarding of information. Mesh networks dynamically self-organize and self-configure, which can reduce installation overhead. The ability of self-configuration enables the dynamic distribution of workloads, when a few nodes fail, contributing to fault tolerance and reduced maintenance costs.

Mesh networks can relay messages using either a flooding technique or a routing technique. While routing, the message is passed along a path by hopping from node to node until it reaches its destination. The network must allow for continuous connections to ensure that all its routes are available. It is able to reconfigure itself around broken paths, using self-healing algorithms such as Shortest Path Bridging. Self-healing ability allows a routing-based network to operate when the node breaks down, or a connection becomes unreliable. As a result, the network is usually reliable, as there is often more than one path between a source and a destination in the network. Although primarily used in wireless situations, the concept also applies to wired
networks and software interaction.

A mesh network whose nodes are all connected is a fully connected network. Fully connected wired networks have security and reliability advantages: problems in a cable affect only the two nodes attached to it.

Wi-Fi is a family of wireless network protocols, based on the IEEE 802.11 family of standards, commonly used for local area networking of devices and Internet access, allowing nearby digital devices to exchange data by Radio waves. Wi-Fi is the most widely used computer networks in the world, used globally in home and small office networks to link desktop and laptop computers, tablet computers, smartphones, smart TVs, printers, and smart speakers together and to a wireless router to connect them to the Internet, and in wireless access points in public places like coffee shops, hotels, libraries and airports to provide the public Internet access for mobile devices.

2. PREVIOUS WORK

Earlier, the research was mainly focused on Bluetooth mesh technology which is a computer mesh networking standard based on Bluetooth Low Energy that allows for many-to-many communication over Bluetooth radio. In Bluetooth low energy (BLE) Beacons the duty-cycle scanning defines the scanning time slot based on the lowest feasible duty cycle unveiled through a comprehensive analysis of energy consumed by advertising and scanning events. The scanning policies, on the other hand, allow each node to explore all possible time slots before locking their scanning event to a particular time slot that is most likely to hear the incoming packet. There is little research study based on Wi-Fi mesh networking, previously for Wi-Fi based wireless mesh networking they used micro-controller ESP32 as Wi-Fi nodes, to create a robust mesh network. The nodes are self-configuring and self-organizing.

3. BLOCKDIAGRAM

Wi-Fi devices are typically either configured as stations or access points. A station is a device like a computer, which can connect to one Wi-Fi network at a time. An access point

is something like a router, which allows many stations to connect to it and acts as a kind of local hub for connecting Wi-Fi devices. The ESP8266 Wi-Fi modules allow a third mode which is a hybrid of the two. A single chip can act as both a station and an access point. However, it limits the total number of connections to 5,4 connections to its access point and its one station connection. The software will be mainly focused on the constraints of the hardware, namely the ESP8266 Wi-Fi modules.

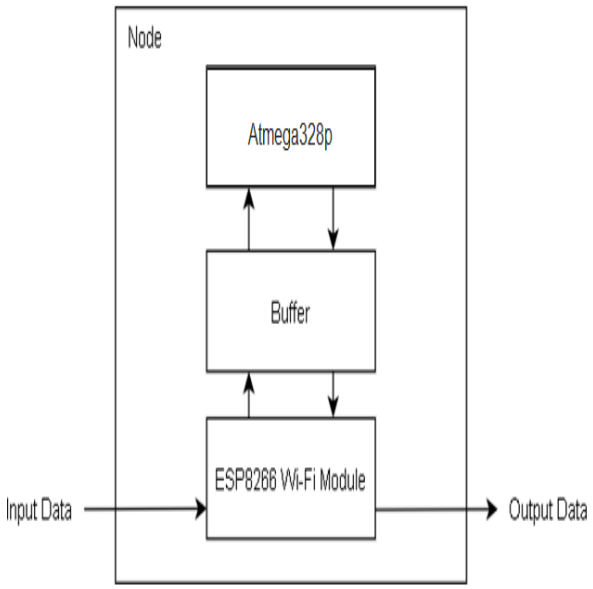


Fig1.General block diagram

With ESP-MESH, the nodes don't need to connect to a central node. Nodes are responsible for relaying each other's transmissions. This allows multiple devices to spread over a large physical area. The Nodes can self-organize and dynamically talk to each other to ensure that the packet reaches its final node destination. The main node sends data packets to the nodes connected to it, which offers the connection for the main Node acts as bub-node, later on the sub-node can act as Main node to the other nodes which transmits the data packets which are received by the sub-node, the data packets which are received by the sub-node are in turn transmitted to the other node maintaining proper data transfers between the mesh. If any node is removed from the network, it is able to self-organize to make sure that the packets reach their destination, that means When a problem occurs to the sub-node it won't be able to transfer the data packets received by the main node in that scenario the other node waits for some duration and tries to establish the connection to the main node if the operation succeeds the other node now becomes the sub-node which is in direct communication with the main node and it can transfer data to the new nodes which can access to the data transmitted by the main node via sub-node.

