EAST WEST UNIVERSITY

Mesh Networking with NodeMCU ESP8266 and painlessMesh

Course Title: Internet of Things

Course Code: CSE 406

Section: 01

Submitted by

Name: Akash Saha

ID: 2022-2-60-081

Date: 10/08/2025

Submitted to

Dr. Raihan Ul Islam (DRUI)

Associate Professor, Department of Computer Science and Engineering East West University

Other Group Members

Jobayer Faisal Fahim	2022-2-60-130
Shafiqul Islam Fahim	2022-2-60-085
Mahir Faysal Dipto	2022-2-60-

Introduction

This lab explores the concept of mesh networking in IoT systems using NodeMCU ESP8266 microcontrollers and the painlessMesh library. A mesh network enables nodes to communicate directly or via intermediate nodes without requiring a central router. This architecture improves network resilience, supports self-healing, and extends communication range through multi-hop routing.

The main objective is to implement and understand broadcasting, direct messaging, and multi-hop routing using painlessMesh, along with interpreting network event callbacks.

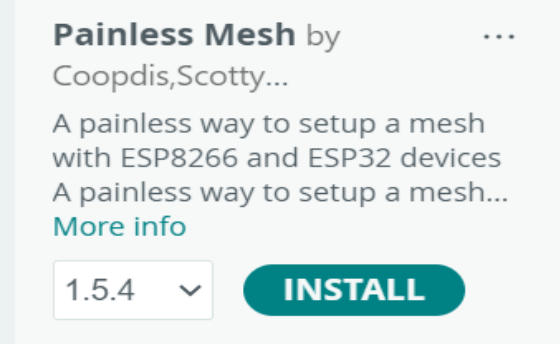
Required Materials

Hardware:

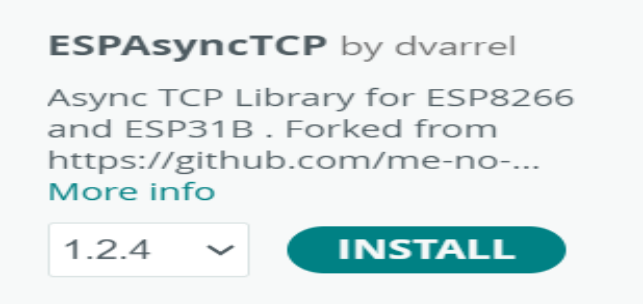
- 3 × NodeMCU ESP8266 boards
- Micro-USB cable

Software:

- Arduino IDE
- *painlessMesh* library (via Library Manager)



- ESPAsyncTCP library (dependency for painlessMesh)



Tools

- **Serial Monitor**: Used for observing node behavior and messages exchanged between nodes during the experiment.

Tasks and Implementation:

Task 1: Message Interpretation

Objective: This task aimed to understand PainlessMesh's standard system messages and demonstrate broadcast operation to reveal the mesh network's internal behavior.

Message Types:

- **New Connection:** Triggered when a new node joins and connects directly.

```
--> startHere: New Connection, nodeId = 1163312619
```

- **Connection Change:** Indicates topology changes such as nodes joining or leaving.

```
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
Changed connections
Changed connections
Changed connections
```

- **Adjusted Time:** Synchronizes clocks across all nodes for coordinated actions.

```
Adjusted time 7650908. Offset = -464380674
Adjusted time 7876955. Offset = -15021
Adjusted time 8102105. Offset = 10107
Adjusted time 8322888. Offset = 49
```

Task 2: **Direct Messaging**

Objective: Modify code to send a message to a single target node using mesh.sendSingle()

```
void sendMessage() {
  uint32_t targetNodeId = 1163276666;
  if(mesh.isConnected(targetNodeId)) {
    String msg = "Direct hello from node " + String(mesh.getNodeId());
    mesh.sendSingle(targetNodeId, msg);
  } else {
    Serial.println("Target node not connected.");
  }
}
```

Output:

```
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
startHere: Received from 1163276666 msg=Hi akash from fahim
```

Task 3: Multi-Hop Messaging

The purpose of multi-hop messaging is to let devices communicate even when they are too far apart for a direct connection.

In a mesh network, the message can "hop" through intermediate nodes (other devices) until it reaches the destination.
This helps:

- **Extend range** — connect devices over longer distances.
- **Avoid dead zones** — use alternate paths if direct links are blocked.
- **Improve reliability** — if one path fails, the message can take another route.

Advantages of Mesh Topology

- **Resilience:** No single point of failure.
- **Scalability:** New nodes can be added without affecting others.
- **Extended Range:** Messages can hop between intermediate nodes.
- **Dynamic Routing:** Automatic path adjustment based on signal conditions.

Potential Applications

- Smart home automation
- Distributed sensor networks
- Disaster recovery communications
- Agricultural monitoring systems

Conclusion

This lab demonstrated the creation and modification of a mesh network using NodeMCU ESP8266 boards and painlessMesh. Broadcasting, direct messaging, and multi-hop routing were implemented and tested. The exercise provided insights into decentralized IoT communication, network self-healing, and adaptive routing mechanisms.