Performance Evaluation of Deployable Bluetooth Low Energy Mesh Network for Monitoring System

Weiwen Wang

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.5.2022

Supervisor

Prof. Riku Jäntti

Advisor

Dr. Reino Virrankoski

Aalto University
School of Electrical Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Weiwen Wang	
Title Performance Evaluation of Deployable Bluetooth Low Energy Mesh Network for Monitoring System	
Degree programme Automation and Electrical Engineering	
Major Electronic and Digital Systems	**Code of major** ELEC3060
Supervisor Prof. Riku Jäntti	
Advisor Dr. Reino Virrankoski	
Date 30.5.2022 **Number of pages** 54	**Language** English

Abstract

The development of wireless sensor network (WSN) technology has promoted automation and monitoring systems. Deploying small, low-power and lightweight nodes in the area can monitor and collect, analyze, and process information from the environment in real-time. In recent years, there have been many critical breakthroughs in Bluetooth technology. Bluetooth Low Energy (BLE) technology provides sensor networks with standardized network protocols, facilitates communication between sensor networks and heterogeneous devices, and reduces development costs. BLE Mesh enables many-to-many types of communication and opens the way to new possible applications. This thesis aims to build a prototype for an outdoor BLE mesh network monitoring system using PIR and acoustic sensors. It can detect the invasion of people by PIR sensing and compute feature vectors from the voice samples collected by the acoustic sensor. This thesis has evaluated the communication performance of the BLE Mesh network, including packet delivery rate (PDR) and round-trip time (RTT) in different network architectures, as well as the power and memory performance of nodes. The results show that the PDR maintained 100% within 13 meters when deploying BLE Mesh nodes outdoors. Adding relaying nodes can extend the reliable communication distance, but the RTT increases by about 25 ms every hop. The battery in this system can support node work for more than 9 hours in low temperatures environment.

Keywords Wireless sensor network, BLE Mesh , PIR sensing , Speaker identification

Preface

This thesis is based on the research work I did as a research assistant in the Department of Communication and Networking (Comnet) at Aalto University from April 2021 until June 2022. My research work is part of the MUTICO project. The MULTICO project is a collaboration between Aalto University, VTT, and several domestic companies to develop an ecosystem using sensor systems connected to minisatellites and multicopter-type of drones and deploy these drones to generate a real-time situation model of the observed environment.

I would like to express my appreciation to all the people who gave their support for this thesis. First of all, I would like to thank my instructor Reino Virrankoski and supervisor Riku Jäntti for allowing me to work on such an exciting topic. Reino Virrankoski is a great advisor who provided me with many research ideas, a pleasant research environment, and many skills in thesis writing and presentation. Also, I would like to thank my colleagues Lauri Mela for giving me advice on batteries, Kinnari Jouko for providing the advice on sensors, Gerhard Malberg for the cooperation of BLE connection between nodes and drones, and Lingyun Yao for the help with experiments. Their kindness and tireless support were essential input for this thesis. Finally, I want to thank my family for their support and encouragement.

Last but not least, I would like to thank Aalto University for providing me with wonderful two years. I have gained knowledge, friendship, open-mindedness, and rigorous thinking here. I will always cherish and miss this wonderful time.

Otaniemi, 20.5.2022

Weiwen Wang

Contents

Abbreviations

WSN	Wireless sensor network
BLE	Bluetooth Low Energy
SIG	Special Interest Group
IoT	Internet of Things
BR/EDR	Basic Rate/Enhanced Data Rate
ISM	Industrial, scientific and medical
GAP	Generic Access Profile
GATT	Generic Attribute Profile
L2CAP	Logical Link Control and Adaptation Layer Protocol
SDU	service data units
HCI	Host Controller Interface
PHY	Physical layer
GFSK	Gaussian frequency-shift keying
LL	Link layer
PDU	Protocol Data Unit
OOB	Out of Band
UUID	Universal Unique Identifier
TTL	Time To Live
AES	Advanced Encryption Standard
SEQ	Sequence Number
PIR	Passive infrared sensor
MFCC	Mel-frequency Cepstral Coefficient
GPIO	General purpose input and output
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface Bus
DK	Development kit
SDK	Software development kit

1 Introduction

The successful applicability of wireless sensor networks (WSNs) [1] on the Internet of Things (IoT) solutions has garnered growing attention recently; this has been due to the robust scalability and flexibility of WSNs. In general, wireless sensor networks contain a set of sensor nodes that can collect information, process data, and communicate with the central server. They are commonly applied to extensive monitoring and automation systems.

New research directions in WSN applications, including localization systems [2], public lighting applications [3], and monitoring systems in smart cities [4], have promoted research of large-scale networks with many nodes. This research has revealed that some problems have arisen when building large-scale networks, including scalability, communication traffic, and available bandwidth. These types of issues have consequently increased the difficulty of communication. One promising approach is the mesh network [5]. One of the key benefits provided by the mesh architecture is multi-path communication. It enables the network to cover much broader areas than the coverage of a single radio link would enable. It also improves the robustness on the network level because there are several alternative communication paths between the nodes. Mesh architecture provides automatic self-configuration and self-healing abilities, constantly adjusting the network to the best condition and performance.

Another challenge regarding WSNs concerns their quality of service as it depends on many factors, including data throughput capacity, communication reliability, communication latency, and battery life of the network [6]. Therefore, improving the quality of service is another crucial topic in WSNs research. Bluetooth Low Energy (BLE) [7] is a short-range wireless communication technology with low energy consumption. In 2017, Mesh Profile based on BLE technology was released by the Bluetooth Special Interest Group (SIG). It provides mesh support for BLE communication. BLE Mesh considers the reduction of energy consumption and optimization for creating large-scale device networks, which improves the network reliability, security, and robustness. Furthermore, it provides developers with a full-stack implementation, reducing development difficulty. BLE Mesh is applied in building automation and commercial lighting solutions [8]. It helps reduce power consumption, optimize space utilization, and create other cost-saving applications for buildings and factories.

However, few studies have addressed BLE Mesh solutions in outdoor monitoring systems. The outdoor environment has many elements, such as interference, radio-path fading effects, low temperatures, rain, and snow (especially in Finland), which affect the performance of the mesh network.

This thesis developed a BLE mesh architecture for passive infrared (PIR) and acoustic monitoring and tested its performance through several experiments. In this system, each node in the network is equipped with acoustic and passive infrared sensors, which can detect the presence of people and possibly their identity. Once the presence of people has been detected, the node will record the voice samples and compute a feature vector for speaker identification.

The rest of the thesis is organized as follows. Chapter 2 presents a brief overview of wireless sensor networks and mesh network architecture. The Bluetooth Low Energy technology BLE Mesh is discussed in Chapter 3. This chapter presents the communication architecture of the network and explains related terminology, which will appear later. Chapter 4 briefly covers the concept of sensing techniques used in the monitoring system. The developed system architecture is presented in Chapter 5. The experiment setup, experimental results, and analysis are discussed in Chapter 6. Finally, chapter 7 presents the conclusions and future work.

2 Wireless Sensor Networks

2.1 General Requirements

Sensors are the foundation and core of the Industrial Internet and a key component of automated intelligent equipment. The vigorous development of the Industrial Internet has brought enormous opportunities to sensor companies but has also put forward new requirements, including accuracy, stability, and robustness [9].

Wireless sensor networks (WSNs) are built to collect data and information from the environment by deploying many sensors in the area over a wireless link. However, it is not enough to observe and collect data in many applications. The system needs to take corresponding operations based on the sensing data. This heralds the emergence of wireless sensor/actuator networks (WSANs) [10]. WSANs consist of sensor platforms and actuators, which can communicate wirelessly. Sensor platforms are called sensor nodes (earlier also motes).

2.1.1 Node Structure and Roles

A sensor node typically has at least a radio, power source, microprocessor or microcontroller, some memory, and one or several sensors. The microprocessor and memory are usually embedded in the sensor node. Figure. 2.1 shows the components in a typical sensor node. The actuator node has a similar composition but replaces the sensor unit with the actuator.

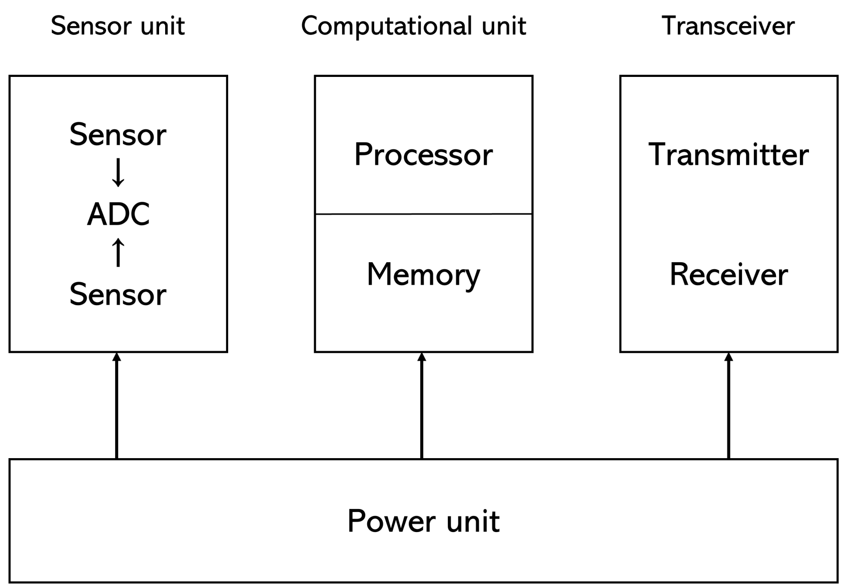

Figure 2.1: Node components

Nodes have different tasks in the network architecture. A sensor node can also function as a clusterhead, a sink, or a gateway. Sensor nodes can be separated into different clusters to improve scalability and energy efficiency [11]. A sensor node in the cluster can be selected as the cluster head. It can combine the sensed data from the sensor nodes in the same cluster and communicate with the actuators or sink nodes. The clusterhead should have the most power supply in the cluster. Sink nodes can gather information from many sensor nodes for further processing. They have relatively stronger processing, storage, and communication capability. Moreover, external users can connect to the sink node to load the raw data the network collect. Figure. 2.2 presents an example of WSAN architecture containing different nodes.

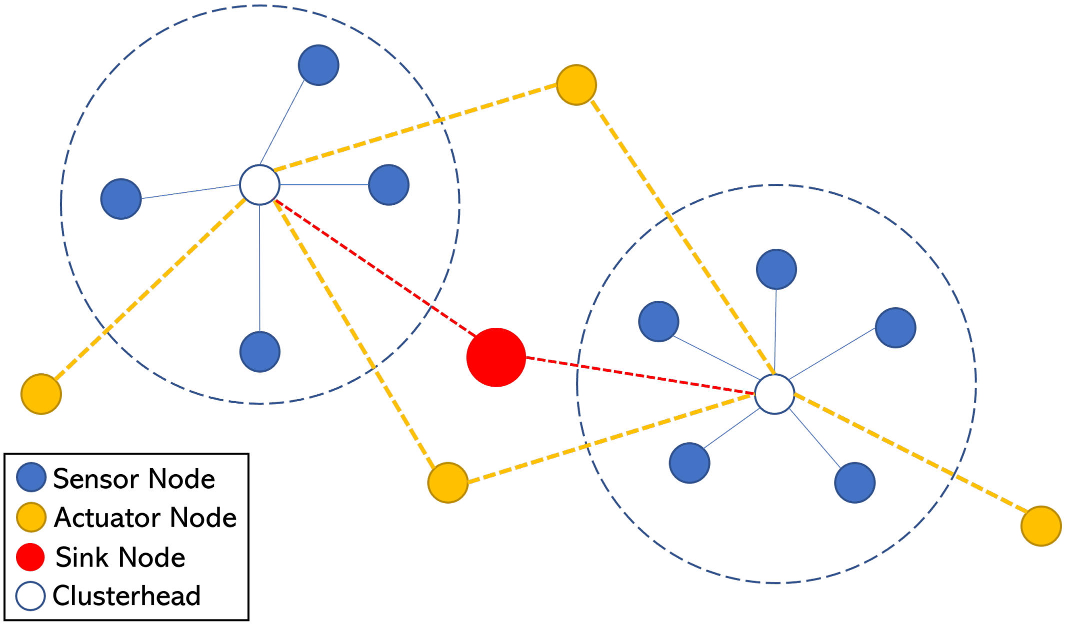

Figure 2.2: An example of WSAN architecture

2.1.2 Applications of WSNs

Wireless sensor networks have been successfully applied in many application domains, including military[12], agriculture[13], environmental monitoring[14], medical[15], and industrial automation[16]:

- Military: the monitoring of enemy forces and equipment, biological and chemical weapons attacks, detection and search of nuclear attacks, real-time monitoring of the battlefield, battlefield assessment, and target positioning.

- Environmental monitoring: WSN can be deployed in areas with complex topography. It is suitable for air pollution monitoring, forest fire detection, meteorological and geographic research, and flood detection. Sophisticated studies of populations can also be conducted by tracking birds, small animals, and insects.

- Agriculture: WSN can be used to monitor the irrigation of crops, soil and air conditions, and the living environment of poultry. Combined with automation technology, such as automatic irrigation systems, it can significantly improve farm efficiency and increase production.

- Areal monitoring: Heat and humidity monitoring, Area security system
- Medical application: The application of WSN in the healthcare system includes detecting various physical conditions of people, and optimal management in the hospital, including patients, physicians, and medications. With the special sensor nodes, the doctor can monitor the patient's condition and rescue them in time when an abnormality occurs. If sensor nodes are installed in various types of medicine dosing devices, the computer system can help identify the prescribed medicines and reduce the chance of medication errors.
- Industrial automation: Machinery condition-based maintenance, Factory automation system
- Smart Home: WSNs can connect embedded sensor nodes in furniture and home appliances to create a user-friendly smart-home environment.

2.1.3 Features and Challenges

The characteristics of WSNs, such as flexibility, self-organization, and rapid deployment, make them widely used in various scenarios. However, WSN technology still has many challenges in achieving reliable and stable operations.

1) The WSN is a large-scale network. Nodes are spread over a large area and are densely deployed. Therefore, WSN has high accuracy and strong fault tolerance. However, managing and maintaining such an extensive network is a challenge.
2) The WSN is a self-organizing and reconfigurable network. It can perform automatic configuration and management during WSN operation and perform adaptive structure change to deal with the damage and movement of nodes due to environmental factors. Topology control is the core technology to realize a reliable WSN. Topology control can be based on routing and power control or node mobility.
3) The individual nodes in WSN are limited resources in terms of energy, computation, communication, and memory. These limitations must be taken into account in network protocols.
4) WSN should be a secure and confidential network. Its security issues include the clandestineness of task execution, the efficiency of data fusion, and communication security. The scarce resources of the sensor nodes are another essential issue in security protocol planning and implementation.
5) Redundant data is a challenge but also a benefit. The redundancy can compensate for missing measurements and get a more precise understanding of the

observed phenomena. On the other hand, the measured data should be processed as close to the measurement as possible and only send that information needed to the upper levels of the system to avoid unnecessary swapping of data back and forth. Data fusion can filter unnecessary data but also cause delays.

