Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Dipanjan Adhikary

Message Prioritization in Bluetooth Mesh Networks

Master's thesis in Communication Technology
Supervisor: Yuming Jiang
Co-supervisor: Omkar Kulkarni, Anders Storrø
July 2023

NTNU
Norwegian University of
Science and Technology

Dipanjan Adhikary

Message Prioritization in Bluetooth Mesh Networks

Master's thesis in Communication Technology
Supervisor: Yuming Jiang
Co-supervisor: Omkar Kulkarni, Anders Storrø
July 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Message Prioritization in Bluetooth Mesh Networks

Adhikary, Dipanjan

Submission date:	July 2023
Main supervisor:	Jiang, Yuming, NTNU
Co-supervisors:	Kulkarni, Omkar, Nordic Semiconductor ASA and Storrø, Anders, Nordic Semiconductor ASA

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title: Message Prioritization in Bluetooth Mesh Networks
Student: Adhikary, Dipanjan

Problem description:

Bluetooth Mesh is an exciting new technology. It is being used for industrial-scale automation. Currently, it is being used mainly for lighting system control but in the near future, the use cases will increase. This means the network traffic will also grow. This might adversely affect the quality of service (QoS) for time-critical operations. All communication in Bluetooth Mesh network is accomplished by sending messages. There are several messages in the mesh network that are not time critical. By designing a message prioritization method it might be possible to maintain the standard of QoS. The goal of this thesis will be to enable a systematic prioritization of mesh messages. The research questions would be to see how prioritization affects performance.

Tests for the thesis will be done in two parts. The first part would be a simulation using the Babblesim physical layer simulator and the second part would be conducted on a test network installed in the Trondheim office of Nordic Semiconductor.

The approaches of prioritization that will be attempted are

- Prioritization Based on Source-Destination Pair
- Prioritization Based on Opcodes

The observations would be the network QoS metrics such as latency, packet delivery ratio etc. Comparison between network performance with and without prioritization will allow us to decide if and how message prioritization is affecting the network performance.

Approved on: 2023-02-23
Main supervisor: Jiang, Yuming, NTNU
Co-supervisors: Kulkarni, Omkar, Nordic Semiconductor ASA and Storrø, Anders, Nordic Semiconductor ASA

Abstract

Bluetooth mesh is a promising new technology with applications in the industrial IoT sector. It allows reliable and scalable communication that requires no central controller.

This thesis work develops two methods for prioritizing Bluetooth mesh messages. Extensive simulations were conducted under diverse scenarios, providing a comparative analysis of network performance under heavy noise, with and without mesh message prioritization.

The study concludes that applying the proposed prioritization methods significantly mitigates the impact of heavy noise on network latency. Additionally, this work demonstrates that further optimization can be achieved by tuning protocol-related parameters, namely Network Transmit Count (NTC) and Relay Retransmit Count (RRC). When applied in conjunction with the prioritization mechanism, these adjustments can bolster the reliability of Bluetooth mesh networks, even in challenging noise scenarios. This research thus presents a valuable contribution to the optimization of Bluetooth mesh technology for industrial IoT applications.

Preface

This thesis has been submitted for completion of the Master of Science degree in Communication Technology at the Department of Information Security and Communication Technology at the Norwegian University of Science and Technology (NTNU). The work for this thesis was started in February 2023 and is the continuation of the specialization project which was done in January 2023.

The research topic was suggested by Nordic Semiconductor ASA. All of the software and hardware tools were provided by Nordic Semiconductor. The work was supervised by Professor Yuming Jiang from NTNU and co-supervised by Omkar Kulkarni and Anders Storrø from Nordic Semiconductor.

Acknowledgment

I would like to thank my supervisor Professor Yuming Jiang for his help and motivation throughout the thesis period.

I am especially grateful to Omkar Kulkarni and Anders Storrø for guiding me through the technical works and showing an extraordinary amount of patience with me.

Finally, my family and friends for supporting me through the hardest ten months of my life.

Contents

List of Figures

List of Tables

List of Acronyms

AppKey Application Key.

BLE Bluetooth Low Energy.

Bluetooth SIG Bluetooth Special Interest Group.

BTM Bluetooth mesh.

IoT Internet of Things.

NetKey Network Key.

NTC Network Transmit Count.

PDU Packet Data Unit.

QoS Quality of Service.

RRC Relay Retransmit Count.

RTOS Real Time Operating System.

SDK Software Development Kit.

TTL Time To Live.

Chapter 1

Introduction

1.1 Problem Description and Motivation

The Internet of Things (IoT) has become integral to modern technology over the last two decades. From smart homes to wireless wearables, the use cases have increased significantly. Various technologies are used for IoT communications, including WiFi, Zigbee, Thread, and Bluetooth. Among Bluetooth technologies, Bluetooth Low Energy(BLE) has become the gold standard for wireless sensor networks and smart wearable devices.

Bluetooth Mesh (BTM) was released in 2017 by the Bluetooth Special Interest Group (SIG). It is built upon the BLE stack and allows many-to-many (m:m) communication over Bluetooth radio. BTM is a new addition to the group of available technologies. It is currently used for industrial lighting systems, sensor networks, and various other IoT applications [BTMIOT].

It is possible to include BLE-compatible devices to create a mesh network that can range to thousands of devices, enabling greater coverage and functionality. The ease of deployment and compatibility with BLE devices makes BTM an exciting technology.

As it is a new technology, only a handful of studies have been conducted on the performance of BTM. Studies conducted so far find that BTM networks suffer from a loss of quality of services (QoS) with the increase of network and packet size [SiLabs]. Network deployment strategy and relay density were also found to be important factors [Ericsson]. The network was found to be vulnerable to congestion for a dense deployment setup [RMGG20].

The use cases of Bluetooth mesh are expected to increase significantly in the near future. Over the next five years, a 115% increase in the number of BTM lighting control devices is expected [Bluetooth Blog]. With the increase in the number of use cases and network size, the network performance may get impacted.

All communication in Bluetooth mesh is done by the exchange of messages. It is possible to develop a prioritization method for these mesh messages. Depending on the use cases there are time-critical and non-time-critical messages. By prioritizing time-critical message transfer over non-time-critical messages, it is possible to obtain a high standard of QoS for time-critical messages. The objective of this thesis is to develop such a message prioritization method. The goal would be to investigate the network performance with message prioritization. The observations would be if and how message prioritization impacts network performance.

In this thesis work the performance of Bluetooth mesh network have been tested under high noise office scenarios. The observations from the tests show that the QoS metrics, such as network latency and packet delivery ratio, get negatively impacted when the amount of noise is high in the network. For the majority of scenarios with high time-critical dependencies, such levels of QoS would be deemed unacceptable. As Bluetooth mesh implements multiple layers of encryption for the messages, prioritization of mesh messages can be challenging. For multi-hop communications, the network uses relay nodes and these nodes would play an integral role in message prioritization. Only the information that can be accessed in a relay node could be used for prioritization. In this work, two novel message prioritization methods were successfully developed.

After conducting extensive simulations and testing under various scenarios, it was observed that using the developed prioritization methods significantly improves network latency without introducing significant network overhead. Also, there are configurable protocol-related parameters such as Network Transmit Count (NTC) and Relay Retransmit Count (RRC) which can help with the improvement of network reliability such as successful packet delivery ratio. By utilizing these protocol-related parameters alongside the prioritization methods, excellent QoS service metrics were possible to obtain.

These implementations would help increase the mesh network's performance even under high traffic and future-proof the network.

This project was done in collaboration with Nordic Semiconductor ASA. It is a

Norwegian fabless technology company. They specialize in designing ultra-low-power wireless communication semiconductors and supporting software for engineers developing and manufacturing IoT products [Nordic Semiconductor].

1.2 Report Outline

This thesis is divided into ten chapters. They are:

- Chapter 1: Introduction. Introduces the problem and motivation for the work.
- Chapter 2: Background. Provides some useful technical background and context.
- Chapter 3: Related Works. Presents the state of the art of current research.
- Chapter 4: Tools and Software. A brief overview of the software and hardware tools used for testing.
- Chapter 5: Methodology. Describes the methodology used in detail.
- Chapter 6: Test Setup. Overview of the various network scenarios used for testing purposes.
- Chapter 7: Results. Presentation of the results and brief analysis.
- Chapter 8: Discussions. Discussions and key takeaways from the work.
- Chapter 9: Conclusion. Draws conclusion.
- Chapter 10: Future Works. Presents possible scopes for future works.

Chapter 2

Background

This section will provide an overview of Bluetooth mesh (BTM) technology, the layered architecture, and some important terminologies.

BTM is built upon the BLE stack and allows many-to-many (m:m) communication over the Bluetooth radio. There is no centralized controller, and the devices communicate directly with each other by exchanging messages. BTM offers greater scalability and reliability since there is no single point of failure.

2.1 Bluetooth Mesh Layered Architecture

Bluetooth mesh has a layered architecture. It is shown in Figure 2.1

Model Layer: The Model layer defines models that are used to standardize the operation of typical user scenarios and are defined in the Bluetooth Mesh Model specification or other higher layer specifications. Examples of higher-layer model specifications include models for lighting and sensors [MshPRF].

Foundation Model Layer: It defines the states, messages, and models required to configure and manage a mesh network [MshPRF].

Access Layer: The access layer defines how higher-layer applications can use the upper transport layer. It defines the format of the application data; it defines and controls the application data encryption and decryption performed in the upper transport layer; and it checks whether the incoming application data has been received in the context of the right network and application keys before forwarding it to the higher layer [MshPRF].

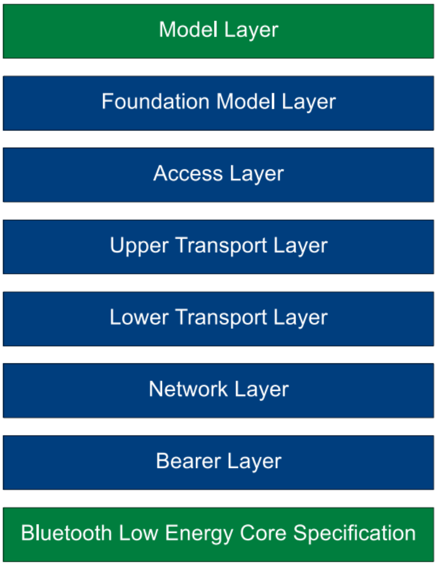


Figure 2.1: Mesh System Architecture: Source [MshPRF].

Upper Transport Layer: The upper transport layer encrypts, decrypts, and authenticates application data and is designed to provide confidentiality of access messages. It also defines how transport control messages are used to manage the upper transport layer between nodes, including when used by the Friend feature [MshPRF].

Lower Transport Layer: The lower transport layer defines how upper transport layer messages are segmented and reassembled into multiple Lower Transport Packet Data Units (PDUs) to deliver large upper transport layer messages to other nodes. It also defines a single control message to manage segmentation and reassembly [MshPRF].

Network Layer: The network layer defines how transport messages are addressed toward one or more elements. It defines the network message format that allows Transport PDUs to be transported by the bearer layer. The network layer decides whether to relay/forward messages, accept them for further processing, or reject them. It also defines how a network message is encrypted and authenticated [MshPRF].

Bearer Layer: The bearer layer defines how network messages are transported between nodes. There are two bearers defined, the advertising bearer and the Generic Attribute (GATT) bearer. Additional bearers may be defined in the future [MshPRF].

BLE Core specification: This is not a layer. This is the full BLE stack which is essential for the basic wireless communications capabilities delivered by the BTM [MshPRF].

2.2 Managed Flooding

Bluetooth mesh uses a method known as “managed flooding” to publish and relay messages. In BTM messages are not routed on a specific path. Instead, all devices that are within the radio range receive the message and the relay devices re-transmit the messages to other devices within its range. By tuning parameters such as Time to Live (TTL) and Relay Re-transmit Count (RRC), it is possible to manage this flooding method [BTM].

2.3 Devices and Nodes

Devices that are part of a BTM network and can transmit and receive messages are called nodes. A provisioning mechanism allows a device to be a part of the network. A provisioner device does the provisioning. A device that is not part of the mesh network is called an unprovisioned device. An unprovisioned device can not transmit mesh messages but will advertise its presence to the provisioner. Elements are addressable entities within a device. A device must have at least one element [MshPRF].

The nodes in the mesh network share four common resources [MshPRF]. They are:

1. **Network Addresses:** It is used to identify the source and destination of messages.
2. **Network Keys:** Network keys are used to secure and authenticate messages at the network layer.
3. **Application Keys:** To secure and authenticate messages at the access layer.
4. **Initialization Vector(IV) Index:** Used to extend the lifetime of the network.

The condition of an element is represented by a value which is known as a state. An element that is exposing a state is called a server [MshPRF] e.g., a Generic On-Off server. An element that is accessing a state is called a client, e.g., a Generic On-Off client [MshPRF].

BTM uses a publish/subscribe method for data exchange between nodes. An element can have three types of addresses, Unicast, Groupcast, and Virtual addresses. For unicast transmission, the target is a single element inside a device, and the message is processed upon reception. For groupcast/virtual addresses, the devices will subscribe to the group/virtual address, and only those subscribed devices will process the received packets. A network can have multiple subnets, which enables area isolation. One node can be part of multiple subnets [MshPRF].

2.4 Bluetooth Mesh Messages

In BTM, communication between nodes is done by the transfer of messages [MshPRF]. They operate in states. For each state, there are defined sets of messages. A message has an opcode, an associated parameter and behavior. An opcode can range from 1-3 octets in length. For the maximum possible payload in special messages, they are one octet long. For standard messages two octets, and for vendor-specific messages, they are three octets long. The transport layer determines the total message size that includes the opcodes. Messages that are up to 11 octets are transferred as non-segmented messages. There is a Segmentation and Reassembly (SAR) mechanism built into the transport layer. Ideally, it will try to fit messages into a single segment in order to reduce SAR overhead. The SAR mechanism is capable of handling a maximum of 32 segments. The maximum message size is 384 octets when using SAR [MshPRF]. Also, messages can be either acknowledged or unacknowledged.

2.5 Packet Data Unit (PDU)

The mesh PDU format is shown in Table 2.1

2.6 Node Types

In BTM a node can have different functionality. There can be four main types of nodes [MshPRF]. They are

1. Relay Node
2. Low Power Node

Field Name	**Bits**	**Notes**
IVI	1	Least significant bit of Initialization Vector (IV) Index
NID	7	Value derived from the NetKey used to identify the Encryption Key and Privacy Key used to secure this PDU
CTL	1	Network Control
TTL	7	Time To Live
SEQ	24	Sequence Number
SRC	16	Source Address
DST	16	Destination Address
TransportPDU	8 to 128	Transport Protocol Data Unit
NetMIC	32 or 64	Message Integrity Check for Network

Table 2.1: Network PDU field definitions [MshPRF]

3. Proxy Node

4. Friend Node

Relay Node: A relay node enables relay features. It is possible to either enable or disable relay functionality in a node. A relay node will forward a network PDU using the advertising bearer to other nodes that are within its radio range. It is possible to significantly improve network coverage by proper relay selection [MshPRF].

Proxy Node: A proxy node supports as the name suggests proxy features. BLE devices that are not mesh-compatible can use the proxy node to be part of a BTM network. These nodes act as intermediaries and use the GATT features in order to allow non-compatible devices to be part of the mesh network [MshPRF].

