MASTER THESIS

Wireless communication in M2M systems

Abiy Tolera Megersa

am304679

Program: Informatics

Specialization: Informatics

SUPERVISOR

Professor Piotr Gaj

Department of Distributed Systems and Informatic Devices

Faculty of Automatic Control, Electronics, and Computer Science

GLIWICE 2023

Thesis title:

Wireless communication in M2M systems

Abstract

This thesis focuses on the use of wireless communications in Machine to machine(M2M) systems. M2M communication refers to the exchange of data between machines, without the need for human intervention. Machine-to-machine (M2M) communication is becoming very important in various applications such as industrial automation, smart homes, transport, agriculture, and healthcare.

The objective of this thesis is to explore the characteristics of wireless communication protocols in M2M systems, including ZigBee and Bluetooth. The thesis also discusses the evaluation of characteristics, challenges, and limitations of wireless communication protocols in M2M systems. To evaluate the performance of the wireless communications protocol's computer program is created using visual studio code. Python Programming language is the main programming language used. The result shows the choice of wireless communications protocols depending on the purpose of the project and its application.

Generally, this thesis gives insight into wireless communication in Machine to machine (M2M) systems and the characteristics, limitations, and changes of wireless communications protocols specifically ZigBee and Bluetooth. This thesis's findings could be used to select appropriate wireless communications protocols for machine-to-machine wireless communications projects. The proposed solution and case study show the potential of M2M systems to transform various industries and pave the way for a more interconnected world.

Keywords: Bluetooth, Python programming language, Wireless communication protocols, ZigBee.

Index

1. INTRODUCTION

Machine-to-Machine (M2M) communication is a form of communication to exchange data between two or more devices without human interactions. This type of communication is widely used in applications where devices are located remote from each other. Wireless communication is an aspect of Machine-to-Machine (M2M) systems, as it allows devices to communicate with each other over large distances without the need for a wired connection. This allows for greater flexibility and scalability, making it possible to connect and communicate with many devices simultaneously.

Nowadays wireless communication is used in Machine-to-Machine (M2M) systems due to their ability to automate processes, reduce cost and improve efficiency in various industries.

The objective of this thesis is to investigate the limitations and challenges of wireless communication in M2M systems and propose the best choice of wireless protocol to address these challenges. This will involve a detailed review of existing literature on this area, developing a framework to evaluate the performance of wireless communication technologies, and investigating the uses of different wireless communication technologies such as ZigBee and Bluetooth and comparing their characteristics.

1.1 Scope & Motivation

The scope of this thesis is to explore communication protocols used in wireless communication in M2M Systems, specifically Zigbee, and Bluetooth. The thesis compares the protocols in terms of their strength and weakness and their suitability for different M2M approaches. Evaluate the characteristics of Zigbee and Bluetooth which involves measuring the latency in different time gaps.

This paper is divided into five chapters, each of which focuses on a particular aspect of the research topic. The first chapter provides an overview of the study, including its purpose and scope. In addition, it describes the problem statement that the research intends to address and the evolution of wireless communication in Machine-to-Machine (M2M) systems. In the second chapter, the Zigbee and Bluetooth protocols are the focus of a literature review that explains their significance in the context of the research. The third chapter describes the research methodology, emphasizing the data collection techniques and describing the configuration parameters. The fourth chapter presents the simulation results derived from the conducted tests, which provide insight into the functionality and efficacy of the implemented tasks. Chapter five concludes the paper with a summary of the main findings, a discussion of their implications, and a conclusion based on the findings of the research.

1.2 Problem Statement

There are a lot of challenges to be addressed to improve the technology for future reliable use. One of the main challenges is ensuring reliable and secure communication between devices. M2M system has various applications such as controlling remote monitoring and automation which requires accurate and timely data exchange. Additionally, wireless communications can be affected by signal degradation, interference, and other factors that lead to transmission errors or even loss

of data. These challenges require careful consideration and planning to ensure that M2M systems operate reliably and securely.

1.3 History of Wireless Communication in M2M Systems

Every industry needs efficient and effective communication between Machines to the machine of production units and services to achieve their goals. This leads a lot of engineers to look forward and engage in this sector by expanding the number of often connected devices such as sensors, controllers, and actuators that are difficult to interconnect and organize in an industrial network. For modern industries to prevail machine-to-machine communication need to be efficiently connected over varying distance in a flexible manner with high security, fast performance, and low cost.

Wired technologies have been widely used over the years in M2M communications. Wired technology is traditional and extensively used when a higher level of security is needed. However, they are static in their setup, and they are not cost-effective. Although cables have been widely used for wired M2M communications for a long time, there are concerns about whether they will be enough for future uses. That is why wireless M2M communication is popular nowadays. Wireless communications are convenient to maintain, install and enhance performance. Wireless technologies reduce cost, provide access to remote areas, and fast transfer of data between nodes. That is why wireless communications are popular in a lot of sectors such as automation, automotive, transport, and so on.

To utilize wireless technologies effectively, it is essential to understand the characteristics of various devices and protocols. As a result, numerous wireless protocols have been developed. Zigbee is widely deployed in RF, Wi-Fi, and Bluetooth applications based on specific requirements. Zigbee provides wireless communication up to 100 meters outdoors and 70 meters indoors. This protocol operates globally on a single 2.4 GHz frequency, providing benefits such as reduced power consumption, minimal latency, and a reduced duty cycle.

Bluetooth Low Energy (BLE) is another short-range wireless communication transmission technology. It is ideally suited for low power consumption, inexpensive, and straightforward communication. BLE is frequently used in industries for device connection and control with a range of up to 10 meters.

LoRaWAN, on the other hand, is a protocol designed specifically for up to 10 miles of long-range communication. It is ideal for industrial applications that require remote monitoring and control. LoRaWAN has low power consumption and reduced latency. The data rate can range between 0.3 and 27 Kbit/s, and the utmost payload size is 222 bytes.

The choice of wireless technology depends on the application requirements, such as range, power consumption, and global coverage. Wireless communication in M2M systems offers several benefits, but it also poses several challenges that require careful consideration to ensure that M2M systems operate reliably and securely.

Table 1.1: Comparison of wireless protocols

Protocol	Range	Frequency	Power consumption	Data rate
Zigbee	Up to 100m	2.4 GHZ	Low	250 kbps
BLE	Up to 100m	2.4 GHZ	Low	2Mbps
WiFi	<100m	Sub-GHZ, 2.4 GHZ, 5 GHZ	Medium	0.1–54 Mbps
LoRaWAN	Up to 10 miles	Sub-GHZ	Low	0.3-27 Kbps

1.3.1 Industrial Communication Backgrounds

Industrial communication is essential for the exchange of data and information in complex industrial settings. Nevertheless, proprietary protocols and limited interoperability posed obstacles to early industrial communication. These obstacles hindered the integration and compatibility of devices from various manufacturers. To address these concerns, standardization organizations such as the International Electrotechnical Commission (IEC), the Institute of Electrical and Electronics Engineers (IEEE), and the International Organization for Standardization (ISO) were established. These organizations were instrumental in creating open, vendor-neutral communication standards, paving the way for the evolution of industrial communication.

Industrial communication protocols and technologies play a crucial role in facilitating seamless and efficient communication between devices and systems in industrial automation environments. Modbus is one of the earliest and most extensively used protocols. 1979's Modbus is a serial communication protocol that enables data exchange between devices connected in a master-slave configuration. It is frequently employed in industrial automation systems for regulating and monitoring devices, accumulating data, and conducting diagnostics.

Profibus, which emerged in the early 1990s as an open fieldbus standard for process automation and manufacturing, is another significant protocol. Profibus enables reliable and rapid device-to-device communication in industrial environments. It supports both decentralized and centralized system architectures, enabling implementations that are flexible and scalable. Profibus is extensively adopted in industries such as manufacturing, automotive, and oil and gas because it offers advantages such as increased productivity, decreased downtime, and streamlined maintenance.

Ethernet/IP is a notable communication technology that utilizes standard Ethernet technology to integrate industrial control systems into enterprise networks. Ethernet/IP facilitates industrial automation data exchange between devices, controllers, and systems by utilizing a familiar Ethernet infrastructure. It provides both real-time control and high-speed data transmission, making it suitable for applications with stringent requirements. Ethernet/IP is extensively deployed in industries such as the automotive, food and beverage, and pharmaceutical sectors, facilitating the integration of industrial automation with enterprise-level systems.

Profinet, on the other hand, is a standard based on Ethernet that was developed specifically for industrial applications. It offers real-time communication, a flexible topology, and high-speed data

transmission, and supports both process and discrete automation. Profinet permits the integration of diverse devices and systems, including controllers, drives, sensors, and actuators, into a single network. It offers increased flexibility, reduced wiring efforts, and enhanced diagnostics, making it a popular option for industrial automation applications.

OPC (OLE for Process Control) is a series of interoperable standards for industrial automation and control systems. OPC enables software applications to communicate and share information regardless of the underlying hardware or communication protocols used. It provides a standardized interface for accessing data from diverse sources, including programmable logic controllers (PLCs), distributed control systems (DCSs), and supervisory control and data acquisition (SCADA) systems. OPC promotes interoperability, simplifies integration efforts, and enables the exchange of real-time and historical data for monitoring, control, and analysis. Industrial processes have been significantly impacted by standardized communication protocols. They have facilitated automation and enhanced industrial efficiency, productivity, and interoperability. Standardization has allowed for the seamless integration of devices and systems from various manufacturers, eliminating compatibility issues. Protocols for industrial communication have revolutionized the operation of industrial automation systems by enabling real-time control, data collection, diagnostics, and process coordination. Standardized communication has accelerated the growth and development of industrial automation, ultimately resulting in increased efficiency and productivity across a variety of industries.[1], [3]

1.3.2Industrial Networks Evolution: From Wired to Wireless

In the early phases of industrial communication, electrical networks were essential for establishing dependable and stable connections within industrial environments. As a result of their demonstrated performance and dependability, wired networks, such as Ethernet and fieldbus systems, were extensively adopted. Ethernet, which was initially designed for office environments, rapidly made its way into industrial settings due to its capacity for high-speed data transmission and compatibility with existing IT infrastructure. It has become the backbone of industrial communication, enabling the exchange of vital data between diverse devices, machinery, and control systems. Similarly, Fieldbus systems, such as Profibus and Modbus, provided a dependable and deterministic communication platform designed specifically for industrial automation and control applications. These wired networks provided the foundation for seamless and efficient data exchange, allowing for real-time monitoring, control, and coordination of industrial processes. The widespread adoption of these technologies paved the way for the subsequent evolution of industrial communication toward wireless technologies.