Many researchers have focused on improving communication protocols and algorithms to address these challenges. For example, the network can work for longer by using energy-aware routing on the network layer and energy-saving mode on the MAC layer [17]. For that reason, they are common research topics in WSNs. Routing tables and data replication can provide reasonable use of memory resources [18]. The trade-off between connectivity and lifetime extension in the problem of relay node deployment strategies has been discussed [19]. Many improved protocols, such as the reliable energy-aware routing (REAR) protocol [20], have been developed and evaluated to provide energy efficiency and reliable wireless communication in WSNs.

2.2 Mesh Architecture

Several topologies can be applied, such as star, tree, and mesh types of structures (Fig. 2.3). A base station can transmit and receive a message to various remote sensor nodes in a star topology. It is simple and low-power but has a limited communication range. Tree topology is also called a cascaded star topology. It is easy to expand and maintain the network. However, if the high-level connection breaks, the network will collapse. Mesh topology allows multi-hop communication, extending the radio transmission range and improving the robustness. It is commonly used in the context of the Internet of Things (IoT) solutions [21]. WSN with mesh topology can be integrated with sensor networks through mesh routers and bridging functions.

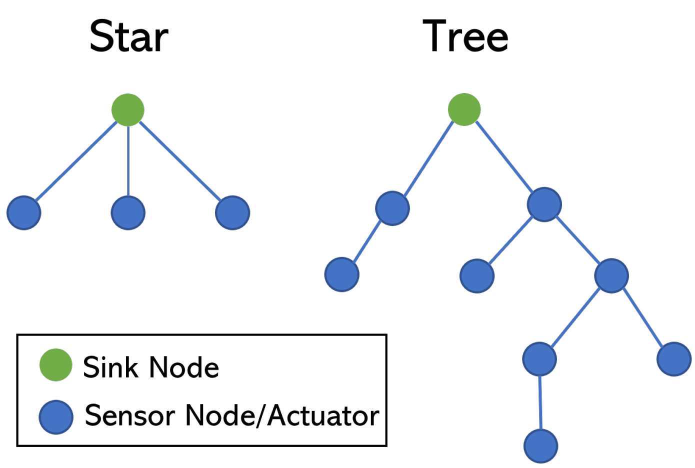

Figure 2.3: Three types of network topologies

Mesh topology has the following characteristics which satisfy the requirements of the application of WSNs under the Internet of Things:

1) Wireless mesh networks (WMNs) have the ability of self-organization and self-configuration. WSN with mesh topology has higher flexibility, easier node deployment and configuration, and higher fault tolerance.

2) In a general networking perspective, the node in the mesh network can become a router or client. Suppose the nodes are out of their direct communication range. They can communicate over multi-hop paths by relaying nodes between the transmitter and receiver node. Routing information between nodes is kept up by routing algorithms, which can be computed in a distributed manner. Wireless mesh architecture can offer higher stability than other WSN topologies. Network self-configuration and self-healing are among the target properties.

3) Wireless mesh network is a multi-hop wireless network. This architecture extends the communication coverage and offers a higher throughput without sacrificing channel capacity.

4) The mesh architecture can reduce WSN deployment costs. Therefore, this kind of network has a low upfront investment and can grow gradually as needed.

Mesh topology offers many advantages. However, there are still issues to be solved, such as communication interference, time-variant communication delays, data traffic, and power management problems. Many algorithms and protocols are continuously improved and developed to address these issues [22]. In general, mesh topology will be one of the most common choices for WSNs from now to the future.

3 Bluetooth Low Energy and Bluetooth Mesh

3.1 Bluetooth Basics

Bluetooth [23] is a wireless technology that supports short-range point-to-point communication. Bluetooth Classic is also termed Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR). EDR is an extended version of BR and provides a faster data rate of 3 Mbps. Bluetooth BR/EDR streams data over 79 channels in the 2.4GHz unlicensed industrial, scientific, and medical (ISM) frequency band. Initially, this technology was used to replace serial data cables between different devices. As the technology matures, large files can also be transferred between PCs and mobile devices using Bluetooth. Nowadays, most phones, tablets, and laptops are equipped with Bluetooth. It is also widely used in wireless headsets, wireless audio, car kits, game consoles, and peripherals such as keyboards and printers. Its typical communication range is less than 10 meters, but it can reach up to 100 meters.

Bluetooth technology supports ad hoc networks, a reasonable property for WSN communication technology. Meanwhile, Bluetooth has many excellent advantages, such as low cost, high security, ease of use, and coexistence with other wireless communication technologies. It consumes less power than WLAN but is still not enough for WSN [24].

3.2 Bluetooth Low Energy

3.2.1 From Bluetooth BR/EDR to Bluetooth LE

The Bluetooth Low Energy (BLE) [25] was created to transmit small data packets while consuming less energy than BR/EDR. It is a significant evolution of Bluetooth technology. BLE focuses on reducing current consumption. The reduced average current ensures that the battery drains slowly. Devices that conform to this standard can operate on a coin cell (or even smaller) battery for months or even years. This is useful in applications that may be difficult to charge frequently but need to extend the battery life of the device. Data communications in BLE are usually in short bursts, suitable for devices with a low requirement of throughput or data streaming [26].

Another advantage of BLE is the faster connection. The connection time of BLE is less than three milliseconds, while that of Bluetooth BR/EDR is up to 20 milliseconds. BLE applies only three dedicated advertising channels to create connections, while Bluetooth BR/EDR uses 32 channels [27]. In addition, BLE can support different communication topologies, including unicast, broadcast, and mesh. BLE is suitable for large-scale applications.

Typical applications of BLE include finding and alerting devices, healthcare, sports and fitness equipment, and mobile payments. This technology also contributes to the Internet of Things (IoT).

3.2.2 BLE Protocol Stack

The BLE protocol stack consists of two parts: host and controller, which are implemented separately. Figure. 3.1 shows the BLE stack architecture.

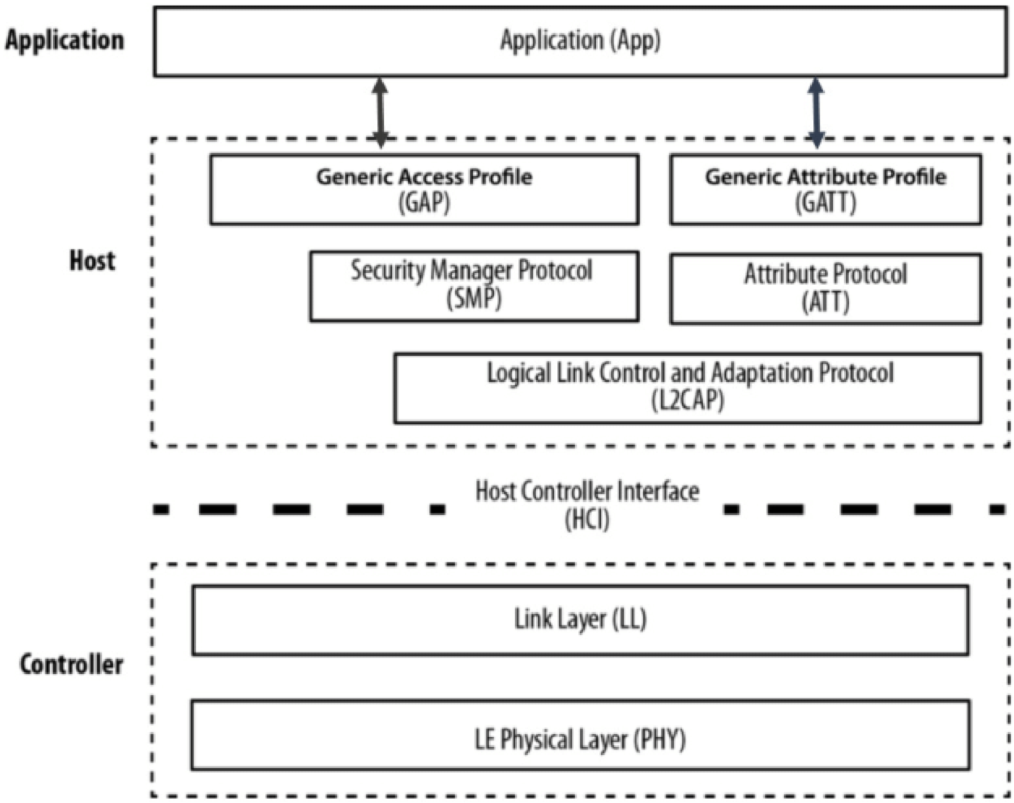

Figure 3.1: BLE Protocol Stack [28]

In the host block in Figure. 3.1,

1. **Generic Access Profile (GAP)**

   The Generic Access Profile (GAP) has two mechanisms: broadcasting and connecting. It determines whether two devices can interact with each other. GAP services contain device and service discovery, association models, connection modes, and security [29].

   In broadcasting, devices are divided into two roles:

   - Broadcaster: Broadcasting public advertising data packets.
   - Observer: Receiving those packets, but no need to connect with the broadcaster.

   In connecting, devices are divided into two roles:

- Peripheral: advertising to central and accepting the connection request from central.
- Central: listening to the advertising packets from peripherals and sending the connection requests.

All the devices can terminate the connections, and central devices can update connection parameters.

2. Generic Attribute Profile (GATT)

The Generic Attribute Profile (GATT) layer handles the data communication between two connected devices. GATT used the Attribute Protocol (ATT) as a transport mechanism. In the Generic Attribute (GATT) profile, the Universally Unique Identifier (UUID) can recognize the profiles, services, and data types. It is a globally unique 128-bit (16-byte) number.

There are two roles in a GATT:

- GATT server: containing the characteristic database.
- GATT client: reading or sending data from or to the GATT server.

The roles in GAP and GATT are independent. Peripheral or central devices can become servers or clients based on the data flow. Once a central-to-peripheral connection has been made over GAP, devices can negotiate a GATT client/server data exchange.

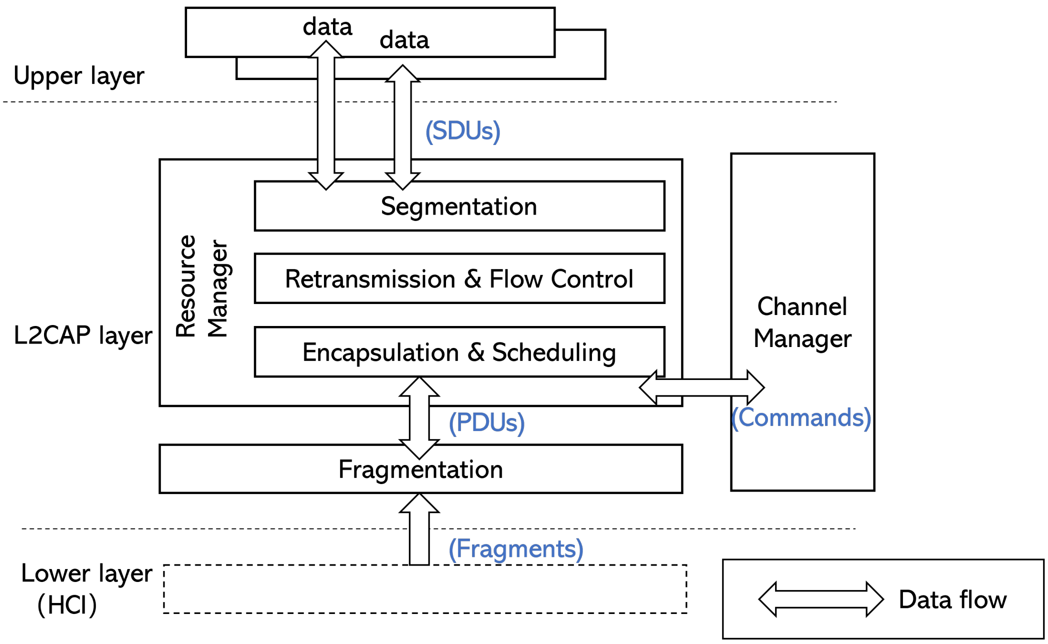

Figure 3.2: Data flow of L2CAP blocks

3. **Logical Link Control and Adaptation Layer Protocol (L2CAP)**

   The role of the L2CAP layer is to transfer data between the lower layers and the upper layers in the host. The functionality includes segmentation, protocol multiplexing capability, and reassembly operation. It uses L2CAP service data units (SDU) when sending and getting upper layer data packets. The BLE stack allows the functions of L2CAP PDUs [30]. As a result, larger packets can be replaced by many small packets at the link layer. Figure 3.2 shows the data flow in the L2CAP blocks between different layers.

4. **Security Manager (SM)**

   The security manager (SM) mainly contains the protocols for pairing and distributing keys. The purpose of pairing is to create keys for link encryption. When the network is reconnected, the keys can rebuild the link, complete the random address resolution, and evaluate the signed data.

In Figure. 3.1, the **Host Controller Interface (HCI)** can transport events and commands between the controller and the host. It can be implemented over different physical transports like UART, SPI, or USB.

In the controller block in Figure. 3.1,

1. **Physical layer (PHY)**

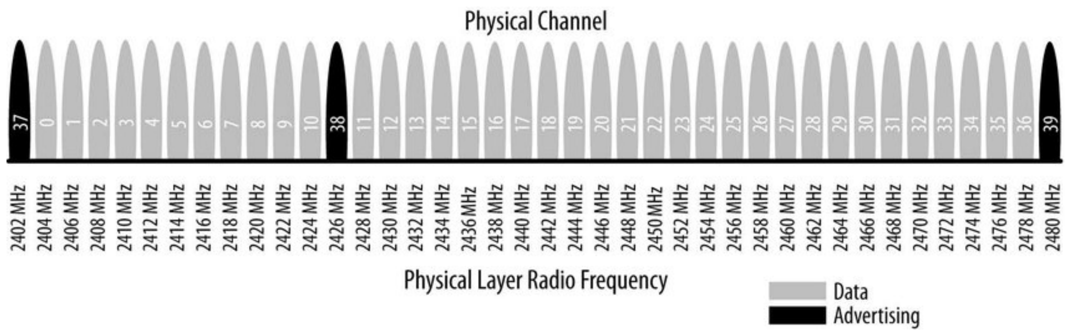

Figure 3.3: BLE frequency channels [31]

   The physical layer (PHY) is the lowest and provides service to the link layer. Its task is to send and receive the packets on the physical channel. The radio uses the 2.4 GHz ISM (industrial, scientific, and medical) band to communicate [32]. The band is divided into 40 channels, shown in Figure. 3.3. Channel 37, 38, and 39 are advertising channels, and channels 0-36 are data channels. Advertising channels are used to discover unconnected devices, build the connection between devices and complete the broadcast transmissions. Data channels provide bidirectional links and robust operation with adaptive frequency hopping. This mechanism can reduce the interference from other devices by transferring packets from the bad channel to the good channel.