Low Power Node: A low-power node (LPN) is a node that supports low-power features. An LPN will work with significantly low power consumption. It is used together with a friend node to provide optimum energy efficiency. It has a sparse hearing time than the other nodes. The friend nodes come in handy in order for the LPN to access important updates or critical commands [MshPRF].

Friend Node: Friendship is a unique feature of Bluetooth mesh. The definition of friendship is taken verbatim from the standard [MshPRF], "Friendship is used by Low Power nodes to limit the amount of time that they need to listen. If a node cannot

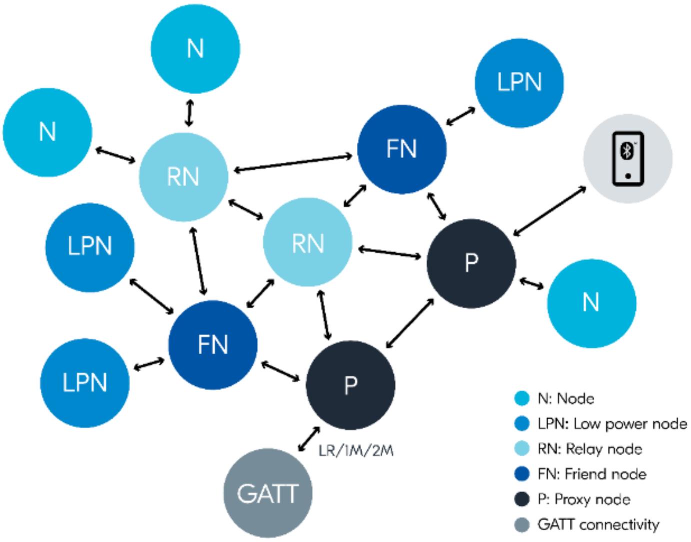


Figure 2.2: Bluetooth mesh Topology: Source [Bluetooth Mesh]

receive continuously, then it is possible that it will not receive mesh messages that it should be processing. This includes security updates required for maintaining the security of the network as well as the normal mesh messages. If the Low Power node does not receive such messages, then it may not operate as desired and it may also fail to keep up-to-date with the latest security state of the network and eventually drop off the network if this security is changed without its knowledge." Thus a friend node can help an LPN to conserve power and serve as an important element for a BTM sensor network.

2.7 Network and Application Key:

Network Key: The network key also known as NetKey is a cryptographic key that is used to encrypt messages at the network layer [MshPRF]. Devices that use the same netkey are part of the same network. In a larger network, there can be multiple netkeys, and thus it is possible to create multiple subnets. When a message is being relayed the relay node can use the netkey to authenticate messages at the network level, but the application content can still remain secure.

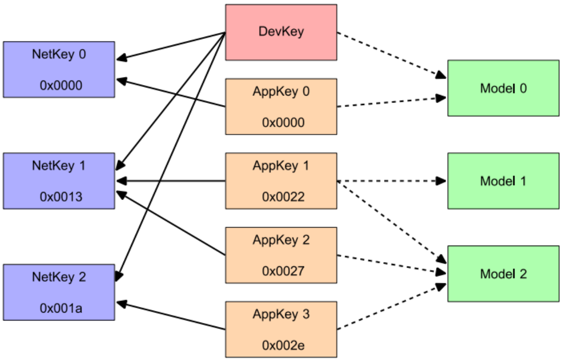


Figure 2.3: Association Between AppKeys, NetKeys and Models: Source [MshPRF]

Application Key: The application key or AppKey is a cryptographic key that is used for encryption in the application layer [MshPRF]. AppKey encryption ensures payload security during the relay procedure. Multiple AppKey can be used in a BTM network. Without knowing the appropriate AppKey the payload can not be decrypted.

Since a node can have multiple AppKeys and NetKeys in BTM key identifiers are used. Using a key derivation function the key identifier is generated from the AppKey or the NetKey. A network key identifier is transmitted in each Network PDU using a 7-bit value, while the application key identifier is transmitted in each Lower Transport PDU using a 6-bit value [MshPRF].

2.8 Network Transmit Count

This parameter Controls the initial number of re-transmissions of original messages in addition to the first transmission. The value can range between zero and seven. The interval between re-transmissions of original messages can be between 10 and 330 milliseconds [MshPRF].

2.9 Relay Retransmit Count

This parameter Controls the initial number of re-transmissions of relayed messages in addition to the first transmission. Its value also ranges between zero and seven. The re-transmission interval can range between 10 and 330 milliseconds [MshPRF].

2.10 2.4GHz ISM band Channel Fading Model

Bluetooth Low Energy (BLE) operates in the 2.4GHz industrial, scientific, and medical (ISM) band. For indoor communications which is the primary focus, the path loss can be modeled by the following equation [ISM]:

$$L_{\text{total}} = 20\log_{10}(f) + N \times 10\log_{10}(d) + L_{f(n)} - 28 \text{ (dB)} \tag{2.1}$$

Here

- d is the distance in meters.
- f is the frequency in MHz.
- $L_{f(n)}$ is attenuation when changing floor.

More specifically $L_{f(n)}$ refers to the wallsfloors attenuation if the transmitter and receiver and not placed in the same roomfloor.

- N is the path loss exponent. The value would be:
  - 2 for big open rooms
  - around 1.8 for corridors
  - around 2.8 as an average for residential buildings

For the testing and simulations, the nRF52 series chip has been used. It has four dBm transmission power, and the receiver sensitivity for BLE long range is -104 dBm [nrf52].

Chapter 3

Related Works

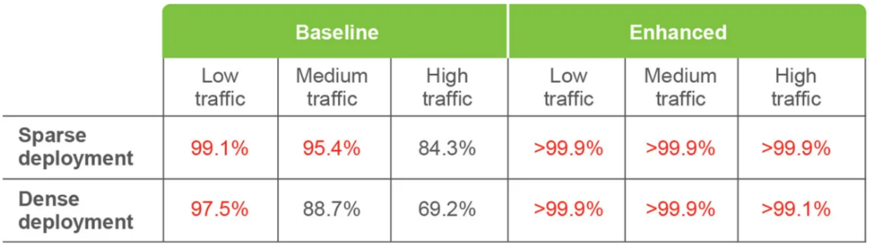
As Bluetooth Mesh is a relatively new technology, there have been only a few studies that look into various aspects of BTM performances. Here some of the relevant studies will be discussed briefly.

3.1 Silicon Labs AN1137: Bluetooth® Mesh Network Performance

Silicon Labs conducted a study to observe BTM performances using a test network spanning a 2230 square meter office area [SiLabs]. There were other wireless networks such as WiFi. The largest test network they had consisted of 192 nodes. They also had network stacks of 24, 48, 96 and 144 nodes. The testing was done over 100 packets. For the different-sized stacks, they observed latency and reliability for set traffic conditions.

They observed that with the increase in stack size, the average latency increases even for small (8-byte) packets. By increasing packet size the latency increases for fixed network size.

They observed excellent latency when the payload is contained in a single packet. The effect of increasing packet size had a higher impact on latency increase than increasing network size. They recommend not using segmented messages for multicast transmission. For large networks, proper relay selection is critical for network performance.



	Baseline			Enhanced		
	Low traffic	Medium traffic	High traffic	Low traffic	Medium traffic	High traffic
Sparse deployment	99.1%	95.4%	84.3%	>99.9%	>99.9%	>99.9%
Dense deployment	97.5%	88.7%	69.2%	>99.9%	>99.9%	>99.1%

Figure 3.1: Ericsson: Bluetooth Mesh Networking Test Reliability Results: Source [Ericsson]

3.2 Ericsson: Bluetooth Mesh Networking

Ericsson published a white paper that includes a large-scale building automation use case and observes the impact of deployment strategy and configuration on BTM networks [Ericsson]. Their test network was deployed in an area of 2000 square meters. It has 879 devices, including window sensors, occupancy sensors, HVAC sensors and actuators, light switches and light bulbs.

They considered three traffic conditions:

1. low traffic (150 bps)
2. medium traffic (1 kbps)
3. high traffic (3 kbps)

They also had an enhanced configuration that also included first-hop message repetitions and randomization of advertising triplet. They had a sparse relay deployment of 12 relays and a dense deployment of 49 relays.

For low traffic and sparse relay deployment, the network had a satisfactory performance obtaining a 99.1% successful message delivery. However, when traffic increases the baseline deployment was not able to perform in a satisfactory way as it was experiencing traffic congestion. Their enhanced configuration had network layer

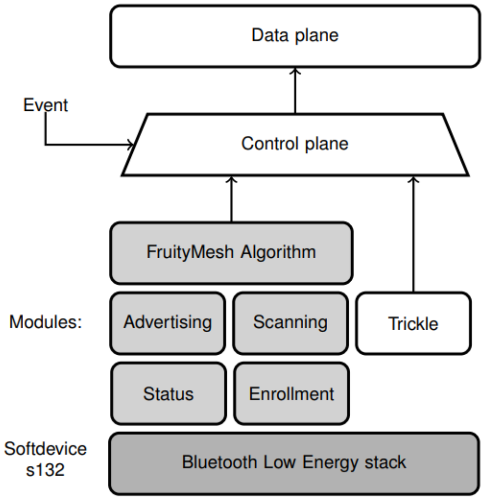


Figure 3.2: Bluetooth Now Architecture: Source [MRC+17]

re-transmissions and randomization. Using that they had the expected level of performance (>99%) for dense deployment and high traffic. For optimal performance, they recommend 1.5% of the total number of nodes to be selected as relays.

3.3 Murillo et al. Bluetooth now or low energy

In this paper, the authors propose a novel method for message propagation in BTM [MRC+17]. By default, BTM uses the managed flooding method. The proposed architecture by the authors is called Bluetooth now. It can automatically switch between a flooding oriented and a connection-oriented network. This switching can be done based on message priority. They considered various QoS metrics like packet delivery ratio (PDR), end-to-end delay and power consumption. For periodic and non-time sensitive data they recommend using connection-oriented mesh to reduce power consumption. Their proposed architecture uses the FruityMesh algorithm [FruityMesh].

3.4 Rondón et al. Understanding the Performance of Bluetooth mesh

In this study, the authors evaluated the Quality of Service (QoS) and scalability of BTM [RMGG20]. Their tests were conducted via simulations. They simulated an office environment that had 28 nodes. They found BTM to be vulnerable to network congestion for a dense deployment setup. There was also an increased probability of packet collision. However, the presence of other BTM devices and interference did not have an impact. But, interference from WLAN devices had an adverse impact on the performance.

3.5 Agnoletto et al. Time Slot Transmission Scheme with Packet Prioritization for Bluetooth Low Energy Devices Used in Real-Time Applications

In this paper, the authors divided the connection interval into equally spaced slots [AJdF20]. They also developed a scheduler that would populate the slots based on message priorities. For time-constrained data, they reserved the first and the last slots and the intermediary slots can contain noncritical data. They found that in order to achieve a good balance between network jitters and reliability the number of ordinary transmissions should be limited. By using this slotted scheme, it was possible to reduce power consumption for battery-powered devices. They were able to achieve a significant improvement in the delay variance for real-time transmissions.

3.6 Basu et al. QoS Enabled Heterogeneous BLE Mesh Networks

In this paper, the authors developed a prioritization method for mesh messages [BBH21]. They introduced a feature to include a priority level for each message and changed the first-in-first-out (FIFO) approach to a priority-based approach. When the timer to start the next message transmission expires in the node (either a source or a relay), the queued packet that has the highest priority in the buffer is transmitted. When the buffer is full, they ensured the enqueuing of high-priority packets by dropping packets that have the lowest priority. The authors also discussed the concept of back-off times. Higher priority packets can be configured with lower backoff times and vice versa. Using this feature helps to predict and improve end-to-end delays. They also observed that proper relay selection can help with reducing network congestion and provide an efficient pathway for messages.

3.7 Qaq et al. Reliable Sensor Data Gathering with Bluetooth Mesh: An Experimental Study

In this paper, the authors provide suggestions to improve the reliability of the data-gathering process in an IoT sensor network based on BTM [QKJ23]. They evaluated the effect of various Bluetooth mesh protocol-related and non-protocol-related parameters. The protocol-related parameters explored in this paper include the Publish Retransmit Count (PRC), the Network Transmit Count (NTC), and the Relay Retransmit Count (RRC). The non-protocol-related parameters include the packet-sending randomization interval and the packet payload redundancy. They found that a high degree of reliability, 99%, can be achieved with a combined use of these parameters. They also found that the placement of the sensor data collector or gateway, relay selection and parameter tuning to be of fundamental importance to achieve high reliability. Their findings conclude that reliability levels can be significantly improved with proper combinations of protocol-related and non-protocol-related parameters.

3.8 Takeaways

Recent research on BTM performance reveals various findings and recommendations. [SiLabs] work demonstrated that network latency increases with packet and network size. They suggest avoiding segmented messages for multicast transmission and emphasizing relay selection in large networks. The [Ericsson] work shows the importance of deployment strategy, configuration, and relay density. [QKJ23] found that by the combined use of protocol-related and non-protocol-related parameters reliability can be improved. [AJdF20] and [BBH21] proposed ideas for message prioritization and their developed methods showed promising results.

[BBH21] developed a form of source destination address-based prioritization that modifies the advertising queue. This has some similarities to the way the advertising queue was modified for the developed prioritization methods in this thesis. However, any kind of prioritization attempt for mesh messages will have to modify the advertising queue for the transmitter and relay node. Despite this similarity, the prioritization methods in this thesis were developed independently. Also, the work by [BBH21] lacks a prioritization method based on opcodes. Their work was conducted for a small setup and considered some simple scenarios. The developed method for this thesis conducted extensive simulations and checked prioritization performance under diverse scenarios thus providing more insights into the performance of message prioritization.

Chapter 4

Tools and Software

This chapter provides a brief overview of the hardware and software tools that have been used for the thesis work.

4.1 Software Tools

The two main software tools used for this thesis work are

- nRF Connect SDK
- BabbleSim Physical Layer Simulator

4.1.1 nRF Connect SDK

The nRF Connect SDK is a software development kit that is developed and maintained by Nordic Semiconductor ASA. It is used to build low-power wireless applications for nRF52, nRF53, nRF70, and nRF91 Series devices [nrfSDK]. The SDK contains optimized cellular IoT (LTE-M and NBIoT), Bluetooth Low Energy, Thread, Zigbee, and Bluetooth mesh stacks. It contains a wide range of examples, samples and extensive documentation. It includes the Zephyr real-time operating system (RTOS), which is built for connected low-power products. It is open source and available through GitHub. For this thesis SDK version 2.2.0 was used.