The implementation of wireless networks in industrial contexts has provided numerous benefits. First, wireless technologies eliminate the need for an extensive cabling infrastructure, which reduces installation and maintenance expenses. This adaptability facilitates the deployment of devices and systems in remote or inaccessible locations. Second, wireless networks facilitate mobility, enabling administrators and personnel to move freely throughout an industrial setting without being tethered to a specific location. This mobility improves operational efficiency and enables real-time monitoring and control from any location within the range of the wireless network.

Wi-Fi, Bluetooth, Zigbee, and WirelessHART are among the wireless networking technologies utilized in industrial environments. Wi-Fi networks are commonly used for local area networking

and connecting industrial devices to the internet. Bluetooth technology facilitates wireless communication over brief distances and is frequently utilized in applications involving mobile devices and sensors. Zigbee provides low-power, low-data-rate wireless communication, which makes it appropriate for industrial automation and control systems. WirelessHART is a wireless sensor network standard designed specifically for industrial process measurement and control.[2], [3]

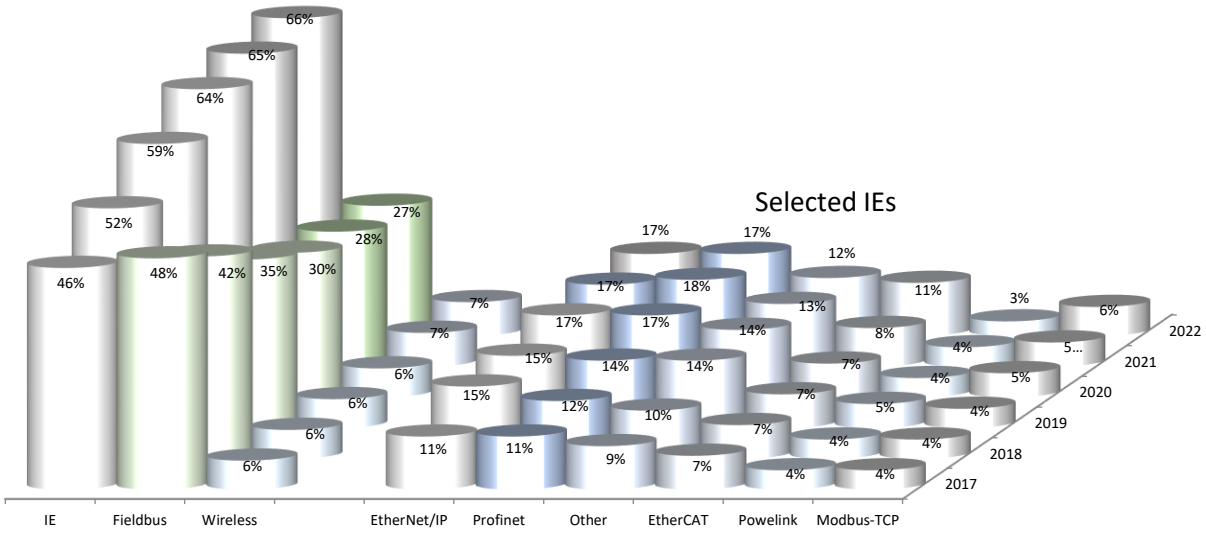

Figure 1.1: Market share of different industrial communication technologies from 2017–2020 [3]

1.4 M2M Systems Requirements

This subsection gives a list of requirements for M2M technologies in this part that are all regarded to be significant requirements and to which all the technologies discussed in this article comply in varying degrees.

A. Scalability

Networks that are built to support many devices must be scalable, which presents several difficulties. A network's likelihood of collisions increases with the number of nodes it contains, which might cause congestion and reduce the network's effectiveness. Additionally, load balancing is a significant issue because a single gateway may not be able to handle the data processing and storage demands. Another aspect to consider is deployment costs, since expanding the network may demand more infrastructure and resources.

Networks that need to grow also need data fusion since it's necessary to aggregate and analyze data from many devices to get insights and make defensible judgments. To accomplish data fusion effectively and with the least amount of latency, the network must be structured in this fashion.

The network's channel access technique is one of the crucial aspects that affect scalability. Contention-based approaches may see an increase in collisions in dynamic circumstances where the number of participants is not static, and nodes are constantly joining and departing the network.

Contention-free approaches, on the other hand, do not need reconfiguration, which may be time-consuming. Additionally, the growing number of devices may make reconfiguration more difficult.

Therefore, effective, and dynamic channel access techniques that can adjust to the shifting network conditions while minimizing collisions, load balancing, and reconfiguration time are necessary to achieve high scalability. The financial effects of adding more nodes to the network, such as the need for extra infrastructure and related maintenance costs, must also be considered.[4]

B. Low power consumption

Due to its dispersed architecture and limited or no access to stable power sources, low power consumption is an essential characteristic for many networked sensors and actuators. In such cases, energy consumption is mostly caused by network activity, with connection using more energy than processing. Therefore, both the physical layer and MAC layers undergo energy-saving optimizations.

MAC techniques concentrate on increasing the effective throughput of transmission and minimizing control overheads and beaconing since at the MAC layer collisions and the exchange of configuration messages are key causes of power depletion. Contention-free systems, on the other hand, strive to reduce control overheads and beaconing, while contention-based mechanisms are heavily impacted by collisions, which may occur often in congested circumstances. [8]

Duty cycling, which enables devices to regularly switch on and off their wireless interfaces, is a common strategy for improving power consumption. Algorithms for duty cycling may be time-based or dynamic, depending on the configurations that the master node receives. These algorithms suggest a time window for "deep sleep," during which the radio interface is off and power consumption is almost zero. Depending on the use scenario, wakeup intervals vary in frequency.

Other strategies, such as radio optimization, data reduction, sleep schemes, energy-oriented, and routing, can be used to improve the energy efficiency of M2M communication. For instance, radio optimization modifies the coding and modulation systems used to increase spectrum efficiency and lower power use. To limit the quantity of data transferred and save power, data must be compressed. Devices are put to sleep using sleep schemes when they are not engaged in communication. Choosing routes with the least amount of energy use is known as energy-oriented routing. Controlling power usage is important for M2M communication, and different strategies may be used at different levels to decrease power consumption and reduce communication energy costs. [4], [8]

C. Reliability

For many network use cases where data transmission must be precise and quick, reliability is a crucial need. By calculating the likelihood that a message will reach a node successfully even if some network connections fail, dependability in networks may be calculated. Link failures and a lack of control mechanisms that would ensure the delivery of data packets are the main causes of a network's unreliability.

The management and topology of wireless networks are essential for maintaining dependability in industrial networks. For instance, mesh redundant topologies, which provide alternative routes for data to reach a node, might reduce the impact of a communication connection failure. This

redundancy makes sure that the data can still be transmitted via an alternative path even if one link fails. On the other hand, networks with simple star or tree topologies are less reliable since even a single link failure might cause one or more nodes to be excluded.[4], [8]

D. Low Cost

Given that a network of many devices, low cost is a crucial requirement. To reduce the cost per unit, this calls for the adoption of inexpensive hardware and components, such as single-chip solutions without costly circuitry. Cost factors also play a role in the selection of channel access mechanisms at the MAC layer. Since TDMA (Time Division Multiple Access) is less complicated and more affordable than CDMA (Code Division Multiple Access)-based methods, it is a viable option in environments free of contention. Due to their complexity, CDMA-based approaches are not appropriate for low-power and low-cost deployments because they would need expensive hardware. Pure FDMA (Frequency Division Multiple Access) methods are also seldom employed in M2M applications because of the high cost of the effective frequency filters needed in each unit's radio hardware. The exception is systems based on OFDMA (Orthogonal Frequency Division Multiple Access). This is because they eliminate the need for filters for each sub-channel and make it simple and affordable to implement the Fast Fourier Transform (FFT) in chips. As a result, it is now feasible to offer simultaneous access for many devices at a reasonable cost.[4], [8]

E. Security

Due to the vulnerabilities that can compromise the reliability and integrity of these systems, securing industrial M2M communication is of utmost importance. Industrial M2M deployments face a vast array of security threats, including potential assaults that can disrupt operations, compromise sensitive data, and undermine the security posture as a whole. With the complexity and extensive deployment of these systems, it becomes even more difficult to implement robust security measures. A comprehensive security strategy for industrial M2M communication must include network security, device security, access control, and data encryption, among other factors. By instituting effective security measures in each of these areas, industrial M2M communication can be protected from unauthorized access, data intrusions, and other security risks, ensuring the dependable and secure operation of these vital systems. [8]

Due to their complexity and pervasive deployment, industrial M2M communication systems are susceptible to numerous security flaws. Network security is essential for preventing unauthorized access and protecting the confidentiality and integrity of data transmitted over a network. Device security is crucial for preventing unauthorized interference with industrial M2M devices, such as sensors, controllers, and gateways, which can compromise the system as a whole. Access control mechanisms must exist to authenticate and authorize users and devices, ensuring that only authorized entities have access to the system. In addition, techniques for data encryption should be implemented to safeguard the confidentiality of sensitive information transmitted over an industrial M2M network. By addressing these security aspects comprehensively, industrial M2M communication systems can operate with confidence, preserving the required levels of integrity, dependability, and security for vital industrial processes.[4], [8]

F. Low Latency

Reduced latency describes how quickly a data packet reaches its destination after being sent. In a network, many variables can affect latency. One of the most crucial elements is the physical deployment of the network, which includes the connection strength between endpoints, the number of hops in a communication line, and the number of nodes in the network. The choice of PHY layer processes, such as spread spectrum methods, modulation and coding schemes, frequency, and geographic diversity, is an additional important consideration.

To reduce latency in a network, the MAC layer channel access technique is also essential. Because they can scale effectively in small networks without requiring reconfiguration, contention-based protocols like CSMA/CA are frequently used in some technologies. They may, however, have astronomically high latency and idle listening in huge networks. Contention-free protocols, like TDMA, on the other hand, are better suited for big networks because they include algorithms that can efficiently use existing resources. However, because a node must be globally reconfigured each time it enters or exits the network, they do not scale effectively.

