DEVELOPMENT AND OPTIMIZATION OF A BLUETOOTH LOW ENERGY (BLE) MESH NETWORK FOR ENVIRONMENTAL MONITORING AND ALERTING

Thesis presentation
Mohamed Mahmud Ali
May 22nd 2024

VAMK

INTRODUCTION

This thesis develops an IoT smart system for integrating devices and sensors, such as air quality , temperature sensors, etc as well as control systems such as lights, gates, fire systems, etc. within a large network. The aim of this thesis is to evaluate the range and data accuracy across various nodes within Bluetooth network.

Challenges:

❑Data Accuracy and Reliability:

- Sending sensor data over long distances can lead to incorrect data delivery.

❑Real-Time Data Processing:

- Delays in data transmission and processing can hinder timely alerting and response actions.
- High volumes of data from numerous sensors require robust and efficient processing systems.

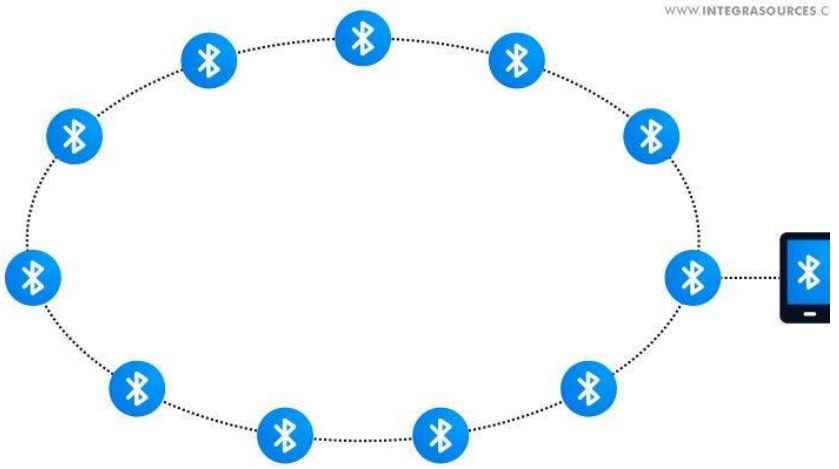
❑Scalability Issues:

- Expanding monitoring systems to cover larger or more varied areas can be technically and financially challenging.
- Integrating new sensors or expanding network coverage often requires significant system reconfiguration.

❑Power Management:

- Remote sensors and devices often have limited power sources, requiring efficient power management to ensure continuous operation.
- Energy consumption becomes a critical issue as the network of sensors expands.

OBJECTIVES



1. **Create a BLE Mesh Network:**
   Design and implement a robust BLE mesh network specifically tailored for environmental monitoring using nRF52 semiconductors microcontrollers.

2. **Ensure Data Reliability**
   Monitor and record the forwarding of messages by each relay node to adjacent nodes within the network.

3. **Optimize Network Performance**:
   Enhance scalability and power efficiency to ensure the network can handle increasing loads with minimal energy consumption.

4. **Test and Validate the System**:
   Conduct comprehensive testing in real-world environments to validate the performance and effectiveness of the BLE mesh network.

nRF52

SOLUTION

Development of a BLE Mesh Network: Suitable specifically for monitoring, alerting and control, leveraging the unique properties of Bluetooth Low Energy technology.

Scalability:

- Supports up to 32,767 nodes /sensors within the network.
- Ideal for complex and extensive environmental monitoring systems.

Reliability

- Data can be relayed across multiple paths, enhancing the network's robustness and reducing the risk of data loss.

Energy Efficiency:

- Optimized for minimal power usage, significantly prolonging battery life in deployed sensor devices.
- Reduces maintenance frequency and operational costs.

Wide Area Coverage:

- Capable of covering large geographical areas.
- Ensures comprehensive environmental data collection across diverse locations.

WHAT IS BLUETOOTH MESH ?

Bluetooth mesh is a network protocol designed for creating large-scale device networks. It is ideal for building automation, sensor networks, and other IoT (Internet of Things) solutions where hundreds or thousands of devices need to reliably and securely communicate with one another.

DIFFERENCES:

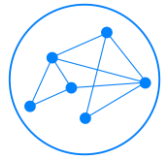
- **Bluetooth classic**
  - exchange data over short distances using radio waves 2.4GHz (10 meters range).
  - Topology : point-to-point or at best star

- **BLE**
  - Compared to Classic Bluetooth, Bluetooth Low Energy is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range. (10-meter range).
  - Topology: point-to-point or at best star

- **BLE mesh**
  - is a protocol build on BLE stack to connect and communicate in mesh network (The more nodes in network, more range).
  - Topology : flooded mesh