4. METHODOLOGY

The software will be mainly focused on the constraints of the hardware, namely ESP8266 Wi-Fi modules. Wi-Fi devices are typically either configured as stations or access points. A station is a device like your computer, which can connect to one Wi-Fi network at a time. An access point is something like a router, which allows many stations to connect to it and acts as a kind of local hub for connecting Wi-Fi devices. The ESP8266 Wi-Fi modules allow a third mode which is a hybrid of the two. A single chip can act as both a station and an access point. However, it limits the total number of connections to 5,4 connections to its access point and its one station connection. Fig 3.6 depicts the control flow of the proposed system node. A closed control loop is present which keeps acting after a specific interval of time,which maintains the connectivity of the nodes.

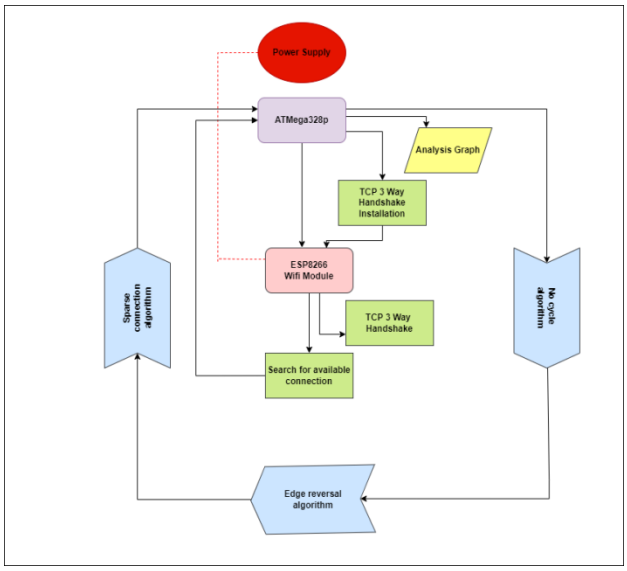


Fig2.Flow Chart

These connectivity restrictions inform the kind of mesh network which can be created. Since the number of edges in the network is limited to the number of devices (since a station can only connect to a single access point) a full fault tolerant network is not possible. Therefore, the focus is on software that tries to connect as many of the devices as possible in a fully connected mesh. The algorithms proposed can allow for the maximum number of nodes to be connected in an arbitrary mesh configuration.

5. HARDWARE AND SOWFTWARE REQUIREMENTS

The hardware components used in the project decides

the proper operation of the project. Thus, NodeMCU is very important in the design procedure.

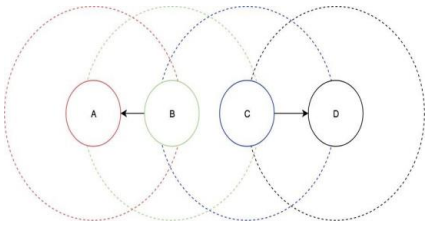
NodeMCU is an open source IoT platform. This based on ESP8266 Wi-Fi Module and it uses Lua Scripting language.This modules has inbuilt USB to serial chip upload codes,3.3v regulator and logic level converter circuit so we can immediately upload codes and connect our circuits.This micro-controller board can easily be programmed using Arduino IDE programming software.

Table1: Software Requirements

Sl.	Components	Description
1.	Arduino1.8.5	Programming the Node MCU and simulating the program
2.	BlynkApp	IoT platform used for controlling and monitoring hardware data

6. DESIGN AND IMPLEMENTATION

In the hardware design, the node consisted primarily of two key components: the Atmega micro-controller and the ESP8266 Wi-Fi module. The Atmega is an inexpensive, powerful chip that would meet the demands of the network. On the other hand, the ESP8266 Wi-Fi-module is a component that is known for its versatility, cost-effectiveness, and powerful capabilities. The nodes are to be designed as compact as possible to minimize the usage, and cost. A NodeMCU is a low cost IoT platform. Using a NodeMCU with Atmega micro-controller and ESP8266 Wi-Fi module with the above design requirements can be achieved.