The network's physical deployment is crucial. Access points and network equipment are strategically positioned to achieve optimum coverage, reduce signal interference or attenuation, and lower latency. Additionally, the network can detect and rectify data transmission errors by adding effective error correction methods, which reduces the need for retransmissions and increases throughput overall. Additionally, more data may be encoded in each transmission as a result of better modulation techniques, substantially boosting the quantity of information transmitted per unit of time. As a result, data transmission speeds are increased. [4], [9]

G. Increased Range of Communication

The current trend in future-generation IoT deployments aiming at the market of monitoring and public welfare is a larger range of radio communication, which translates to a bigger area deployment. Knowing that the nominal range is sometimes insufficient to determine how broad a deployment may be, such a feature is a must for many use cases. The range is often correlated with the spectrum frequency bands and modulation encoding techniques due to indoor settings, obstructions, and spatial cohabitation with other technologies. In addition to being intended for moderately reliable data transport, the 2.4 GHz frequency bands have several non-negligible disadvantages for long-range IoT situations. Due to its nature, it can support higher data rates more readily, but it also struggles more with obstacles, and indoor deployments, and needs more power to be pushed over long distances.

Furthermore, in situations with a high network population, the recent overuse of such frequency bands is not helpful. These and other factors are increasing interest in a network in sub-GHz band technology. In cohabitation with other cellular technologies like LTE, 5G, UMTS, and GSM, almost all long-range technologies use either unlicensed bands like the 868 MHz or licensed channels around 800 MHz.

Additionally, an increased range is frequently chosen over lower data rates at the cost of increased power consumption. Numerous applications of the next generation require extremely low consumption and low data rates, making the emerging narrowband long-range solutions made for wide-area deployments convenient.[4], [9]

1.5 M2M Systems Applications

With the advent of low-power and low-cost M2M devices, there has been a significant increase in demand for M2M applications that can automate various tasks and improve our daily lives. M2M applications, for instance, may be utilized in administrative work to automate rote operations like data input, processing, and analysis, as well as report generation. M2M devices may be utilized in the house to automate chores like regulating lights, security systems, and HVAC (Heating, Ventilation, and Air conditioning) systems. This increases energy efficiency and lowers expenses.

M2M apps may be utilized in human activities like healthcare to remotely monitor patients' health state and notify medical personnel of any emergencies. By keeping an eye on traffic and giving drivers and traffic cops up-to-date information, they may also be utilized in transportation to increase efficiency and safety.M2M applications provide a variety of advantages by automating repetitive operations and enhancing efficiency, which may save costs, boost output, and enhance. [5]

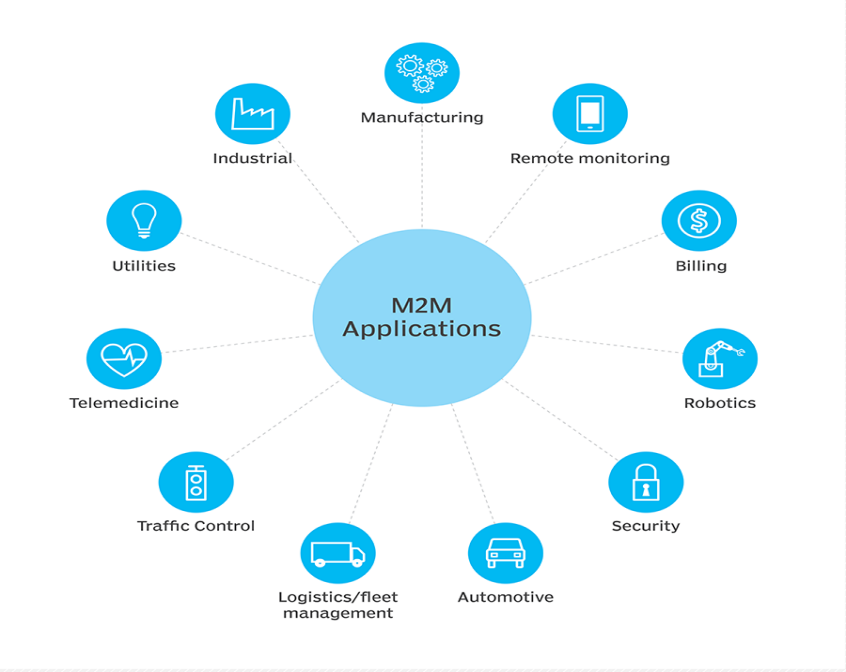

Figure 1.2: Application of Machine to Machine [5]

1.6 Wireless Communication Standards for M2M

Wireless communication standards for M2M (Machine-to-Machine) systems are a collection of protocols and specifications that define how M2M network devices communicate wirelessly with one another. These standards guarantee interoperability, compatibility, and seamless communication between devices from a variety of manufacturers and industries. Here are several important aspects of M2M wireless communication standards:

Protocols and Specifications: Wireless communication standards for M2M include various protocols and specifications that regulate various aspects of communication, including data transmission, device discovery, security, and network administration. These standards define the norms and procedures for data exchange, enabling devices to communicate effectively and reliably.

Range and Coverage: Various wireless communication standards offer varying ranges and coverage capabilities. Bluetooth and Wi-Fi, for instance, typically provide short-range communication within a few meters or tens of meters, which makes them suitable for local area networks. Cellular networks, on the other hand, offer extensive coverage and can facilitate M2M communication across a larger geographical area. [13], [14]

Data Rate and Bandwidth: Wireless communication standards also vary in terms of data rate and available bandwidth. Some standards, such as cellular networks, offer higher data rates and greater bandwidth, thereby facilitating quicker and more data-intensive communication. Other standards, such as low-power wide-area networks (LPWAN), prioritize long-distance communication with reduced data rates, which are optimized for low-power and low-bandwidth M2M applications.

Power Consumption: Power efficiency is an important factor for M2M systems, particularly for devices that operate on battery power or have limited energy resources. The power consumption requirements of different M2M wireless communication standards vary. Bluetooth Low Energy (BLE) and low-power wide-area network (LPWAN) technologies are designed to reduce power consumption, enabling devices to operate for extended periods.

Security and Authentication: Considering the sensitive nature of the data exchanged, ensuring secure communication is crucial for M2M systems. Encryption, authentication mechanisms, and secure key exchange protocols are incorporated into wireless communication standards to safeguard the integrity and confidentiality of transmitted data. [13]

1.7 Future Trends and Obstacles

The development of wireless M2M communication is influencing the future of numerous industries. The deployment of 5G networks and forthcoming technologies such as 6G will transform M2M communication by providing quicker throughput, reduced latency, and higher device densities. This will allow for more efficient and pervasive M2M deployments, enabling monitoring, control, and automation in real time. In addition, the incorporation of edge computing will facilitate processing and decision-making at the network's periphery, thereby reducing latency and enhancing M2M system efficiency. The incorporation of artificial intelligence and machine learning algorithms will further improve wireless M2M communication's data analytics, predictive maintenance, and intelligent decision-making.

Despite the bright future of wireless M2M communication, several obstacles must be overcome. The growing number of connected devices makes M2M networks susceptible to cyberattacks, posing significant security and privacy concerns. Interference and spectrum limitations can result in degraded performance, necessitating spectrum management techniques that are effective. For M2M devices with limited power sources, power consumption, and energy efficiency are crucial considerations. As the number of devices and applications increases, scalability and network management become increasingly complex, necessitating efficient provisioning and centralized management systems. Standardization and interoperability initiatives are essential for ensuring the seamless integration of M2M devices and systems. Unlocking the full potential of wireless M2M communication in industrial settings will necessitate continuous research, collaboration, and technological advances to overcome these obstacles. [5]

2. LITERATURE REVIEW

In the context of industrial communication, numerous wireless connectivity protocols are available. Nevertheless, the Zigbee and Bluetooth protocols were chosen for this study due to their widespread adoption and importance in industrial communication. Both protocols offer unique benefits and capabilities that make them highly suited for a variety of industrial applications. Based on the IEEE 802.15.4 standard, Zigbee is renowned for its low-power and low-data-rate wireless communication capabilities. It is ideal for connecting numerous low-power devices in industrial automation and control systems because it excels at establishing dependable wireless mesh networks. Zigbee's characteristics, which include low latency, power efficiency, and robust security measures, contribute to its widespread use in applications requiring extended battery life and dependable data transmission.

Bluetooth was included in this study due to its adaptability, usability, and extensive application in industrial communication. Due to its compatibility and support for short-range wireless connectivity, Bluetooth technology, which was originally utilized for personal area networking and consumer applications, has made its way into industrial settings. Bluetooth is a cost-effective method for connecting industrial devices like mobile computers, barcode scanners, and wearable devices. Its sturdiness, ease of integration, and support for several profiles allow for the exchange and control of data between industrial devices and systems. Bluetooth's frequency-hopping spread spectrum technique assures reliable transmission in the presence of interference, making it suitable for industrial settings with multiple wireless devices in use.

2.1 Overview of Zigbee Protocol

Zigbee is a low-power, 250 kbps, wireless communication protocol that is designed for machine-to-machine applications. It operates in the 2.4 GHz band and is used in a variety of industrial applications. Zigbee is popular in industrial mesh networking for connecting control systems and sensors. It is an open global, packet-based protocol designed for wireless communication which is secure, reliable, and low power. It allows for the creation of wireless networks that can be used in a variety of applications. Zigbee uses the IEE 802.15.4 standard as a low-data-rate wireless networking standard as a foundation. [6], [7]

Equipment can be placed anywhere and still communicate with the rest of the system, and it can also be moved around without affecting the network. This makes it a very flexible and adaptable technology for industrial control. Zigbee RF4CE is an extension of the IEE 801.15.4 standard that provides a simple networking layer and standard application profiles. This allows for the creation of interoperable multi-vendor customer electronic solutions. This means that devices from different manufacturers can be used together in the same system, which can be very convenient for consumers. [6], [7]

Zigbee is a wireless communication protocol that has been specifically designed to meet the needs of low-power, low-data rate devices such as sensors, actuators, and controllers. It is gaining popularity as a global control/sensor network standard due to its unique features some of which are:

- Low cost
- High density of nodes per network
- Simple protocol, global implementation
- Low power consumption

2.1.1 Zigbee Logical Device Types

In a ZigBee wireless network, there are three main logical device types:

- Coordinator
- Router
- End device

Coordinator

The coordinator is the central device that manages the network. It is responsible for starting the network, assigning network addresses, managing the routing of messages between devices, and handling other network management functions. There can only be one coordinator in a ZigBee network, which acts as the network's root. [6]

Router

Routers are intermediate devices that extend the range of the ZigBee network by forwarding messages between other routers and end devices. Routers can also participate in network management functions such as security key management and channel selection. The router can function as a parent to other end devices or as a child of the coordinator or another router. [6]

End device

End devices are devices that have limited processing power, and memory, and typically operate on battery power. They can gather a variety of data via switches and sensors. They can communicate with their parents (the coordinator or router) but are unable to transmit information from other devices. There is a chance that their costs might be decreased due to their decreased functionality. Better low-power models are supported by them. Unlike the gadgets in the other two categories, these devices are not required to always remain awake. Up to 240 end nodes, or distinct apps sharing a radio, may be present on each end device. [6]