4.1.2 BabbleSim

BabbleSim is a physical layer simulator. It was developed and maintained by the Zephyr project [Babblesim]. It supports heterogeneous devices and methods and allows a user to run real device codes. Completely different types of devices can be run together for a simulation. For a device to be able to run in a simulation, all it needs

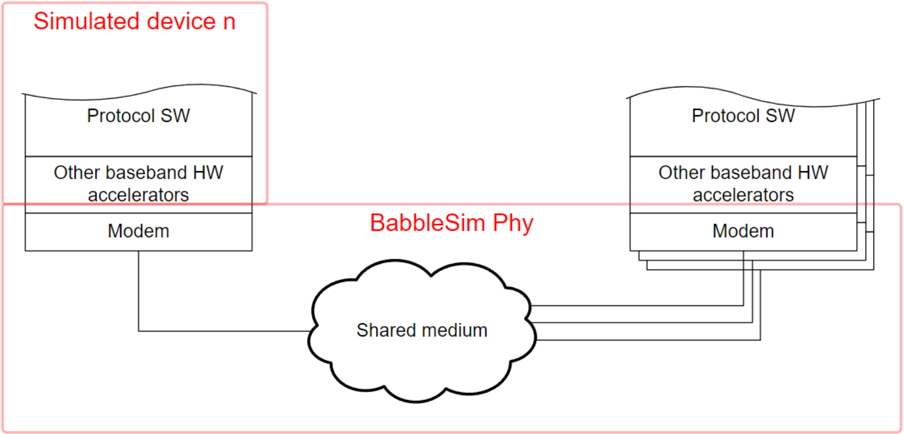


Figure 4.1: BabbleSim Phy-device interface: Source [Babblesim]

is to relay to the BabbleSim Phy its transmissions and reception attempt [Babblesim].

In BabbleSim, the physical layer simulation (Phy) and each device run as independent programs each in their own Linux processes [Babblesim]. The Phy is in charge of emulating the channel/shared medium and modem (analog and digital demodulation) for each of the devices and handling the coordination of the device in that medium. Babblesim is highly modular and allows easy debugging capabilities.

For BLE development, BabbleSim includes [Babblesim]:

1. A physical layer for BLE devices: The 2G4 Phy
2. Interferers models: Burst interferer, WLAN activity model
3. 3 selectable channel models
4. 2 selectable modem models
5. A device which can re-play back the activity of any device in a previous simulation run

For the simulation work, Zephyr's nrf52_bsim board was used to execute Zephyr and BabbleSim's 2G4 Phy was used to simulate the BLE communication over the 2.4GHz ISM band [Babblesim].

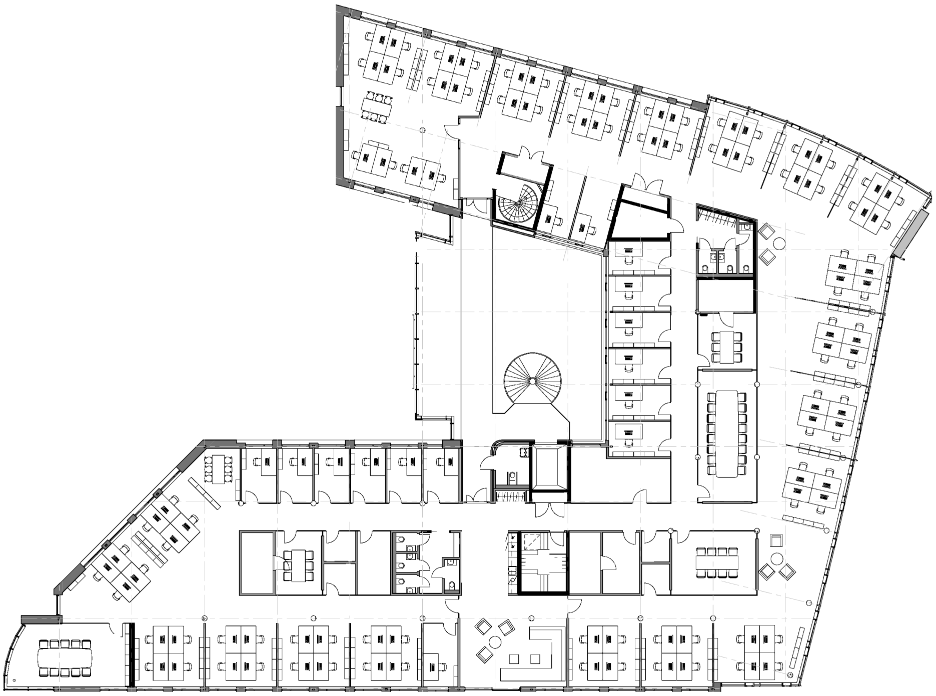


Figure 4.2: Nordic Semiconductor Test Network [Source: Nordic Semiconductor Internal]

4.2 Hardware Tools

The Nordic Semiconductor test network, Figure 4.2, consists of 100 kits installed on an area of 1400 m2 in the Nordic Semiconductor office in Trondheim. Each kit is a custom-made internal Power Over Ethernet (POE) development board labeled POE dev board Figure 4.3.

Nordic nRF52840 microchip is installed on the kits. With the Ethernet port and wiznet chip they can receive commands over Ethernet. Each kit has an ID that is associated with its MAC address. Each node is connected to one of three

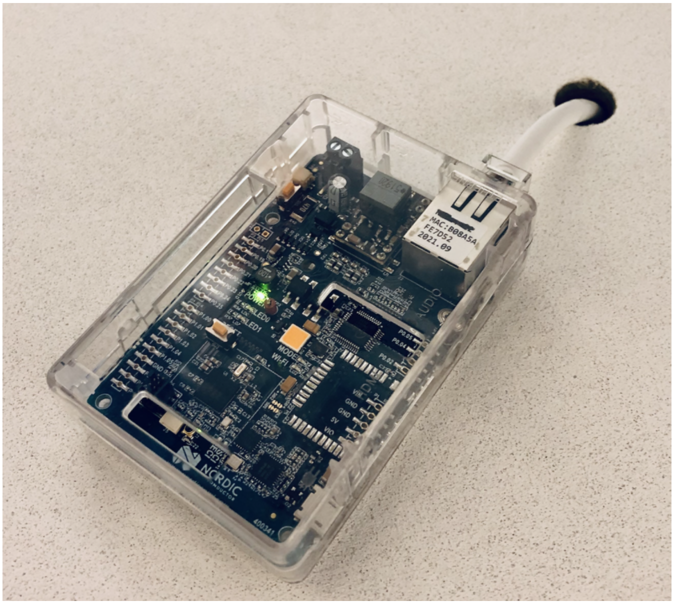


Figure 4.3: POE Dev Board [Source: Private Photo]

POE switches that operate together as one virtual device connected to a dedicated computer also known as the remote PC. The scripts for testing the performance or performing other commands such as network parameters configuration etc can be launched from the connected computer's terminal.

Chapter 5

Methodology

In this chapter, the research methodology will be described. First, a brief recapitulation of the previous attempts at prioritization done by Agneletto et al. and Basu et al. will be described. Then the prioritization hypothesis for this thesis work will be presented. This hypothesis will be tested and validated on a simple six-node setup. If successful, then the tests will be extended to a larger more complex setup.

5.1 State of The Art of BTM Message Prioritization

There are only a few research works focused on Bluetooth mesh message prioritization. In the related works chapter two of such attempts were introduced. The study conducted by Agneletto et al. attempted to divide the connection interval into equally placed slots [AJdF20]. Then the slots would be populated based on message priorities. The work done by Basu et al. attempted to change the FIFO-based transmission process into a priority-based process [BBH21]. Their methods enqueue a priority-labeled message into the buffer ahead of non-prioritized ones.

5.2 Hypothesis

When it comes to prioritizing Bluetooth mesh messages, the most important part would be the relay nodes. Since Bluetooth mesh uses multiple layers of encryption, access to information is limited in a relay node. To get the network layer information the NetKey must be available to the relay node and to get the application layer information it will require the AppKey.

During initial research done for the specialization project for this thesis, two possible methods of prioritization were hypothesized:

1. Prioritization Based on Source-Destination Pair

2. Prioritization Based on Opcodes

Elaboration on them is given below.

5.3 Source-Destination Pair Based Prioritization

Prioritization based on source-destination pair, hereafter denoted as SRC-DST prioritization, has the potential to be a powerful tool to ensure excellent QoS in a BTM network. For time-critical scenarios, if it is imperative to transfer messages to a destination node, SRC-DST prioritization should be able to deliver that with high reliability and low latency. Prioritizing messages only for certain destination nodes without also taking into consideration the source of that message would not yield any benefits. Because, in the network, it is possible that multiple nodes are transmitting messages to that particular destination node. So, for this type of prioritization to work both the source and destination address must be taken into account.

If the relay node has access to the same NetKey that is used by both the designated source and destination node, then it would be able to access the network layer data. By default, relay nodes will perform the network layer decryption. So, using prioritization here would not introduce extra overhead.

For SRC-DST prioritization, three parameters are necessary:

1. Prioritization Flag
2. Priority Source Address
3. Priority Destination Address

Among these, the source and destination addresses are available as part of the network layer data. The prioritization flag is a parameter that has to be set if certain conditions are met. The priority destination address can be either a unicast or a groupcast address.

It is possible to enable or disable the SRC-DST prioritization during runtime. If prioritization is enabled, first there will be a check in the network layer. It will check the source and destination address for the current packet. If the addresses match the priority source and destination, then the prioritization flag will be set.

The next check is done in the bearer layer before the advertisements are sent. Before the buffer is enqueued in the advertising queue, it will check for the prioritization flag. If the flag is set, then the advertising queue would be modified. In BTM the advertising queue is a first in first out (FIFO) queue. So, in order to ensure the priority buffer is sent out first some improvisation was necessary. First, the advertising queue was emptied out, and the contents were stored in a temporary queue. Then the priority buffer is enqueued, and then the contents from the temporary queue are added to the advertising queue in FIFO order. This method adds minimal overhead to the system but ensures the transmission of the priority packets. Figure 5.1 shows a simple flowchart describing the process of the SRC-DST-based prioritization

5.4 Opcode Based Prioritization

In Bluetooth Mesh, relay nodes are designed to operate at the network layer, and their primary function is to extend the range of the network by re-transmitting messages. Relay nodes are not intended to access or interpret the contents of the upper transport layer. The opcode is part of the upper transport PDU. If a message is transmitted as a segmented message, then the opcode is contained in the first segment of the message. Relay nodes will receive individual segments, but without reassembly, they won't have access to the full upper transport PDU. Thus without reassembling, it is not possible to obtain the opcodes in the relay nodes.

There was a workaround attempt of Opcode-based prioritization using mesh models. Models use opcodes to define the types of messages they can send or receive. For example, a Generic OnOff Server model can receive messages with an opcode that represents a "Generic OnOff Set" command. So, the attempt was to encrypt different models with different AppKeys. The AppKey index was initially thought to be accessible from the relay nodes. To give more context, in the nRF connect SDK implementation of Bluetooth mesh a structure called "bt_mesh_msg_ctx" is used in the stack implementation. IT is used for message transmission and relaying functions. This structure contains the AppKey index. However, in the implementation of nRF connect SDK the AppKey index is hardcoded to a default value of "0xffff" for nodes that are not the destination node of the intended message. So, it was not possible to go ahead with this method without making significant changes to the implementation of the SDK, and it is expected that this would introduce significant overhead to the system.

However, another workaround using NetKeys can be possible by taking advantage of the association between AppKeys and NetKeys.