2. Link layer (LL)

Diagram 3.4 depicts the process of building a unicast connection between two BLE hosts. The two hosts are initially in the standby state. Then, one becomes an advertiser to send advertising packets. The other host becomes the scanner to receive the packets. After that, the scanner becomes the initiator and sends a CONNECT_REQ advertising packet to the advertiser. The connection is successfully built once the advertiser accepts the request. As a result, the advertiser becomes enslaved while the initiator becomes the master. The hosts, which send packets, are called broadcasters, and the scanners are called observers for the broadcast connection.

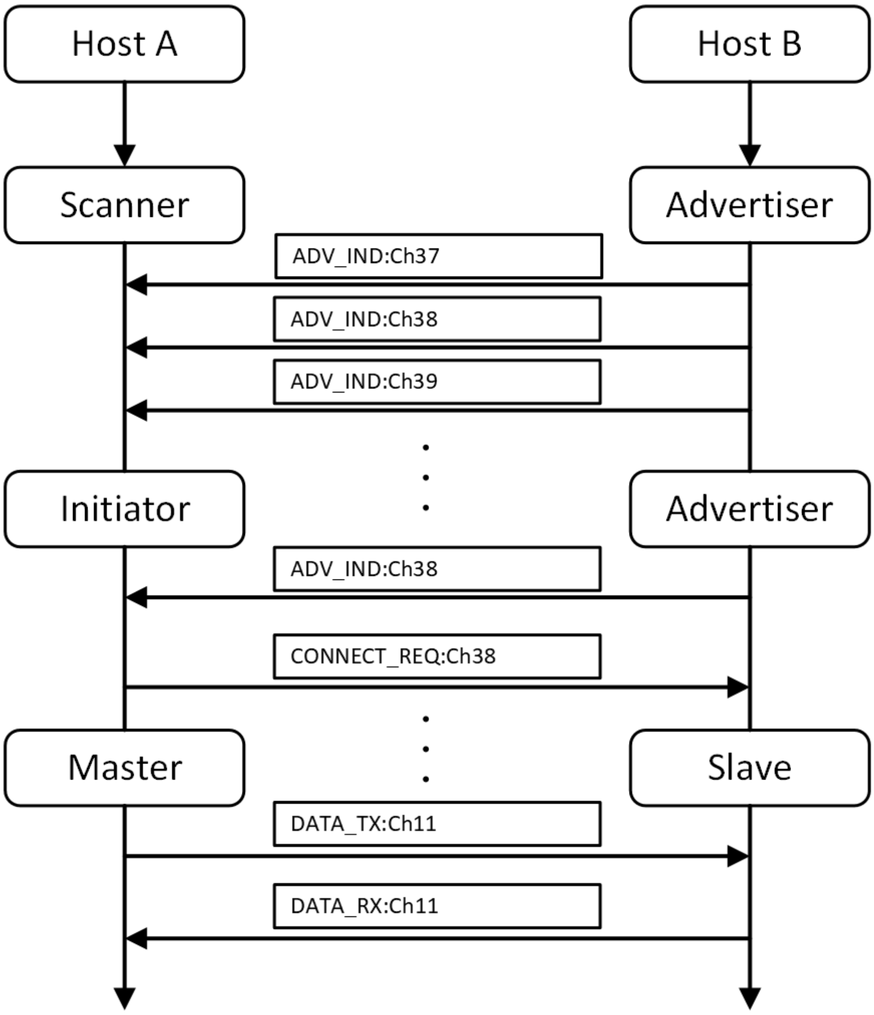

Figure 3.4: Unicast connection in Link layer [33]

Only one packet format is applied in the link layer, as shown in Figure 3.5. Advertising channels and data channels have different protocol data units (PDU). Advertising PDUs mainly support broadcasting, discovering, and connecting hosts. The maximum payload of data PDUs is 246 bytes because of other protocol overhead.

BLE Packet

Preamble	Access Address	Protocol Data Unit(PDU)	CRC
1 Byte	4 Bytes	2-257 Bytes	3 Bytes

Advertising Channel PDU

Header	Payload
2 Bytes	0-37 Bytes

Data Channel PDU

Header	L2 He	Op	Payload	MIC
2 Bytes	4 Bytes	1 Byte	0-246 Bytes	4 Bytes

Figure 3.5: Packet format in BLE link layer

3.3 Bluetooth Mesh

Bluetooth Mesh technology [34] is based on Bluetooth Low Energy. It enables a many-to-many topology, in which every pair of devices can communicate with each other by sending and receiving messages. If the devices are not directly connected over a single radio link, the messages can be routed over a multi-hop path from the transmitter to the receiver to extend the end-to-end communication range. This is why the BLE Mesh network can be applied in large-scale systems. The Bluetooth Mesh profile specification [35] is the official file published by Bluetooth SIG to introduce the fundamental concepts and define the requirements from a developer perspective.

3.3.1 Operations of Bluetooth Mesh

In the Bluetooth Mesh, provisioning is the process of adding a new device to a mesh network [35]. Security is considered a critical factor in the design of the provisioning process. A provisioner will drive the provisioning process. Provisioning has the following five steps (Fig. 3.6):

1. **Beaconing**: An unprovisioned device will advertise packets containing "Mesh Beacon" to inform the provisioner of the existence.

2. **Invitation**: After the provisioner knows the existence of the unprovisioned device, it will send a provisioning invite PDU to the device. The unprovisioned device will then respond to the invitation with provisioning capabilities PDU.

3. **Exchange Public Keys**: The unprovisioned device and the provisioner exchange the public keys in two ways: direct method or Out of Band(OOB) method.

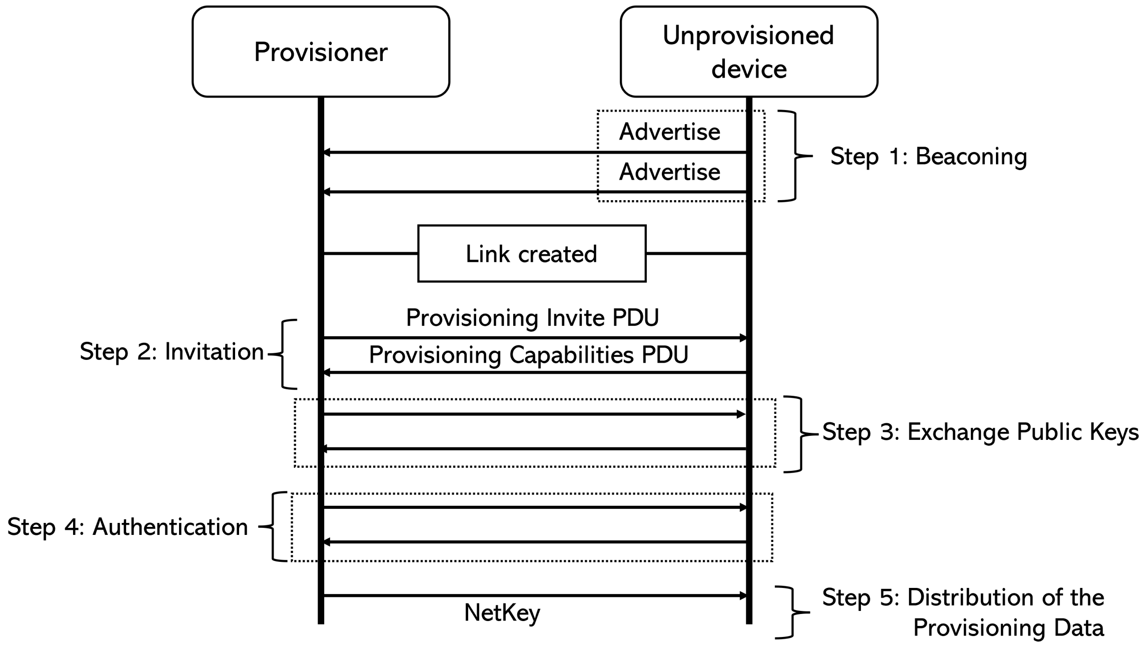

Figure 3.6: Bluetooth mesh provisioning process

4. **Authentication**: The unprovisioned device will output a single, random, or multi-digit number, which is exchanged with the provisioner to complete the authentication. This step ensures the security of the network building.

5. **Distribution of the Provisioning Data**: Session keys are generated after the successful authentication and used to protect the subsequent distribution of the data for the configuration.

After the provisioning, the node joins the mesh network. The nodes possess the Network Key (NetKey) of the network, which the provisioner allocates [35]. All the nodes can send and receive mesh messages. Additionally, nodes have four specific features in BLE Mesh networks [35], including relay, proxy, friend, and low power. Each node can support any combination of features simultaneously. Nodes with relay features can retransmit mesh messages they receive from other nodes, extending the range of the network. Nodes with the proxy feature allow other BLE devices without mesh stack to interact with mesh network nodes using the GATT (Introduced in chapter. 3.2.2). Most Bluetooth devices on the market support BLE, and nodes with proxy feature bring convenience to the construction of mesh networks. Nodes with friend and low power features usually work together. This operation is called friendship [35], a vital optimization mechanism in Bluetooth Mesh. Nodes with friend feature are called Friend nodes, and nodes with low power feature are called low power nodes (LPN). Friend nodes store the messages addressed to LPNs and send them when the LPNs are ready. LPNs transmit and receive data intermittently. The cooperation of these two kinds of nodes can significantly reduce the power consumption of LPNs and avoid missing messages. Figure. 3.7 shows the friendship messaging process.

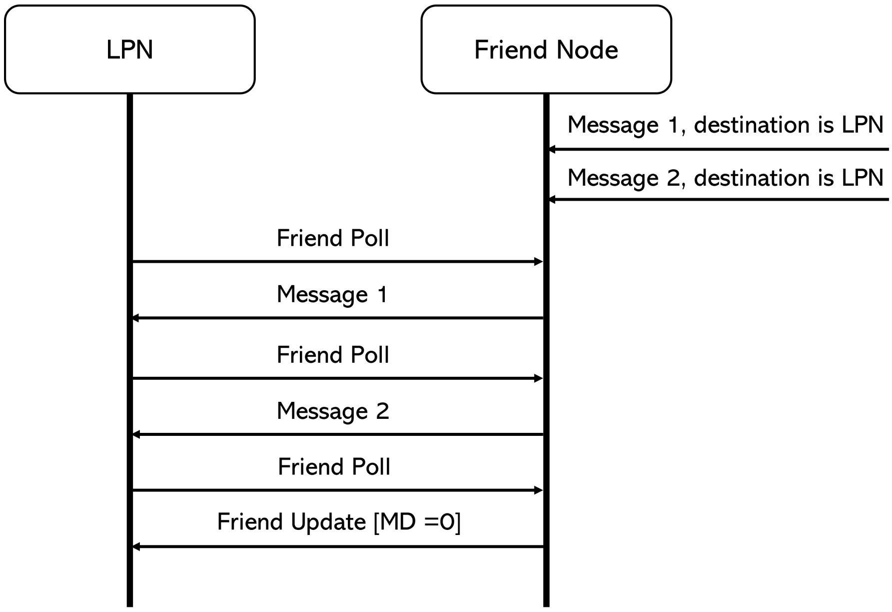

Figure 3.7: BLE Mesh friendship messaging [36]

BLE Mesh node software architecture may contain many parts. These parts are called elements [35]. Elements can be identified by using a system of various address types. Many types of addresses are defined in BLE Mesh. The first type is a unicast address, which is mainly used in the provisioning process. The second type is group address. It is a multicast address to control many elements. The Virtual Address uses a 128-bit UUID value to identify the elements.

All the communication of the BLE Mesh is message-oriented. Messages have different types defined by different layers and have their unique opcode. BLE Mesh follows a publish-subscribe model to transmit and receive the messages in the network. The operation of transmitting messages is called publishing. The operation of receiving messages is referred to as subscribing.

BLE Mesh used managed flooding to deliver messages. This is different from other communication technologies like WiFi, which uses a central node called a router [37]. In the BLE Mesh network, every node can receive messages from other nodes in the radio range. Managed flooding can transmit messages over multi-hop paths in the network, increasing network coverage. BLE Mesh uses the following measures to ensure the efficiency and reliability of managed flooding [35]:

- Heartbeats and Time To Live (TTL): Heartbeat messages are transmitted by devices periodically. They contain information on the network topology and the length of the multi-hop paths. All BLE Mesh packets consist of a field called Time To Live (TTL), limiting the maximum number of hops a message is relayed. Therefore, heartbeats help devices set an optimal TTL and avoid unnecessary delays.

- Message Cache: Message cache is included in the node software. It is used to avoid unnecessary processing by discarding the same messages found in the cache.

- Friendship: Low power nodes and friend nodes work together to improve energy efficiency.

A message usually contains the operation on the states, which are the conditions of the elements. A typical example is the use of three types of messages: GET, SET, and STATUS. The GET messages demand states of the elements. The elements respond to the GET messages by sending STATUS messages containing the information of states. SET messages ask the elements to change their states. Different state operations can implement various functions.

Models define all the states, state operations, and messages. Three models are recognized. The server model and client model work together. The server model defines the states and messages, while the client model defines the GET, SET, and STATUS messages. Another type is the control model. It shares the functionality of the server model and client model. A model may extend another model. The combination of the three types of models can generate different mesh applications. A typical example is the application of smart lighting systems. The light uses the server models, and the switch uses the client models. GET messages can obtain the state of the lights, and SET messages can turn on or off the lights. As a result, the network can control the lights in batches.

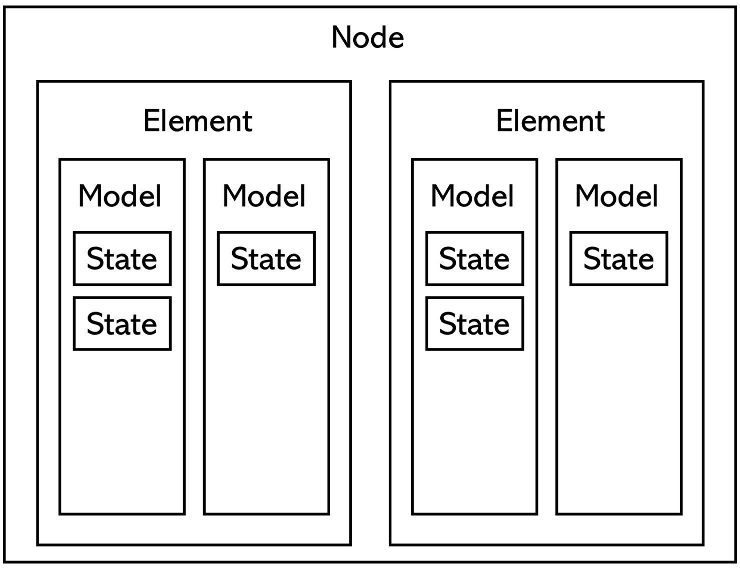

Figure 3.8: BLE Mesh node software composition

Defining the mesh node composition is the priority for a mesh developer. The developer needs to decide on elements used in node software architecture and the models used in each element depending on different applications. Fig. 3.8 illustrates an example of node composition.