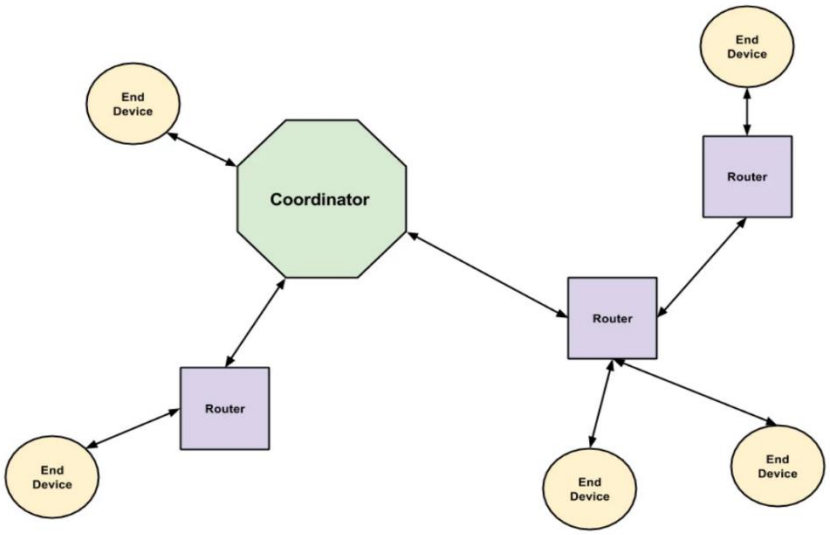

Figure 2.1: Zigbee logical device types [6]

2.1.2 Access Modes

Beacon and non-beacon networks are the two types of multi-access in ZigBee networks. Nodes in non-beacon networks may broadcast data whenever the channel is vacant, with no time slots. In beacon-enabled networks, however, the coordinator assigns guaranteed time slots (GTS) to each device to transmit data. The devices must be synced with one another to utilize their own time, which is accomplished through the coordinator's beacon signals. [7], [6]

Devices in non-beacon networks do not need to be synchronized and may broadcast data anytime the channel is available. As a consequence, these networks are unable to provide GTS or contention-free times. This increases the likelihood of collisions and retransmissions, which might result in increased energy consumption and shorter battery life for the devices. [7], [6]

In contrast, in beacon-enabled networks, devices are assigned GTS, allowing them to broadcast data without interfering with other devices. The coordinator provides beacon signals to the devices to synchronize them, which lowers collisions and retransmissions and increases energy efficiency and battery life. Because the devices only wake up during their assigned time slot, they can save energy by remaining in sleep mode during other times. As a result, the beacon-enabled network consumes less power than the non-beacon network. [7]

2.1.3 Zigbee Protocols Stack

The Open System Interconnection (OSI) model, which outlines a seven-layer architecture for communication systems, served as the foundation for the construction of the ZigBee protocol stack. But just a portion of these levels—specifically layers 1-3, or the Physical Layer (PHY), Media Access Control (MAC), and Network layers—are used by the ZigBee protocol stack. The ZigBee Alliance specifies the Network layer and Application layer in addition to the PHY and MAC layers.

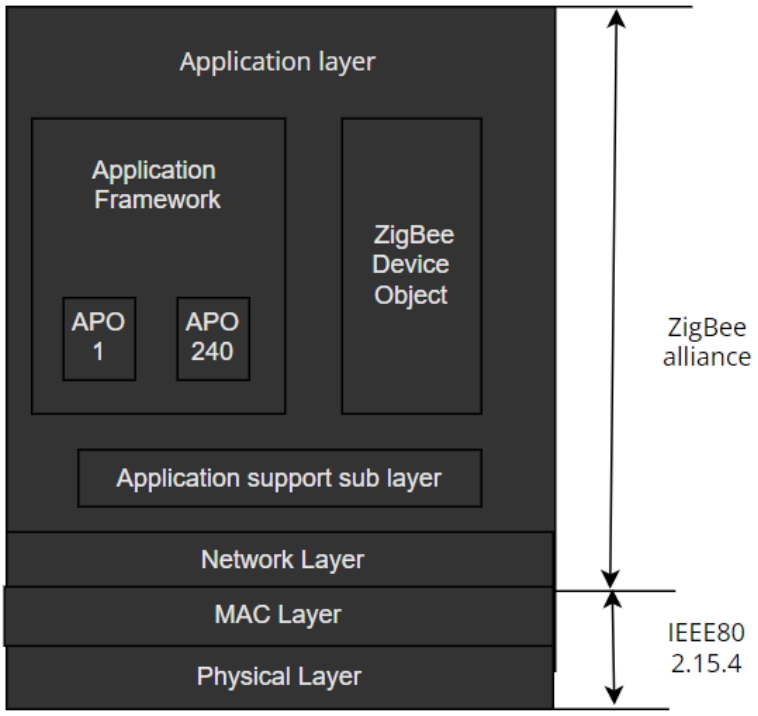

Figure 2.2: ZigBee Protocol Stack

A. Physical Layer

Multiple layers of the ZigBee protocol architecture define the communication framework for ZigBee devices. Utilizing direct-sequence spread spectrum (DSSS), Zigbee operates in the 2.4 GHz ISM band. In order to enable simultaneous usage of several channels, it splits the frequency into 16 equally sized channels of 5 MHz. The Physical Layer, located at the lowest level, is responsible for the transmission and reception of data over the frequencies. Physical Layer responsibilities include modulation/demodulation, frequency selection, power management, error detection, and correction. It specifies the radio frequency's physical characteristics, including the frequency band, channel access mode, data rate, and transmission capacity. The Physical Layer ensures dependable and efficient wireless communication between ZigBee devices, enabling them to transmit and receive data in a robust and energy-efficient manner, which is crucial for constructing low-power and cost-effective ZigBee networks. [7], [6]

B. MAC Layer

The ZigBee protocol stack's MAC (Media Access Control) layer is responsible for coordinating communication within a ZigBee network. It controls the network's topology, device association, and channel access. When multiple devices attempt to transmit data, the MAC layer implements the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) mechanisms to prevent collisions. In addition, it manages duties such as frame acknowledgment, retransmission, and security measures to ensure dependable and secure data transmission. The MAC layer plays a

crucial role in enabling coordinated and efficient communication between ZigBee devices, thereby facilitating the establishment of scalable, robust ZigBee networks. [7], [6]

C. Network Layer

The network layer manages the network infrastructure and provides for managing the network infrastructure and providing efficient routing and addressing services. It manages network discovery, device addressing, and routing decisions to ensure that data reaches its intended destination. The network layer employs protocols to establish and maintain to establish and maintain network connections, calculate routing, and administer network resources. It enables ZigBee devices to communicate regardless of their physical proximity by identifying the most efficient data transmission paths. The network layer plays a crucial role in enabling reliable and seamless communication within ZigBee networks, thereby enhancing the scalability and adaptability of the entire system. [7], [6]

D. Application Layer

The application layer of the ZigBee protocol architecture defines the specific functionalities and interactions between applications operating on ZigBee devices. It permits the implementation of various application profiles that define specific use cases and network requirements. Application layer services include device discovery, data encoding and decoding, and application-specific capabilities. Within the ZigBee network, it enables applications to exchange data, control devices, and perform specific duties. The application layer facilitates interoperability and enables the seamless integration of ZigBee devices into larger systems or platforms by acting as a bridge between the underlying network layers and the specific applications. [7], [6]

Application Support Sub-Layer

In ZigBee, the Application Support (APS) sublayer provides a collection of services that allow safe communication between the Network (NWK) and Application (APL) levels. To provide secure communication, the APS sublayer coordinates frame transmission and reception as well as cryptographic keys. Through its data and management entities, the APS sublayer delivers services to the top levels. [6]

Establish Key, Transport Key, Update Device, Remove Device, Request Key, Switch Key, Entity Authentication, and Permissions Configuration Table are among the services provided by the APS Layer Security. The APS sublayer can use these services to create and maintain cryptographic keys, authenticate entities, and adjust authorization settings for secure communication. The top layers communicate with the APS sublayer through the use of primitives that request certain services. [6]

The Application Objects (APO)

The application objects (APO) in the ZigBee protocol stack are in charge of controlling and managing the protocol layers in a ZigBee device. Each APO is given a unique endpoint number that other APOs can use to communicate with it. A single ZigBee device can have up to 240 APOs. A ZigBee application must conform to an established application profile authorized by the ZigBee Alliance to enable compatibility across various ZigBee devices. A message format and protocol for interactions between application objects are defined by an application profile. The application

profile architecture enables various vendors to create and sell ZigBee devices that can interoperate with each other in a specific application profile independently.

ZigBee devices may interact with one another predictably and dependably by adhering to a common application profile, independent of the individual brand or model of the devices involved. This universality is a significant benefit of the ZigBee protocol stack, allowing for the creation of a diverse variety of wireless M2M applications.[7]

ZigBee Device Object

The ZigBee Device Object is an important part of the ZigBee protocol stack. It serves three purposes: service discovery, security, and binding. Using unicast messages, the service discovery function locates nodes and determines the MAC address of the coordinator or router. It also makes it easier to discover certain services based on their profile identifiers. Profiles are crucial in this context because they establish the message formats and protocols for interactions between application objects.

The network manager function in the coordinator is in charge of connecting to an existing Personal Area Network (PAN) or building a new PAN. This enables devices to connect to the network and interact with one another. In the ZigBee Device Object, the binding manager function is in charge of binding nodes to resources and applications, as well as devices to channels. This helps to guarantee that devices can effectively interact with one another.[8]

2.2 Overview of Bluetooth Protocol

Bluetooth is used wireless communication protocol that operates in the 2.4 GHz frequency spectrum and has a significant impact on industrial applications. Bluetooth's short-range communication capabilities, which typically extend up to 100 meters, enable a reliable and secure wireless connection between various industrial devices. It is widely used in industrial environments where devices must communicate over relatively brief distances due to their adaptability and usability. Bluetooth technology facilitates seamless connectivity between devices, enabling efficient data exchange and command transmission. It supports multiple profiles, including the Serial Port Profile (SPP) and the Human Interface Device (HID) Profile, enabling standardized communication between various types of devices. This interoperability ensures that Bluetooth-enabled devices from different manufacturers can function in the same system, providing users with flexibility and convenience.