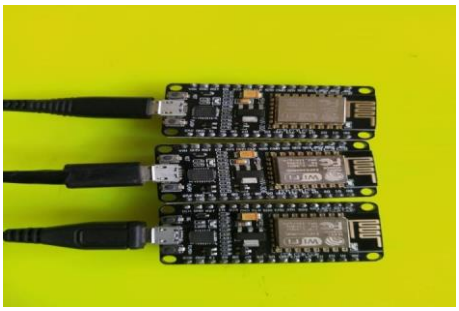


Fig3. Hardware Implementation: NodeMCU

The software will be mainly focused on the constraints of the hardware, namely ESP8266 Wi-Fi modules. Wi-Fi devices are typically either configured as stations or access points. The focus is on software that tries to connect as many of the devices as possible in a fully connected mesh. The algorithms proposed can allow for the maximum number of nodes to be connected in an arbitrary mesh configuration. In order to accomplish this, the software is split into 4 logically separate layers: Wi-Fi, Routing, Topology, and Application.

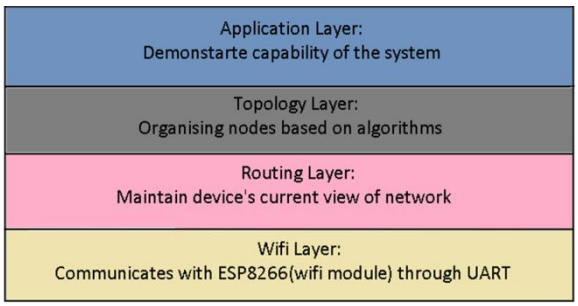


Fig 4. Network Architecture Stack

A. Wi-Fi Layer: The Wi-Fi layer is mainly concerned with setting up the Wi-Fi module and handling connections and disconnections from stations and access points. When it sets up the Wi-Fi module, the Wi-Fi layer reads the module's MAC address and assigns itself a unique ID and IP address based on the MAC address. It then sets up a TCP server at the IP address and broadcasts itself to other Wi-Fi modules by setting its name to "ESP8266-Mesh". It then exposes a way to scan for other mesh devices to the above layer. It also exposes a way to connect to an access point, disconnect from an access point, get a list of the direct neighbors that the device is connected to, and a way to register a handler for when a different mesh device connects to its access point.

B. Routing Layer: The algorithm is primarily concerned with connection creation and destruction.
When station S connects to access point A, node A sends S a bootstrap packet. This bootstrap packet contains node A's current graph. Node S will then figure out the differences between A's graph and its own graph. It will then flood the new edges from A to its side of the network and flood edges from S's side of the network to A. Finally, it will flood the S-A edge to the whole network.

As a concrete example, consider the following scenario. Node T is connected to node S and node A is connected to node B. So nodes T and S see the network graph as an edge connecting each other and nodes A and B see the network graph as an edge connecting each other. Node S connects to node A, and the messages look like so:

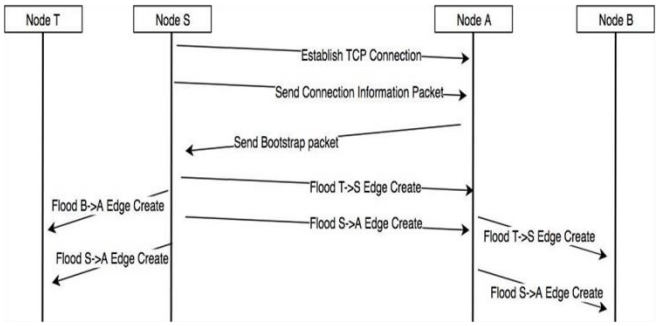


Fig 5: TCP connection and Flooding

After the flood of messages has subsided, all four nodes in the network know about all three edges in the fully connected network and they are all able to produce the same graph. When two nodes disconnect from each other they both flood their respective sides of the network with an edge removal message. This allows each node in the network to update its graph to reflect the removed edge.