3.3.2 Mesh Stack Architecture

As defined in the Mesh Profile specification [35], BLE Mesh sits on top of the BLE stack, which provides wireless communication software. BLE Mesh networks can add all BLE-enabled devices. Figure 3.9 presents the BLE Mesh stack and the functionality of each layer.

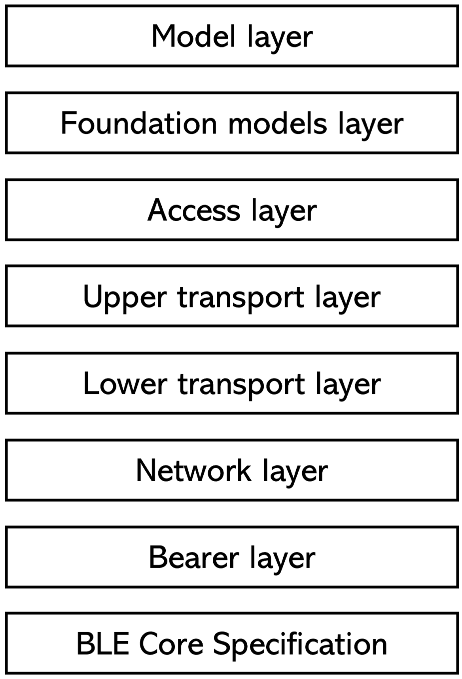

Figure 3.9: BLE Mesh architecture [38]

In Figure 3.9, the bearer layer defines how to broadcast messages between nodes in the BLE Mesh network. Two bearers are used in the layer to meet different requirements: The advertising and the GATT. With the support of the advertising bearer, nodes can directly apply the GAP advertising and scanning features of BLE technology to broadcast and receive messages. GATT bearers are used to help the device without advertising bearers support communicate indirectly with other nodes in the mesh network.

The network layer standardizes various network address types and message formats to ensure communication between the transport and bearer layers. At the same time, some input and output filters are designed to decide which messages need to be forwarded, processed, or rejected. In addition, the network layer also adds encryption and authentication of network messages. This layer can implement the relay and proxy features.

The transport layer includes a lower transport layer and upper transport layer. The main jobs of the Lower Transport Layer are to perform segmentation of long packets from the transport layer and assign them to the network layer and form short packets from the network layer into a suitable transport layer PDU and transport them to the upper layer. The upper transport layer is concerned with the security and confidentiality of messages. It encrypts, deciphers, and authorizes application data for messages from the access layer. In addition, it defines the transport control messages between nodes, such as friendship messages and heartbeats.

The access layer defines how higher-layer applications communicate to the upper layer, including applying the data format, the encryption and decryption of the upper transport layer application data, and verification of the received application data.

The foundation model layer defines the state, messages, model configuration, and mesh network management. The model layer contains the relevant models that define standardized operations for typical user scenarios.

3.3.3 Security

BLE Mesh is designed for industrial applications, which typically have high requirements for security. It has the following fundamentals to ensure the security [35]:

1. The messages are all encrypted and authenticated in the BLE Mesh network.

2. Different keys are used to protect the security of the network, application, and device.

   The network key (NetKey) serves as the ticket to enter the BLE Mesh network, which each node possesses. The node with the NetKey can carry out some network functions. But decryption of application data is not allowed. A network can be separated into several subnets with their own Netkeys. This property can be used to isolate specific areas.

   Nodes can only decrypt application data with the correct application key (AppKey). Nodes with different functions have different AppKeys, but one AppKey only connects with one NetKey.

   The device key (DevKey) is only used for the provisioning process (introduced in chapter. 3.3.1) to secure communications between the provisioner and nodes.

3. After removing a node in the BLE Mesh network, the provisioner will initiate a Key Refresh Procedure process to replace all security keys. Even if someone obtains a node that has joined the BLE Mesh network before, the network cannot be deciphered or attacked through the information in the node.

4. The network uses a privacy mechanism called obfuscation which utilizes Advanced Encryption Standard (AES) to encrypt the source address, sequence numbers, and other header information using a private key. It can increase the difficulty of tracking nodes.

5. The network uses Sequence Number (SEQ) and IV Index to respond to replay attacks.

4 Applied Passive Infrared and Acoustic Sensing

4.1 PIR Sensor and Human Detection

The passive infrared (PIR) sensor is a pyroelectric sensor that can detect infrared radiation. In general, PIR sensors can detect the movement of humans or animals in a specific range. The PIR sensor has two slots. These two slots receive the same amount of infrared radiation when there are no objects in the detection area. When a human or animal gets into the sensing area, the level of IR radiation received in one slot is higher than that in the other. As a result, the PIR sensor outputs a signal.

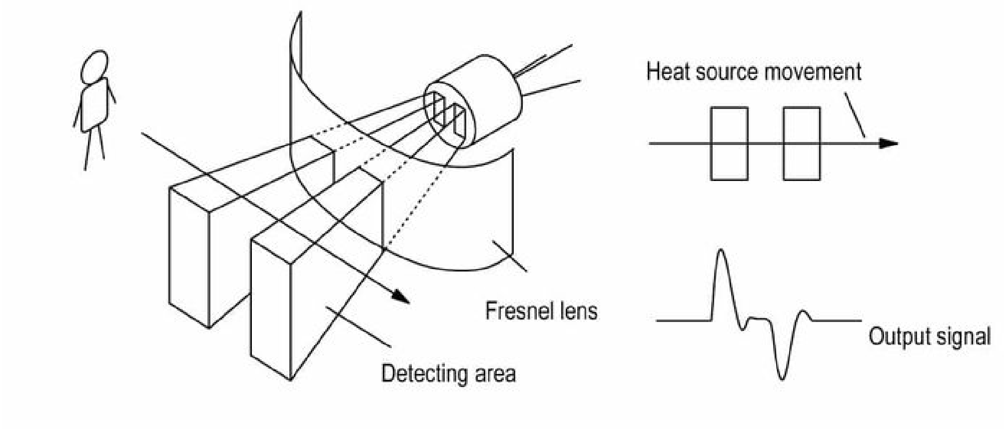

Figure 4.1: Passive infrared sensor principle

PIR sensors have become a popular option in IoT applications because of their small size, low price, and low-power consumption. They are commonly used, e.g., in security alarms [39] and automatic lighting applications [40].

4.2 Acoustic Sensor and Speaker Identification

4.2.1 Speaker Identification Overview

Speaker identification is a technology to identify a person from the characteristics of the voice [41]. People can be authenticated by the similarity comparison of the sensed voice samples and the stored voice samples. In addition to the human voice, the same technology can be used to identify some other interesting voices, such as vehicles and animals.

Mel-Frequency Cepstral Coefficients (MFCC) [42] have been widely applied in speaker identification. Mel scale is based on the auditory perception of the human ear and is more suitable for speaker identification. The linear sound spectrum needs to be mapped to Mel nonlinear spectrum through mel-scale filter banks to obtain the coefficients. Cepstrum analysis is a commonly used method for extracting sound recognition properties. Cepstrum can be obtained by Fourier transform, logarithmic operation, and Fourier inverse transform.

4.2.2 Feature Vector Computation

This thesis uses a lightweight method [43] to characterize the voice features by cepstral analysis. Proceeding of computation is presented in Figure. 4.2.

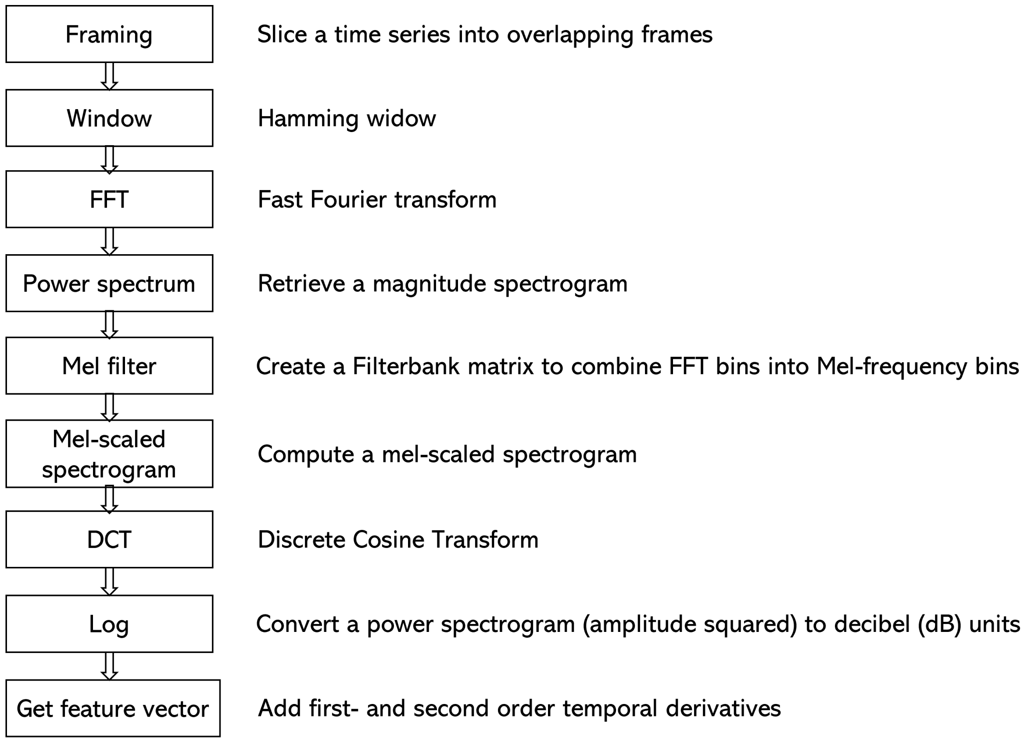

Figure 4.2: MFCC extraction process

A voice signal with N samples is transferred into vector:

$$\mathbf{x} = [x(\mathbf{1}), ..., x(N)] \tag{1}$$

A pre-emphasis process is processed to the signal by a high-pass filter to enhance the high frequencies of the spectrum.

$$x_p(i) = x(i) - \alpha x(i - 1) \qquad i = 2, ..., N \tag{2}$$

$\mathbf{x_p}$ is the enhanced signal vector. α is a pre-defined parameter in the range of [0.95,0.98].

It is meaningless to Fourier transform the entire signal because the frequencies in a signal usually vary with time. Assume that the frequencies in the signal are stationary for a short time. Concatenating adjacent frames by taking a Fourier transform over a short time frame can obtain a good approximation of the frequency profile of the signal. Each frame contains $t_\omega f_s$ points and then is multiplied by a

Hamming window. This step can increase the continuity of the frames both on the left and right sides. The shift between two consecutive windows is 2/3 of the window length. f_s is the sampling frequency, and t_ω is a windowing parameter that the user can define. The total number of windows is:

$$N_\omega = [\frac{N}{\frac{2}{3}t_\omega}] \qquad i = 1, ..., t_\omega \tag{3}$$

Y is the windowed signal matrix, and each column is one window of the signal, L_ω is the signal window length depending on the sample points.

$$\mathbf{Y} = [y(i,j)] \qquad i = 1, ..., L_\omega \qquad j = 1, ..., N_\omega \tag{4}$$

$\tilde{\mathbf{F}}$ is calculated by appling the Discrete Fourier Transform to each column of **Y**:

$$\tilde{\mathbf{F}} = [\mathcal{F}(\mathbf{y}(\mathbf{1}))^{\mathbf{T}}...\mathcal{F}(\mathbf{y}(N_\omega))^{\mathbf{T}}] \tag{5}$$

Each column of $\tilde{\mathbf{F}}$ has N_{bins} elements. $\tilde{\mathbf{F}}$ can be reduced to **F** with the first $\frac{N_{bins}}{2}$ rows because the result is a symmetric spectrum.

The power spectrum of the voice signal is obtained by taking the quadratic residue of each element:

$$\mathbf{P_W} = [|F(i,j)|^2] \qquad i = 1, ..., \frac{N_{bins}}{2} \qquad j = 1, ..., N_\omega \tag{6}$$

Then, the power spectrum is passed through a set of Mel-scale triangular filter bank $\mathbf{B_f}$. The purpose is to smooth the spectrum and remove harmonics to highlight the formants of the original speech. Smoothed spectrum matrix is:

$$\mathbf{P_S} = \mathbf{P_W}\mathbf{B_f} \tag{7}$$

And transform $\mathbf{P_S}$ into decibels $\mathbf{P_{dB}}$:

$$P_{dB}(i,j) = \begin{cases} 20log_{10}(P_s(i,j)) & , P_s(i,j) > 0 \\ 0 & , P_s(i,j) = 0 \end{cases} \tag{8}$$

The mel-cepstral coefficients$\mathbf{C_p}$ are obtained by Discrete Cosine Transform to every vector in $\mathbf{P_{dB}}$:

$$C_p(k,l) = a(k) \sum_{i=1}^{\frac{N_{bins}}{2}} P_{dB}(i,l) cos(\frac{\pi(2i-1)(k-1)}{N_{bins}}) \tag{9}$$

where $1 \leq k \leq N_{cep}, \qquad 1 \leq l \leq N_\omega$

$$a(k) = \begin{cases} \sqrt{\frac{N_{bins}}{2}} & , k = 1 \\ \sqrt{\frac{4}{N_{bins}}} & , 2 \leq k \leq N_{cep} \leq \frac{N_{bins}}{2} \end{cases}$$

The first cepstral coefficient of each window represents the overall average energy of the spectrum, which can be ignored to reduce the computation. The remaining mel-cepstral coefficients are centered by subtracting the average value of each signal window. The centered mel-cepstral matrix is:

$$\mathbf{C} = \begin{bmatrix} C_p(2,1) - \mu_1 & \cdots & C_p(2,N_\omega) - \mu_{N_\omega} \\ \vdots & \ddots & \vdots \\ C_p(N_{cep},1) - \mu_1 & \cdots & C_p(N_{cep},N_\omega) - \mu_{N_\omega} \end{bmatrix} \tag{10}$$

To de-emphasized the highest and lowest coefficients, a vector $\mathbf{M}$ is multiplied by each column of $\mathbf{C}$. The result is $\mathbf{C_s}$.

$$M(i) = 1 + \frac{N_{cep} - 1}{2} sin(\frac{\pi i}{N_{cep} - 1}) \tag{11}$$

where $i = 1, \cdots, N_{cep} - 1$.

A further processing is to separate the mel-cepstral vectors from the noise. The criteria matrix $\mathbf{C_N}$. Each column of $\mathbf{C_N}$ is the normalized average of that of $\mathbf{C_s}$. The processed matrix $\mathbf{C_{sp}}$ is:

$$\mathbf{C_{sp}} = \mathbf{C_s}(j) | C_N \geq \mu(\mathbf{C_N}) \qquad j = 1, \cdots, N_\omega \tag{12}$$

The final mel-cepstral coefficients $\mathbf{C_{cep}}$ are computed by taking the row-vise average of $\mathbf{C_{sp}}$:

$$\mathbf{C_{cep}} = \begin{bmatrix} \mu\{C_{sp}(1,1) & \cdots & C_{sp}(1,n)\} \\ \vdots & & \\ \mu\{C_{sp}(N_{cep}-1,1) & \cdots & C_{sp}(N_{cep}-1,n)\} \end{bmatrix} \tag{13}$$

n is the number of columns chosen from $\mathbf{C_s}$ to $\mathbf{C_{sp}}$ following (12).