In addition to its versatility, Bluetooth provides durability in tough industrial environments. Its frequency hopping spread spectrum method mitigates interference from other wireless devices operating in the same frequency band, ensuring reliable transmission, and minimizing the impact of potential signal interruptions. This makes Bluetooth suitable for industrial applications requiring the simultaneous operation of multiple wireless devices. Bluetooth also provides secure communication through encryption and authentication mechanisms, protecting sensitive data in industrial environments. These security features are essential for safeguarding sensitive data and preventing unauthorized access, thereby enhancing the overall dependability and integrity of industrial communication.

Bluetooth Low Energy (BLE), a power-efficient Bluetooth variant, has acquired traction in industrial environments. BLE is designed for low-power applications, allowing for extended battery life in wireless devices and reducing energy consumption. This makes it suitable for numerous industrial applications, such as industrial monitoring systems and asset tracking, that require power-efficient wireless connectivity. BLE provides interoperability with Bluetooth devices, enabling seamless integration into existing Bluetooth ecosystems. Its minimal power consumption makes it an ideal component industrial device, allowing for long-term operation without frequent battery replacements or recharging. Bluetooth and its low-power variant BLE continue to play a crucial role in facilitating wireless communication in industrial environments due to their efficiency, dependability, and compatibility. [10]

2.2.1 Bluetooth Logical Device Types

Bluetooth supports a variety of logical device categories, each of which serves a distinct function within the Bluetooth ecosystem. These logical device categories are defined to facilitate specific functions and facilitate device-to-device communication. The following are the most common logical device categories in Bluetooth. [12]

A. Host/Controller

The Host/Controller is accountable for the overall operation of a Bluetooth system. It consists of both host and controller components. The host is responsible for higher-level tasks such as device management, protocol stack, and application interfaces, whereas the controller is responsible for lower-level tasks such as radio frequency (RF) transmission and reception.

B. Central

The Central device is responsible for initiating and coordinating Bluetooth communication. It is responsible for scanning peripheral devices, connecting them, and controlling them. Smartphones, tablets, and computers are typically the central devices that establish connections with peripherals such as Bluetooth headphones and speakers.

C. Peripheral

The Peripheral device serves as the counterpart to the Central device. It enables other devices to communicate and interact with it, such as smartphones and tablets. Peripheral devices include fitness monitors, wearables, and sensors that transmit data for processing or display to a central device.

D. Dual mode

Dual-mode devices can function as both central and peripheral devices. These devices are capable of switching between duties based on the needs of the communication scenario. A dual-mode smartphone, for instance, can function as a Central device when connected to a peripheral speaker, but as a Peripheral device when connected to a computer for file transfer.

E. Bridge

A Bridge device acts as an intermediary between Bluetooth and other forms of communication technology. It allows Bluetooth devices to communicate with non-Bluetooth devices or networks. In IoT (Internet of Things) applications where Bluetooth devices must communicate with devices

using other wireless technologies, such as Wi-Fi or Zigbee, bridge devices are frequently employed. These logical device categories provide the infrastructure required to establish and manage Bluetooth connections in a variety of contexts. Each device type has its unique function and purpose, contributing to the adaptability and compatibility of Bluetooth technology. Bluetooth technology supports multiple access modes that dictate how devices connect and exchange data. These access modes specify the type of connection and degree of device interaction. Here are the common Bluetooth access modes: [12]

Classic Mode is the traditional Bluetooth access mode that supports point-to-point and point-to-multipoint connections. In this mode, devices establish a master-slave relationship, with one device acting as the master and initiating and controlling the connection, while the other(s) act as captives and respond to the master's commands. Applications such as audio streaming, file transfer, and device synchronization commonly use Classic Mode.

Bluetooth Low Energy (BLE) Mode is a power-efficient access mode designed for low-power devices with limited processing capabilities. One device (the central device) initiates connections and communicates with one or more peripheral devices using a client-server architecture. BLE is optimized for intermittent data transmission, making it optimal for battery-sensitive applications such as fitness monitors, smart home devices, and wearable technology. [12]

Dual Mode: Dual Mode devices support both Classic Mode and BLE Mode, enabling them to connect with a wider variety of devices. Based on the requirements of the application and the devices they are connecting to; these devices can transition between the two modes. Dual Mode devices offer both legacy compatibility with older Bluetooth devices and low-power communication with BLE devices. [12]

Bluetooth Mesh is a network access mode designed for large-scale deployments such as intelligent lighting systems and industrial automation. In this mode, devices establish a self-healing mesh network in which multiple devices (nodes) can communicate to relay messages and extend network coverage. Bluetooth Mesh enables robust and scalable communication, allowing for extensive connectivity and control over a vast area.

These access modes provide flexibility and accommodate various Bluetooth communication requirements. The choice of access mode is dependent on the specific application, power consumption requirements, and device compatibility.[11]

2.2.2 Bluetooth Protocol Stack

Bluetooth node architecture is the hierarchical structure of a Bluetooth device that allows it to communicate with other Bluetooth devices.

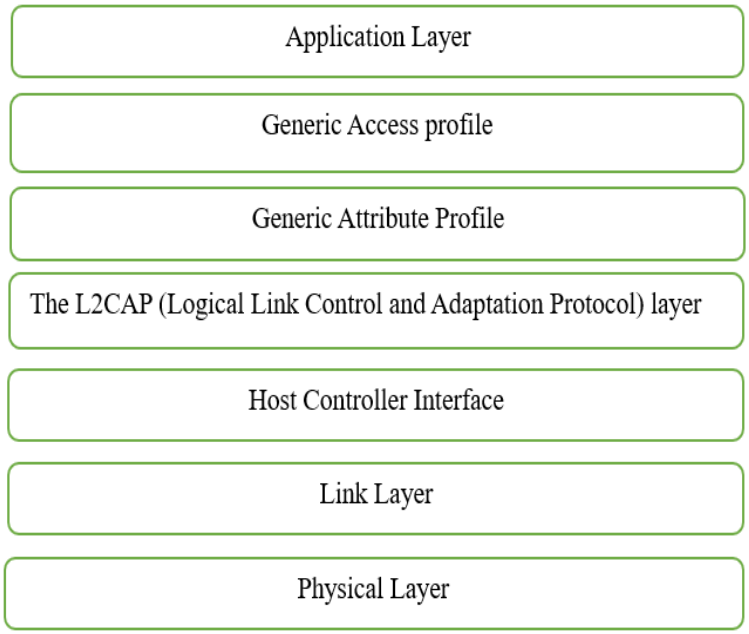

Figure 2.3: Bluetooth Architecture

A. Physical Layer

The Bluetooth architecture's physical layer is its foundational layer. A Bluetooth communication system controls the lowest level of operation and is responsible for radio signal transmission and receiving. A variety of critical functions are carried out at the physical layer to guarantee a dependable wireless connection. One of its main purposes is modulation, which transforms digital data into a format appropriate for radio broadcast encoding the data onto a carrier wave entail changing the wave's properties (such as amplitude, frequency, or phase) to represent the data. The physical layer extracts the encoded data from the received signals at the receiving end and prepares it for further processing by the Bluetooth upper levels.

The Physical layer of the Bluetooth design runs in the 2.4 GHz Industrial, Scientific, and Medical (ISM) band. The 2.4 GHz ISM band is ideal for Bluetooth communication because it provides enough bandwidth for data transmission while reducing interference from other devices. Bluetooth uses frequency hopping spread spectrum (FHSS) as part of its physical layer to reduce the impacts of interference and increase signal dependability. FHSS allows Bluetooth devices to swiftly transition between different frequencies within the 2.4 GHz ISM band, hopping over numerous channels in a coordinated fashion. This dynamic frequency hopping approach aids in the avoidance of congestion and the reduction of the impact of interference, hence improving the overall stability of Bluetooth wireless communication. [11]

B. Link Layer

In the Bluetooth architecture, the link layer is critical in establishing connections between Bluetooth devices and managing the connection state during the communication process. It is

responsible for several elements of link-level activities, including encryption, security, packetization, and error management.

Connection management is one of the link layer's key duties. It speeds up the formation of connections between Bluetooth devices by commencing a process known as advertising and scanning. Devices that want to communicate send out advertising packets, and other devices scan for these packets to find and connect to them. Once a connection is established, the link layer monitors the connection state, which includes monitoring link quality, synchronization, and processing link-level events. [12]

C. Host Controller Interface (HCI):

The Host Controller Interface (HCI) connects the host, which is often an application processor or a computing device, to the controller, which is the Bluetooth radio. The HCI layer is critical for providing compatibility between various host platforms and Bluetooth radio implementations. Because Bluetooth radios differ between devices and manufacturers, the HCI offers a defined set of instructions and protocols through which the host may connect with the controller. This standardization assures that host platforms from different suppliers may coexist with multiple Bluetooth radio implementations. [11]

The host can use the HCI to send commands to the Bluetooth controller to conduct different functions such as searching for nearby devices, starting connections, transferring data, and controlling the operating settings of the BLE radio. The Bluetooth SIG (Special Interest Group), the body in charge of creating and maintaining Bluetooth protocols, defines the HCI commands. Depending on the implementation, these commands are transferred from the host to the controller via a transport layer, which can be a physical connection or a wireless link.

D. The L2CAP (Logical Link Control and Adaptation Protocol) Layer

L2CAP is a crucial element of the protocol stack. L2CAP operates at the highest stratum of the Bluetooth hierarchy, above Baseband and Link Manager. It provides several fundamental functions that allow higher-level protocols and applications to exchange data over Bluetooth connections. Establishing and managing logical channels between Bluetooth devices is the primary function of L2CAP. These channels serve as paths of communication that facilitate the exchange of data between devices. L2CAP supports both connection-based and connectionless communication protocols. [11]

E. Generic Attribute Profile

The Generic Attribute Profile is a Bluetooth standard that defines data organization and sharing between Bluetooth devices. GATT defines the structure of data as well as the operations that may be done on it, allowing devices to communicate and interact with one another.

GATT extends the Attribute Protocol (ATT), which specifies a hierarchical data structure for organizing information in a Bluetooth device. Data in GATT is arranged into a set of characteristics, each of which represents a unique piece of information. Devices can execute several operations on the characteristics to share data via GATT, including reading, writing, and subscribing to notifications or indications. Devices can retrieve or update attribute values via read and write

operations, whilst notifications and indicators allow devices to receive asynchronous updates when the value of characteristic changes.