Bluetooth Classic

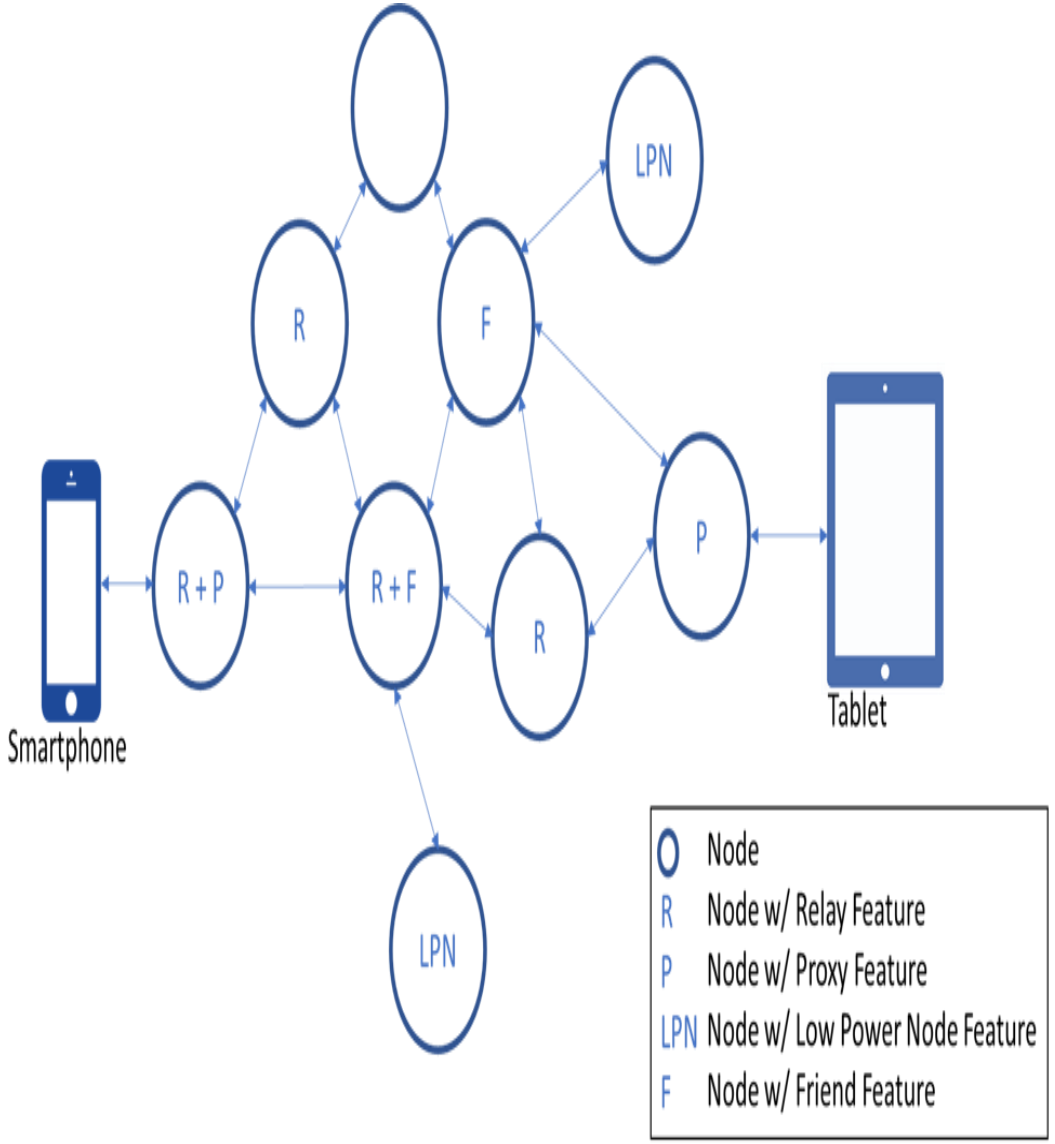
Bluetooth low energy



BLE-mesh

FEATURES AND ROLES

- **Relay Node**
  - Enables a node to relay messages over the advertising bearer.
- **Low Power Node (LPN)**
  - conserves energy by allowing devices to sleep, waking only to check for messages with a Friend node.
- **Friend Node**
  - will listen for any messages that are relayed in the network and intended for an associated Low-Power Node.
- **Proxy Node**
  - The Proxy feature allows a node to relay messages between the GATT (General ATTribute) and advertising bearers.
- **Provisioner**
  - is a device that add unprovisoned nodes in the network