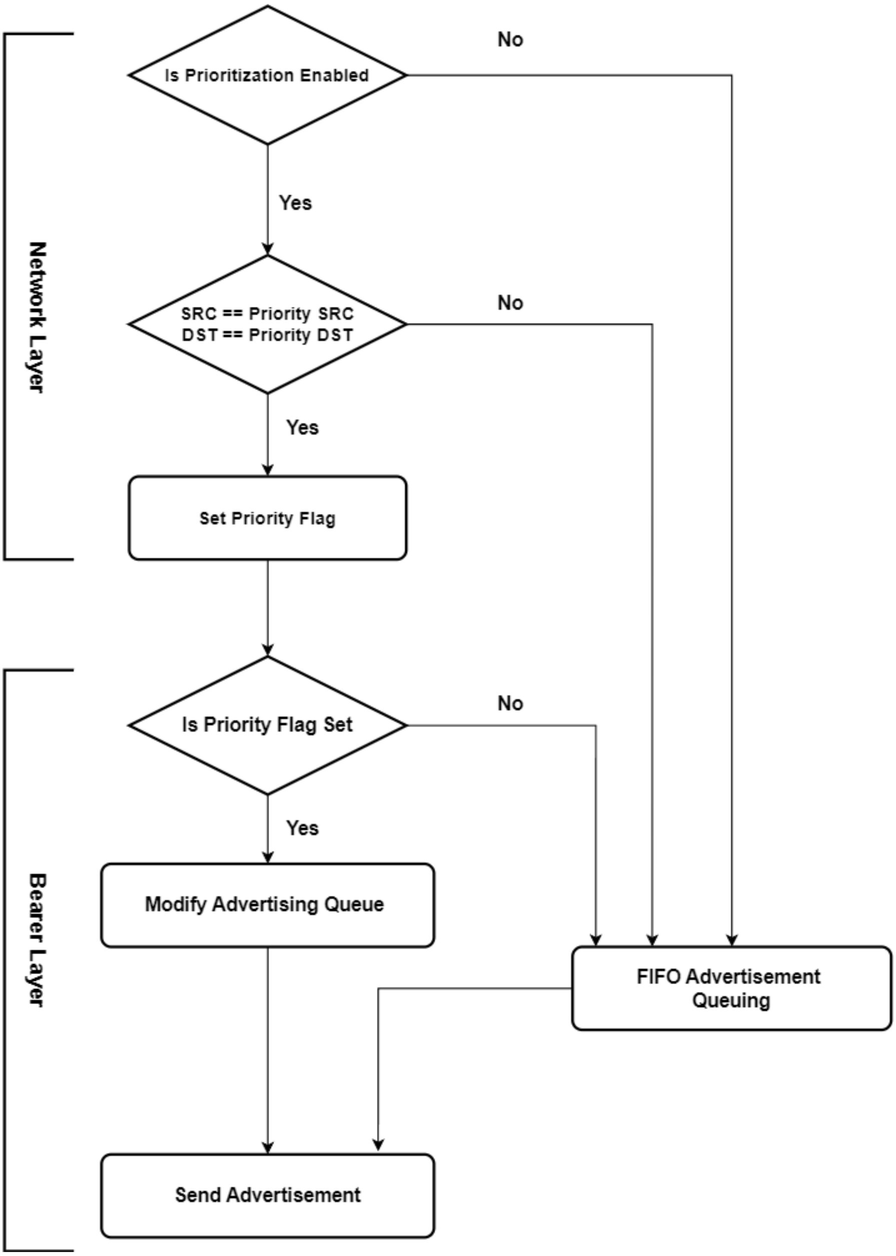


Figure 5.1: Flowchart Logic For SRC-DST Prioritization

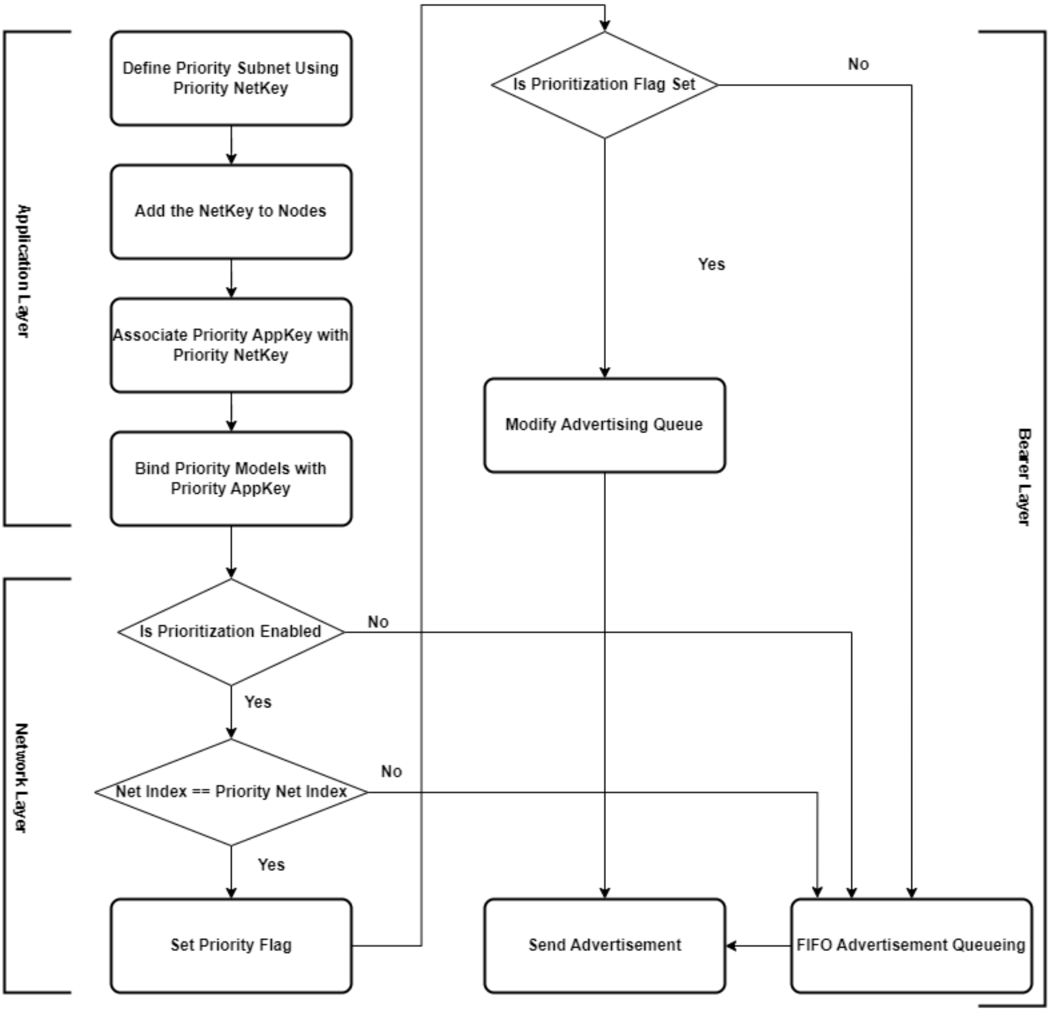


Figure 5.2: Flowchart Logic For Opcode Based Prioritization

The steps would be as follows:

1. Defining a priority subnet using a dedicated NetKey.
2. Add this NetKey to the relay nodes and the intended transmitter and receiver nodes so that they can be part of the dedicated priority subnet.
3. Add a "Priority AppKey" and associate this with the priority NetKey.

4. Bind the message models that need to be prioritized with this priority AppKey.

As it is possible to access the NetKey index in the relay nodes, now it would be possible to use an Opcode-based prioritization indirectly. Since every model has an associated opcode, this indirect model-based prioritization and opcode-based prioritization can be used interchangeably. For communication between two nodes that are using multiple models, one model can be prioritized over the others by binding that model with the priority AppKey and associating that AppKey with the Priority NetKey.

The modification of the advertising queue would be similar to the method for SRC-DST pair-based prioritization. Similar to that process a priority flag would be set, and priority buffers will be enqueued in the advertising queue ahead of non-priority buffers. Figure 5.2 shows the procedure for Opcode-based prioritization.

5.5 Quality of Service Metrics

The quality of service (QoS) metrics that are being considered to observe the impact of prioritization are

1. **Latency:** The difference between the time a packet is sent from the source node and the time that packet is processed at the destination node.
2. **Packet Delivery Ratio (PDR):** The ratio of transmitted packet count and processed packet count.

5.6 Test and Data Collection Procedure

5.6.1 BabbleSim Simulation

Running tests on Babblesim requires several steps. The first step is to write the machine code that will be run on the simulated boards. For the tests, the codes will be run on nRF52 series boards. The boards are part of the BabbleSim library. Then, the physical medium between the virtual devices has to be set. The 2.4 GHz physical channel was chosen since that is the channel in which BLE operates. BabbleSim provides an N-to-N attenuator with independent attenuations per path. With the help of this physical separation between nodes can be simulated. However, a "distance matrix" file is necessary. This is a text file that contains the attenuation value for every possible link in the test setup. So for a setup with six nodes, there are 15 unique links, and the distance matrix will have to contain attenuation values for both directions. The attenuation values are calculated using equation 2.1.

The implementation of the BTM mesh stack in the SDK has debug log options that can be manually configured. In order to measure latency, the debug logs for the access layer and the network layer needs to be enabled. After every simulation, the generated logs are saved. A Python script that has been prepared to parse these logs will calculate end-to-end latency and successful packet delivery ratio for any pair of source-destination nodes. For efficient and fast parsing regular expressions were used.

A 24-bit value contained in the SEQ field of the Network PDU is called the sequence number. One packet transmission between two intended nodes can be tracked with the help of this sequence number. The timestamps of the moment a mesh packet is sent and the moment that packet is decrypted in the target node is tracked, and the difference between them is recorded as the end-to-end latency. The parser also tracks if a packet is not processed at the target node and these events are considered to be an unsuccessful packet delivery. As the number of sent packets is known the PDR can be calculated from this information.

Prioritization can be enabled or disabled during runtime. In order to observe the impact and performance of prioritization tests have to be conducted for both cases. Keeping every other tunable parameter the same, tests will be done once without prioritization and once with prioritization. Recording the values of latency and PDR for each case will give insight into the performance of prioritization.

5.6.2 Test Network

For running the tests on the test network, first, the machine code needs to be written. Device Firmware Upgrade (DFU) of the boards must be done with the machine code. After DFU-ing power over Ethernet (POE), commands are sent from the remote PC. With the POE commands nodes can be controlled to act as noise generators that will broadcast messages to all of the nearby nodes. Controlling which nodes act as noise generators are done by passing a list during runtime. The list contains the MAC addresses of the intended noise generator devices. The noise generation interval and the noise packet amount are also passed. By setting a set amount of noise packets, it is ensured that the nodes do not continue to generate noise packets when the tests are finished. The noise generators will immediately start generating noise when they receive the command. Another script will control the transmitter-receiver pair. An Ethernet command will be sent to the intended transmitter node with the unicast address of the designated destination node. Transmission interval and the number of messages to be transmitted are also sent. For now, the priority addresses and the priority net index are hard coded into the machine code that is available to every node when the DFU is done. Enabling or disabling the prioritization is also hard coded. So, in order to change the prioritization address pair and the net index or to enable/disable prioritization the devices have to be DFU-ed again. After the transmitter node receives the command, it would start transmission. It will log the timestamp and the SEQ of each packet sent. The receiver node upon receiving every packet will also log the timestamp and the SEQ. These logs are sent via Ethernet to the remote PC where a Python script parses these logs. Packets are matched with the help of the SEQ, and the latency is calculated. If one SEQ is not found in the receiver log then that is considered as an unsuccessful transmission. In this way, the latency and PDR are calculated.

5.7 Validation of Hypothesis

To validate the two prioritization hypotheses simulations were conducted on Babblesim physical layer simulator with a small network of six nodes. The topology is shown in Figure 5.3.

5.7.1 Validation of SRC-DST Pair Based Prioritization

In this setup, there is one pair of transmitter and receiver. The transmitter node is denoted as NT and the receiver is denoted as NR. The nodes denoted as N are noise generators that will broadcast messages to all of the nodes within its radio range. Throughout this report, the term **noise** would mean background Bluetooth traffic. In the figure, the nodes connected by dotted lines are within each other's radio range. The source-destination pair is three hops away from each other.

NT will be transmitting messages every 100 milliseconds. The tests will be done for some high-noise scenarios. Latency and PDR will be recorded for both priority-enabled and disabled setups. To clarify latency and PDR will be calculated for the priority packets. NTC and RRC are set to zero. The results are recorded in Table 5.1 and 5.2.

Noise Generation Interval (ms)	Prioritization Disabled Latency (ms)	Prioritization Enabled Latency (ms)
30	481	21
40	241	18
50	41	17

Table 5.1: SRC-DST Based Prioritization: Average Latency (NT-NR) For Six Node Setup

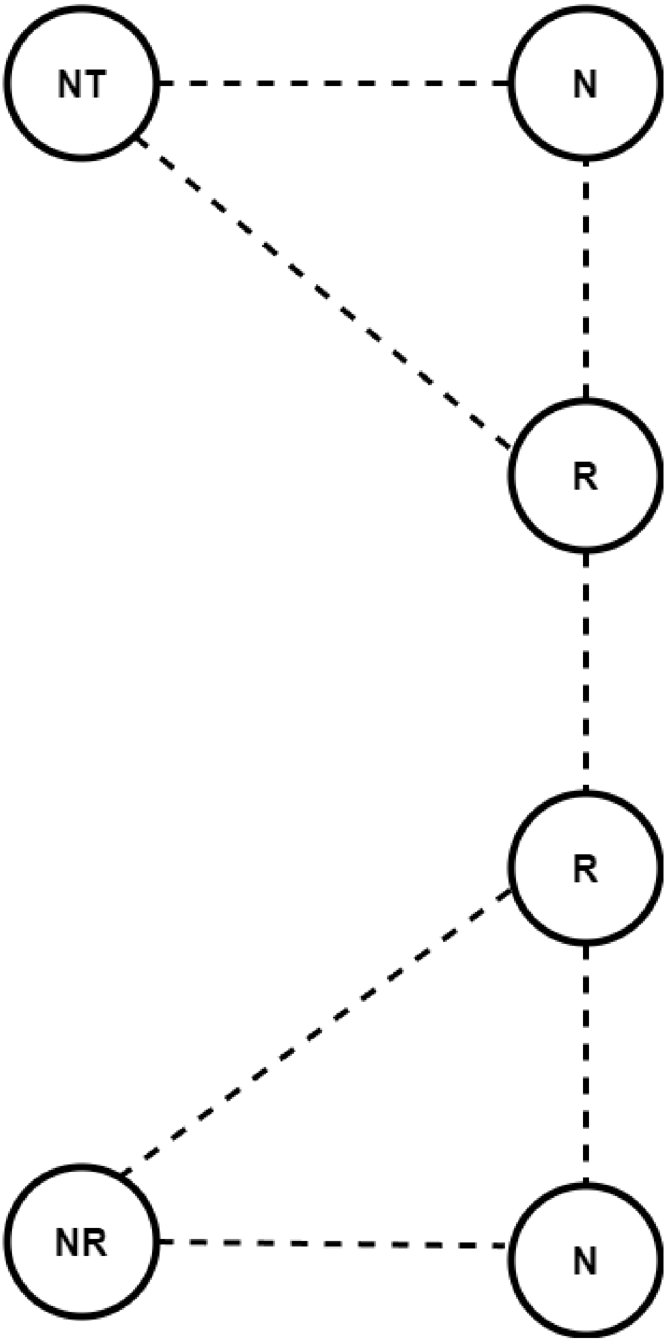


Figure 5.3: Six Node Setup For Validation

Noise Generation Interval (ms)	Prioritization Disabled PDR	Prioritization Enabled PDR
30	85%	65%
40	85%	85%
50	75%	70%

Table 5.2: SRC-DST Based Prioritization: PDR (NT-NR) For Six Node Setup

From the results it is evident that using the SRC-DST pair-based prioritization significantly improves end-to-end latency. However, when it comes to PDR, the results are inconclusive. But, when used in conjunction with NTC and RRC, it is expected that the PDR will improve.

5.7.2 Validation of Opcode Based Prioritization

To validate the Opcode-based prioritization hypothesis the same topology is being used. However, this time NT is transmitting two different test models to NR. The transmission is being done every 100 milliseconds. There are two noise generators and two relay nodes.

For the test one of the models will be prioritized over the other, and that model will be sent using the priority subnet. The latency and PDR will be recorded for the transfer of the priority model for prioritization-enabled and disabled setup. The NTC and RRC are set to zero.

Noise Generation Interval (ms)	Without Prioritization Latency (ms)	With Prioritization Latency (ms)
30	581	22
40	250	36
50	96	29
60	65	35

Table 5.3: Opcode-Based Prioritization: Average Latency For Priority Model (Six Node Setup)

Noise Generation Interval (ms)	Without Prioritization PDR in Percentage	With Prioritization PDR in Percentage
30	80	80
40	80	80
50	70	70
60	50	90

Table 5.4: Opcode-Based Prioritization: PDR For Priority Model (Six Node Setup)

From Table 5.3 and 5.4 it is clear that significant improvement in terms of latency can be achieved. But the PDR improvement is inconclusive. But in further testing, this will be tested in conjunction with NTC and RRC parameters. It is anticipated that the PDR will improve when used together with the protocol-related parameters.

From this six-node setup, both SRC-DST-based prioritization and Opcode-based prioritization methods have been validated. In the next chapter, more extensive simulations with complex topologies will be conducted. The effect of changing NTC and RRC will also be observed.

5.8 Challenges

5.8.1 For Simulation Setup

For BabbleSim simulation the main challenge was some limitations of the simulator. Creating topologies with physical separation between nodes requires creating a distance matrix file. Creating this file becomes challenging with the increase in network size.

5.8.2 For Test Setup

Before running the tests in the 100-node network, it was important to make the test stable for the test setup. For that, a small desk setup of four nodes shown in figure 5.4 was used for prototyping. Among the four devices in the desk setup, two were selected as noise generators, and two were selected as transmitters and receivers.

