NexLattice: Secure, Plug&play Offgrid Wi-Fi Mesh for IoT

Submitted by

Debayangshu Sen

Roll No.: ADTU/2022-26/BCS/049

B.Tech CSE 7th Semester

Vivek Upadhaya

Roll No.: ADTU/2022-26/BCS/048

B.Tech CSE 7th Semester

Rajdeep Roy

Roll No.: ADTU/2022-26/BCS/046

B.Tech CSE 7th Semester

Under the Guidance of

Dr. Debashis Dev Misra

Associate Professor

Faculty of Computer Technology

Assam down town University

Guwahati-26, Assam.

Contents

Introduction

- The Internet of Things (IoT) has rapidly evolved into a network of billions of interconnected devices, but most current systems depend heavily on centralized servers or gateways.
- This dependency creates latency, reliability, and security issues when connectivity is disrupted.
- **NexLattice** aims to overcome these challenges by proposing a lightweight, secure, and universally compatible Wi-Fi–based mesh networking layer for embedded IoT devices.
- It focuses on enabling direct, peer-to-peer communication without relying on external infrastructure, ensuring low-latency and resilient data transfer across nodes.

Problem Statement

Lack of Universal Protocols

There's a lack of universal, secure, and lightweight Wi-Fi mesh protocols for IoT nodes.

Vendor Lock-in

Current solutions often depend on vendor-specific technologies, creating integration challenges.

Need for Self-Healing Networks

A need exists for a plug-and-play, self-healing network architecture for IoT ecosystems.

Lack of plug-and-play

The lack of a plug-and-play **Wi-Fi mesh protocol** for embedded systems creates a critical gap that NexLattice aims to address by establishing a plug-and-play communication framework across multiple hardware platforms.

Theoretical Study

(Existing Framework Analysis)

LoRa Mesh :

Study of LoRa Mesh to understand its application in long-range, low-power IoT networks including strengths and weaknesses.

Cons : Very low speed, high latency, needs dedicated transceivers.

ZigBee :

Investigation of ZigBee for its mesh capabilities, scalability, and secure communication protocols.

Cons : Low data rate, short range, requires special hardware.

ESP-Mesh :

Analysis of ESP-Mesh, focusing on its advantages and limitations within the Espressif ecosystem.

Cons : ESP-specific, not compatible with other Wi-Fi devices.

Motivation

- The motivation behind NexLattice stems from the increasing need for **secure, resilient, and infrastructure-independent** IoT communication.
- Having experience in troubleshooting and understanding network behavior, we aims to design a solution that combines simplicity with technical strength.
- The system will serve as a practical demonstration of how everyday Wi-Fi devices can self-organize into a reliable communication network—useful for smart cities, remote sensor deployments, and research applications and mainly offgrid settlers.

Objectives

The primary objective of NexLattice is to develop a **universal Wi-Fi mesh layer** that allows IoT devices to communicate securely and efficiently without relying on cloud infrastructure.

Specific goals include:

- Designing a **lightweight routing and discovery protocol** for embedded nodes.
- Implementing **encryption and authentication mechanisms** to ensure data security.
- Creating a **plug-and-play architecture** compatible with MicroPython-enabled devices.
- Building a **dashboard interface** to visualize node connections and communication flow.

Plan of work

Phase 1 (Oct–Nov 2025)

Literature review and theoretical design of mesh protocol.

Phase 2 (Dec 2025)

Node communication demo using ESP32s.

Phase 3 (Jan–Feb 2026)

Implement discovery, routing, and encryption.