The GATT protocol is based on a client-server approach, with one device serving as the GATT client and another serving as the GATT server. The client can send requests to the server to read or write characteristics or subscribe to alerts, and the server will reply appropriately. [11]

F. Generic Access Profile

The Bluetooth protocol includes the Generic Access Profile (GAP), which defines the fundamental features of device identification and connection setup in a Bluetooth network. GAP is a defined protocol that allows devices to announce their presence and other devices to discover and connect to them. Bluetooth devices promote their capabilities and identities to surrounding devices via advertising packets that adhere to the GAP standard. These packets typically include information such as the device's name, services provided, and other pertinent data. Devices on the receiving end can check for advertising packets from adjacent devices, extract the information contained inside, and determine whether to start a connection. GAP also provides connection formation protocols, allowing devices to negotiate connection parameters and implement the appropriate security mechanisms for secure communication.[11], [12]

G. Application Layer

The application layer is the topmost element of the protocol stack in Bluetooth architecture. It includes the applications, services, and profiles that use the Bluetooth stack's lower layers to provide specific functionality and services to end users. The application layer defines the behavior, features, and interactions of Bluetooth devices in different usage scenarios. Here are the essential Bluetooth application layer components:

Applications: Bluetooth applications are software applications or services that operate on Bluetooth-enabled devices. These applications utilize the underlying Bluetooth protocols and services to provide specific functionality and user experiences. Bluetooth applications include applications for file transfer, hands-free communication, fitness monitoring, and smart home control.[11], [12]

Services: Bluetooth services are logical entities that characterize a collection of capabilities and behaviors offered by a Bluetooth device. Services are responsible for specific functionalities, such as audio streaming, data synchronization, or device control, and are implemented using profiles (discussed below). Bluetooth devices can simultaneously provide multiple services, enabling diverse applications and use cases.

Profiles: Bluetooth profiles define the functionality and behavior of a particular application or service. Profiles define the protocols, procedures, and data formats required for interoperable communication between Bluetooth devices. Each profile is customized for a specific use case or application domain, such as the Hands-Free Profile (HFP) for phone call control, the Advanced Audio Distribution Profile (A2DP) for stereo audio streaming, and the Human Interface Device Profile (HID) for connecting input devices such as keyboards and mouse. [12]

User Interfaces: The application layer contains user interfaces (UI) that allow consumers to interact with Bluetooth devices and services. User interfaces can differ based on the type of device and

application. They may include smartphone applications with graphical user interfaces, voice-activated assistants, or tangible controls on Bluetooth devices. User interfaces play a crucial role in delivering a seamless and intuitive Bluetooth user experience.

Application Programming Interfaces (APIs): In the Bluetooth architecture, the application layer is where the actual applications, services, and profiles reside. It specifies the specific functionalities and behaviors with which users interact while leveraging the Bluetooth stack's lower layers for communication and data exchange. By providing a standardized framework for application development and interoperability, the Bluetooth application layer facilitates a vast array of wireless applications and services across a variety of domains. [12]

2.2.3 Bluetooth Operation

Bluetooth networking employs low-powered radio waves to wirelessly transmit data between devices. Communication occurs within a specific 2.45 gigahertz (GHz) frequency band, which ranges from 2.402 GHz to 2.480 GHz. International agreement has designated this frequency band for use by industrial, scientific, and medical devices (ISM). The ISM band at 2.45 GHz was designated for unlicensed use, so devices operating within this frequency range do not require specific licenses or regulatory approvals in the majority of countries. This allocation enables multiple device types to utilize this frequency band for their communication requirements, fostering innovation and extensive adoption. [11], [12]

In the context of Bluetooth, a "piconet" is a network made up of one primary device and one or more secondary devices (sometimes referred to as slaves). The primary device is often a device, such as a computer, smartphone, or other device that starts and manages the connection, while the secondary devices are devices that connect to the primary device and communicate in accordance with its instructions. The word "piconet" is often used in relation to Bluetooth technology, which is frequently utilized for wireless communication between devices close together.

In a piconet with a single secondary, TDMA operation is uncomplicated. The time is divided into 625-microsecond intervals. The primary device transmits data in slots with even numbers (0, 2, 4, etc.), while the secondary device receives data in slots with odd numbers (1, 3, 5, etc.). In other words, the primary and secondary devices alternate sending and receiving data, with no additional devices involved. [11]

In a piconet with multiple secondaries, however, the procedure becomes more complicated. Each secondary device can only send in the next odd-numbered slot if the packet in the preceding slot was designated to it. The primary device continues to use the even-numbered slots. In other words, if a transmission is sent to a specific secondary device, that device will respond in the next available slot with an odd number. If no packets are addressed to a secondary device, it will merely listen on slots with an even number.

Only one secondary device can transmit in each slot with an odd number, requiring multiple secondaries to compete for the opportunity to send data. A system of priority levels handles this, with higher-priority devices taking precedence over lower-priority ones. If two or more devices have the same priority level, a random backoff algorithm will be used to determine who gets the available positions.[10]

2.3 Summary of Zigbee and Bluetooth Protocols

Zigbee is a low-power, low-data-rate protocol designed for short-range wireless communication. It employs the IEEE 802.15.4 physical radio standard and the Zigbee network and application layer stacks. Zigbee is appropriate for applications requiring low power consumption, and minimal data throughput, including home automation, industrial monitoring, and smart energy systems. Multiple devices can establish a network and route messages to each other using mesh networking. Zigbee networks are scalable and can support a significant number of devices (up to thousands). The protocol provides interference resistance and facilitates secure communication via encryption and authentication mechanisms.

The Frequency Hopping Spread Spectrum (FHSS) method is used by Bluetooth. It creates 79 channels, each with a bandwidth of 1 MHz, within the range of frequencies that are accessible. By switching frequencies 1,600 times per second, the Bluetooth device quickly switches between various channels. This technique for frequency hopping reduces interference from other devices using the same frequency band. The impact of signal interference can be reduced, and a steady connection can be maintained by Bluetooth devices by frequently shifting channels. FHSS facilitates dependable communication over brief to medium distances and offers resilience against interference.

ZigBee, however, makes use of Direct Sequence Spread Spectrum (DSSS) technology. It separates the 2.4GHz frequency spectrum into 16 channels, each with a 2 MHz bandwidth. ZigBee devices use a particular channel within the available frequency spectrum to transmit data, as compared to Bluetooth's frequency hopping. To increase resistance to interference and enable the coexistence of several devices within the same frequency range, DSSS distributes the signal throughout the whole channel bandwidth using a special coding. With fewer channels than Bluetooth, ZigBee is better suited for applications like wireless sensor networks and home automation that need lower data rates and less power.

Table 2.1: Comparison of Bluetooth and ZigBee Protocols

	Bluetooth	ZigBee
Version	V5.1, IEEE 802.15.1	IEEE 802.15.4
Frequency Band	2.4GHz	2.4GHz
Number of Channels	79(1MHz)	16(2MHz)
Spreading	FHSS	DSSS
Channel bandwidth	1MHz	2MHz

3. RESEARCH METHODOLOGY

3.1 Zigbee Protocol

Configuring the Digi XBee3 802.15.4 TH modules to operate with the Zigbee protocol was part of the experimental configuration. These modules were programmed and installed to establish a Zigbee communication network. The Zigbee network was established using two computers, each of which contained a Digi XBee3 802.15.4 TH module. Various configuration steps ensured compatibility and network functionality. This involved configuring parameters including network identification, channel selection, security settings, and addressing. The modules were programmed with the configuration parameters required to establish a secure and dependable communication link.

After configuring the modules, USB cables were used to connect them to their corresponding computers. This allowed for efficient communication between the modules and computers. With the configuration and connections established, the Zigbee network became capable of data transmission. Through their respective Digi XBee3 802.15.4 TH modules, the two computers were able to communicate via the Zigbee protocol. This allowed for a seamless exchange of data between the devices through the Zigbee network.

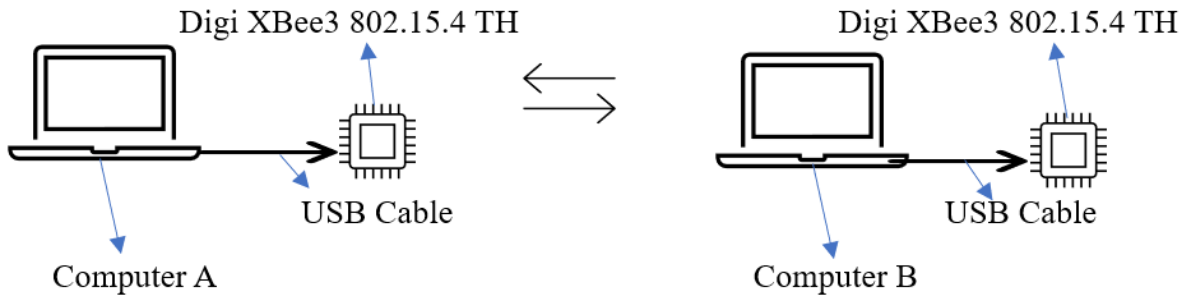

Figure 3.1: Zigbee Network Used

3.1.1 Device Configuration Tool

Digi International has developed XCTU (Xbee Configuration and Test Utility) as a tool for configuring and administering Xbee wireless modules. XCTU provides an intuitive graphical user interface for communicating with Xbee modules connected to a computer via serial or USB. Xbee modules are a popular option for wireless communication in a variety of applications, including the Internet of Things (IoT), industrial automation, and remote sensing.

XCTU's Main Features and Functions

Module Configuration: XCTU permits the configuration of various parameters of Xbee modules, such as network settings, security options, data rate, channel, addressing modes, and other operational settings. You can modify these parameters to meet the needs of your specific undertaking.

Firmware Updates: XCTU enables the firmware update of Xbee modules. Firmware updates can introduce new features, problem corrections, and performance enhancements to a module.

The software makes it simple to locate the most recent firmware versions and update the connected modules.

Network Management: XCTU offers management and monitoring utilities for Xbee networks. You can discover and visualize the network topology, monitor network traffic, and view the status of individual modules. This feature is especially useful for troubleshooting and ensuring that your wireless network operates reliably.

Diagnostic Tools: The software provides diagnostic tools for evaluating the health and functionality of Xbee modules. You can conduct range tests to evaluate signal strength and interference, signal quality metrics, and link tests to validate the connectivity of modules.

Data Monitoring and Communication: XCTU enables the monitoring and transmission of data via Xbee modules. Configuring modules to receive and transmit data packets, monitor data traffic, and perform fundamental data analysis is possible. This feature is useful for validating and verifying the communication between Xbee modules in your application.

Scripting and Automation: XCTU's scripting API facilitates scripting and automation. Using popular programming languages such as Python or JavaScript, you can construct custom scripts to automate duties and streamline your workflow.

Implementation

Discovery of Radio Modules:

The initial stage is to identify the Xbee modules connected to the computer. Initiate the XCTU software and establish a serial or USB connection between the computer and the XBee module. Utilize XCTU's discovery feature to determine the connected module's interface within the software.