Unfortunately, the goal of running the test on the 100-node test setup was not possible to achieve. The test code showed some unexpected behavior. The logging mechanism was not working properly. Due to time constraints debugging the code and fixing these issues were not possible. Since the code was not stable for the desk setup, deployment to the 100-node network was too risky. The nodes in the test

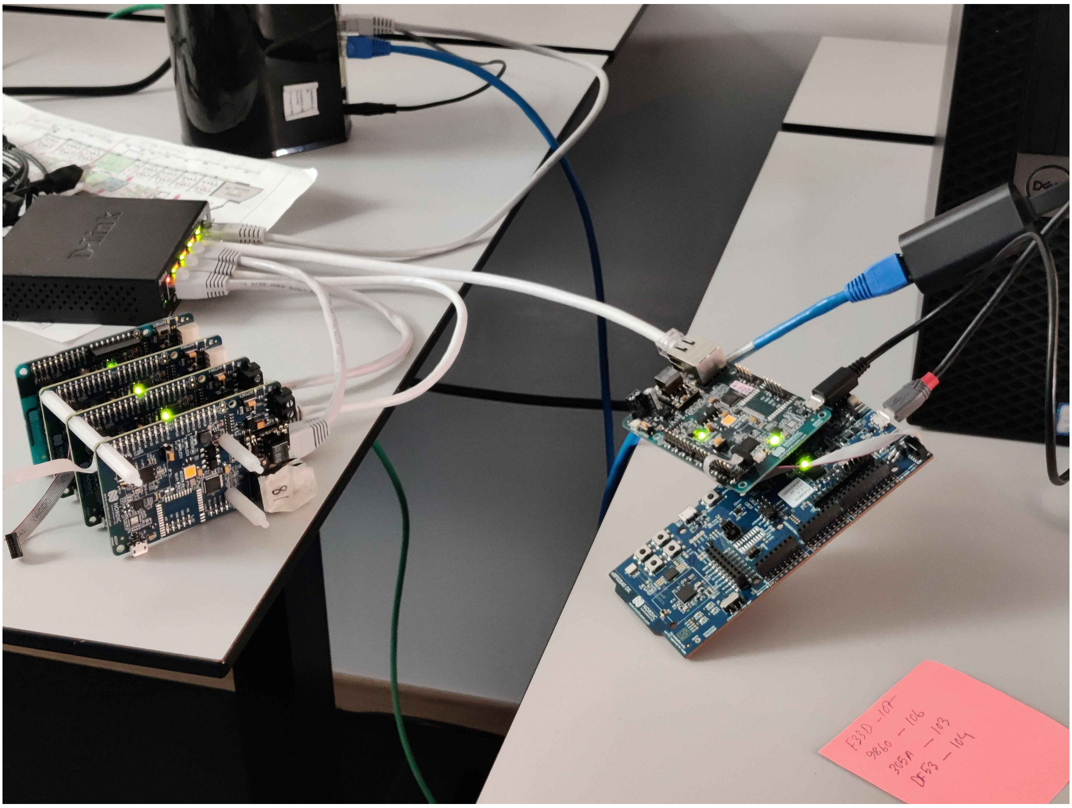


Figure 5.4: Four Node Desk Setup Source: Private Photo

network have mechanisms that lock the devices if they encounter any kind of assertion error during runtime. To fix them the nodes then have to be manually reset. Since employees in the Nordic Semiconductor office use the test network for their work and locking the devices by running unstable code would lead to problems for everyone, it was avoided. As a result, the tests were not conducted on the 100-node setup. Consequently, the impact of prioritization in a real environment setting was not possible to observe. However, the BabbleSim simulation, despite some limitations, provides valuable insight into the performance of the developed prioritization method. It is expected that the results will translate to real-world observations with a high degree of similarity.

Chapter 6

Test Setup

In the previous chapter, the two hypotheses were validated using a simple six-node setup. As the results were promising the simulation will be extended with a larger and more complex network. The network topology and different scenarios used for the simulations are described in this chapter.

6.1 Simulation Setup Layout

Figure 6.1 shows the floor layout of an office. It has multiple office rooms, meeting rooms, a kitchen, a bathroom and a commonplace. The nodes are physically separated from each other. There are concrete walls, and the physical separation between devices means there are devices that will be outside the radio range of each other. So, communication between them will depend on proper relay selection. The blue circles in Figure 6.1 show the nodes. There are a total of 20 nodes. This setup is the initial setup. Three different scenarios will be used in the simulation. They are

1. Setup 1: Noise generators crowding the transmitter
2. Setup 2: Noise generators crowding the receiver
3. Setup 3: Noise generators crowding the relays

6.2 Setup 1: Noise Generators Crowding the Transmitter

In this setup, the noise generator nodes will be positioned closer to the transmitter node. Figure 6.2 shows the simulation topology.

NT denotes the transmitter node, NR denotes the receiver node. R denotes the relays, and N denotes the noise generators. When the simulation runs the relay nodes will receive packets from all of the nodes within its radio range. The impact of prioritization from this scenario can be observed. When prioritization is enabled,

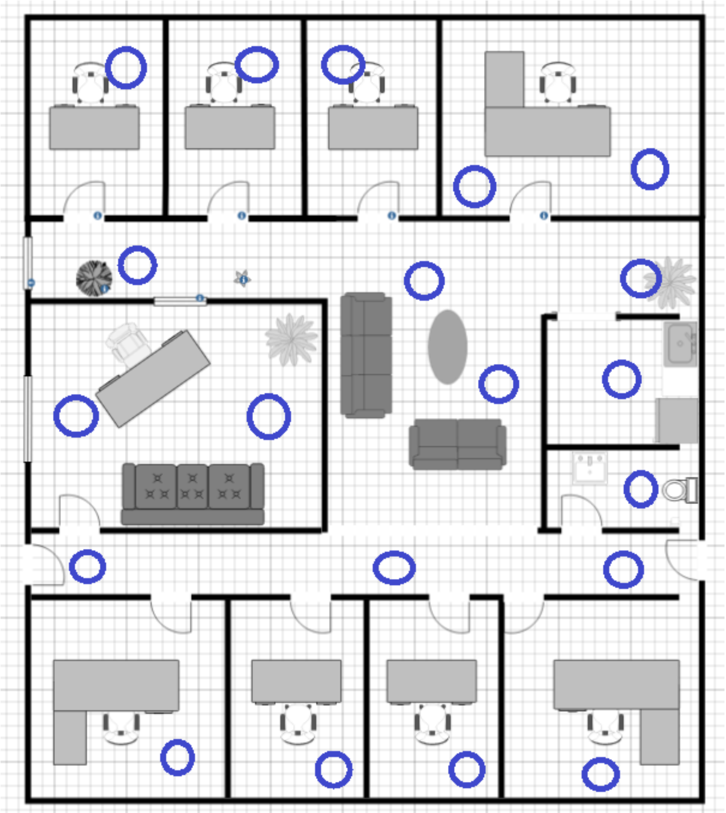


Figure 6.1: Office Floor Plan With Bluetooth Mesh Nodes Used for Simulations

the relay nodes will prioritize packets from the transmitter node. The results would indicate how prioritization behaves when the noise traffic is high around the transmitter.

6.3 Setup 2: Noise Generators Crowding the Receiver

Figure 6.3 shows the network topology when the noise generators are positioned around the receiver node. Similar to the previous setup the impact of prioritization when noise traffic is high around the receiver will be observed from this setup.

6.4 Setup 3: Noise Generators Around Relays

Relay nodes are very important for the prioritization method. When they are crowded by noise generators meaning when the traffic is high around them, the behavior can be observed from this setup. The topology is shown in Figure 6.4.

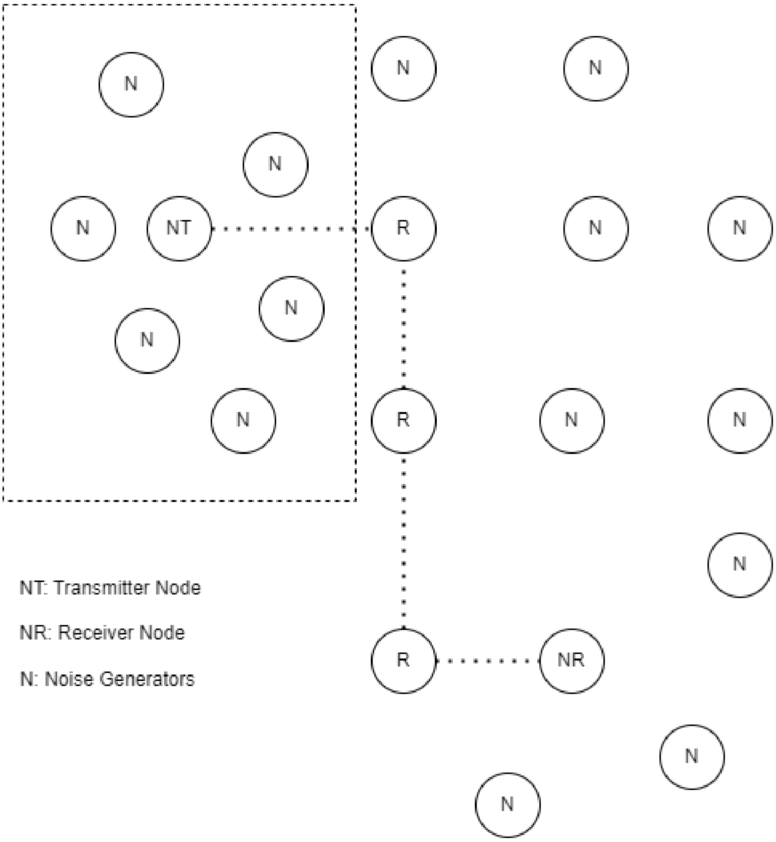


Figure 6.2: Noise Generators Around Transmitter

6.5 Simulation Parameters

In the simulation by default, the values of Network Transmit Count (NTC) and Relay Retransmit Count (RRC) are set to zero. But different combinations of NTC and RRC will be tried, and their impact will be observed.

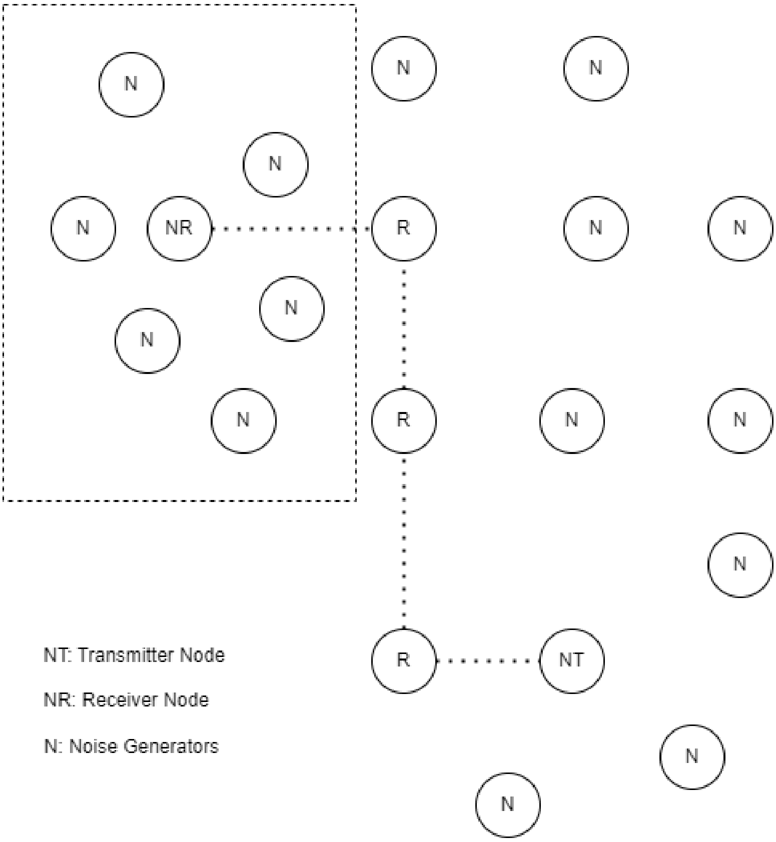


Figure 6.3: Noise Generators Around Receiver

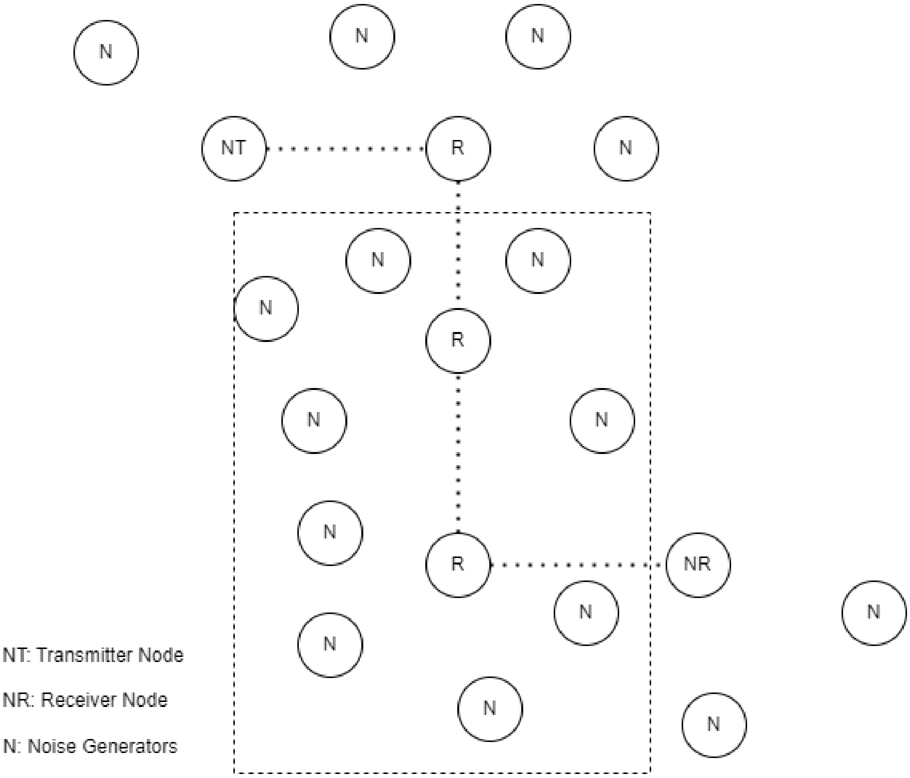


Figure 6.4: Noise Generators Around Relays

Chapter 7

Results

In this chapter, the results of the simulations for the different scenarios described in the previous chapter will be shown.

7.1 Simulation Preset

As shown in the different scenarios described in the previous chapter, there are 20 nodes in the setup. For testing purposes, one pair of transmitters and receivers has been chosen for the work while the rest of the nodes are acting as noise generators.

As shown in Figure 7.1 the NT is the transmitter node and NR is the receiver node. The transmitter and receiver are set 4 hops away from each other. There are three relay nodes denoted by R. The analysis of the impact of prioritization for this pair will be observed. The nodes labeled N are the noise-generating nodes. They are broadcasting messages to all nodes within their radio range. The transmitter is sending packets every 100 milliseconds. There is a small amount of randomization introduced for the 100 milliseconds transmission interval.

The number of advertising buffers and relay buffers was set to 32. A high number of advertising buffers increases the reliability of the mesh network. A low number of advertising buffers reduces the message latency on the Relay Node but at the same time increases the amount of packet drops. Because of this choice of buffer count, it is expected that for high noise/traffic, the latency value would be quite high. But this will help with observing the impact of prioritization more clearly.

In order to test the Opcode-based prioritization three test models were included in the simulation setup. Each model has an associated opcode with them.

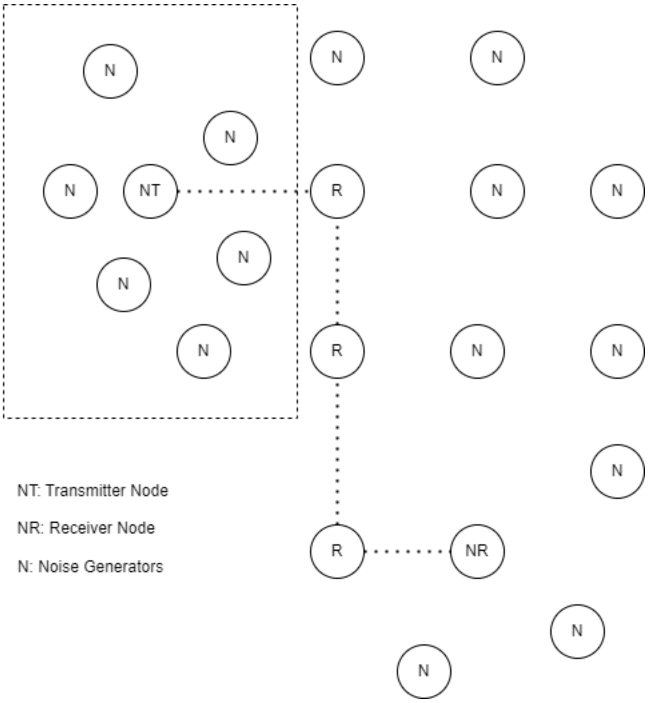


Figure 7.1: Simulation Setup: Noise Generators Around the Transmitter

Calculation of end-to-end latency also referred to as just latency and packet delivery ratio (PDR) will be done only for the prioritized packets unless specified otherwise.