The cepstral coefficients' first and second order derivatives are also computed to characterize the dynamic speech features.

$$\Delta C_s(i,j) = \frac{\sum_{k=-\Theta}^{\Theta} k C_s(i, j+k)}{\sum_{k=-\Theta}^{\Theta} k^2} \tag{14}$$

where $1 + \Theta \leq j + k \leq N_\omega - \Theta$ and $1 \leq i \leq N_{cep} - 1$.

$$\Delta\Delta C_s(i,j) = \frac{\sum_{k=-\Theta}^{\Theta} k \Delta C_s(i, j+k)}{\sum_{k=-\Theta}^{\Theta} k^2} \tag{15}$$

where $1 + 3\Theta \leq j + k \leq N_\omega - \Theta$ and $1 \leq i \leq N_{cep} - 1$.

The final feature vector combines the mel-cepstral coefficients and their first and second order temporal derivatives:

$$\mathbf{F_s} = \begin{bmatrix} {C_{cep}}^T & \Delta {C_{cep}}^T & \Delta\Delta {C_{cep}}^T \end{bmatrix} \tag{16}$$

The speaker identification is accomplished by comparing the similarity between the feature vector computed based on the recorded voice sample and that from the voice sample in the database.

5 Developed Architecture

5.1 Network Architecture

This thesis built a monitoring system with deployable nodes by BLE Mesh technology. As presented in Figure. 5.1, nodes are deployed in designated locations. These nodes form a BLE Mesh network and can communicate with each other. When the node receives the operation command, it will continuously detect the presence of people in the network coverage area by PIR sensing. Once the presence of people is detected, the acoustic sensor in the node will record the voice of the person. The node will then compute the feature vector through cepstral analysis as presented in Chapter 4, and store it. A drone will stop from time to time to collect the computed feature vectors over a short-distance link.

A mesh network consisting of five nodes was used in the experimental deployment of this thesis work. The network performance is evaluated by using a set of test scenarios.

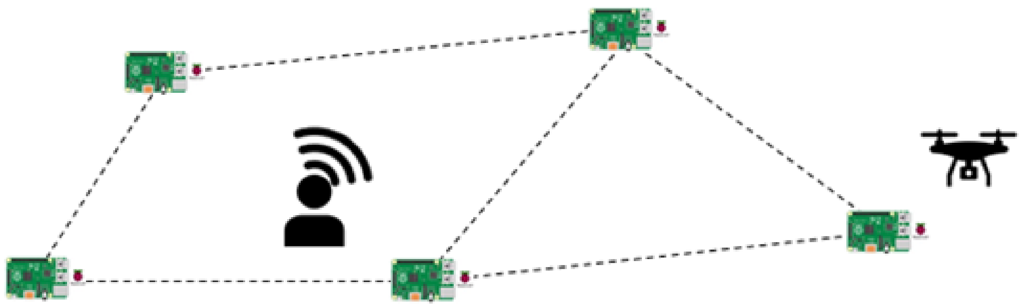

Figure 5.1: Deployable BLE Mesh monitoring system

5.1.1 Unicast Architecture

Unicast is the most straightforward architecture of a BLE Mesh network. This architecture aims to test the length of the reliable communication range between two nodes and test the communication reliability and latency in the unicast situation.

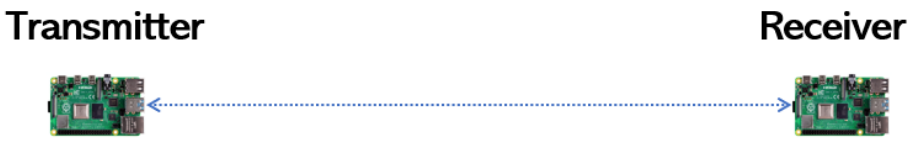

Figure 5.2: Unicast architecture

5.1.2 Clustered Architecture

Clusters can be used in general. In Figure. 5.3, the nodes in the network have three roles: provisioner, general node, and controller. The provisioner is responsible for adding devices to the mesh network. In this setup, a smartphone with the nRF MESH app served as a provisioner for convenience. A node itself can also become a provisioner. The controller can send commands to general nodes to start or end operations and has access to the states of each general node, including its operating mode, whether it detects people, and whether it has feature vectors to send. The general nodes in the same cluster can receive the messages from the controller simultaneously. This can improve efficiency and flexibility. The design of clustered architecture can personalize and extend the monitoring area. Communication reliability and latency in the case of the clustered architecture are investigated in this thesis.

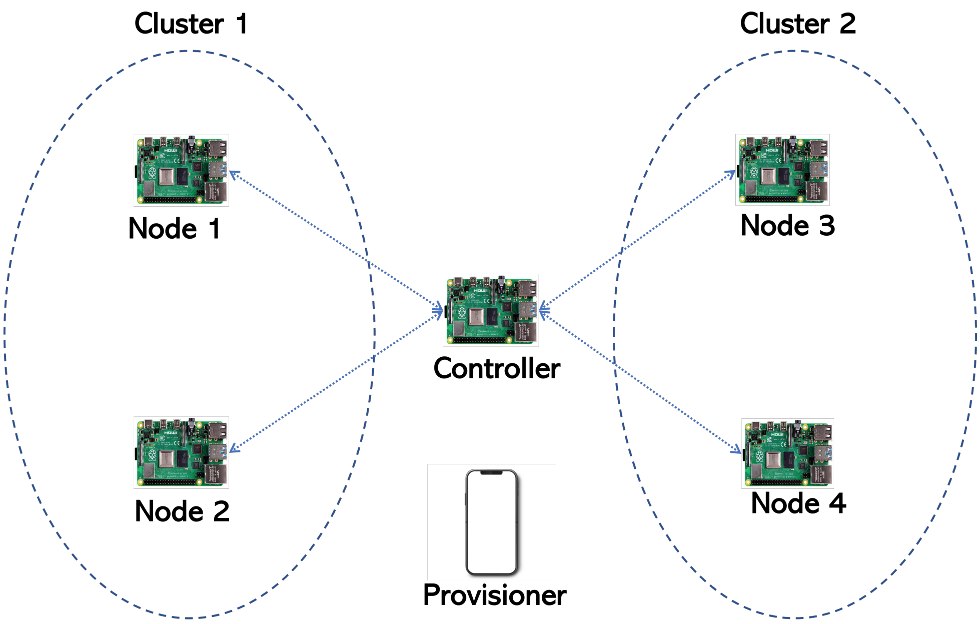

Figure 5.3: Clustered architecture

5.1.3 Multi-hop Architecture

If the transmitter and receiver do not have a direct radio link, the messages can be transmitted over multi-hop paths. These paths consist of several nodes and radio links between them. Figure. 5.4 shows an example of the multi-hop architecture. This architecture can implement multi-hop communication. Time To Live (TTL) can be set in messages to limit the maximum number of hops.

Adding relaying nodes in a cluster can extend the communication range. Using different clusters can improve the flexibility of the network and save power consump-

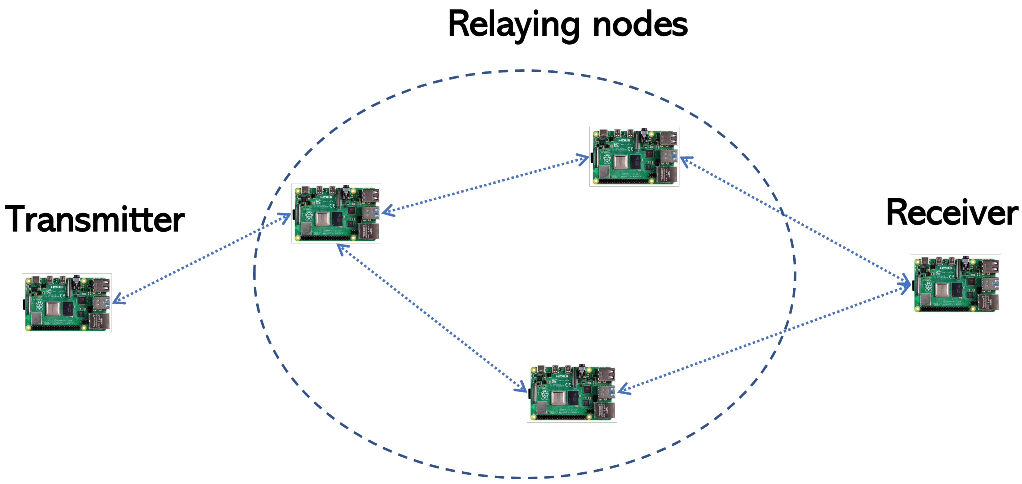

Figure 5.4: Multi-hop architecture

tion by stopping unnecessary cluster work. In practice, the deployable BLE Mesh network can be a combination of those mentioned above three basic architectures.

5.2 Hardware Architecture

5.2.1 Node Components

In the developed BLE Mesh network, each node consists of Raspberry Pi 4 Model B, nRF52840 radio module, acoustic sensor, PIR sensor, and a battery.

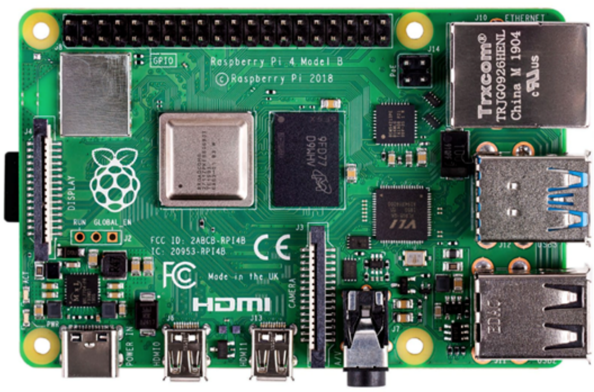

Figure 5.5: Raspberry Pi 4 Model B

Raspberry Pi 4 Model B (Fig. 5.5) is used as the processing unit and platform of the node. Raspberry Pi contains a cheap, robust, and low-power CPU, program memory, and many interfaces, including two USB 2.0 ports, a 3.5mm audio jack, and 26 GPIO (general purpose input and output) pins supporting SPI (Serial Peripheral Interface Bus) and I2C (Inter-Integrated Circuit). Raspberry Pi has excellent flexibility for different applications, a powerful data processing capability, and large memory. Furthermore, it uses Raspbian as an operating system, which is similar to Linux, thus significantly reducing the difficulty of programming and bringing the advantages of a PC. It is a good option as a sensor platform and has been used successfully in many WSN applications [44].

The nRF52840 DK (Fig. 5.6) is used as a radio module in the BLE Mesh network. The product specification [45] defines the features and functions. Figure. 1 shows the crucial properties of Bluetooth in nRF52840.

Table 1: Properties for Bluetooth in nRF52840

-95 dBm sensitivity in 1 Mbps Bluetooth Low Energy mode
-103 dBm sensitivity in 125 kbps Bluetooth Low Energy mode (long-range)
-20 to +8 dBm TX power, configurable in 4 dB steps
Supported data rates: 2 Mbps, 1 Mbps, 500 kbps, and 125 kbps
4.8 mA peak current in TX (0 dBm)
4.6 mA peak current in RX
Capability to measure RSSI (1 dB resolution)

Figure 5.6: nRF52840 Development Kit

Power and clock management (PMU) in nRF52840 can efficiently allocate power in supply regulators and clock sources by switching suitable operation modes. The rich peripheral interfaces provide the possibility to transmit information to other devices or sensors in different ways. These properties make nRF52840 a good choice for being used as a radio module in the BLE Mesh. Nordic Semiconductor creates nRF5 SDK (Software development kit) and nRF5 SDK for Mesh for the developers to implement BLE Mesh applications on the nRF5 Series boards.

As mentioned in Chapter 4, there is an acoustic sensor and PIR sensor in each node. The PIR sensor is connected to Raspberry Pi by GPIO pins. Figure. 5.7 shows the HC-SR501 PIR Motion Sensor Module. It is widely used in battery-powered products. This sensor will output a signal when people move in the monitoring area.

Its sensing range is less than 120 degrees and within 7 meters in the specification. The working temperature is – 15 to +70 °C. Three PIR sensors are used in one node to cover a circular monitoring area with a diameter of seven meters (Fig. 5.8).

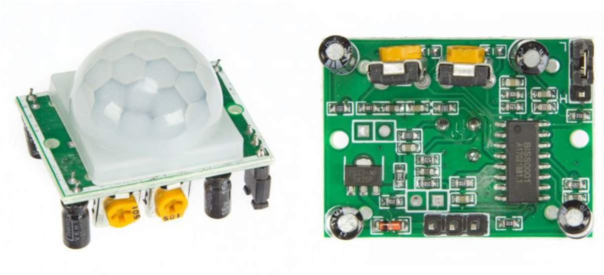

Figure 5.7: HC-SR501 PIR Motion Sensor Module

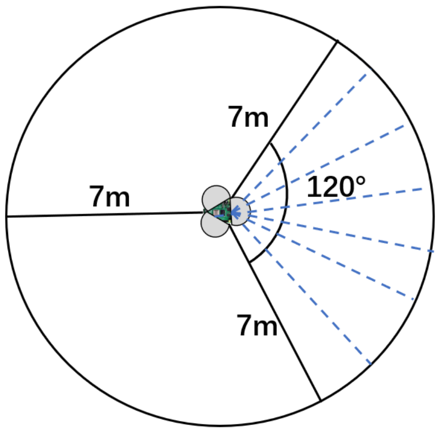

Figure 5.8: Sensing range of HC-SR501 PIR Sensor

The acoustic sensor is connected to Raspberry Pi by a USB interface. The type of the sensor is Maono AU-410 USB Lavalier Microphone (Fig. 5.9), which has a frequency response from 30 Hz to 18 kHz, and a sampling rate of 48 kHz and 44.1 kHz [46].

Powerbank is a typical portable power device suitable for node power supply. Raspberry Pi needs a minimum of 3 A of current supply. PowerBank MP15MA (Fig. 5.10) with a capacity of 10000mAh is used as the battery of the node.

5.3 Software Architecture

The developed system has two software parts. One part is to establish the BLE Mesh communication implemented on the nRF52840 DK. The other part is to control the sensors and process the data. It is implemented on the Raspberry Pi. Figure. 5.11 shows a schematic diagram of the software architecture.

Figure 5.9: Maono AU-410 USB Lavalier Microphone

Figure 5.10: PowerBank MP15MA

Software development kit for the nRF52 Series and nRF51 Series SoCs (nRF5 SDK) and Software development kit for BLE Mesh (nRF5 SDK for Mesh) are used in this thesis. New models and applications have been modified and developed based on these two kits. As presented in chapter. 5.1, the nodes in the BLE Mesh network can operate in different roles, such as a transmitter, receiver, and controller. They all share the same mesh stack code but use other models based on their roles.