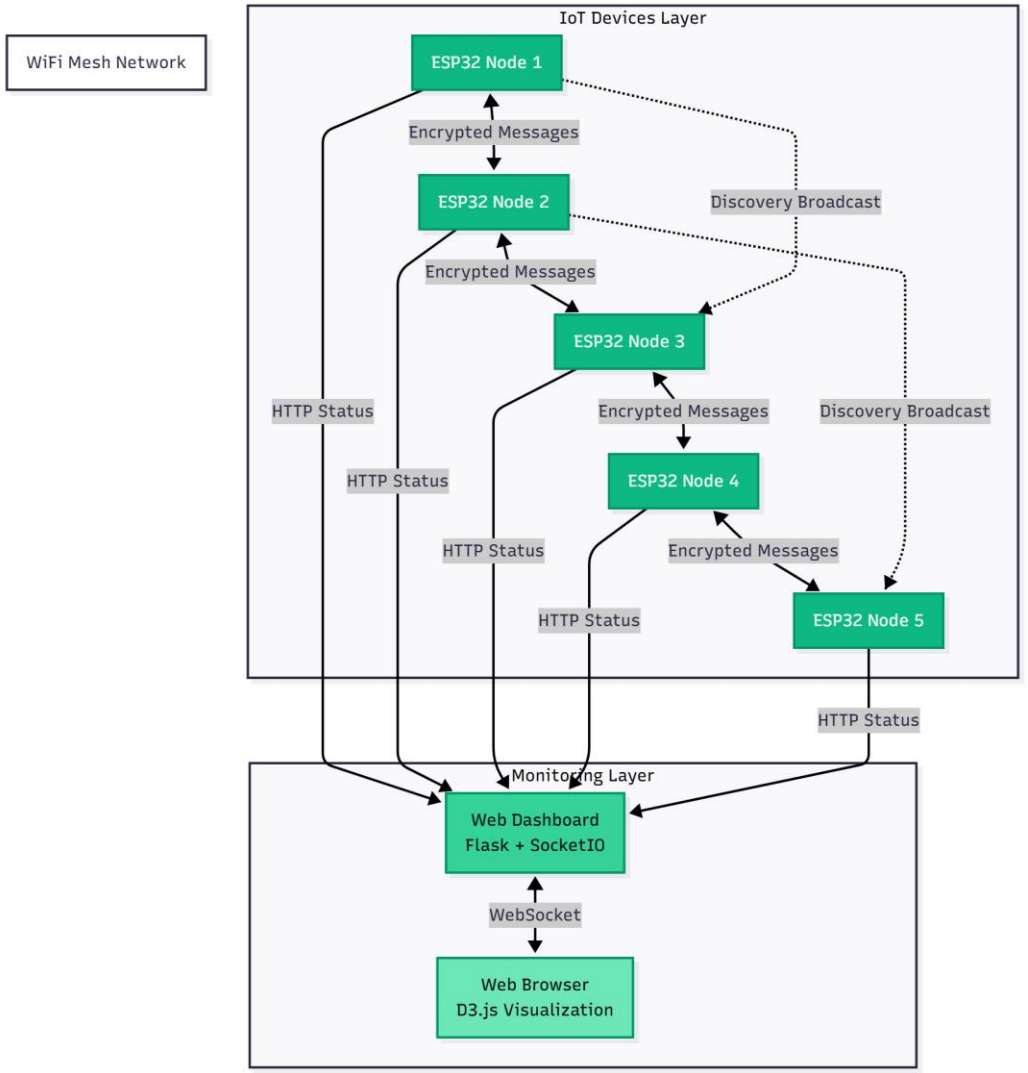
Phase 4 (Mar 2026)

Dashboard development and integration.

Phase 5 (Apr 2026)

Testing, documentation, and final evaluation.

Architecture Overview



Project Structure

```
NexLattice/
├── devices/                    # ESP32 MicroPython code
│   ├── node_main.py           # Main node logic
│   ├── network_manager.py     # WiFi and communication
│   ├── crypto_utils.py        # Encryption and keys
│   ├── message_router.py      # Routing algorithm
│   └── config.json            # Node configuration
├── dashboard/                  # Web dashboard
│   ├── app.py                 # Flask backend
│   ├── templates/
│   │   └── index.html         # Dashboard UI
│   └── static/
│       └── dashboard.js       # Real-time visualization
├── simulator/                  # Virtual testing
│   └── network_simulator.py   # Software simulation
├── docs/                       # Documentation
│   ├── architecture.md        # System architecture
│   ├── protocol_design.md     # Protocol specification
│   └── setup_instructions.md  # Deployment guide
├── tests/                      # Test plans
│   └── test_plan.md           # Comprehensive testing
├── logo/                       # Project logos
└── requirements.txt            # Python dependencies
```

Expected Outcomes

Functional Prototype

- A **5-node Wi-Fi mesh network** demonstrating secure, autonomous peer-to-peer communication. (Demo)

Performance Validation

- Measure **packet delivery**, **latency**, and **routing stability** in real-time tests.

Security & Reliability

- Encrypted, authenticated data exchange between independent IoT nodes.

Theoretical Model

- Framework showing how **Wi-Fi-based mesh** can **replace or supplement** cloud-dependent IoT systems.

Vision for the Future

- A step toward **universal, offgrid, plug-and-play IoT networking** — independent, scalable, and infrastructure-free.

Thank You