For the noise generators, they will generate noises at different rates. For simulation, the rate will start from a noise generation rate of 30 milliseconds per noise packet to 120 milliseconds with a ten milliseconds increment. This value also includes a small amount of randomization. Each noise generation interval forms a distinct simulation case. The simulations will be run with different sets of NTC and RRC pair. For NTC and RRC, the re-transmission interval is kept to the default value of twenty milliseconds.

Tests for SRC-DST-based prioritization will be done using all three setups. In order to understand the impact of noise localization around the transmitter, receiver and relay nodes all of these setups are being used. For the Opcode-based prioritization test, setup 1 will be used, and three different test models will be used in that test.

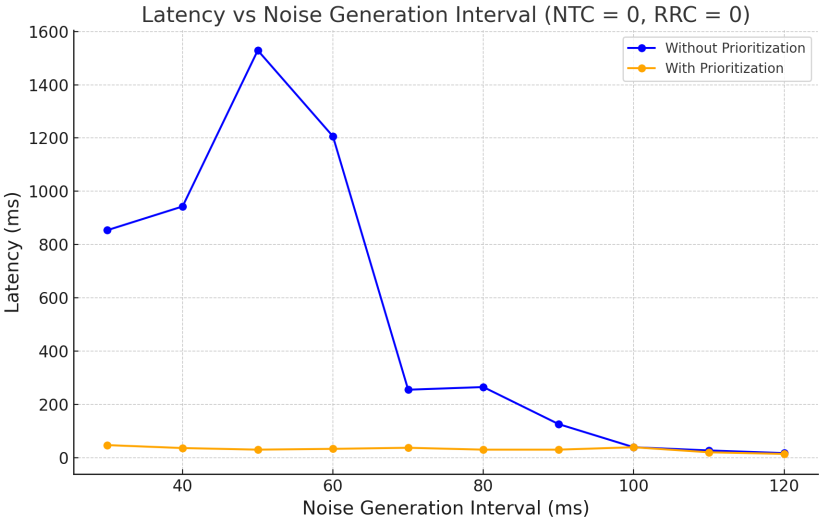


Figure 7.2: Setup 1: Latency vs Noise Generation Interval (NTC = 0, RRC = 0)

7.2 SRC-DST Pair Based Prioritization

7.2.1 Setup 1: Noise Generators Crowding the Transmitter

From Figure 7.1 it can be seen that six noise generators are positioned around the transmitter node. These nodes will broadcast messages to all of the nodes that are within their radio range. As the transmitter node is positioned close to them, it will experience a very high traffic/noise scenario.

Simulation Results for NTC = 0, RRC = 0

Figure 7.2 and Figure 7.3 show the latency and packet delivery ratio (PDR) respectively. The blue lines represent values obtained without using SRC-DST-based prioritization, and the orange line represents values obtained by using SRC-DST-based prioritization. It is obvious looking at the figures that using prioritization significantly improves the latency for the prioritized packets. However, when it comes to improving PDR the impact of prioritization is inconclusive.

Simulation Results for NTC = 1, RRC = 1

Network Transmit Count (NTC) and Relay Retransmit Count(RRC) are parameters in Bluetooth Mesh that can be used to improve the reliability of the network. How-

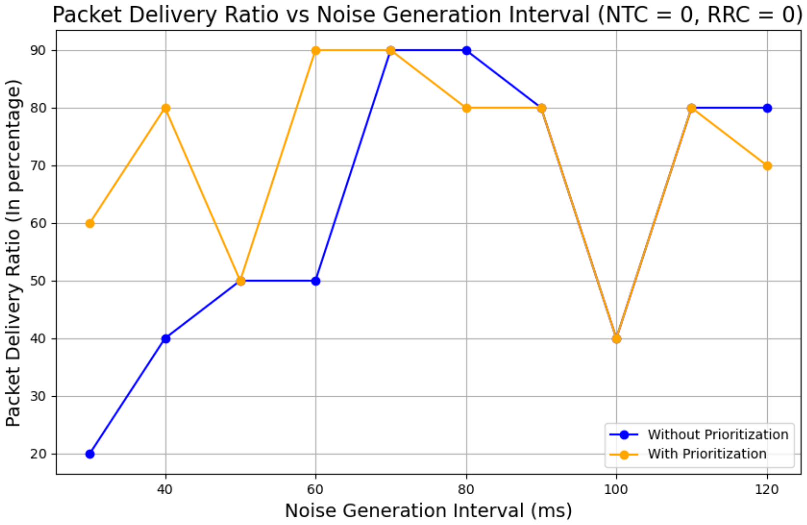


Figure 7.3: Setup 1: PDR vs Noise Generation Interval (NTC = 0, RRC = 0)

ever, increasing the values of NTC and RRC will also increase the network traffic and cause congestion. This will result in higher latency but should improve successful PDR.

Figure 7.4 and 7.5 show the latency and the PDR respectively when NTC and RRC are set to 1. From the figures, it can be seen that increasing the NTC and RRC values significantly increased the network latency. As the network traffic increased and due to the presence of the noise generators, the PDR did not improve despite increasing NTC and RRC. However, when prioritization is enabled, the latency stabilizes to a much-improved range. Significant improvement in PDR is observed as well.

Simulation Results for NTC = 2 RRC = 2

To obtain even better reliability NTC and RRC have been increased to 2. But this will also increase the overall traffic and cause congestion. But used in conjunction with prioritization, the results from Figure 7.6 and Figure 7.7 indicate that it is possible to achieve a very high level of successful PDR and a lower and stable latency.

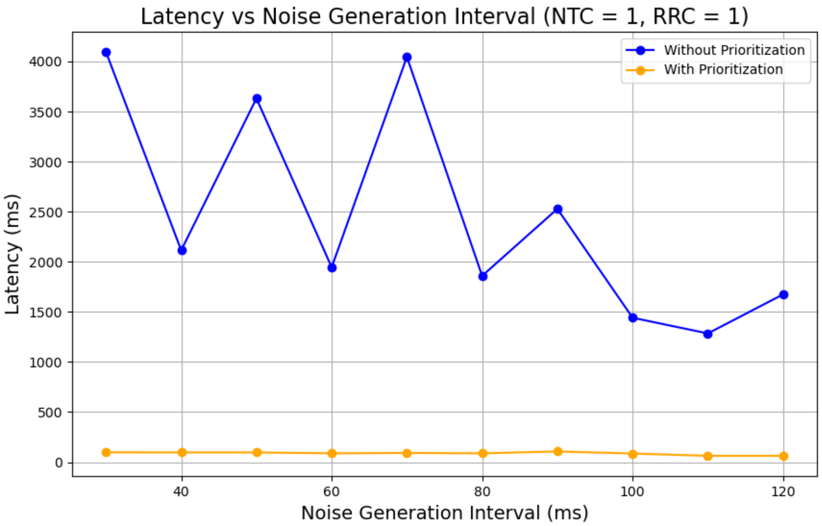


Figure 7.4: Setup 1: Latency vs Noise Generation Interval (NTC = 1, RRC = 1)

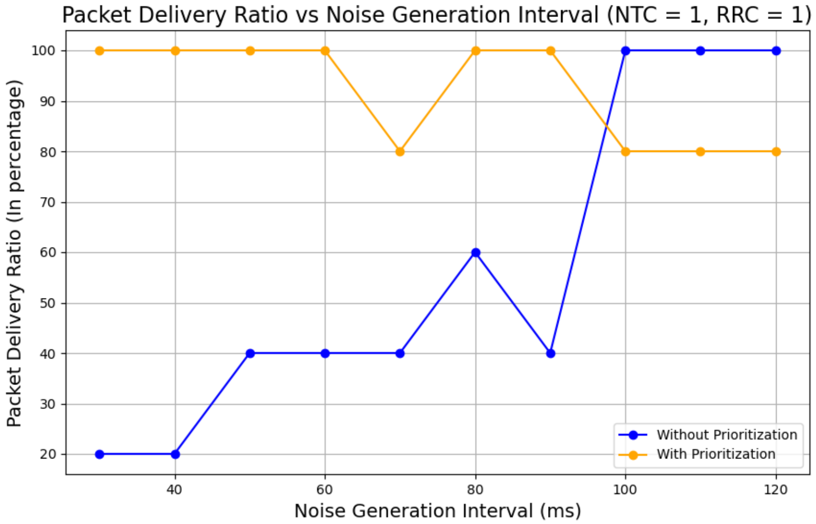


Figure 7.5: Setup 1: PDR vs Noise Generation Interval (NTC = 1, RRC = 1)

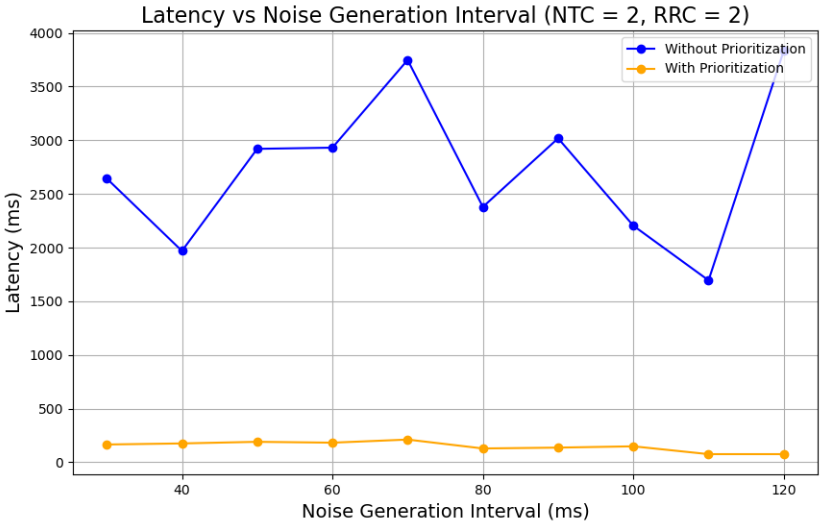


Figure 7.6: Setup 1: Latency vs Noise Generation Interval (NTC = 2, RRC = 2)

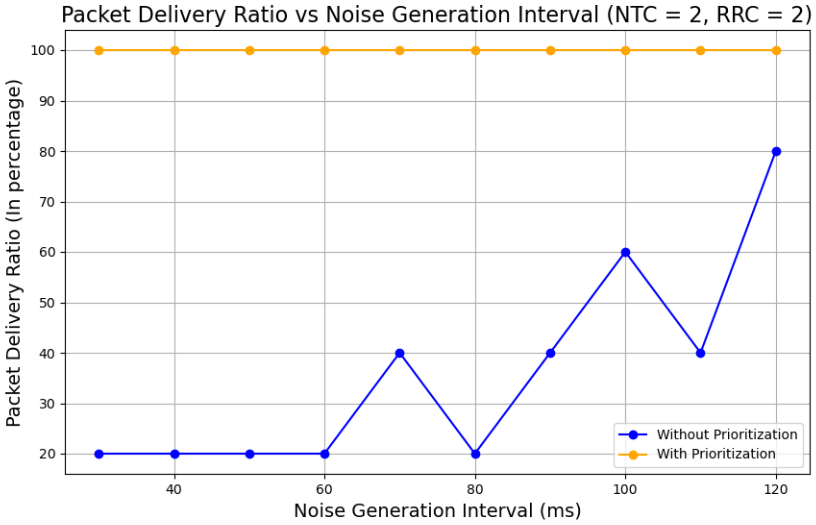


Figure 7.7: Setup 1: PDR vs Noise Generation Interval (NTC = 2, RRC = 2)

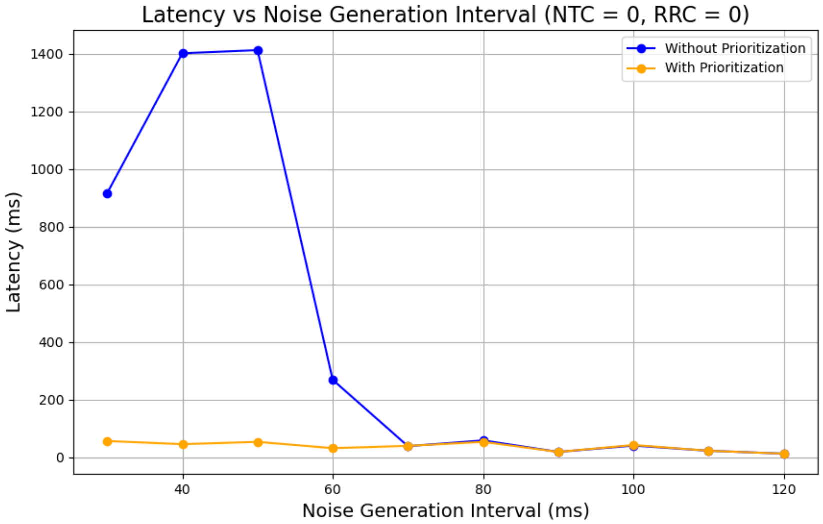


Figure 7.8: Setup 2: Latency vs Noise Generation Interval (NTC = 0, RRC = 0)

7.2.2 Setup 2: Noise Generators Crowding the Receiver

From Figure 6.3, six noise generator nodes are positioned around the receiver. Similar to setup 1, they will broadcast messages to all of the nodes that are within their radio range. As the receiver is positioned among these nodes, it will experience heavy traffic.

Simulation Results for NTC = 0, RRC = 0

Figure 7.8 and Figure 7.9 show latency and PDR vs the noise generation interval respectively. Similar to the previous setup significant improvement in terms of latency has been achieved. However, the PDR improvement is inconclusive.

Simulation Results for NTC = 1, RRC = 1

Unlike the scenario where the noise generators are positioned around the transmitter, in this scenario, it was possible to achieve significant improvement in PDR by using a smaller value of NTC and RRC. A stable and lower range of average latency is also observed. The results are shown in Figure 7.10 and Figure 7.11.