The Raspberry Pi uses Python as the primary coding language because it is powerful, versatile, and easy to use. Python is preinstalled on Raspbian, which is the operating system of Raspberry Pi. Python libraries are used to implement the control of sensors and data processing.

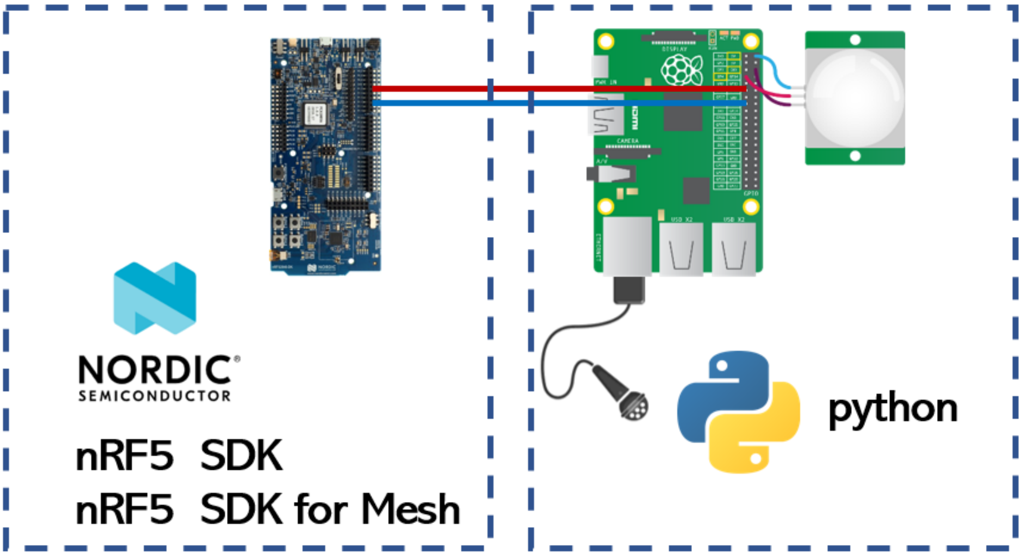

Figure 5.11: Software Architecture

5.3.1 BLE Mesh Communication

Figure. 5.12 presents the software architecture of the BLE Mesh module used in this thesis. The nRF5 SDK for Mesh contains the source code and unit tests for the BLE Mesh stack [47]. It supports most of the features of BLE Mesh, including provisioner role, node role, relay feature, proxy feature, low power feature, friend feature, and foundation models. The coding in mesh access organizes the models and communication. The core module contains the transport and network layer. It provides mesh-specific transportation for the messages. The coding of the provisioning protocol provides an easy approach to adding devices to a mesh network.

In this thesis, the node model is based on the Generic On-Off models in nRF5 SDK for Mesh. This model enables the remote control of Boolean states on a mesh device. It contains two kinds of types. One is called the server model, and the other is the client model. The server model includes three states (Fig. 5.13). State 1 is controllable. It indicates the operating status of the node. The client node can remotely control the server node. States 2 and 3 are read-only; they indicate whether the node detects human intrusion in the monitoring area and whether feature vectors are waiting to be transmitted in the node.

The software model of the environment is built based on SEGGER Embedded Studio, an all-in-one solution for managing, building, testing, and deploying embedded applications [48]. It is an excellent toolchain to develop the BLE Mesh network on nRF5 series boards. Other toolchains can be found in the Nordic Semiconductor document [49].

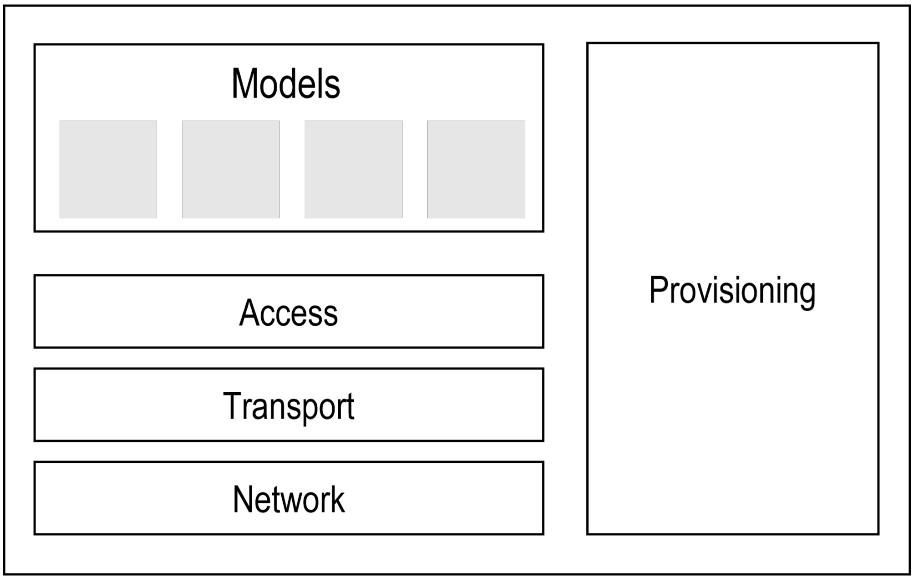

Figure 5.12: Software architecture of mesh module

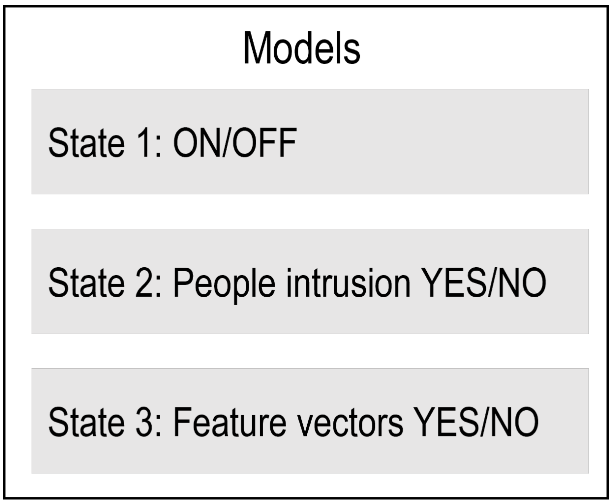

Figure 5.13: States in the model

5.3.2 Monitoring Function

In the developed software architecture, every node except the controller can have five modes:

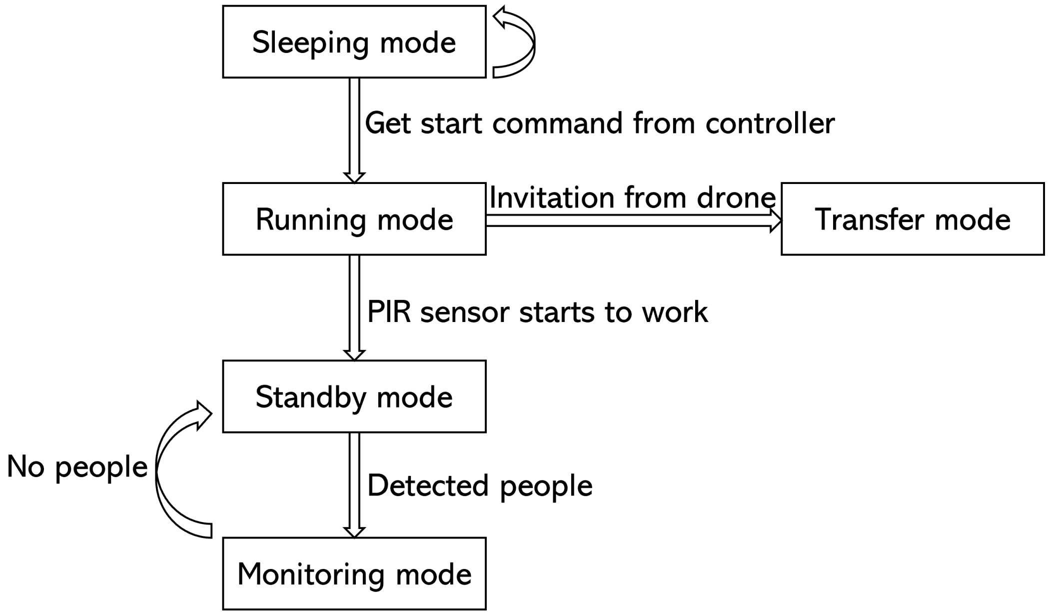

Figure 5.14: Modes of the node

1. Sleeping mode: In this mode, the nodes are waiting for the commands from the controllers and consume little energy.
2. Running mode: After receiving the command from the controller, the node will switch from sleeping mode to running mode. Sensors are activated.
3. Standby mode: In this mode, the PIR sensor works continuously to detect the presence of people in the monitoring area.
4. Monitoring mode: When the PIR sensor detects the people in the monitoring area, the acoustic sensor in the node starts to record the voice of people, immediately compute a feature vector from the voice sample, and store the vector in the Raspberry Pi. This mode will cycle until the PIR sensor detects no people in the monitoring area or the node is accidentally turned off. The computation of feature vectors is presented in Chapter 4.2.
5. Transfer mode: In this mode, the node will send the feature vectors stored to the drone after certifying the invitation from the drone.

Multi-mode operation reduces the node energy consumption and increases the operating time. The first four modes are mainly used in the developed application. The data transmission between the nodes and the drone is temporarily independent. Through the Bluetooth Mesh network, each node can be remotely controlled.

6 Experiments

6.1 Performance Evaluation Metrics

The following metrics are used in the experiments to evaluate the performance of the system:

1. **Packet Delivery Rate (PDR)**

   The packet delivery rate is the ratio between received and transmitted packets between the sender and the receiver.

   $$Packet\ Delivery\ Rate\ (PDR) = \frac{P_{received}}{P_{transmitted}} \tag{17}$$

   $P_{received}$ means the number of packets the receiver got. $P_{transmitted}$ means the number of packets the transmitter sent.

2. **Round Trip Time (RTT)**

   The round trip time is calculated in the transmitter by the time difference between the time when the transmitter starts to send the messages and the time when the transmitter gets the acknowledgment from the receiver.

   $$t_{RTT} = t_{ACK} - t_{send} \tag{18}$$

   t_{ACK} means the time when the transmitter gets the acknowledgment from the receiver. t_{send} implies the time when the transmitter starts to send the message.

6.2 Experiment Setup

6.2.1 Outdoor Setup

The experimental site is on a 60 meters long lawn in the middle of the car park in Aalto campus. Indoor and outdoor radio environments are different. Since the BLE Mesh network is targeted to be deployed outdoors, the experimental results in such a setup are more realistic and practical. Figure. 6.1 shows the experiment environment.

6.2.2 Packet Delivery Rate

Different BLE Mesh architectures (mentioned in Chapter. 5.1) are considered in the experiments. At first, unicast architecture tested the packet delivery rate between two nodes at different distances. The two nodes were set at Direct Test Mode (DTM). One node acted as a transmitter, and the other one as a receiver. The following experiments could choose a suitable distance for different architectures based on the results.

Then, nodes were connected to the BLE mesh network. Single-hop and multi-hop architectures were used in the experiments. Both of them had one transmitter and

Figure 6.1: Experiment environment

one receiver with unicast communication. There were some relaying nodes between the transmitter and the receiver in multi-hop cases. We considered two-hop and three-hop scenarios. Each node was equidistant from its neighboring nodes. The message contained the Time to Live (TTL) information, and its default value is set to 4. Whenever the message was forwarded through a relaying node, the TTL would reduced by one. The TTL could be used to determine the number of hops. The package delivery rate was calculated by the packets number ratio recorded on the transmitter and the receiver.

The cluster architecture (Fig. 6.2) had one controller (functioned as the transmitter) and one cluster with two nodes (served as the receivers). The two nodes in the cluster had the same distance from the controller. The packet delivery rate is the ratio between the sum of packets received by the two nodes and the packets sent by the controller.

6.2.3 Round Trip Time

The RTT testing applied the same architectures used in the PDR testing. However, the calculated round trip time will equal the pre-set timeout when the messages fail to send. Only RTTs in the distance with 100% PDR were presented and analyzed. In single-hop and multi-hop cases, the transmitter would record the timestamp when sending the packets and receiving the acknowledgment and calculate the time difference. In the cluster case, the controller would record the timestamp when sending the packets and receive the last acknowledgment from the nodes in the cluster, and calculate the time difference.

The following table shows the default setting of critical parameters in the PDR and RTT experiments (unless stated otherwise).

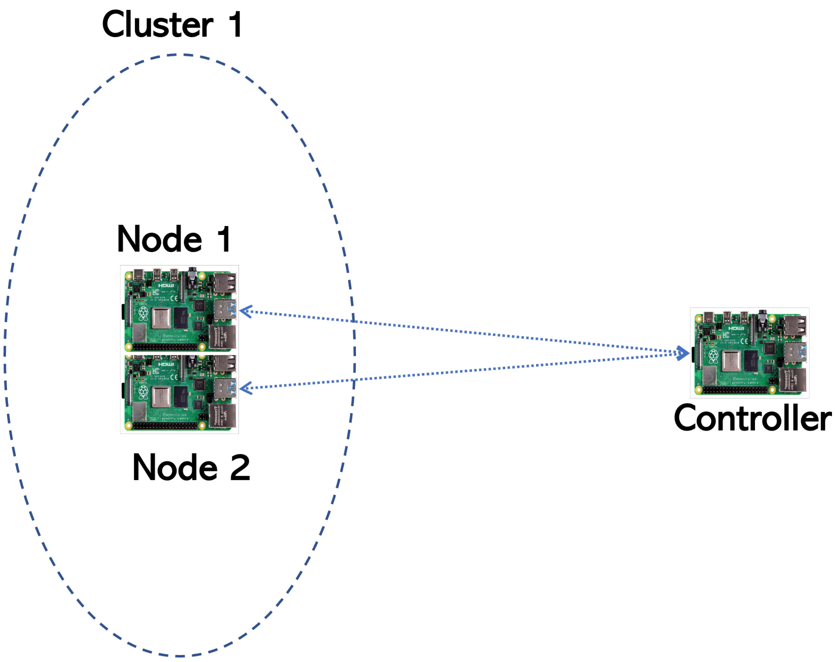

Figure 6.2: Cluster architecture in the experiments

Table 2: Default parameters and configuration

Parameter	Value
Transmission power	0 dBm
Transmission interval	20 ms
BLE radio throughput	1 Mbps
Packet size	58 Bytes
ACK packet size	57 Bytes
Time to Live (TTL)	4

6.3 Result Analysis and Discussion

6.3.1 Packet Delivery Rate

Communication reliability between two nodes was evaluated in this experiment. First, two nRF52840 boards set at Direct Test Mode (DTM) applied the unicast architecture to test the packet delivery rate. The transmitter sent 16,000 packets to the receiver in 10 seconds. The size of one packet was 8 Bytes, and the structure was 11110000.