Firmware Update:

It is essential to update the XBee module's firmware to obtain problem corrections, feature enhancements, and enhanced performance. Within XCTU, search for the most recent compatible firmware version for the Xbee module. Select the appropriate firmware and proceed with updating the module's firmware.

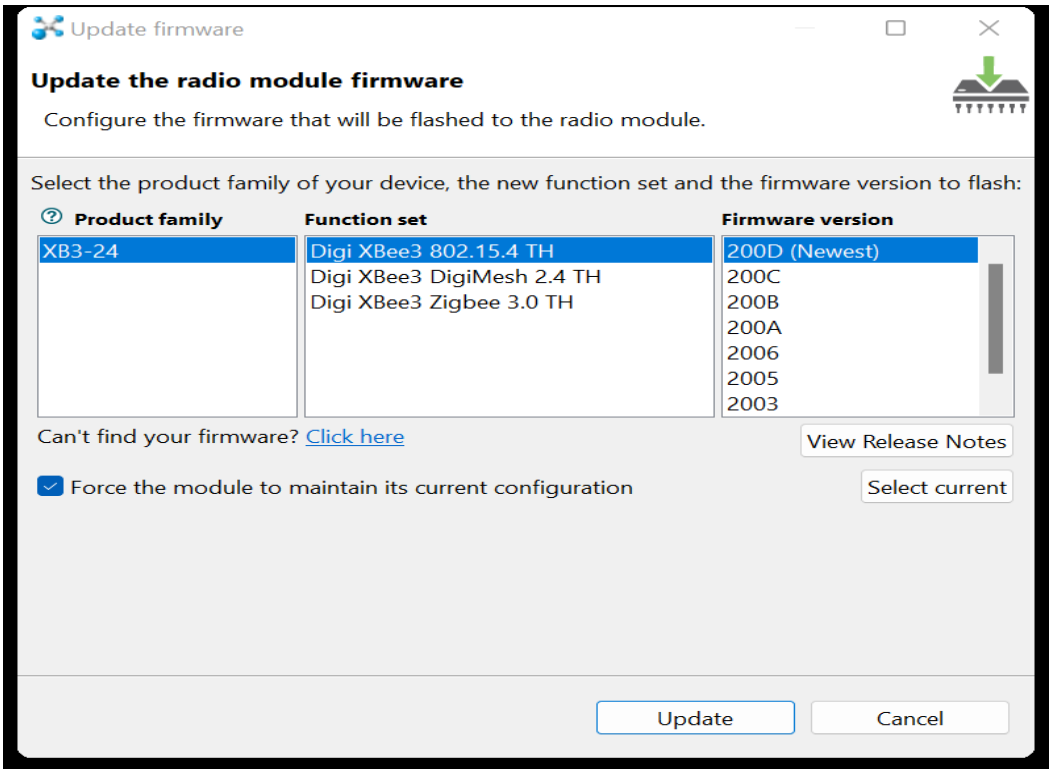

Figure 3.2: Firmware update

Configuration working Mode:

It determines how the module processes and transmits data and how it interacts with the wireless network. There are two typical modes of operation in XCTU:

When operating in transparent mode, an XBee module acts as a serial line replacement. All data received through the serial input is immediately transmitted over the air. When the XBee module receives wireless data, it is sent out through the serial interface exactly as it is received. Communication in transparent mode yields the same result as if the two modules were connected by a wire, but wireless communication makes that physical wire unnecessary.

API Mode (Application Programming Interface Mode): In API mode, the XBee module provides more sophisticated data processing and communication features and capabilities. It enables greater control and flexibility in the management of data packets, routing, addressing, and other network-related operations. API Mode enables developers to interact with the XBee module via API commands and frames.

For communication and data processing in this thesis, The API (Application Programming Interface) mode is used. This mode gives greater control over the communication process by allowing us to interact with the XBee module using specific API frames and commands.

Baud rate:

The speed at which data is transmitted in XBee modules and other serial communication devices is referred to as the baud rate or data rate. Data is transmitted at a slower rate of 9600 bits per second using the 9600 baud rate. It is appropriate for uses where lower data rates are sufficient and power consumption is a consideration. Due to the extra processing time given to each bit, this baud rate may be more tolerant of noisy or unstable communication channels. The 115200 baud rate, on the other hand, is quicker, transmitting data at 115,200 bits per second. It is used for applications that need faster data transmission rates, particularly real-time or high-bandwidth ones. Higher baud rates, however, could be more susceptible to errors in noisy environments or across longer distances since there is less time available to analyze each bit. The baud rate used for the serial connection of the XBee module in the implementation for this thesis is principally 9600 baud. Additionally, a test utilizing a baud rate of 115200 was carried out.

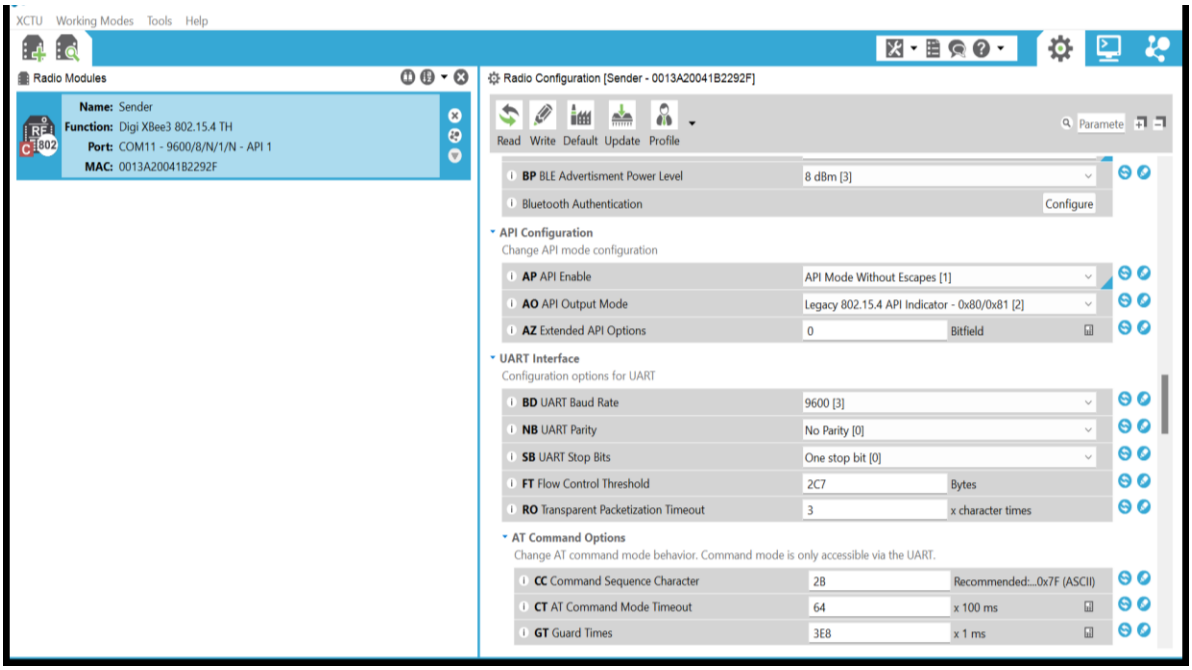

Figure 3.3: API mode and Baud Rate configuration.

Networking Settings: This tab allows you to configure networking-related parameters, including the operating mode (Coordinator, Router, or End Device), addressing mode (16-bit or 64-bit addressing), and network encryption settings. These settings determine the module's behavior within a wireless network.

During the implementation of this thesis, networking-related parameters for XBee modules were also configured. The node identifier 'Sender' was allocated to the sender XBee module to uniquely identify it within the network. It was assigned the function of coordinator, which entails operating as the network controller and instigating and managing communication within the network.

In contrast, the node identifier 'Receiver' was designated to the receiver XBee module to distinguish it from other network devices. It was configured as an End Device, which typically serves as a data

source or receiver within the network and relies on other devices such as routers or coordinators to facilitate communication and data transmission.

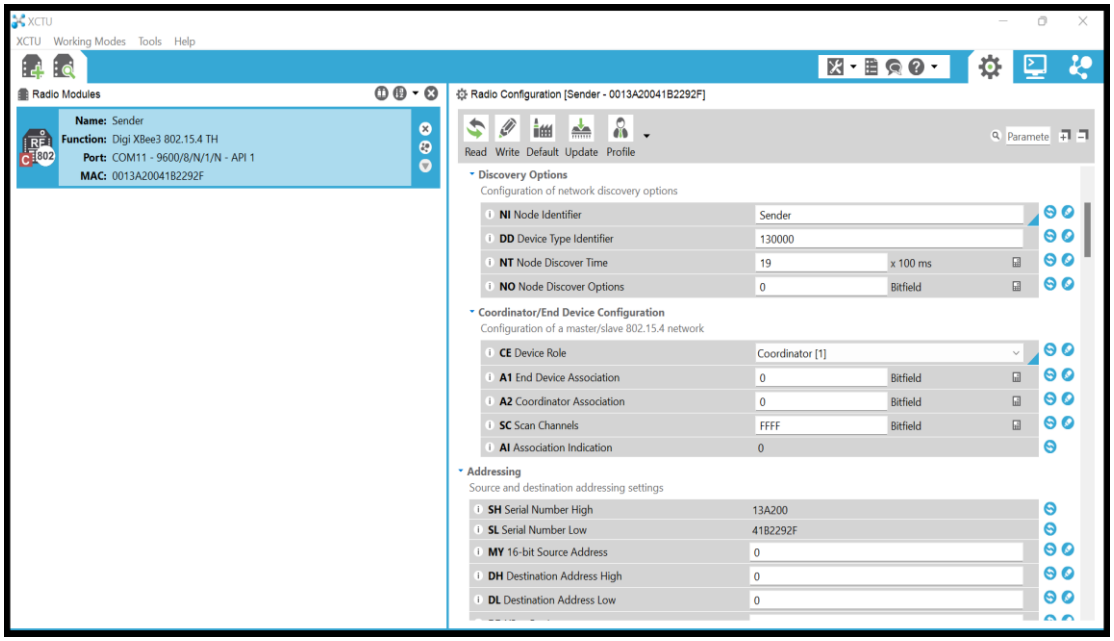

Figure 3.4: Networking Settings.

PAN ID: The PAN ID (Personal Area Network ID) was used to generate a unique identifier for the network of XBee devices in the implementation of this thesis. The PAN ID is essential for communication between devices on the same network. By configuring the XBee devices with the PAN ID, it is ensured that they are part of the same network and can exchange data without interruption.