In order to stop flood messages from infinitely propagating throughout the network we use the concept of sequence numbers. Each node maintains its own monotonically increasing sequence number, which it increases with each message that it creates. Every flood message has an origin node and the sequence number of that origin node when the flood message was created. Each node keeps track of the highest sequence number that it has seen from each node in the network. When a node sees a new message from origin node O with sequence number N, it checks to see if N is greater than the largest sequence number previously seen from O. I fit is, then the node accepts the message and updates the largest sequence number to N. Otherwise, the node discards the message.

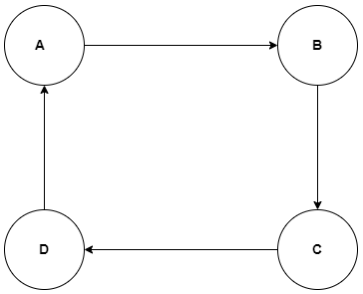


Fig 6. No cycle algorithm

C. Topology Layer: The no cycle algorithm inspects a node's current network graph and determines if there is a cycle in the graph that it is a part of. If it finds one, the node will disconnect from its access point if it is the lowest ID node in the cycle.

Every node will see that it is in a cycle, but node 15 will disconnect from node 21 since node15 has the smallest ID.

Fig 7. Before Edge reversal algorithm

The edge reversal algorithm is necessary because the network could in advertently partition itself. As a simple toy example, assume we have nodes A, B, C, and D. Nodes A and B can see each other, nodes B and C can see each other, and nodes C and D can see each other. Assume that node B connects to node A and node C connects to node D. Since each node can only have a single station connection, the network is now partitioned.

Then a node N that runs the edge reversal algorithm looks for nodes which are visible to N but not in N's graph. If such a node exists, N tries to find a reversible path in its graph, which is a path ending in a node without degree 0. It will then tell every node along that path to disconnect from its access point and connect to the previous node in the path. After this is done, the network will still have the same connectivity and now N will have out degree 0, meaning it can connect to the node that was visible but not in the network. In the above network, let's assume that B completed the edge reversal process first. Then the resulting network will look like so:

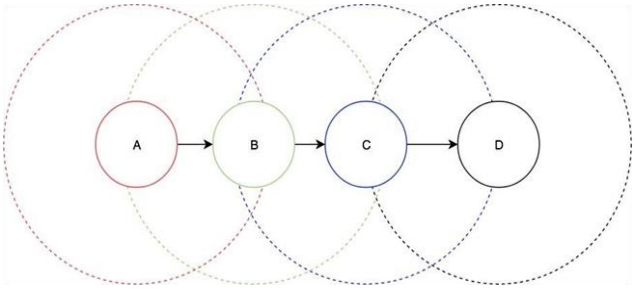


Fig 8. After edge reversal algorithm

Finally, the sparse connection algorithm attempts to remedy the 5-connection limit imposed on the Wi-Fi modules. As a motivating example, assume the network has the following configuration and that modules are limited to 3 connections:

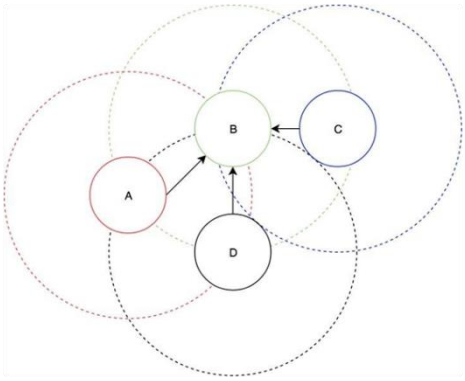


Fig 9. Maximum node connection scenario

Now assume a 5th node, node E comes online in the following position:

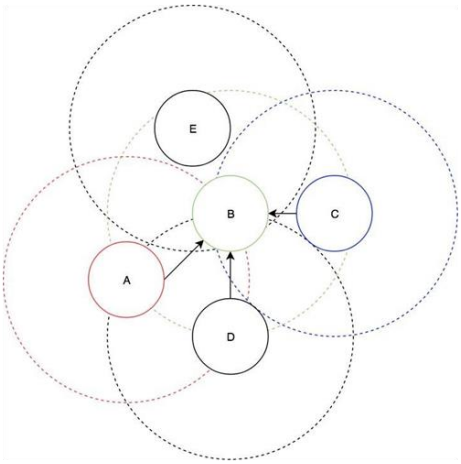


Fig 10. Inclusion of new node in a maximum node scenario

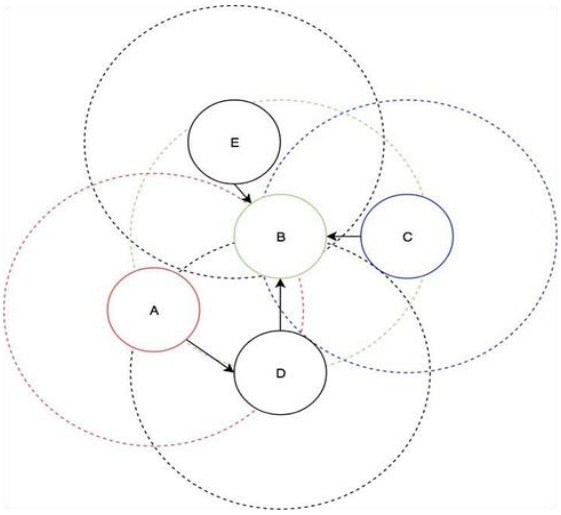


Fig 11. Inclusion of new node using Sparse connection algorithm

Node E wants to connect to node B, but node B already has 3 connections, so it will refuse the connection. However, we can still fully connect the network if A makes a connection to D instead. The way that A does this is that it realizes that node B has a large degree and that A sees another node that is the network(D). A will then choose to disconnect from node B and connect to node D in order to decrease the degree of node B. Then, node E can connect to node B and we now have a fully connected network.

7. RESULTS

The proposed idea could be implemented successfully, and the system could fulfill the requirements. The mesh network is analyzed using 3 nodes for the implementation. When a node in this system is powered ON, it searches available connections, which can either be a predefined internet access-point or other system nodes. Once the node connects with other node/access-point, it tries to connect with Blynk Cloud server. If it is not able to connect it retries again and if it still is not able to connect to the server, the node disconnects the current Wi-Fi connection and tries to connect with an alternate node.

The implemented mesh network is capable of data transfer, but in this implementation there is no data source. Thus, for performance analysis of the network the nodes have been classified as main-node and sub-node. The main-node sends data packets to the other nodes connected to it, which offers the connection for the main-node acts as sub-node. The data packets which are received by the sub-node are in turn transmitted to the other nodes maintaining proper data transfers between the mesh. If any node disconnects from the network, it is able to self-organize to make sure that the packets reach their destination.

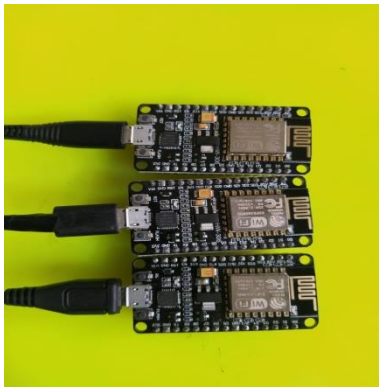


Fig 12. NodeMCU used as node

The performance analysis is represented through a mobile application UI. The UI consists of mainly two parts: Node status part and Data Log graph. When a new node is connected or disconnected to the network, the UI displays the respective connection status. The UI represents two sub-nodes used in the performance analysis. The node representation in the UI flicker once when a data packet is received. The Data Log graph represents the data packet being sent by the main-node to the two sub-nodes and the data packet received by the sub-nodes.

Fig 13. Performance Analysis UI

The main-node tries connecting to a access-point as soon as the device is powered ON. The main-node connects to the blynk cloud server as shown in Fig 14, when the node hasinternet access.

Fig 14. Main-node connecting to blynk cloud server

Similar to the main-node, the sub-nodes also try connecting to a access-point as soon as the device is powered ON. The sub-node also connects to the blynk cloud server as shownin Fig 15, when the node has internet access.

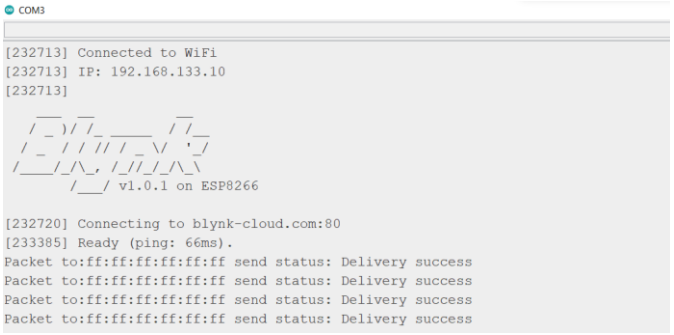


Fig 15. Sub-node connecting to blynk cloud server

When the main-node is connected to the network, it starts broadcasting data packets to each sub-node in the network as shown in Fig 16

Fig 16. Data packets being sent by main-node

The sub-nodes starts receiving data packets from the main-node connected in the networkas shown in Fig 17. The sub-nodes send the x and y coordinates to the blynk cloud server,which depicts the time and size, for the UI to plot the Data Log graph.