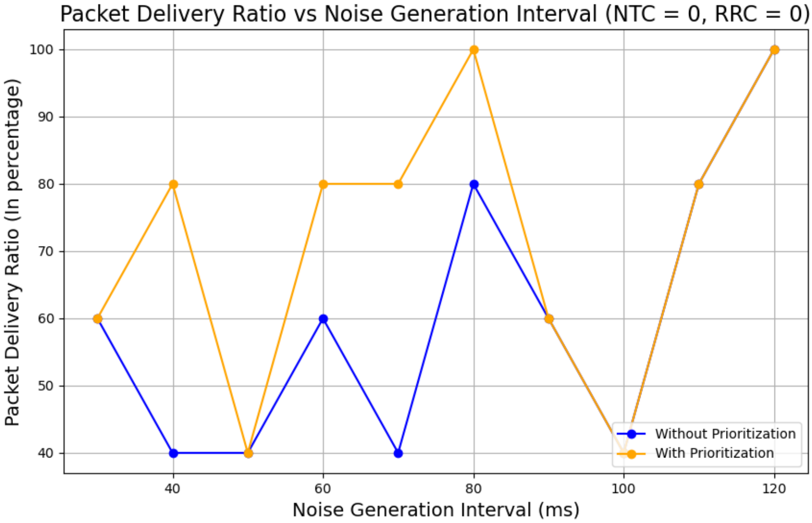


Figure 7.9: Setup 2: PDR vs Noise Generation Interval (NTC = 0, RRC = 0)

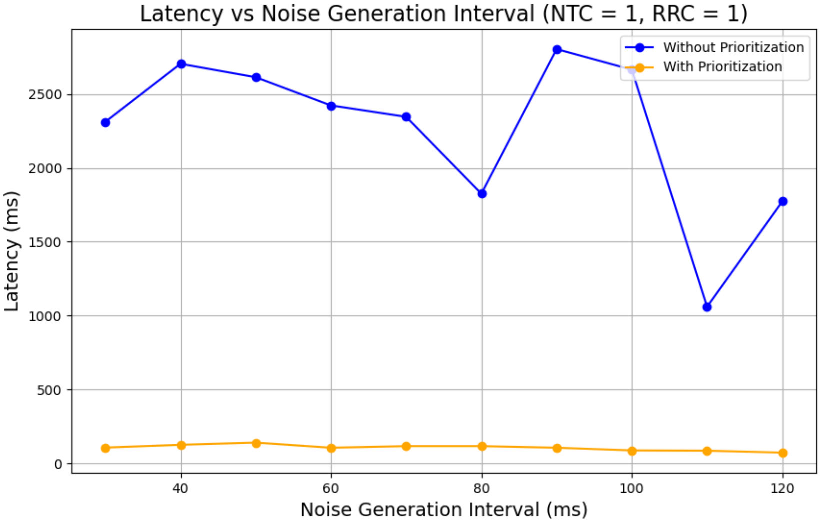


Figure 7.10: Setup 2: Latency vs Noise Generation Interval (NTC = 1, RRC = 1)

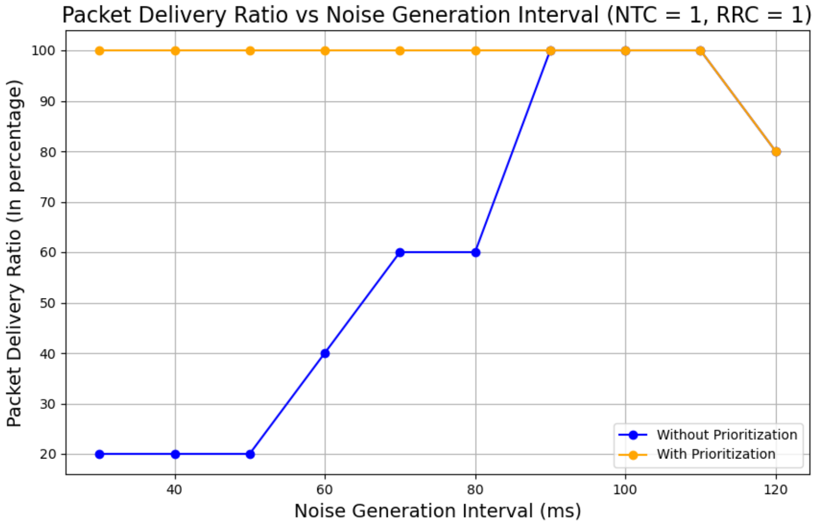


Figure 7.11: Setup 2: PDR vs Noise Generation Interval (NTC = 1, RRC = 1)

7.2.3 Setup 3: Noise Generators Around Relays

When it comes to prioritizing Bluetooth mesh messages, the relay nodes are of utmost importance. In order to see how the high noise rate around the relay nodes impacts the network. The setup is shown in Figure 6.4. In this setup, there are three relays and several noise generator nodes positioned around the relay nodes.

Simulation Results for NTC = 0, RRC = 0

Figure 7.12 and Figure 7.13 show the network latency and PDR respectively. For a high noise generation rate the average latency is high. But when prioritization is enabled, very good latency values are achieved, and the range is very stable as well. Just like the previous simulation with NTC and RRC set to zero, the impact of prioritization in PDR improvement is inconclusive.

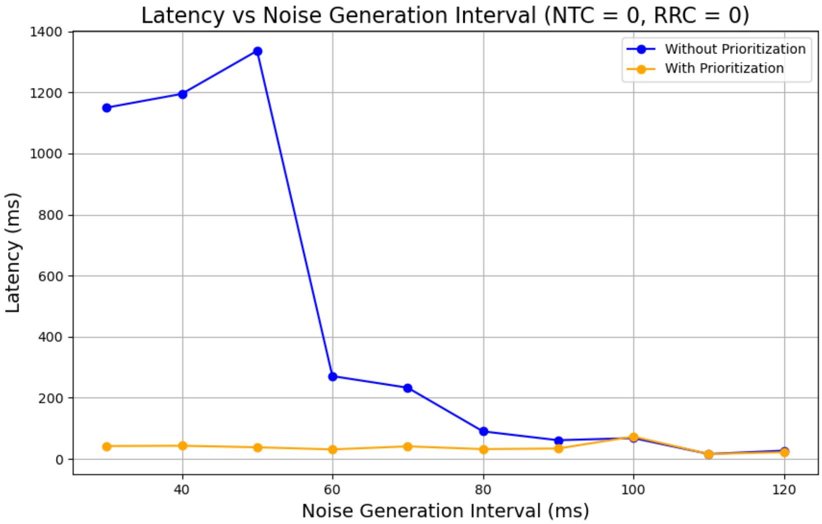


Figure 7.12: Setup 3: Latency vs Noise Generation Interval (NTC, RRC = 0)

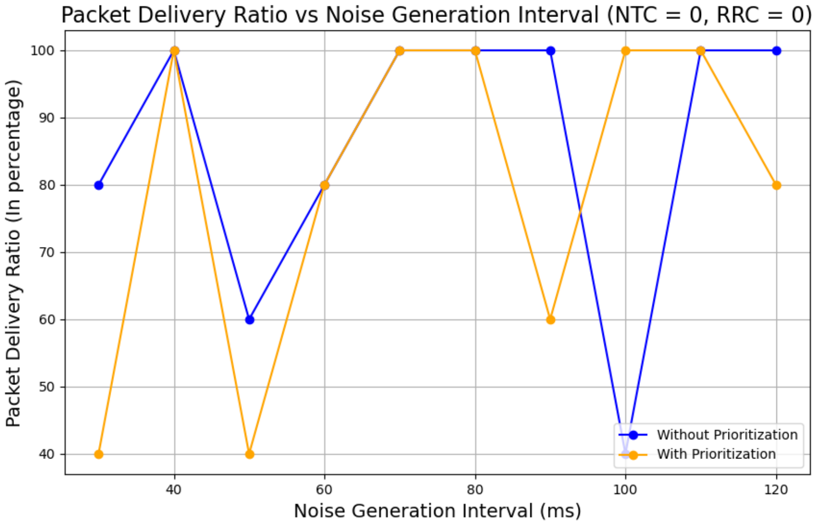


Figure 7.13: Setup 3: PDR vs Noise Generation Interval (NTC, RRC = 1)

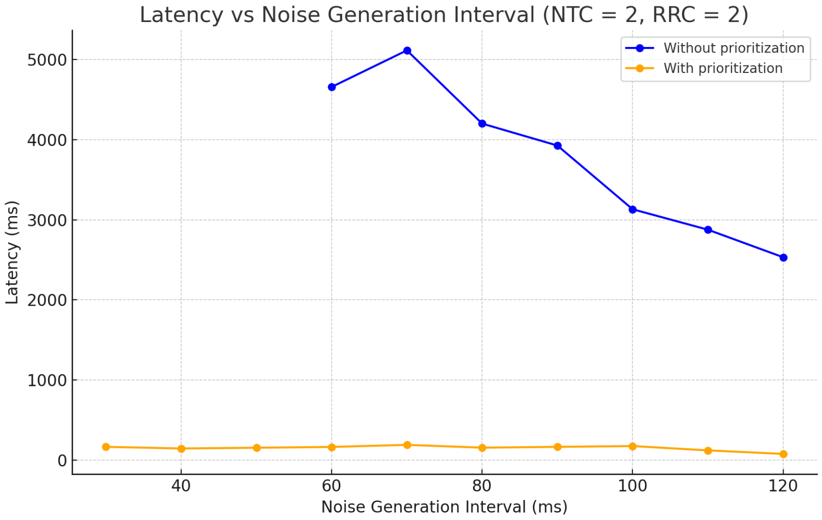


Figure 7.14: Setup 3: Latency vs Noise Generation Interval (NTC, RRC = 2)

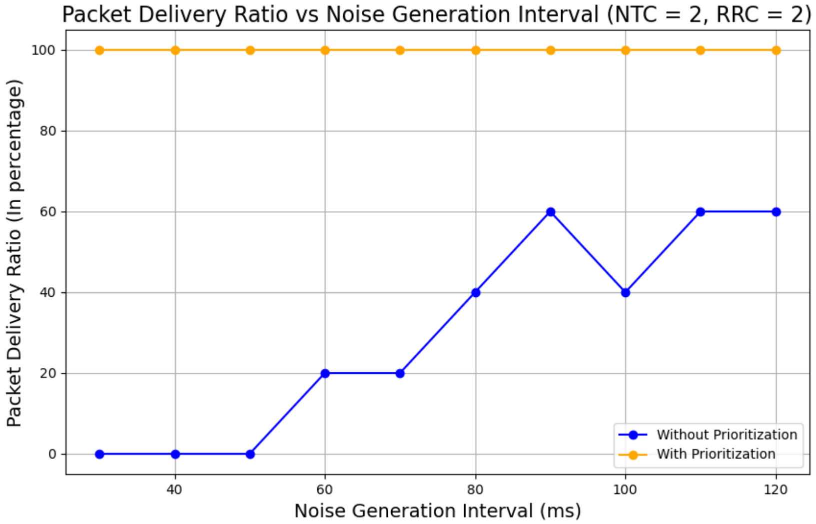


Figure 7.15: Setup 3: PDR vs Noise Generation Interval (NTC, RRC = 2)

Simulation Results for NTC = 2, RRC = 2

Since the relay nodes are experiencing high noise, RRC is increased to 2, and the impact can be seen in Figure 7.14 and Figure 7.15. After increasing the NTC and RRC, for very high noise generation rates the PDR was zero percent when prioritization was disabled. For these cases, the latency values were not available in Figure 7.14. But by enabling prioritization, 100 percent PDR was achieved even for those cases.

7.3 Opcode-Based Prioritization

In order to test the impact of Opcode-based prioritization setup 1 shown in Figure 6.2 is being used. There are six noise generators positioned around the transmitter node NT. NT is transmitting three different test models to NR every 100 milliseconds. These models have associated opcodes with them and by the procedure shown in Figure 5.2 one of the test models would be prioritized in the communication. It can be assumed that test model 1 is necessary for a time-critical operation that needs to be transported to the receiver node NR with low latency and high reliability. As there are multiple models being transmitted between the same source and destination address, using SRC-DST-based prioritization here will not be beneficial. So, in order to improve performance Opcode-based prioritization has been applied.

Simulation Results for NTC = 0, RRC = 0

Figure 7.16 and 7.17 show the network latency and PDR respectively for the prioritized test model. It is clear from the results that enabling prioritization significantly improves network latency. However, when NTC and RRC are zero the improvement in PDR is inconclusive.

Simulation Results for NTC = 2, RRC = 2

From Figure 7.18 and 7.19 it can be concluded that increasing the NTC and RRC values to two increases network latency significantly when prioritization is disabled. But by enabling Opcode-based prioritization the latency for the priority model can be reduced significantly, as well as increasing the PDR substantially.

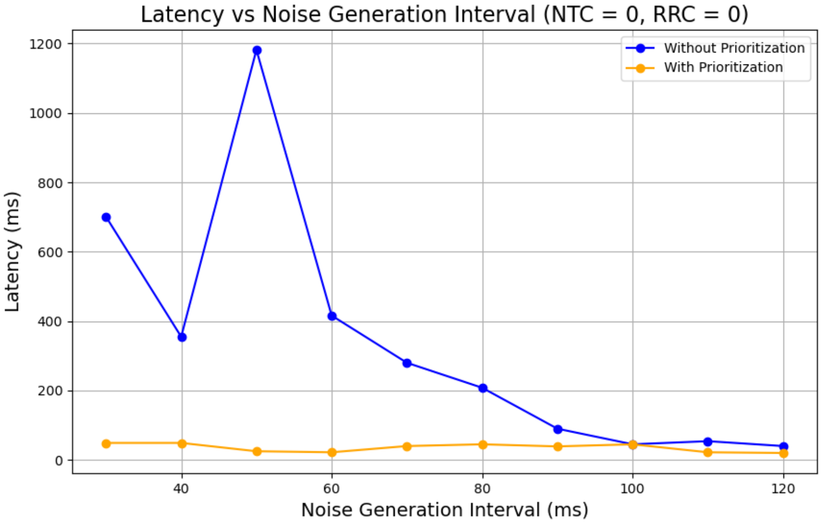


Figure 7.16: Opcode Based Prioritization: Latency vs Noise Generation Interval (NTC = 0, RRC = 0)

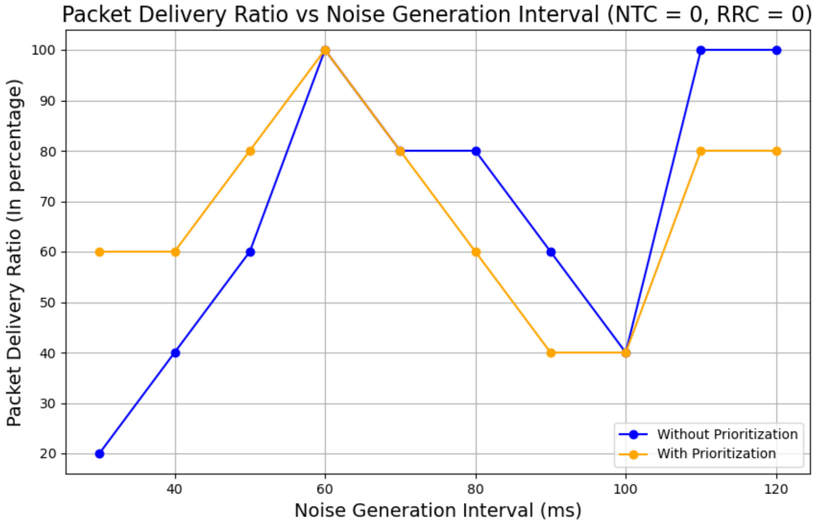


Figure 7.17: Opcode Based Prioritization: PDR vs Noise Generation Interval (NTC = 0, RRC = 0)

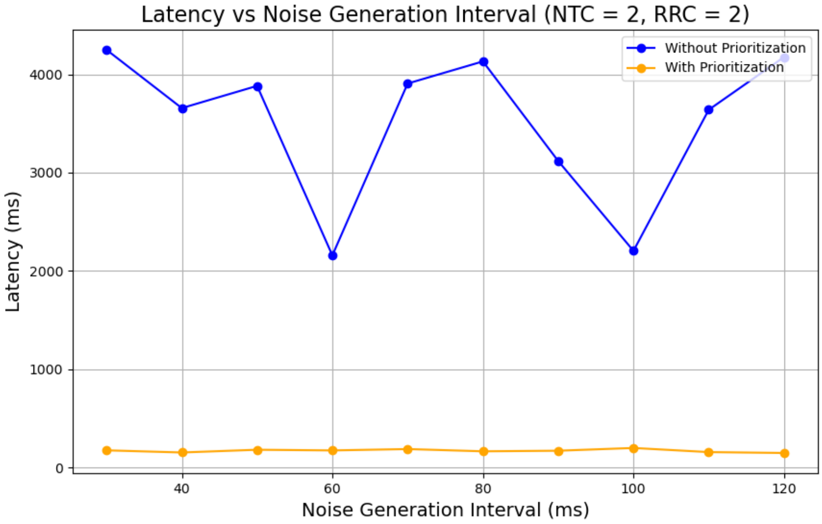


Figure 7.18: Opcode Based Prioritization: Latency vs Noise Generation Interval (NTC = 2, RRC = 2)