RELAYING

Relay devices retransmit messages

- Maximum of 127 hops
- Enough to relay across an enormous physical area

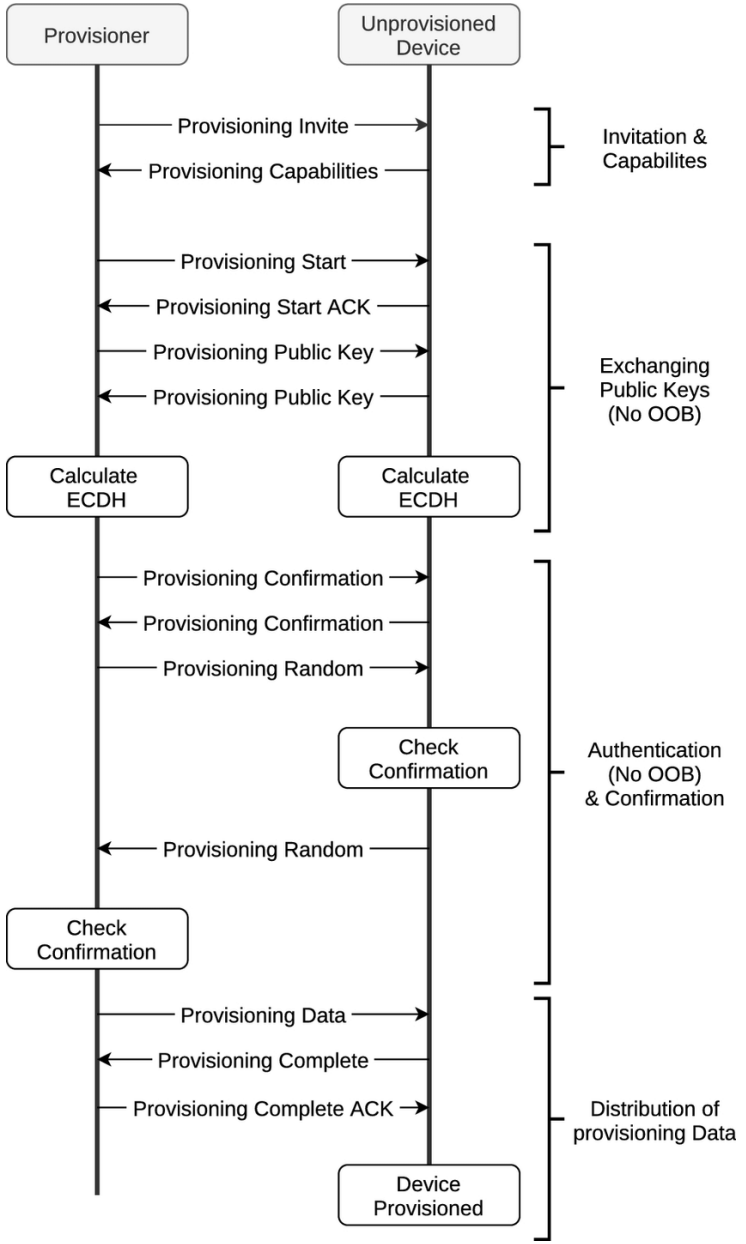
Managed flooding

- **Retransmit the message to all other devices in range**
- **All packets include a time-to-live (TTL)**
- **Message cache to reduce overhead**

PROVISIONING

Provision is the process of providing new devices in the Bluetooth mesh network with the information they need to join a network. To become a node and participate in the Bluetooth mesh communication, each device must be provisioned. Provision following two roles:

- The provisioner represents the network owner, and is responsible for adding new nodes to the mesh network.
- The provisionee is the device that gets added to the network through the Provisioning process. Before the provisioning process starts, the provisionee is an unprovisioned device.



SECURITY

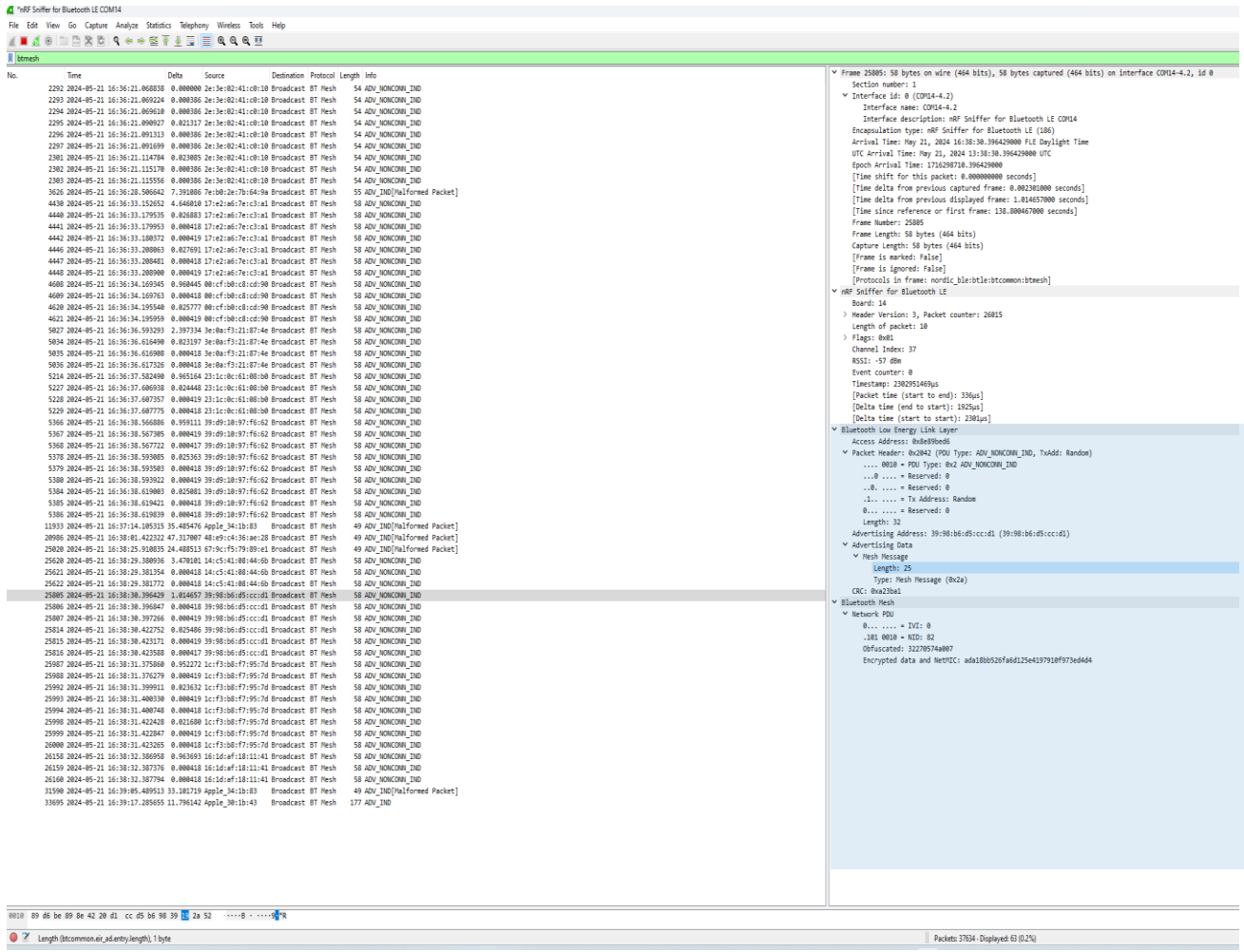
Network layer

- Every packet is encrypted and authenticated
- Sequence numbers to avoid replay attacks
- Man-in-the middle protection
- Protection against trash-can attacks
- Security keys are refreshed on regular basis