Fig 17. Data packets being received by sub-node

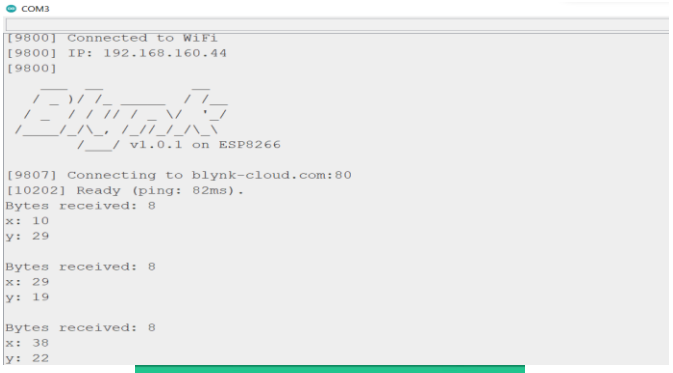
When a sub-node is connected to the network the mobile application UI displays the message "wifimesh connected". The live Data Log Graph represents the data packets being send by the main node and data packets being received in the receiving node. Fig 18 represents a single sub-node connected to the network with a main-node. The graph representsa very good connection between the nodes with minimal data losses.

Fig 18. UI status display of sub-node connected to the network

Fig 19. Data packets being received by a single sub-node

The sub-node connected to the network starts receiving data packets from the main-node. Due to internet fluctuation and other Wi-Fi connection uncertainties, there will be data losses, as represented in Fig 19.

The "wifimesh connected" message is displayed in the UI, when a second sub-node is connected to the network. When the new node is connected it starts receiving data packets from the main-node as shown in Fig 20.



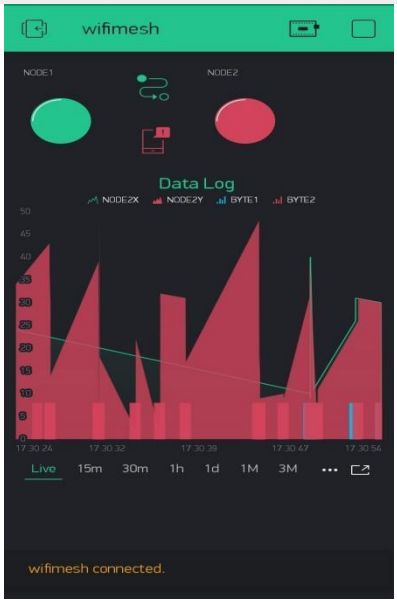


Fig 20. Second sub-node connection with the network

The two sub-nodes in the network receives data packets from the main-node. High degree of variation in data losses is present in the initial stage of a new node being connected in the network as shown in Fig 21.

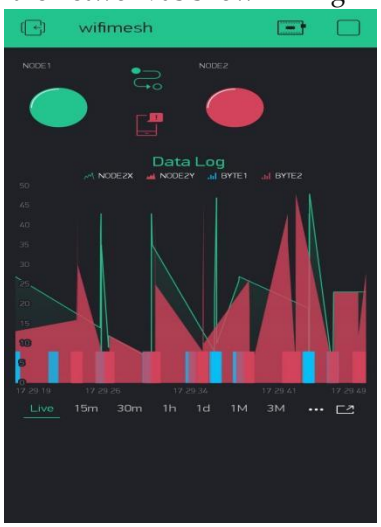


Fig 21. Data Log soon after a new node connected

After a period of time the data losses in both the nodes are reducing as the connections in the network are stabilizing. This is depicted in Fig 22. where the variations in the data received is reduced.

