

## **LAB - 04**

Name: Halima Sadia

Student ID: 2021-3-60-280

Course Code: CSE406

### **Introduction**

In this lab we are setting up a mesh network using ESP8266 NodeMCU boards and the `painlessMesh` library. A mesh network allows devices to communicate directly with each other without the presence of a central router. When two nodes are out of range then, the information can traverse through the intermediate nodes- this is called multi-hop routing. This creates a more reliable, self-healing and covering a greater network.

We tested three main features:

1. Broadcasting messages to all connected nodes.
2. Sending direct messages to a specific node.
3. Multi-hop messaging, where messages pass through intermediate nodes.

### **2. Equipment Used**

- 4 × NodeMCU ESP8266 boards
- Micro-USB cables
- Arduino IDE (version 1.8 or later)
- `painlessMesh` library (installed via Arduino Library Manager)
- Serial Monitor (built into Arduino IDE)

### **3. Background of study**

A mesh network is simply interconnected so that each node joins several other nodes, and there are no centrally organized communication backbones that are required. This is as opposed to star topology whereby all nodes depend on a central access point. Creating such a network with Arduino-compatible microcontrollers is simplified by the `painlessMesh` library that deals effectively with Connection and matching of nodes.

1. Received signal strength routing (RSS).
2. Coordination between the time on gadgets.
3. Transmission of message (broadcast or direct).

The library uses callback functions to let the program know about some events:

- New Connection - Activated when a new node has been connected.

- Activated paths Changed Connections - Being activated in the presence of a change in the network structure.
- Adjusted Time- Set when the synchronization of time is made.

## 4. Methodology

### 1. Preparation:

- Installed the package of ESP8266 board in Arduino IDE.
- Took in the painlessMesh library.
- Set the same MESH\_PREFIX, MESH\_PASSWORD and MESH\_PORT to each NodeMCU.

### 2. Hardware Setup:

- Connected individual NodeMCUs to a computer via USB both to program and monitor.
- Placed nodes in different geographical locations in order to simulate different connection strengths and test multi-hop feasibilities.

### 3. Programming and Testing:

- **Task 1:** Uploaded the broadcast message program for all their nodes and ensured that they are getting the messages on the devices.
- **Task 2:** The code was modified to introduce direct messaging. Sent special messages only to a specific node ID and so that only the concerned specific node would show them.
- **Task 3:** Placed nodes in a position whereby some of them were unable to communicate without use of an intermediate node. Sent direct messages and saw messages relayed automatically by painlessMesh.

### 4. Observation Recording:

- Used Serial Monitor to log all received messages, connection changes, and time adjustments.
- Collected screenshots for evidence.

## 5. Explanation of Code

The base program included the following main sections:

### Library and Object Setup:

```
#include "painlessMesh.h"
```

```
painlessMesh mesh;
```

This includes the `painlessMesh` library and creates the `mesh` object to control the network.

### **Network Configuration:**

```
#define MESH_PREFIX    "meshNetwork"
```

```
#define MESH_PASSWORD  "meshPassword"
```

```
#define MESH_PORT      5555
```

Nodes need to share a prefix, a password and a port to be able to communicate with each other.

### **Setup Function:**

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    mesh.init(MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT);
```

```
    mesh.onReceive(receivedCallback);
```

```
    mesh.onNewConnection(newConnectionCallback);
```

```
    mesh.onChangedConnections(changedConnectionCallback);
```

```
    mesh.onNodeTimeAdjusted(timeAdjustedCallback);
```

```
}
```

- Initializes serial communication at 115200 baud.
- Starts the mesh network with our credentials.
- Attaches callback functions to handle events.

### **Sending Messages:**

1. **For broadcasting:** `mesh.sendBroadcast("Hello from node " + String(mesh.getNodeId()));`

**2. For direct messaging:** `mesh.sendSingle(targetNodeId, "Hello specific node");`

The `targetNodeId` is the ID of the node you want to send to.

### **Receiving Messages:**

```
void receivedCallback(uint32_t from, String &msg) {  
  
    Serial.printf("Received from %u msg=%s\n", from, msg.c_str());  
  
}
```

This is an automatic function that executes whenever a message has been received printing out a message, together with the sender ID on to the Serial Monitor.

## **6. Output**

**Task 1 - Broadcast Messaging:** We also coded the MCUs on NodeMCU to say every so often Hello on node ID. Each node in the network received these messages and displayed them in the Serial Monitor. This was a check which worked to confirm that the simple mesh configuration was active.



**Task 2 - Direct Messaging:** It altered the code to ensure that a message is only passed to a designated target node by calling sendSingle. Our messages were like Bristyr din e Lab e ashlam and Sir Valo achen? Ami Dipto!!!!!!!!!!!!!!". These messages were only received and displayed by the desired node, which is indicative of the fact that direct communication was functioning successfully.