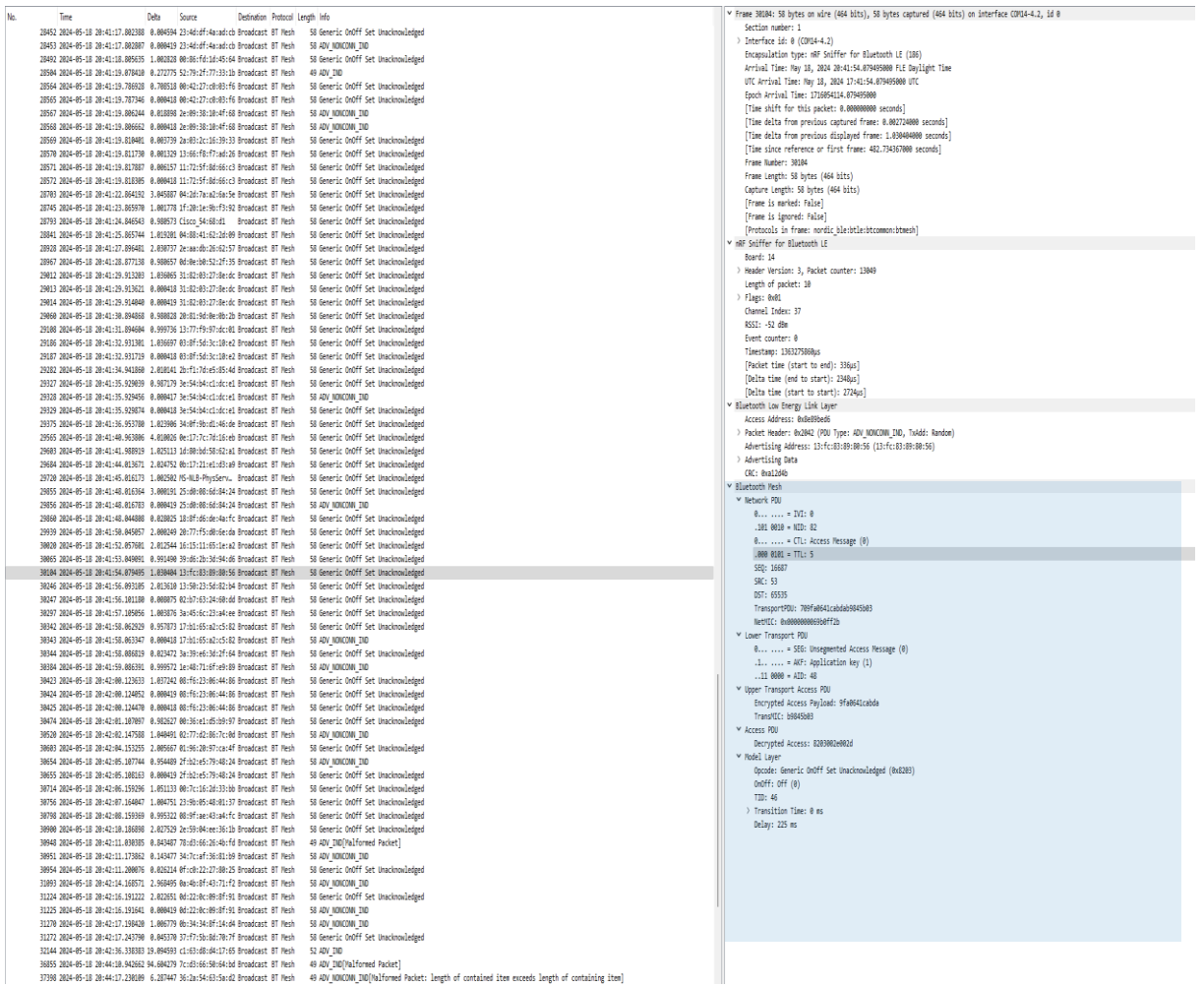
Access layer

- Separation into network keys, application keys and device keys.