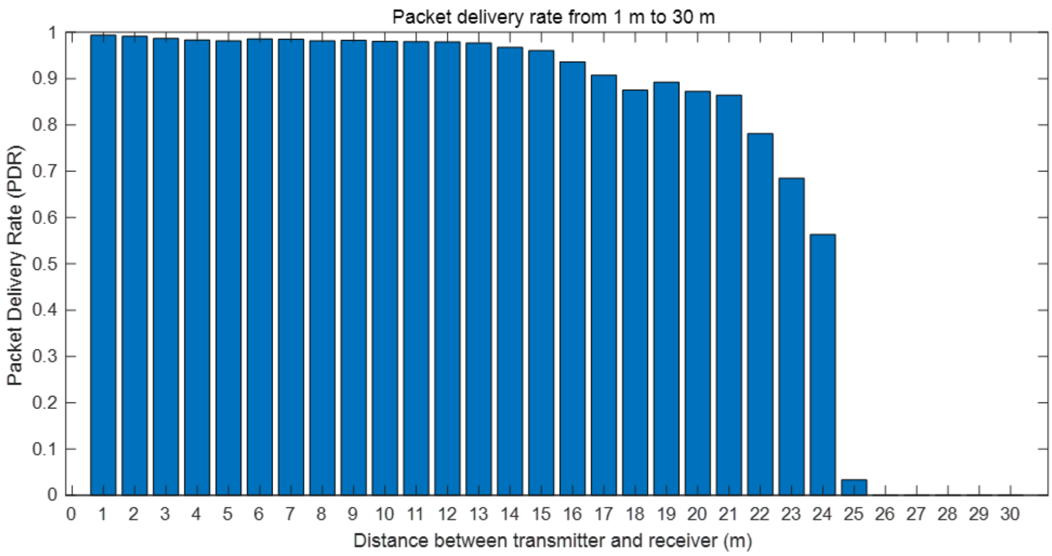

Figure 6.3: Packet delivery rate from 1 m to 30 m

The result is presented in Figure. 6.3. Up to 15 meters, the packet delivery rate remained over 96%. From 15 to 21 meters, PDR had a fluctuation but stabilized between 10% - 20%. From 22 meters, the PDR started to be less than 50%, and the connection became unreliable. At 25 meters, only 3.3% of packets were received by the receiver. From 29 meters, there were no received packets. There was a sudden decrease of PDR from about 50% to 0% in a small distance increase from 24 to 26 meters. Based on these results, 20 meters was used as a maximum reliable communication distance for a single radio link in the rest of the experiments.

Then, we still used unicast communication but considered both single-hop and multi-hop scenarios. Two nodes acted as transmitter and receiver, respectively, and the rest of the nodes were relaying nodes within the communication path, making the maximum number of hops 3. All the nodes were connected in a BLE Mesh network. The message in this experiment was acknowledged, and the transmitter would keep sending it to the receiver until it gets the acknowledgment or the sending time is over 5 seconds. The transmitter sent 100 mesh packets and recorded the acknowledgment numbers. According to the first experiment, the reliable communication distance between two radio modules was about 20 meters. Therefore, in the unicast case, the packet delivery rate was measured from 1 meter to 20 meters using one-meter increments. In the multi-hop case, a longer distance between transmitter and receiver was applied, but the maximum length of the single radio links was kept at 20 meters foremost.

Figures. 6.4, 6.5 and 6.6 present the PDR in single-hop, 2-hop, and 3-hop cases.

In the unicast case, the PDR was 100% within 13 meters. Starting from 14 meters, the communication reliability started to deteriorate and dropped to about 80%. At 16 and 17 meters, the PDR decreased to 40.9% and 50.3%, respectively but returned to about 80% at 19 meters. Some interference may have affected the result when the distance was 16 and 17 meters.

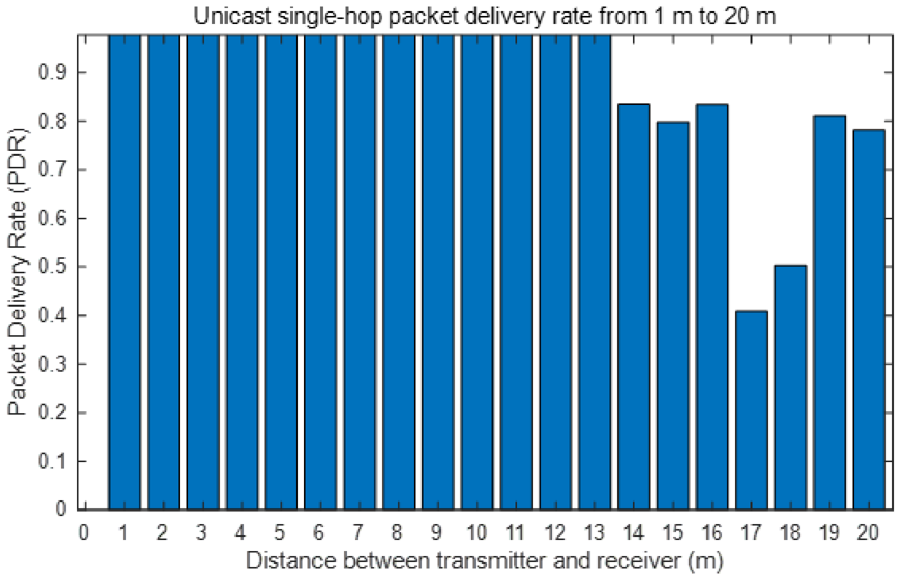

Figure 6.4: Unicast single-hop packet delivery rate from 1 m to 20 m

In the 2-hop case, the PDR remained 100% up to 16 meters and dropped about 15% from 20 to 32 meters. The connection between transmitter and receiver was lost when the distance was over 36 meters. Adding a relaying node between two nodes can extend the communication distance between nodes, but 100% PDR communication distance was only extended 3 meters compared to a single-hop case. In the 3-hop case, PDR remained 100% up to 24 meters and remained over 85% when the distance was 42 meters.

In a cluster case, each cluster had two receiver nodes sharing the same cluster address. The transmitter sent messages to one cluster once. The transmitter would keep sending until it gets two acknowledgments or the sending time was over 5 seconds. From Fig. 6.7, PDR remained 100% until 13 meters. This is the same as the result in the single-hop case. The PDR was irregularly distributed from 14 to 20 meters, and the worst PDR was only 52.3% in 20 meters.

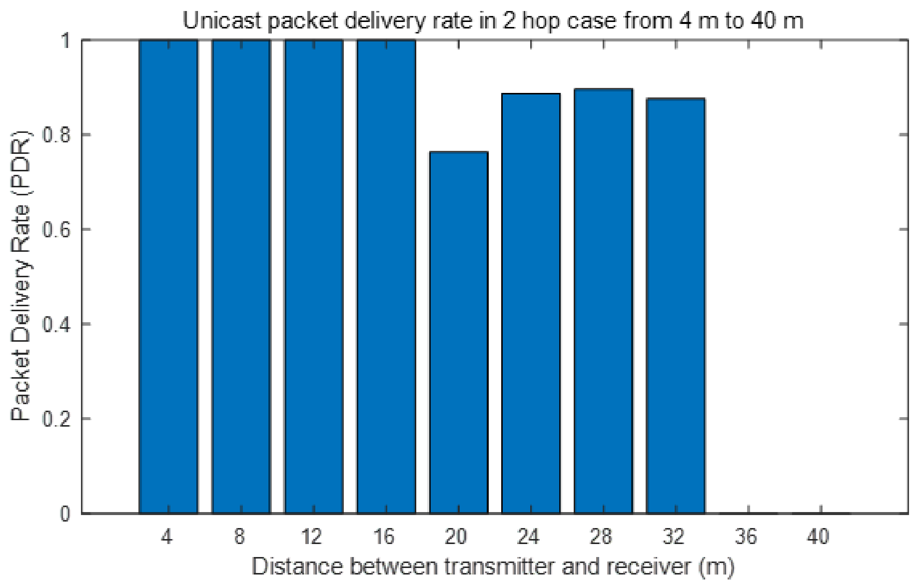

Figure 6.5: Unicast packet delivery rate in 2 hop case from 4 m to 40 m

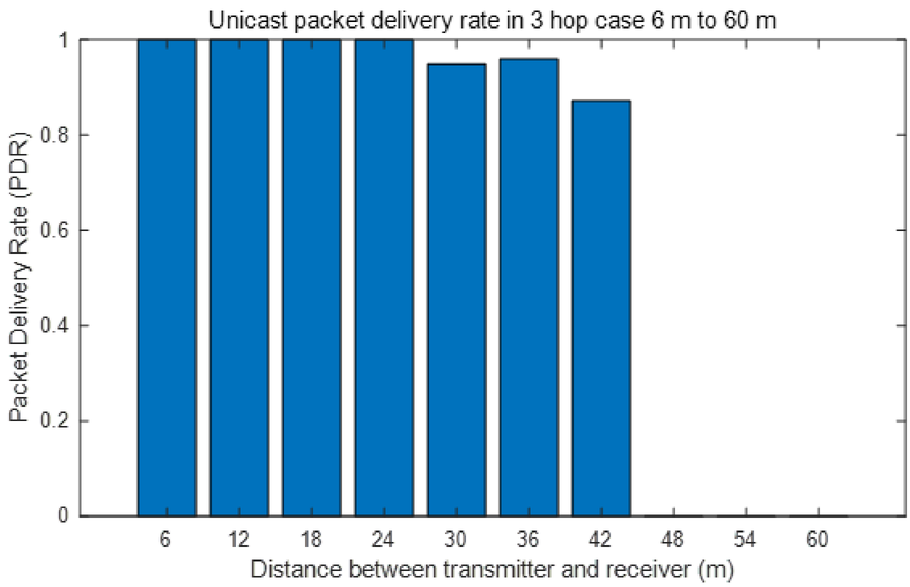

Figure 6.6: Unicast packet delivery rate in 3 hop case 6 m to 60 m

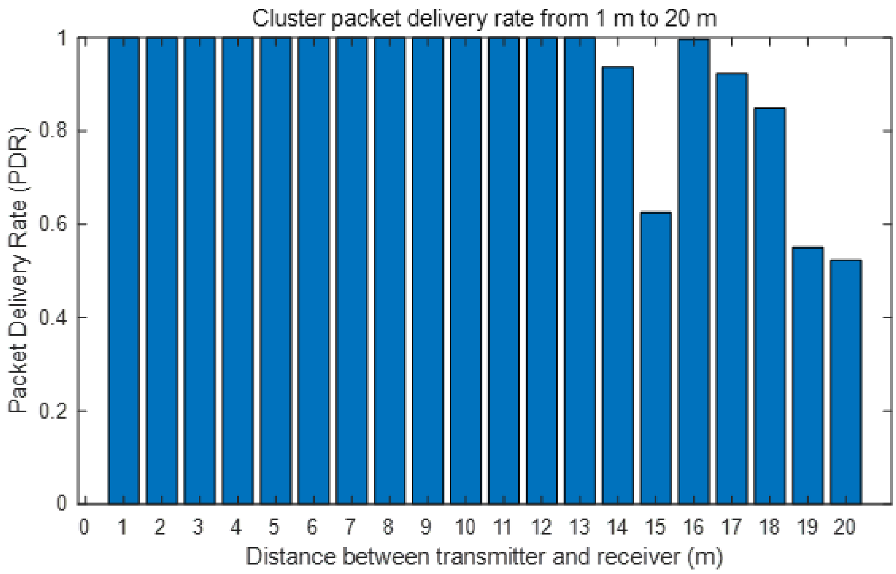

Figure 6.7: Cluster packet delivery rate from 1 m to 20 m

6.3.2 Round Trip Time

Communication latency is evaluated by measuring the round trip time (RTT). Figures. 6.8, 6.9 and 6.10 show the average RTT and standard deviation in unicast communication with different numbers of hops. The dot in the middle is the average RTT at each distance, and the error bars present the standard deviation around the average. In a single-hop case, the average RTT was about 55 ms within 10 meters, and the standard deviation was slight, less than 5. Then some huge RTTs occurred, reaching even 183.51 ms at 13 m. The average RTT was about 78 ms in 2-hop cases and 105 ms in 3-hop cases. The RTTs were relatively stable, with similar standard deviations at different distances. The average RTT increased about 25 ms when the number of hops increased by 1.

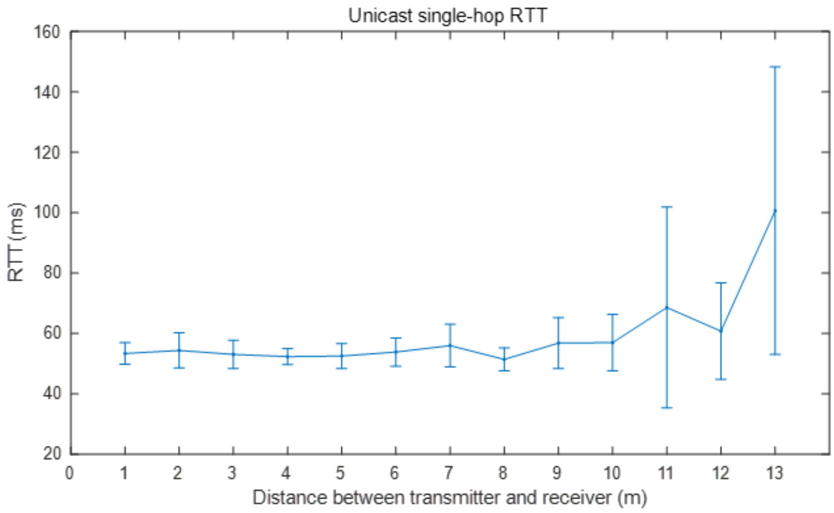

Figure 6.8: Average RTT and standard deviation in unicast single-hop case

In the cluster RTT testing experiment, the RTT was calculated by the time difference between the transmitter starting to send the message and the transmitter getting the last acknowledgment from the nodes in the cluster. When the transmitter and receivers were close, the difference between the average RTTs in the cluster and unicast cases was only a few milliseconds (3 ms). However, the standard deviation of RTT in cluster cases was more significant, more than 8 within 10 meters. As the communication distance increased, both average RTTs and the standard deviations rose. The maximum RTT was up to 140 ms at 13 meters.