```
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163269996 msg=Hello from node 1163269996
Changed connections
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
Changed connections
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163286391 msg=Bristyr din e Lab e ashlam :((((((((1163286391
Changed connections
startHere: Received from 1163312619 msg=Hello from node 1163312619

--> startHere: New Connection, nodeId = 1163269996
startHere: Received from 1163279684 msg=Hello from node 1163279684
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163286391 msg=Sir Valo achen? Ami Dipto!!!!!!!!!!!!!! 1163286391
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163279684 msg=Hello from node 1163279684
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163286391 msg=Sir Valo achen? Ami Dipto!!!!!!!!!!!!!! 1163286391
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163286391 msg=Sir Valo achen? Ami Dipto!!!!!!!!!!!!!! 1163286391
Changed connections
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163274111 msg=Hello from node 1163274111
startHere: Received from 1163286391 msg=Sir Valo achen? Ami Dipto!!!!!!!!!!!!!! 1163286391
startHere: Received from 1163269996 msg=Hello from node 1163269996
Changed connections
startHere: Received from 1163274111 msg=Hello from node 1163274111
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163286391 msg=Sir Valo achen? Ami Dipto!!!!!!!!!!!!!! 1163286391
```

**Task 3 - Multi-Hop Messaging :** We made a movement of the nodes, which entailed that Node

A and Node C were too far apart to do a direct connection. They had Node B in between them. As the message was sent directly between Node A and Node C, painlessMesh with the help of Node B rerouted this message. This was confirmed in Serial Monitor logs and proved the self-healing and support of multi-hop communication in the network.

```
Output Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM5')
Carriage Return 115200 baud

0x0wG0_awaG`00 00 [000]qB0 ``0wG0LlM000d0_0WGS_0Dw'wg'000 ``DWG0_awa'0_aw'0_00000G0wG'0000Gd0 ``G00startHere: Received from 1163286391 msg=Hello from node 1163286391
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163274111 msg=Hello from node 1163274111
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163286391 msg=Hello from node 1163286391
startHere: Received from 3299373438 msg=Hello from node ABC3299373438
Adjusted time 654058341. Offset = -101
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 3299373438 msg=Hello from node ABC3299373438
startHere: Received from 1163274111 msg=Hello from node 1163274111
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 3299373438 msg=Hello from node ABC3299373438
startHere: Received from 1163286391 msg=Hello from node 1163286391
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 3299373438 msg=Hello from node ABC3299373438
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163286391 msg=Hello from node 1163286391
startHere: Received from 1163274111 msg=Hello from node 1163274111
startHere: Received from 3299373438 msg=Hello from node ABC3299373438
startHere: Received from 3299373438 msg=Hello from node ABC3299373438
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163286391 msg=Hello from node 1163286391
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163274111 msg=Hello from node 1163274111
```

```

Output   Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM5')

startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163263447 msg=Hello from node 1163263447
startHere: Received from 1163274263 msg=Hello from node 1163274263
startHere: Received from 1163269996 msg=Hello from node 1163269996
startHere: Received from 1163312619 msg=Hello from node 1163312619
Changed connections
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163286391 msg=Bristyr din e Lab e ashlam :((((((((1163286391
Changed connections
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163286391 msg=Bristyr din e Lab e ashlam :((((((((1163286391
startHere: Received from 1163279684 msg=sobai poralekha koro, na hole jhamela hobe 1163279684
startHere: Received from 1163312619 msg=Hello from node 1163312619
startHere: Received from 1163286391 msg=Bristyr din e Lab e ashlam :((((((((1163286391
startHere: Received from 1163279684 msg=sobai poralekha koro, na hole jhamela hobe 1163279684
Changed connections
Changed connections
Changed connections
--> startHere: New Connection, nodeId = 1163274111

```

## **7. Observations**

Events were clear on the Serial Monitor in every stage. All nodes received broadcast messages, only the intended recipient received direct messages, and an intermediate node relayed multi-hop messages without any additional encoding. The adaptability of the network was observed immediately as the logs of changed connections as well as the new Connection were displayed immediately when nodes were moved.

## **8. Discussion**

The experiment revealed that the mesh networks are more flex and reliable as opposed to star topologies. Two nodes may be out-of-range, yet communication can be performed by means of the other nodes. Through the `painlessMesh` library this can be configured without having to write custom code to program routing or connection logic. We also noticed the topology of the network having changed in real time because of the messages in the form of changed connections. This can be handy in practical situations where the devices might get moved around or connections might degrade over the course of its operation. The experiment has confirmed that mesh networking proves to be a good means of establishing strong and scalable IoT systems.