ENCRYPTED DATA



DECRYPTED DATA



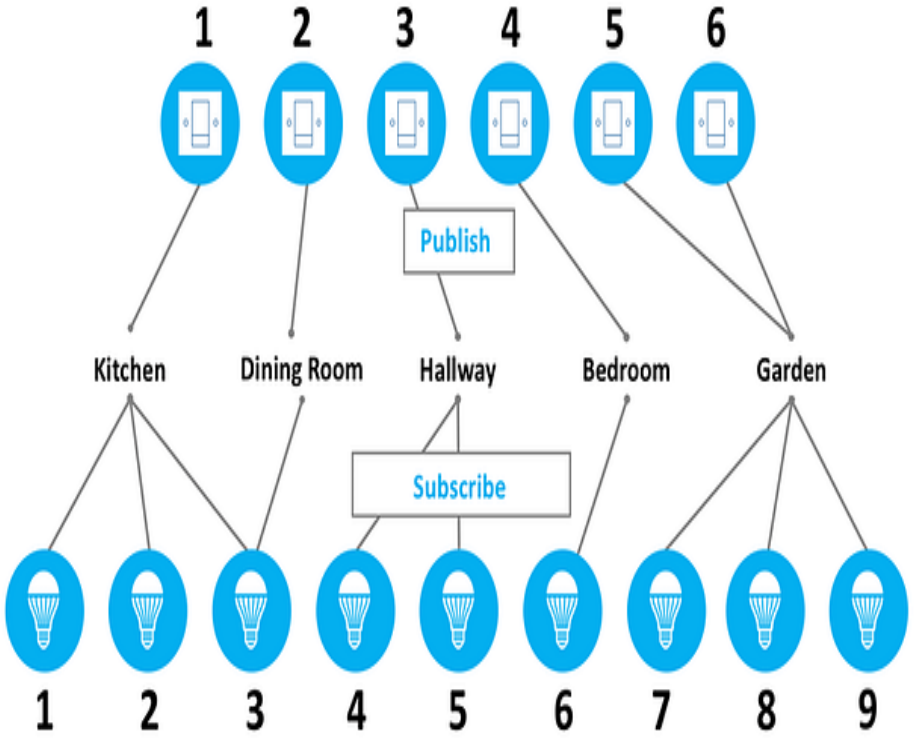
PUBLISH AND SUBSCRIBE

Publishing

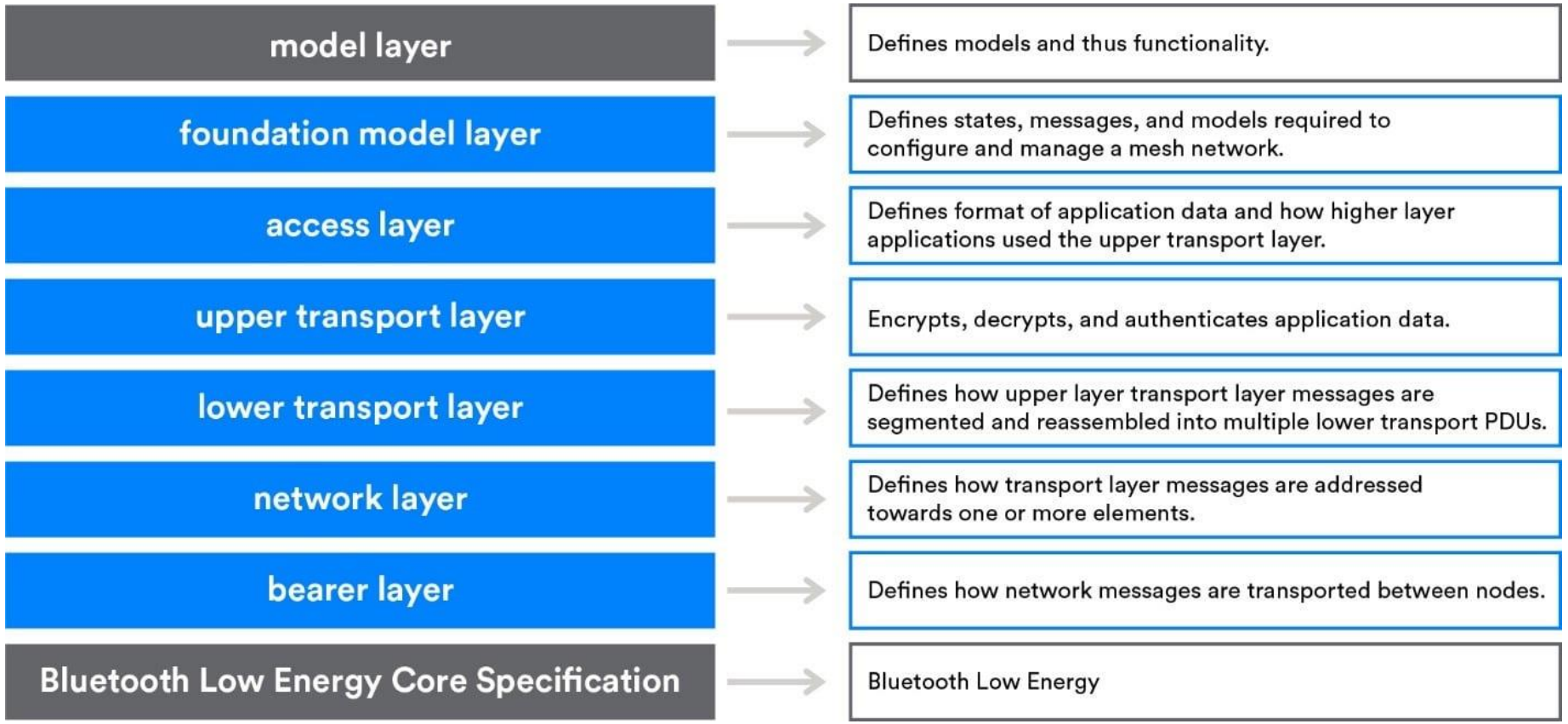
- Devices may send messages to addresses (group), all the other devices that subscribed to that address will receive a copy of it, process it and react in some way.

Subscribing

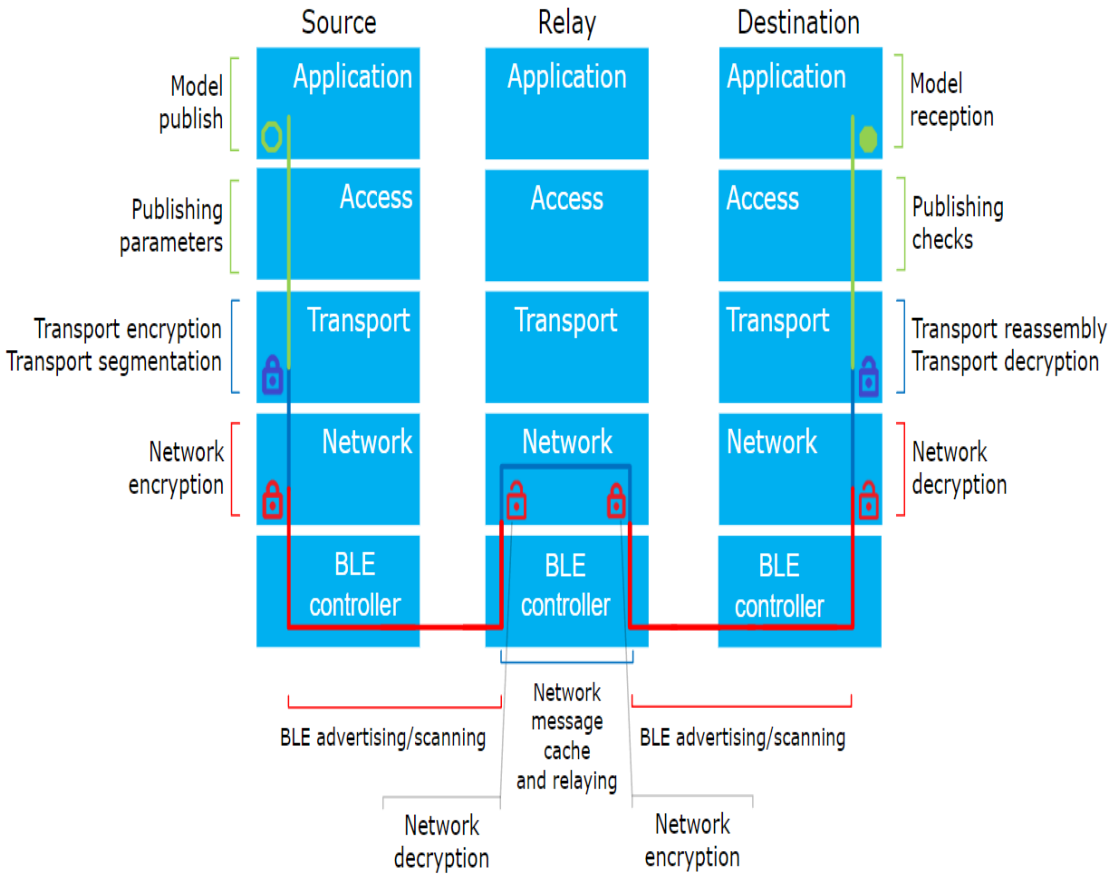
- Devices can be configured to receive messages which were sent to addresses (group) by other device.