6.3.3 Energy and Memory Performance

The developed BLE Mesh network is for outdoor use. In Finland, temperatures are low most of the year, and low temperatures can significantly impact the battery. Therefore, the performance of the node equipped with the same battery was tested at different temperatures. The expected running time was around ten hours, so the

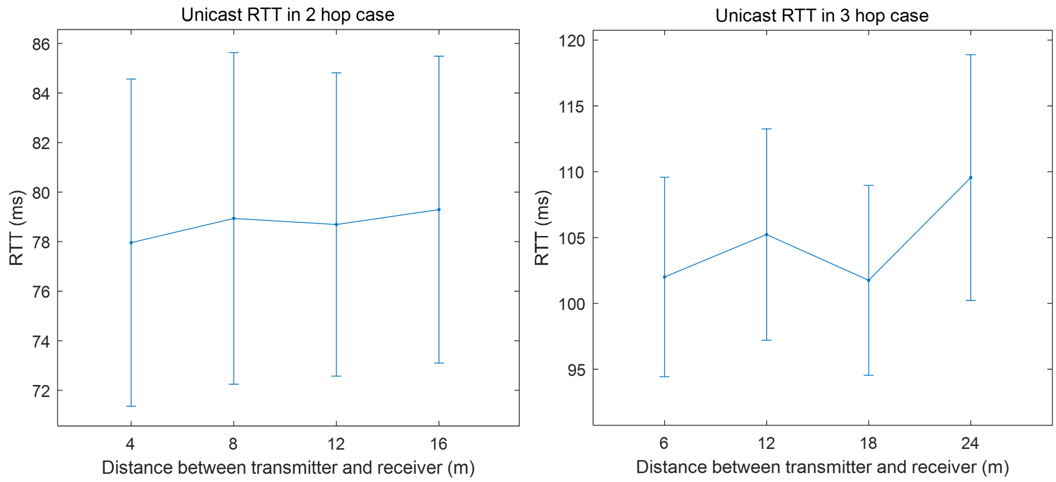

Figure 6.9: Average RTT and standard deviation in 2-hop case (Left) and 3-hop case (Right)

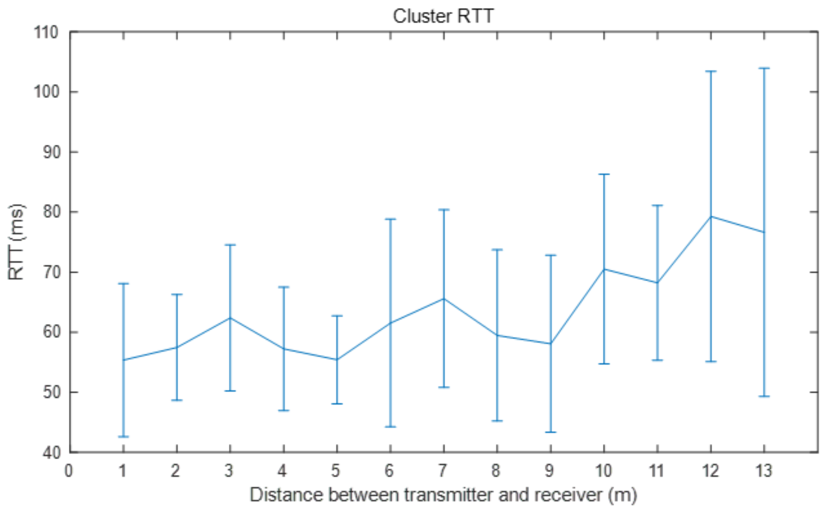

Figure 6.10: Average RTT and standard deviation of in cluster case

experiments were conducted at night. The local meteorological office in Finland provided temperature data.

In practice, the node is in standby mode for a long time and only switches to monitoring mode when people are present (Chapter. 5.3.2). In this experiment, a node kept operating in the monitoring mode until the battery was out of power. The monitoring mode consumes the most power because it needs to maintain BLE Mesh communication, record the acoustic samples and calculate feature vectors. The acoustic sensor recorded the voice for 7 seconds once, and the sampling rate was 44.1 kHz. The dimensions of each feature vector were 1 x 297, and the memory footprint was 5.60 KB. The total running time of the node, the average computation time for the feature vector, the number of the feature vectors, and the total memory are presented in Table. 3.

Table 3: Power and memory performance in different temperature

Parameter	Test 1	Test 2	Test 3	Test 4
Temperature (°C)	-8 ~ -2	-8 ~ -3	0 ~ 2	3 ~ 7
Feature vector number	4451	4400	4967	4949
Operating time (h)	9.44	9.34	10.55	10.51
Average computation time (ms)	238.8	240.9	240.4	239.5
Total memory (MB)	24.930	24.642	27.824	27.716

This experiment proved that low temperature could reduce the power supply capacity of the node battery. The node could run for more than 9 hours when the temperature is under 0 °C. When the temperature is above 0 °C, the node can operate for more than ten hours, which is about one hour longer than at low temperature. The average computation time for a feature vector was 240 ms. A node could collect 4000 to 5000 feature vectors if it keeps working, requiring about 24 to 27 MB of memory.

7 Conclusions and Future Work

7.1 Conclusions

This thesis developed a BLE Mesh network for acoustic and PIR sensing. A feature vector computation for speaker identification was implemented in the nodes.

Developed network performance was evaluated through several experiments. These experiments were set in an outdoor environment to be close to practical use. The maximum reliable communication distance for a single radio link between two nodes is 20 meters in the experiments. Packet delivery rate and round trip time of BLE Mesh communication have been tested under different network architectures. In unicast signal-hop and cluster architecture, the nodes had 100% PDR up to 13 meters. The average RTTs in these two cases only had a difference of 3 ms, but the unicast RTTs had more minor standard deviations of less than 5 while cluster RTTs had the standard deviations of more than 8. Extensive and more reliable communication could obtained by multi-hop communication paths by deploying relaying nodes in the network. However, as the number of hops increased by 1, the RTT also increased about 25 ms in the experiments.

In terms of power and memory performance, The nodes were able to operate in low temperatures down to -8 °C for more than 9 hours in the most power consumption mode. At temperatures above 0 °C, the operating time was extended by about one hour. The nodes could compute over 4000 feature vectors in their working time with a total memory footprint of about 24 MB. Since the node memory capacity is 64 GB, it was well sufficient for the feature vectors storage. The average computation time of each feature vector was 240 ms in the experiments.

7.2 Limitations and Future Work

In the thesis work, the functional implementation of the sensor in the node has been completed, but its sensing performance has not yet been evaluated. For the PIR sensing, the sensor used in the developed architecture (HC-SR501) is cheap and simple. After roughly testing, it performed well indoors, but in outdoor environments, the accuracy of PIR sensing was significantly reduced for unknown reasons. Therefore, other PIR sensors more suitable for the outdoor environment may be used and tested in the future. For the acoustic sensing, the computed feature vectors will be transmitted to the database for speaker identification. The quality of the calculated feature vectors is affected by the distance between the people and the acoustic sensor. The algorithms can be improved by filtering useless voice samples to reduce redundant computation and some noise cancellation processing. In addition, multiple nodes can record the same human voice, and redundancy can be utilized. Optimal node deployment can also affect speaker identification accuracy.

This thesis mainly focuses on communication performance of the BLE Mesh network. However, for the latency, only round trip time was tested. One-way latency is also meaningful to the latency analysis. The performance evaluation can be improved by implementing time synchronization to the network.

References

[1] S. R. Jino Ramson and D. J. Moni, "Applications of wireless sensor networks — a survey," in *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, 2017, pp. 325–329.

[2] M. J. et al., "Bluetooth mesh networks for indoor localization," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, 2019, pp. 397–402.

[3] A. Veiga and C. Abbas, "Proposal and application of bluetooth mesh profile for smart cities' services," *Smart Cities*, vol. 2, no. 1, pp. 1–19, 2019. [Online]. Available: https://www.mdpi.com/2624-6511/2/1/1

[4] H.-C. Lee and K.-H. Ke, "Monitoring of large-area iot sensors using a lora wireless mesh network system: Design and evaluation," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2177–2187, 2018.

[5] A. Zakrzewska, L. Koszalka, I. Pozniak-Koszalka, and A. Kasprzak, "Analysis of routing protocol performance in wireless mesh networks," in *2010 International Conference on Computational Science and Its Applications*, 2010, pp. 307–310.

[6] Z. Nurlan, T. Zhukabayeva, M. Othman, A. Adamova, and N. Zhakiyev, "Wireless sensor network as a mesh: Vision and challenges," *IEEE Access*, vol. 10, pp. 46–67, 2022.

[7] B. SIG, "Bluetooth technology website," https://www.bluetooth.com/learn-about-bluetooth/tech-overview/, 2019.

[8] B. Hussain, C. Qiu, and C. P. Yue, "Smart lighting control and services using visible light communication and bluetooth," in *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, 2019, pp. 1–2.

[9] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[10] A. Rezgui and M. Eltoweissy, "Service-oriented sensor-actuator networks [ad hoc and sensor networks]," *IEEE Communications Magazine*, vol. 45, no. 12, pp. 92–100, 2007.

[11] M. Patil and R. C. Biradar, "A survey on routing protocols in wireless sensor networks," in *2012 18th IEEE International Conference on Networks (ICON)*, 2012, pp. 86–91.

[12] C. Grumazescu, V.-A. Vlăduţă, and G. Subaşu, "Wsn solutions for communication challenges in military live simulation environments," in *2016 International Conference on Communications (COMM)*, 2016, pp. 319–322.

[13] S.-m. Xiong, L.-m. Wang, X.-q. Qu, and Y.-z. Zhan, "Application research of wsn in precise agriculture irrigation," in *2009 International Conference on Environmental Science and Information Application Technology*, vol. 2, 2009, pp. 297–300.

[14] M. T. Lazarescu, "Design of a wsn platform for long-term environmental monitoring for iot applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.

[15] N. Sharma and R. Bhatt, "Privacy preservation in wsn for healthcare application," *Procedia Computer Science*, vol. 132, pp. 1243–1252, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918307725

[16] A. El Kouche, L. Al-Awami, H. Hassanein, and K. Obaia, "Wsn application in the harsh industrial environment of the oil sands," in *2011 7th International Wireless Communications and Mobile Computing Conference*, 2011, pp. 613–618.

[17] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proceedings. 10th IEEE international symposium on modeling, analysis and simulation of computer and telecommunications systems.* IEEE, 2002, pp. 129–136.

[18] G. Fersi, W. Louati, and M. Ben Jemaa, "Distributed hash table-based routing and data management in wireless sensor networks: a survey," *Wireless networks*, vol. 19, no. 2, pp. 219–236, 2013.

[19] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Relay node deployment strategies in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 2, pp. 145–159, 2010.

[20] K.-Y. Shin, J. Song, J. Kim, M. Yu, and P. S. Mah, "Rear: Reliable energy aware routing protocol for wireless sensor networks," in *The 9th International Conference on Advanced Communication Technology*, vol. 1, 2007, pp. 525–530.

[21] Z. Nurlan, T. Zhukabayeva, M. Othman, A. Adamova, and N. Zhakiyev, "Wireless sensor network as a mesh: Vision and challenges," *IEEE Access*, vol. 10, pp. 46–67, 2022.

[22] J. Fan, S.-t. Yin, Q. Wu, and F. Gao, "Study on refined deployment of wireless mesh sensor network," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010, pp. 1–5.

[23] N. K. Gupta, *Inside Bluetooth Low Energy, Second Edition*. Artech House, 2016.

[24] R. G. Garroppo, L. Gazzarrini, S. Giordano, and L. Tavanti, "Experimental assessment of the coexistence of wi-fi, zigbee, and bluetooth devices," in *2011*

IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2011, pp. 1–9.

[25] N. K. Gupta, *Inside Bluetooth Low Energy, Second Edition*. Artech House, 2016.

[26] P. Kindt, D. Yunge, M. Gopp, and S. Chakraborty, “Adaptive online power-management for bluetooth low energy,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2695–2703.

[27] M. Woolley, “Bluetooth core specification v5. 1,” in *Bluetooth*, 2019.

[28] F. Almenárez-Mendoza, L. Alonso, A. Marín-López, and P. Cabarcos, “Assessment of fitness tracker security: A case of study,” *Proceedings*, vol. 2, p. 1235, 10 2018.

[29] T. Instruments, “Bluetooth low energy protocol stack,” https://software-dl.ti.com/lprf/simplelink_cc2640r2_sdk/1.00.00.22/exports/docs/blestack/html/ble-stack/, 2016.

[30] Z. Project members, “Bluetooth stack architecture,” https://docs.zephyrproject.org/latest/guides/bluetooth/bluetooth-arch.html, Mar 2022.

[31] K. Townsend, C. Cufí, R. Davidson *et al.*, *Getting started with Bluetooth low energy: tools and techniques for low-power networking*. " O’Reilly Media, Inc.", 2014.

[32] R. Shorey and B. Miller, “The bluetooth technology: merits and limitations,” in *2000 IEEE International Conference on Personal Wireless Communications. Conference Proceedings (Cat. No.00TH8488)*, 2000, pp. 80–84.

[33] M. Technology, “Ble link layer roles and states,” https://microchipdeveloper.com/wireless:ble-link-layer-roles-states, 2021.

[34] S. M. Darroudi and C. Gomez, “Bluetooth low energy mesh networks: A survey,” *Sensors*, vol. 17, no. 7, p. 1467, 2017.

[35] B. SIG, “Mesh profile version 1.0,” https://www.bluetooth.com/specifications/mesh-specifications, 2017.

[36] K. Ren, “Bluetooth mesh networking: Friendship,” https://www.bluetooth.com/blog/bluetooth-mesh-networking-series-friendship/, 2017.

[37] R. Muhendra, A. Rinaldi, M. Budiman, and Khairurrijal, “Development of wifi mesh infrastructure for internet of things applications,” *Procedia Engineering*, vol. 170, pp. 332–337, 2017, engineering Physics International Conference 2016 – EPIC 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817311566

[38] K. Ren and M. Woolley, "The fundamental concepts of bluetooth mesh networking," https://www.bluetooth.com/blog/the-fundamental-concepts-of-bluetooth-mesh-networking-part-2/, 2017.

[39] P. N. Saranu, G. Abirami, S. Sivakumar, K. M. Ramesh, U. Arul, and J. Seetha, "Theft detection system using pir sensor," in *2018 4th International Conference on Electrical Energy Systems (ICEES)*, 2018, pp. 656–660.

[40] Tina, Sonam, Harshit, and M. Singla, "Smart lightning and security system," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019, pp. 1–6.

[41] G. Doddington, "Speaker recognition—identifying people by their voices," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1651–1664, 1985.

[42] A. Charisma, M. R. Hidayat, and Y. B. Zainal, "Speaker recognition using mel-frequency cepstrum coefficients and sum square error," in *2017 3rd International Conference on Wireless and Telematics (ICWT)*, 2017, pp. 160–163.

[43] M. Bocca, R. Virrankoski, and H. Koivo, "Text and language independent speaker identification by using short-time low quality signals," in *Workshop on Wireless Communication and Applications (WoWCA 2008)*, 2008.

[44] C. Bell, *Beginning sensor networks with Arduino and Raspberry Pi.* Apress, 2014.

[45] N. Semiconductor, "nrf52840 product specification v1.7," https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.7.pdf, 2021.

[46] Maono, "Au-410 usb lavalier microphone specification," https://www.maono.com/products/usb-lavalier-microphone-maono-au-410, 2022.

[47] N. Semiconductor, "nrf5 sdk for mesh," https://www.nordicsemi.com/Products/Development-software/nrf5-sdk-for-mesh, 2022.

[48] SEGGER, "About segger microcontroller," https://www.segger.com/about-us/the-company/, 2022.

[49] N. Semiconductor, "Installing the toolchain," https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.meshsdk.v5.0.0/md_doc_getting_started_how_to_toolchain.html, 2020.