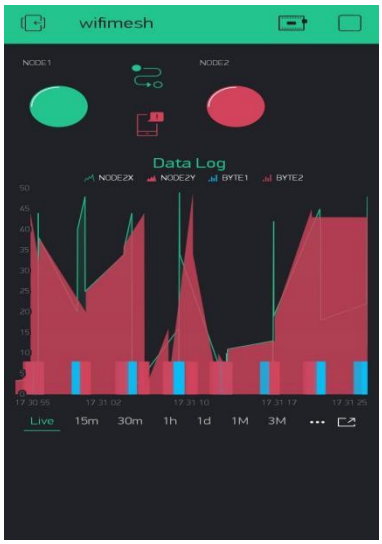


Fig 22. Data Log a while after a new node connected

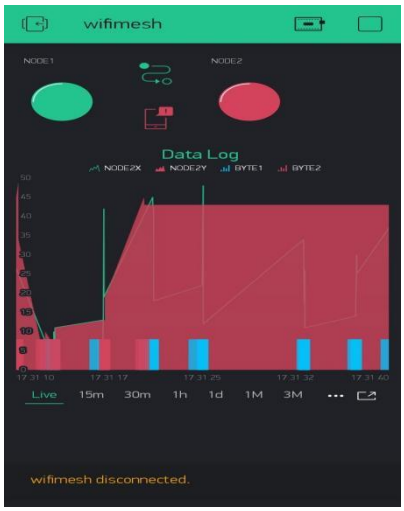


Fig 23. One sub-node disconnected from the network

When a sub-node is disconnected from the network the UI displays the message "wifimesh disconneted" as shown in Fig 23. The network re-configures if any nodes were connected through the disconnected node. After a node is disconnected from the network the remaining nodes reconfigure the network structure and continue receiving data packets from the main-node. As shown in Fig 24. the data losses may vary initially after the disconnection of a node and then the data losses gradually decrease.

Fig 24. Data Log graph after a sub-node is disconnected

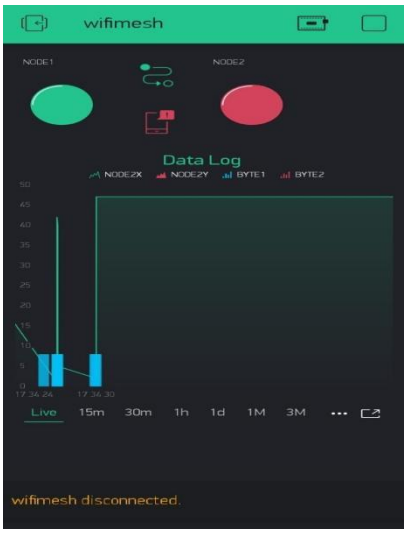


Fig 25. Both sub-nodes disconnected

With the disconnection of all sub-nodes from the network, the "wifimesh disconnected" message is displayed in the UI as shown in Fig 25. After the disconnection of the all sub-nodes, the Data Log graph display blank.

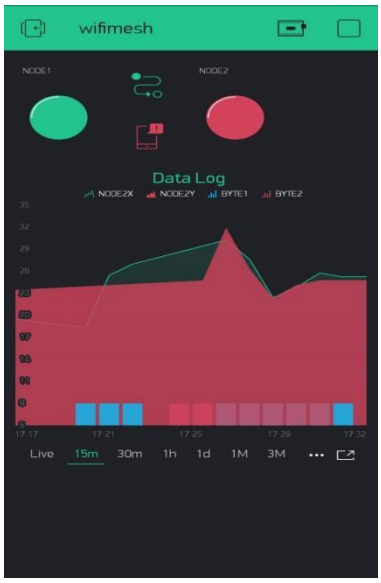


Fig 26. Data Log graph 15 minute overview

Fig 27. Data Log graph 30 minute overview

The overview of the Data Log graph for 15 minutes and 30 minutes shown in Fig 4.15 and 4.16 respectively.

8. CONCLUSION

A wireless mesh network is a communication network having mesh routers and mesh clients connected in a mesh topology. The Wireless mesh network experience link failure due to application bandwidth demands, channel interference etc. These failures will cause performance degradation. Reconfiguration is required to protect the network from dynamic link failure. The gateway will send the reconfiguration plan to the main node. This main node will send this plan to all other nodes in the group. The group members can change their link settings according to the reconfiguration plan. This scheme will move to the neighboring nodes.

In particular, we proposed a policy that, when run on each device in the network, would be able to reconfigure each device in such a way to create the network and keep it working and self-optimizing.