BLE MESH ARCHITECTURE



BLE MESH NETWORK DATA FLOW



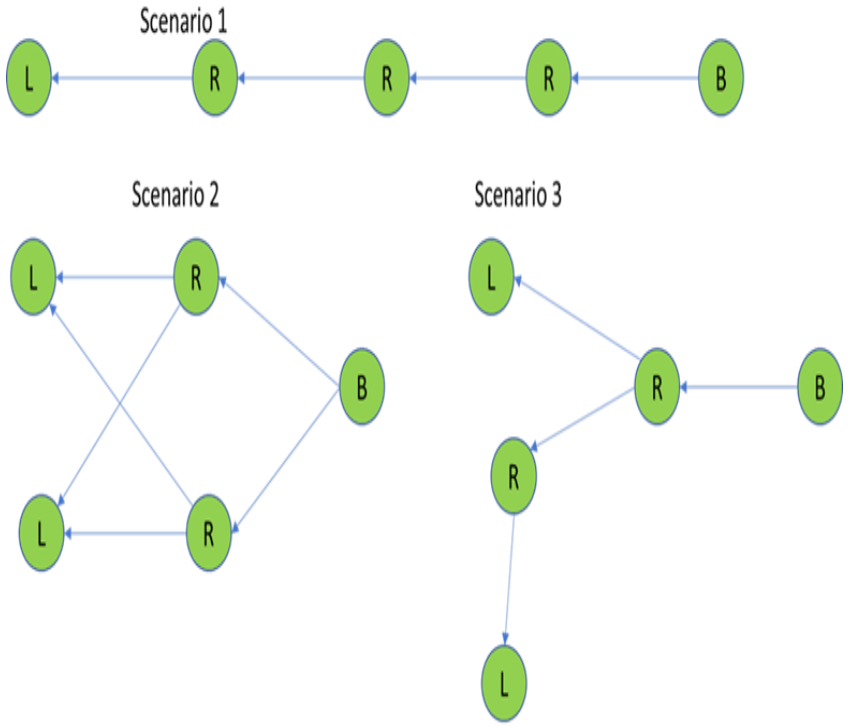
MODELS

The applications are implemented as a set of mesh models. The Bluetooth SIG defines some generic and reusable models in the Bluetooth Mesh model specification, but vendors are free to define their own models.

EXPERIMENT SETUP

System contains 3 roles:
B: broadcaster
R: Relay
L: Listener

- Broadcaster initiate a sequence of messages to all nodes.
- Relays receives and calculate PLR (Packet Lose Ratio) and forward them to adjacent nodes.
- Listeners will record these messages and calculate PLR also.



SCENARIOS TESTS

Links between the nodes will be 15 and 30 meters.

- Scenario 1:

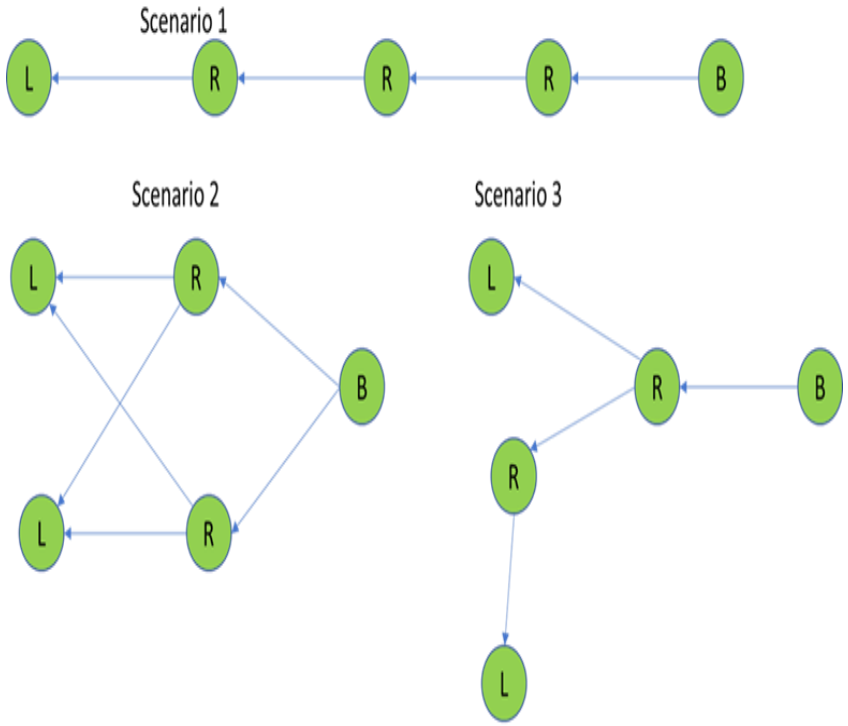
  The broadcaster will send 60, 120, and 600 messages over a 60-second period. Each node will record the Packet Loss Ratio (PLR).