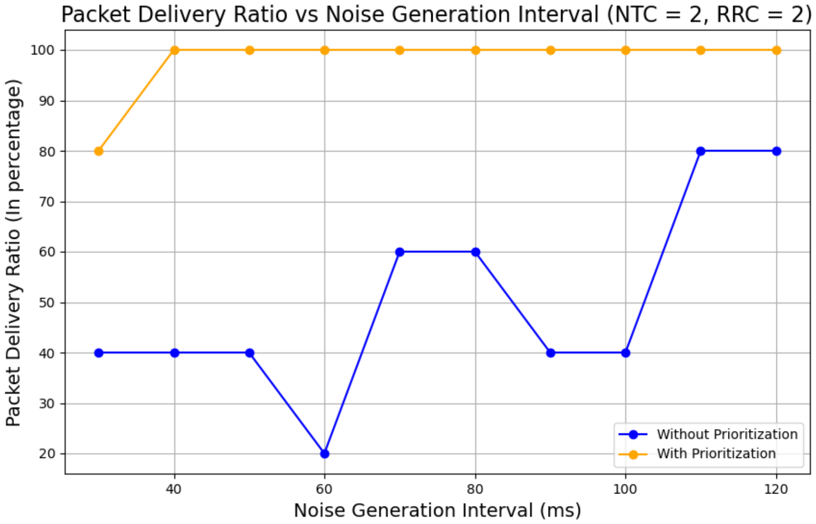


Figure 7.19: Opcode Based Prioritization: PDR vs Noise Generation Interval (NTC = 2, RRC = 2)

Chapter 8

Discussion

The goal of this thesis was to observe the impact of prioritization of Bluetooth mesh messages. How prioritization impacts the QoS, such as latency and packet delivery ratio, was the main goal. Extensive simulations were conducted considering different network topologies and scenarios. The results were included in the previous chapter. Key takeaways from those results will be discussed here.

8.1 Choice of Prioritization Method

Two different types of prioritization methods were developed in this thesis. The SRC-DST pair-based prioritization and Opcode-based prioritization. Both of them share the underlying property of modifying the FIFO advertising queue. The choice of which prioritization method to use depends on the use cases. For transmission between a set pair of source-destination devices where all available models must be sent with low latency and high reliability the SRC-DST pair-based prioritization would be the ideal choice. In events where a certain type of message containing a time-critical model, for example when a generic on-off model must be sent to the target node with high QoS metrics the Opcode-based prioritization would be the best choice. As it would ensure the delivery of the priority model even in scenarios where multiple types of models are being exchanged between the same pair of nodes.

8.2 Node Selection

In order to get the best possible results from the developed prioritization methods, it is important to choose the source and destination nodes properly. If the nodes are just one hop away from each other, then using prioritization is unnecessary. To show how the latency values look like, SRC-DST-based prioritization is applied between two nodes that are one hop distance away from each other. The receiver node in Figure 6.2 was moved and placed at a distance that is within the radio range of the transmitter node NT. Table 8.1 shows the average latency for some high-noise scenarios.

Noise Generation Interval (ms)	Average Latency (ms) (Without Prioritization)	Average Latency (ms) (With Prioritization)
30	3.28	3.28
40	2.97	2.97
50	2.94	2.94
60	3.04	3.04

Table 8.1: Average Latency For One Hop Communication

From the values shown in Table 8.1 it is clear that for nodes that are one hop away using prioritization is redundant.

8.3 Relay Selection

Relay selection is very important to ensure successful packet delivery. In all of the different scenarios described in the previous chapter, the relay nodes were selected in such a way that it was possible for the source and destination nodes to communicate through the relays. Without proper relay selection prioritization would not yield any beneficial result.

8.4 Impact on Non-Prioritized Nodes and Models

In the implemented prioritization methods the FIFO advertising queue gets modified if the prioritization condition is fulfilled. For SRC-DST-based prioritization, This results in an improved latency value for the prioritized set of source-destination pair. But since the buffer is being modified, intuition tells that the latency for the non-prioritized source-destination pair should be impacted in some way. Similarly for Opcode-based prioritization, non-prioritized models will experience a decline in QoS. To test out this for SRC-DST pair-based prioritization one extra set of transmitter-receiver pair was included in the network topology shown in Figure 8.1.

These are shown as NT_2 and NR_2 in the figure. They are two hops away from each other and use one of the relays. The NTC and RRC values are set to zero for this test, and NT_2 is generating packets every 100ms. The average latency between them for high-noise scenarios is shown in Table 8.2. The latency values are rounded to the nearest integer.

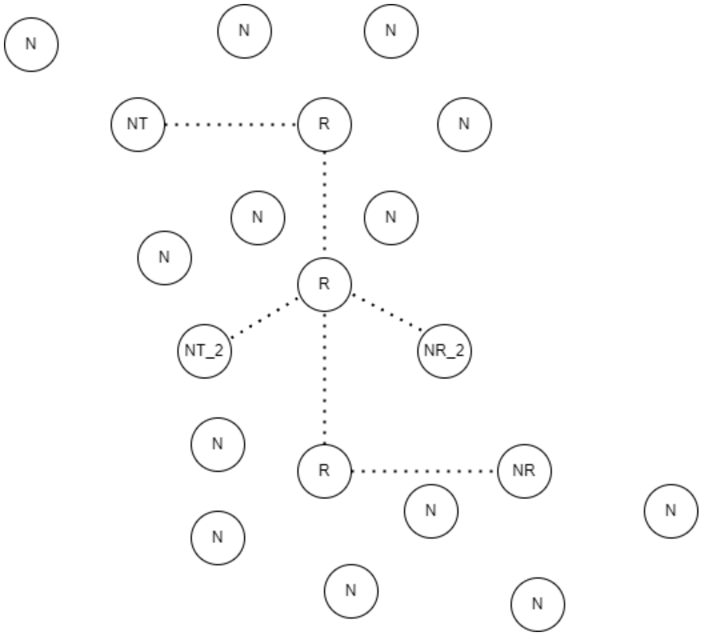


Figure 8.1: Topology With Additional Transmitter Receiver Pair

Noise Generation Interval (ms)	Average Latency (ms) (Without Prioritization)	Average Latency (ms) (With Prioritization)
30	659	718
40	410	475
50	308	378
60	252	306
70	68	104

Table 8.2: Average Latency For Non-Prioritized Pair of Transmitter and Receiver

From the table, it can be seen that when prioritization is enabled, the latency increases for a set of non-prioritized transmitter-receiver pair. Similarly in Opcode-based prioritization latency will increase for non-prioritized models.

8.5 Comparison Between Three Different Setups

Three different network setups were used in the simulations:

1. Noise Generators Crowding Transmitter
2. Noise Generators Crowding Receiver
3. Noise Generators Around Relays

They are shown in Figure 6.2, 6.3 and 6.4.

In all three setups, the position of the transmitter, receiver and relay nodes are constant. The noise generator nodes were moved around. The impact of crowding the noise generators around the transmitter, receiver and relay nodes offers some useful insight.

For the same values of NTC and RRC, the latency for some high-noise scenarios is shown in Table 8.3. These values were obtained when prioritization was enabled. The values are quite close to each other. Due to randomness introduced in the machine code and also randomness built into the nRF Connect SDK some differences in these cases can be expected. However, it is clear that the values are quite close. In the tests, the source and destination are four hops away from each other, and by enabling prioritization, it was possible to obtain a minimal amount of latency for the four-hop communication.

Noise Generation Interval (ms)	Setup 1 Latency (ms)	Setup 2 Latency (ms)	Setup 3 Latency (ms)
30	47	57	42
40	36	46	43
50	30	54	38
60	33	32	31
70	37	40	41

Table 8.3: Average Latency For Three Setups (NTC = 0, RRC = 0)

8.6 Prioritization and Protocol Related Parameters

As mentioned previously, the NTC and RRC allow BTM networks to achieve high reliability. However, since increasing the NTC and RRC also increases the number of transmissions the overall network traffic will increase. For a high noise scenario, this can cause network congestion and decrease the QoS. But when used together with the SRC-DST pair-based prioritization, the network reliability in terms of successful PDR can be improved, and the latency can be kept to a low and acceptable value. From the simulations this pattern is obvious.

Noise Generation Interval (ms)	Without Prioritization PDR	With Prioritization PDR
30	0%	100%
40	0%	100%
50	0%	100%
60	20%	100%
70	20%	100%
80	40%	100%

Table 8.4: PDR Improvement: Prioritization + Protocol-Related Parameters

For example Table 8.4 shows the improvement of PDR when SRC-DST-based prioritization was applied in setup 3. The NTC and RRC are set to 2, and the results are from high-noise scenarios. When prioritization was disabled, the PDR was very poor. But with the help of prioritization the improvement in PDR can be seen clearly.

For the same setup, the improvement in terms of latency is described in Table 8.5. The values are rounded to the nearest integer.

Noise Generation Interval (ms)	Without Prioritization Latency (ms)	With Prioritization Latency (ms)
30	NA	164
40	NA	144
50	NA	153
60	4660	163
70	5117	189
80	4202	154

Table 8.5: Latency Improvement: Prioritization + Protocol-Related Parameters

Chapter 9
Conclusion

The Bluetooth mesh network is capable of providing very high reliability when relays are selected properly, and the network traffic and noises are low. However, from the simulation results, it is observed that when network traffic and noise are high, the network performance reduces. In time-critical operations, this can present unforeseen challenges.

Using the two developed prioritization methods can solve these issues for time-critical scenarios. Only using the prioritization method offers a significant improvement in terms of end-to-end latency. Using this together with the protocol-related parameters such as NTC and RRC allows the network to obtain both very good and stable end-to-end latency alongside reliability in terms of a successful packet delivery ratio of greater than 99.99%.

From the various different scenarios that have been tested in the simulations it is clear that prioritization can even nullify the impact of localization of noise.

However there are some recommendations for using these prioritization methods:

1. **Appropriate Selection of Prioritization Method:** The choice of the appropriate prioritization method will depend on use cases. For transmission of priority models between the same set of source and destination pair Opcode-based prioritization would be preferable, and for prioritizing a set of transmitters and receivers, SRC-DST-based prioritization would be preferable.
2. **Proper Relay Selection:** Relay selection is paramount for successful message delivery. If the relay nodes are not properly selected, prioritization would not be effective as the messages would not be able to reach the desired destination.

3. **Source and Destination Node Selection:** For one-hop communication using prioritization is redundant. So, it should only be used for nodes that will require multi-hop communication.
4. **Drawback and Proper Usage:** Prioritizing one pair of nodes or a specific model/opcode will cause an increase in latency for the non-prioritized ones using the same relay nodes. So, this must be kept in mind while using prioritization and the recommendation would be to use it only for time-critical communication.
5. **NTC and RRC Usage:** Using an increased value of NTC and RRC alongside the developed prioritization methods would be the recommendation as it would provide both a high PDR and low and stable end-to-end latency.

Chapter 10

Future Works

For this thesis work, it was only possible to conduct tests on the BabbleSim simulation environment. The next step would be to debug and complete the current code setup for running the tests on the 100-node test network.

The current prioritization methods modify the advertising procedure in the Bluetooth mesh stack. These modifications should be further reviewed and be made part of the next enhancement of the Bluetooth mesh specification.

The developed SRC-DST pair-based prioritization and the Opcode-based prioritization method can be expanded by defining vendor-specific model messages. This will allow better control for a network manager to control message prioritization.

References

[AJdF20] D. Agnoletto, M. Jonsson, and E. P. de Freitas, «Time slot transmission scheme with packet prioritization for bluetooth low energy devices used in real-time applications», *International Journal of Wireless Information Networks*, 2020.

[Babblesim] Babblesim. [Online]. Available: https://babblesim.github.io/ (last visited: Jul. 16, 2023).

[BBH21] S. S. Basu, M. Baert, and J. Hoebeke, «Qos enabled heterogeneous ble mesh networks», *Journal of Sensor and Actuator Networks*, vol. 10, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2224-2708/10/2/24.

[Bluetooth Blog] Maintaining Lighting Control Means More Than Managing Lighting. [Online]. Available: https://www.bluetooth.com/blog/maintaining-lighting-control-means-more-than-managing-lighting/ (last visited: Jul. 16, 2023).

[Bluetooth Mesh] Bluetooth mesh. [Online]. Available: https://www.nordicsemi.com/Products/Bluetooth-mesh/What-is-Bluetooth-mesh (last visited: Jul. 16, 2023).

[BTM] An Intro to Bluetooth Mesh. [Online]. Available: https://www.bluetooth.com/blog/an-intro-to-bluetooth-mesh-part2 (last visited: Jul. 16, 2023).

[BTMIOT] Bluetooth Mesh for Industrial IoT. [Online]. Available: https://blog.nordicsemi.com/getconnected/bluetooth-mesh-for-industrial-iot (last visited: Jul. 16, 2023).

[Ericsson] Bluetooth mesh networking. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/bluetooth-mesh-networking (last visited: Jul. 16, 2023).

[FruityMesh] FruityMesh. [Online]. Available: https://www.bluerange.io/docs/fruitymesh/index.html (last visited: Jan. 23, 2021).

[ISM] ISM Path Loss Model. [Online]. Available: https://github.com/BabbleSim/ext_2G4_channel_Indoorv1/blob/master/docs/2G4Indoorv1_Description.pdf (last visited: Jul. 16, 2023).

[MRC+17]	Y. Murillo, B. Reynders, *et al.*, «Bluetooth now or low energy: Should ble mesh become a flooding or connection oriented network?», in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6.
[MshPRF]	Mesh Profile 1.0.1. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457092 (last visited: Jul. 16, 2023).
[Nordic Semiconductor]	Nordic Semiconductor. [Online]. Available: https://en.wikipedia.org/wiki/Nordic_Semiconductor (last visited: Jul. 16, 2023).
[nrf52]	nRF52811 SOC. [Online]. Available: https://www.nordicsemi.com/products/nrf52811 (last visited: Jul. 16, 2023).
[nrfSDK]	nRF Connect SDK. [Online]. Available: https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/%20index.html (last visited: Jul. 16, 2023).
[QKJ23]	H. Qaq, O. Kulkarni, and Y. Jiang, «Reliable sensor data gathering with bluetooth mesh: An experimental study», Apr. 2023.
[RMGG20]	R. Rondón, A. Mahmood, *et al.*, «Understanding the performance of bluetooth mesh: Reliability, delay, and scalability analysis», *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2089–2101, 2020.
[SiLabs]	AN1137: Bluetooth® Mesh Network Performance. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1137-bluetooth-mesh-network-performance.pdf (last visited: Jul. 16, 2023).

NTNU

Norwegian University of
Science and Technology