REFERENCES

[1] Junhao Hu, Lin Cai, Jianping Pan, Mesh Network Reliability Analysis for Ultra-Reliable Low-Latency Services, 2021, IEEE.
[2] Pai Chet Ng, James She and Petros Spachos, Energy-Efficient Overlay Protocol for BLE Beacon-based Mesh Network,2021, IEEE.
[3] Lihui Zhang, Xiaolin Ma , Pei Zhang and Zeyu Guo. Dust monitoring and processing system based on Wi-Fi Mesh network distributed backup routing algorithm, 2021, IOPScience.
[4] Mukhtiar Bano, Amir Qayyum, Rao Naveed Bin Rais and Syed Sherjeel A. Gilani, Soft-Mesh: A Robust Routing Architecture for Hybrid SDN and Wireless Mesh Net-works ,2021, IEEE.
[5] Muhammad Umar Farooq and Muhammad Zeeshan, Connected dominating set Enabled On-Demand Routing (CDS-OR) for Wireless Mesh Networks,2021, IEEE.
[6] Mahrokh Abdollahi, Wei Ni, Mehran Abolhasan, Shenghong Li, Software-Defined Networking Based Adaptive Routing for Multi-Hop Multi-Frequency Wireless Mesh,
2021, IEEE.
[7] Cheng Qiao, Kenneth N. Brown, Fan Zhang and Zhihong Tian, Federated Adaptive Asynchronous Clustering Algorithm for Wireless Mesh Networks,2021, IEEE.
[8] Jose Cecílio LASIGE, AQUAMesh: A Low-Power Wide-Area Mesh network protocol for Remote Monitoring Applications in Water Environments,2021, IEEE.
[9] Emanuele Giacomini, Francesco D'Alterio, Andrea Lacava and Frances ca Cuomo, BLUES: A Self-organizing BLE Mesh-network Paradigm for IoT Environments,
2020, IEEE.
[10] Daniel J. Dubois, Yosuke Bando, Konosuke Watanabe, Henry Holtzman, Lightweight Self-organizing Reconfiguration of opportunistic infrastructure mode Wi-Fi Networks, 2013, IEEE.
[11] Gollu Appala Naidu and Jayendra Kumar, Wireless Protocols: Wi-Fi SON, Bluetooth, ZigBee, Z-Wave, and Wi-Fi, 2019, ResearchGate.
[12] Marco Mirabilio, Alessio Iovine, Elena De Santis, Maria Domenica Di Benedetto and Giordano Pola, Scalable Mesh Stability of Nonlinear Interconnected Systems,2022, IEEE.

[13] P. Sharnya and Jennifer S. Raj, Self Organizing Wireless Mesh Network, 2013, IJIAS.
[14] Yu Liu, Kin-Fai Tong, Xiangdong Qiu, Ying Liu and Xuyang Ding, Wireless Mesh Networks in IoT networks, 2017 International Workshop on Electromagnetics: Applications and Student Innovation Competition.
[15] Y. Peng, L. Guo, Q. Deng, Z. Ning, and L. Zhang, A novel hybrid routing forwarding algorithm in SDN enabled wireless mesh networks, 2015, IEEE.
[16] Junfeng Wang, Yiming Miao, Ping Zhou, M.Shamim Hossain, SkMd Mizanur Rahman, A software-defined network routing in a wireless multihop network, 2017, ELSEVIER.
[17] Usman Ashraf, Placing controllers in software-defined wireless mesh networks, 2018, SCIRP.
[18] Akram Hakiri, Aniruddha Gokhale, and Prithviraj Patil, A software-defined wireless networking for efficient communication in smart cities,2017, Semanticscholar.
[19] Hisham Elzain and Yang Wu, Software Defined Wireless Mesh Network Flat distribution control plane, 2019, MDPI.
[20] Nithin Shastry and T.G. Keerthan Kumar, Enhancing the performance of software defined wireless mesh networks, 2020, Springer.
[21] Weijia Wang, Chang-Heng Wang, and Tara Javidi, Reliable shortest path routing with applications to wireless software-defined networking, 2018, IEEE.
[22] Junho Kim, Seung Gwan Lee, and Sungwon Lee, Mesh network convergence management system using software-defined network, 2018, ELSEVIER.
[23] Ke Bao, John D. Matyjas, FeiHu, and Sunil Kumar, Intelligent software-defined mesh networks with link-failure adaptive traffic balancing, 2018, IEEE.
[24] Mohamed Labraoui, Michael Boc and Anne Fladenmuller, Self-configuration mechanisms for SDN deployment in Wireless Mesh Networks, 2017,IEEE.