- Scenario 2:

  Two relay nodes will be positioned centrally to facilitate message forwarding to the listeners.

- Scenario 3:

  One listener will receive messages through a single hop, while another listener will receive messages through two hops.



PLR CALCULATION

```c
static int gen_onoff_set_unack(const struct bt_mesh_model *model,
                               struct bt_mesh_msg_ctx *ctx,
                               struct net_buf_simple *buf)
{

    if (buf->len < 4) {
        printk("Buffer too short\n");
        return -EINVAL;
    }
    uint8_t val = net_buf_simple_pull_u8(buf);
    uint16_t seq = net_buf_simple_pull_le16(buf);
    uint8_t tid = net_buf_simple_pull_u8(buf);

    int32_t trans = 0;
    int32_t delay = 0;
    printk("Receiving onoff:%d, tid:%d ,src:%d ,seq:%u \n", val, tid, ctx->addr ,seq);

    PLR(seq);
    /*if (ctx->recv_ttl < 1) {
        rebroadcast(model, ctx, seq, ctx->recv_ttl);
    }*/


    if (buf->len) {
        trans = model_time_decode(net_buf_simple_pull_u8(buf));
        delay = net_buf_simple_pull_u8(buf) * 5;
    }

    /* Only perform change if the message wasn't a duplicate and the
     * value is different.
     */
    if (tid == onoff.tid && ctx->addr == onoff.src) {
        /* Duplicate */
        return 0;
    }

    if (val == onoff.val) {
        /* No change */
        return 0;
    }
```

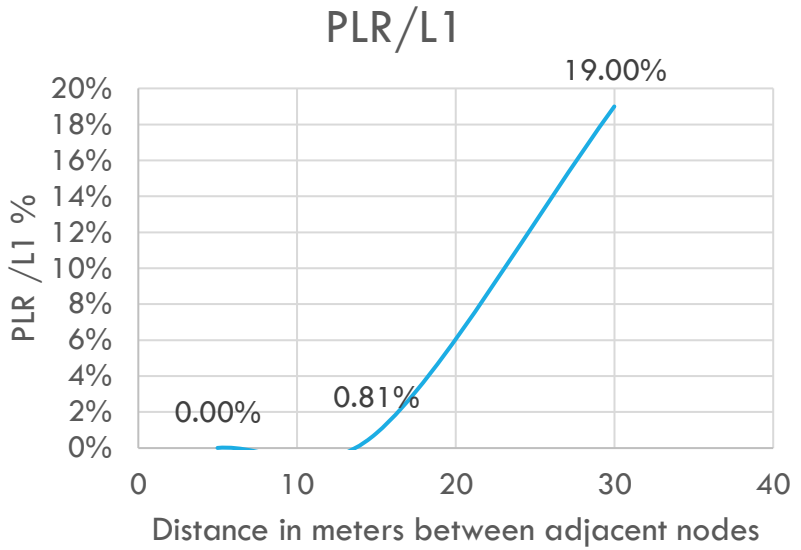
```c
static int PLR(uint16_t seq_num) {
    // Static variables to keep track of sequence and statistics
    static uint16_t last_seq_num = 0;
    static uint16_t first_seq_seq = 0;
    static bool is_first_packet = true;
    static uint16_t total_missed_packets = 0;
    static uint16_t total_received_packets = 0;

    // Handling the first packet received
    if (is_first_packet) {
        first_seq_seq = seq_num;
        last_seq_num = seq_num ;  // Initialize the last sequence number with the first received sequence
        is_first_packet = false; // Mark that the first packet has now been received
        total_received_packets++; // Start counting packets with the first one received
    } else {
        // Normal handling for all subsequent packets
        if (seq_num > last_seq_num) {
            // Calculating missed packets if there's a gap
            total_missed_packets += (seq_num - last_seq_num - 1);
            last_seq_num = seq_num; // Update the last received sequence number
            total_received_packets++; // Increment the total received packets count
        } else if (seq_num == last_seq_num) {
            // Handle duplicate packet
            printf("Duplicate packet received, seq_num: %u\n", seq_num);
        } else {
            // Handle out of order packet
            printf("Out of order packet received, seq_num: %u, last_seq_num: %u\n", seq_num, last_seq_num);
        }
    }

    // Calculate Packet Loss Ratio (PLR)
    uint32_t total_packets_considered = total_received_packets + total_missed_packets;
    uint32_t plr = (total_missed_packets * 10000) / total_packets_considered;

    printk("First received packet: %u\n", first_seq_seq);
    printk("Total missed packets: %u\n", total_missed_packets);
    printk("Total received packets: %u\n", total_received_packets);
    printk("Received seq_num %u, PLR: %u.%02u%%\n", seq_num, plr / 100, plr % 100);
    printk("......................\n");

    return 0;
}
```

RESULTS

Scenario1:

- At **~15 meters**:
  - Packet Loss Ratio (PLR) was minimal across all message sizes (60, 120, 600), with nearly zero loss at the relay nodes and minor loss at the listener.
  - Received Signal Strength Indicator (RSSI) ranged from -60dBm to -53dBm, indicating strong signal strength.
- At **~30 meters**:
  - PLR increased notably, especially with higher message counts (120, 600), showing increased loss primarily at the listener.
  - RSSI decreased to as low as -81dBm, suggesting weaker signals at greater distances.

Distance	payload	PLR/L	PLR /R3	PLR /R2	PLR /R1	RSSI
~15m	60	1.44%	0%	0%	0%	~-60dBm
	120	0.83%	0%	0%	0%	~-52dBm
	600	0.17%	0.34%	0.52%	0.0%	~-53dBm
~30m	60	16.3%	11.7	3.27%	3.27%	~-68dBm
	120	26%	4.68%	0%	0%	~-81dBm
	600	14.74%	5.16%	5.16%	1.06%	~-77dBm

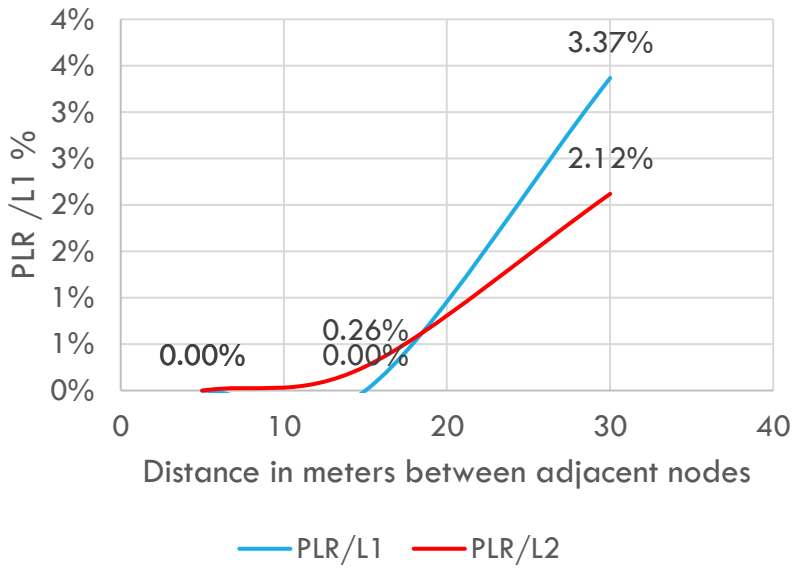


RESULTS

Scenario2:

- At **15 meters**:
  - Very low to zero PLR across all nodes and message sizes, with excellent RSSI values around -51dBm to -54dBm.
- At **30 meters**:
  - PLR varied, with higher message sizes showing increased loss, particularly notable in messages directed to listeners.
  - RSSI showed some decrease but remained relatively strong, suggesting that the double relay setup helps maintain signal integrity over greater distances.

Distance	payload	PLR /L1	PLR /L2	PLR /R2	PLR /R1	RSSI
15m	60	0%	0%	0%	0%	~-51dBm
	120	1.3%	0.78%	6.3%	3.9%	~-57dBm
	600	0%	0%	0.16%	0.16%	~-54dBm
30m	60	0	0%	4.76%	0%	~-57dBm
	120	8.13	0.80%	0%	0%	~-60dBm
	600	1.98%	1.32%	1.48%	0.49%	~-64dBm

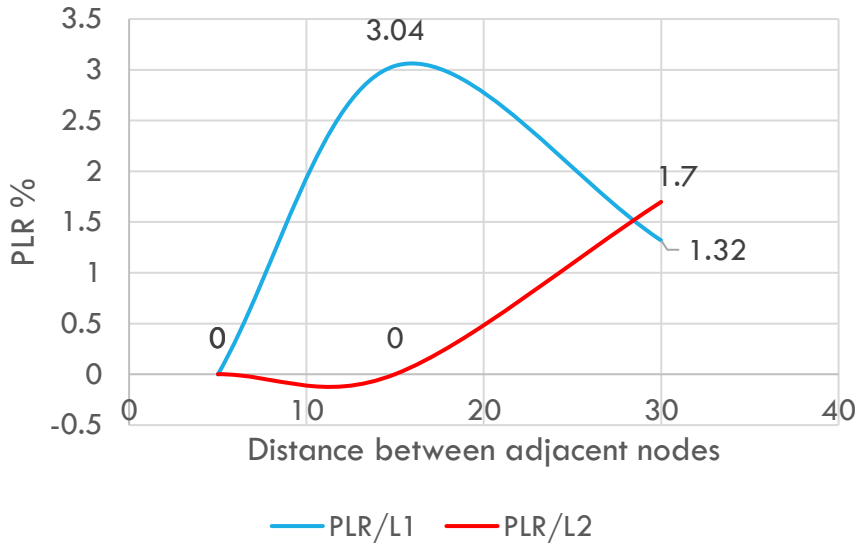


RESULTS

Distance	payload	PLR /L1	PLR /L2	PLR /R2	PLR /R1	RSSI
15m	60	4.54	0	0	0	~-63dBm
	120	2.15	0	0	0	~-57dBm
	600	2.43	0	0	0	~-58dBm
30m	60	0	0	3.27	0	~-58dBm
	120	0.78	0	0.78	0	~-55dBm
	600	3.18	5.20	5.03	0.50	~-63dBm

Scenario 3:

- At **15 meters**:
  - Higher PLR for the listener requiring more hops (L1), especially as message size increased, indicating that additional hops can lead to increased packet loss.
  - RSSI was slightly lower but still within a good range, averaging around -57dBm to -63dBm.
- At **30 meters**:
  - Significant PLR increase for both listeners as message size increased, particularly for L2.
  - RSSI values showed more substantial drops, highlighting challenges with signal strength at longer distances and multiple hops.



Q&A

THANK YOU