In this specific implementation, PAN ID 3332 was used. This particular PAN ID was selected to establish a separate network for the XBee devices used in the thesis project. By configuring the PAN ID to 3332, the XBee devices were able to recognize and communicate with one another within the designated network, allowing for efficient data exchange and communication. Setting the PAN ID is a necessary step in configuring XBee devices, as it enables devices to identify and communicate with each other within the specified network.

Table 3.1: Summery of main parameters used.

	Sender	Receiver
Network PAN ID	3332	3332
MAC Mode	802.15.4 + Digi header w/ACKS[0]	802.15.4 + Digi header w/ACKS[0]
Device Role	Coordinator [1]	End Device [0]
API Enable	API Mode Without Escapes[1]	API Mode Without Escapes[1]
UART Parity	9600[3]	9600[3]

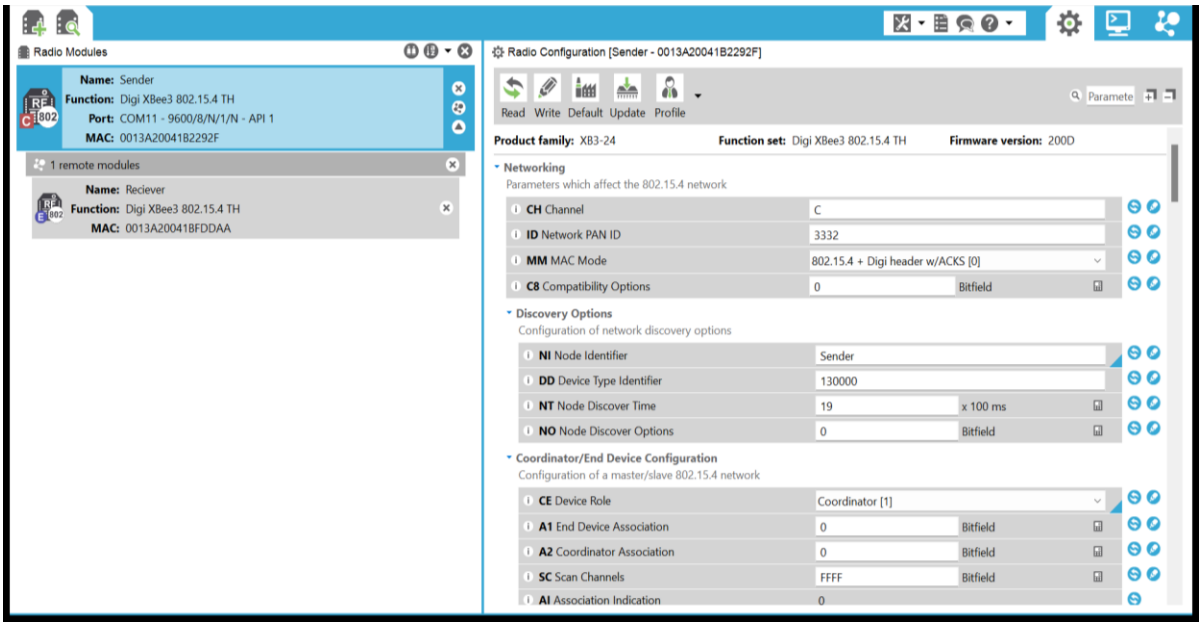

Figure 3.5: PAN ID and Device Role of the Sender

The implementation for this thesis consisted of configuring the receiver XBee module with the same PAN ID of 3332 as the sender module. By utilizing the same PAN ID for both the sender and receiver devices, it was ensured that they belonged to the same network and could establish a connection to exchange data without interruption. This configuration enabled the receiver module to identify and communicate with the sender module within the specified network, facilitating efficient data exchange and establishing a reliable communication link between the XBee devices.

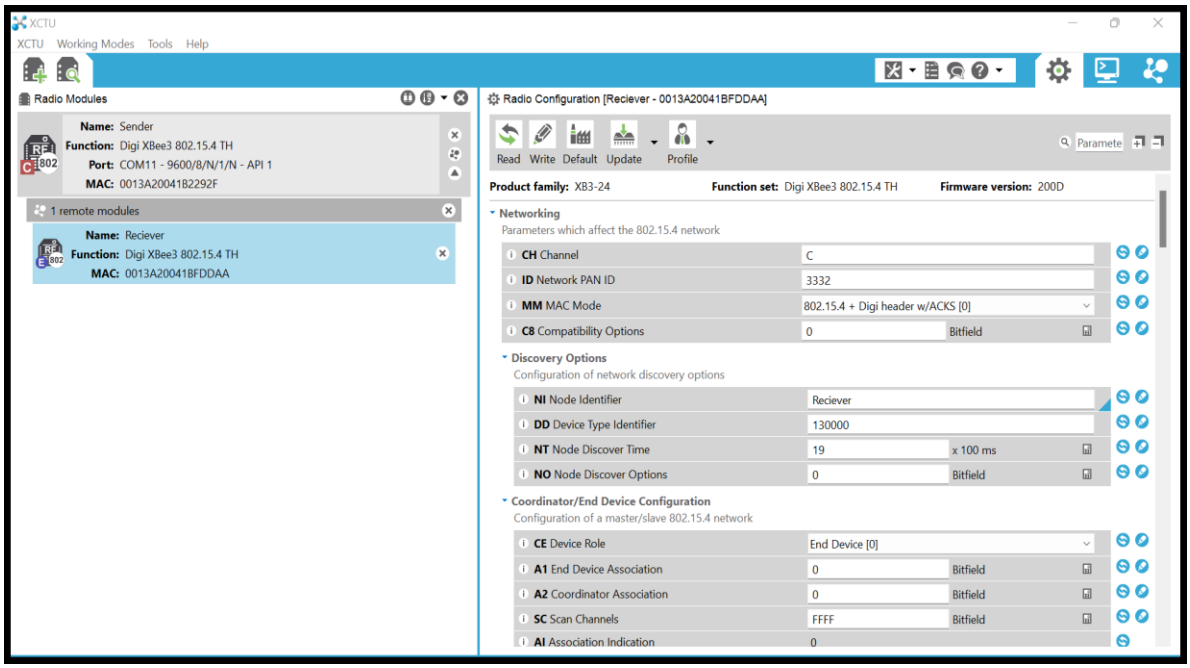

Figure 3.6: PAN ID and Device Role of the Receiver.

3.1.2 Methods Used to Collect Data

In the context of this thesis, it is necessary to provide prior information on the data acquisition instruments and programming language used for the study. Visual Studio Code, a widely recognized and robust code editor that supports multiple programming languages, facilitated the deployment of code. Python was chosen as the primary language for code development in this thesis due to its flexibility and extensive library support for effective data processing and analysis.

XBee modules were employed to facilitate communication between devices. These modules are wireless communication devices that facilitate the transfer of data between various networked devices. XCTU (Xbee Configuration and Test Utility) was used to configure and test these XBee modules. XCTU is a potent software application designed to configure, test, and manage XBee devices. It provides an intuitive interface for configuring various XBee module parameters, including network settings and communication protocols.

After configuring the XBee modules with XCTU, a Python script was developed to generate a frame. A frame functions as a container or payload in wireless communication, transporting data from the originator to the receiver. The Python script generated and organized the desired data frame for wireless transmission between the devices. The content and structure of the frame are determined by the requirements of the employed communication.

Using Visual Studio Code, XCTU, and Python, the thesis project was able to establish effective communication between XBee modules-equipped devices. The Python script facilitated the generation and formatting of frames for wireless data transmission, whereas XCTU enabled the configuration and testing of XBee modules without any complications. The combination of these tools and technologies provided a solid foundation for conducting experiments and analyzing the performance of the thesis's communication system.

The Functionality of The Scripts

The Sender Script

The code provides handling of wireless communication between two XBee devices. One device function as the sender and the other as the receiver. The primary objective of the script is to evaluate the characteristics of this communication by transmitting a predetermined number of frames between the sender and receiver. The code measures the round-trip time (RTT) for each frame, which is the amount of time it takes for a frame to travel from the sender to the receiver and back again. It accomplishes this by documenting the start time before transmitting the frame and the end time shaft receiving the response. The RTT is the difference between the two timestamps.

In addition, the code verifies that the received frame matches the original sent frame. If they match the frame is considered effectively received. If the received frame differs from the sent frame, the frame is considered lost. In such circumstances, the code maintains notes of the lost frame content and writes the information to a CSV file for further analysis.

The code calculates the total time required for all frames, the number of received frames, and the number of missing frames throughout the procedure. After concluding the frame transmission and receiving cycle, it divides the total time by the number of frames to determine the average round-trip time. The packet delivery ratio is also computed by dividing the number of received frames by

the total number of frames. The code provides informative output by printing the average latency, total time, packet delivery ratio, and if any, the number of lost frames. This information aids in evaluating the efficacy and dependability of the XBee module-based communication between the two devices.

The Receiver Script

Upon implementation, the code configures the serial port's parameters, including the port's name and baud rate. This ensures that the sender module and XBee module have a reliable connection. After the serial port has been initialized, an XBee object is created to enable interaction with the XBee module. This object provides functions for transmitting and receiving communications at a high level. The code then specifies the XBee sender's originating address. This address is used to verify the origin of communications sent by the recipient.

The code waits for a message to be received from the receiver within the main loop. Once a message has been received, the source address is compared to the expected sender's address. If the addresses match, the code extracts and stores the message frame from the response. The sender then uses the XBee object to transmit the received frame back to its address as a response. This allows for bidirectional communication between the originator and recipient.

The code also compares received frames to send frames and keeps note of the number of received and missing frames. If a received frame matches the transmitted frame, it is marked as successfully received. Nevertheless, if the received frame differs from the sent frame, it is considered lost, and its content is written to a CSV file for further analysis. By monitoring the received and lost frames, this section enables the evaluation of the communication's reliability and the analysis of any discrepancies in data transmission.

3.2 Bluetooth Protocol

To evaluate the characteristics of Bluetooth communication, a test configuration composed of two Raspberry Pi 4 modules was used. One module functioned as the sender, while the other served as the receiver. This configuration permitted the evaluation of Bluetooth connectivity and data transmission between the two devices. The sender module was responsible for transmitting data packets over Bluetooth, whereas the receiver module was responsible for receiving and processing these packets. Using Raspberry Pi 4 modules that support Bluetooth functionality, a controlled and reliable testing environment was created.

The receiver module uses its Bluetooth capabilities to listen for data packets sent by the transmitter module. It utilized the proper software components to receive and process the data transmissions.

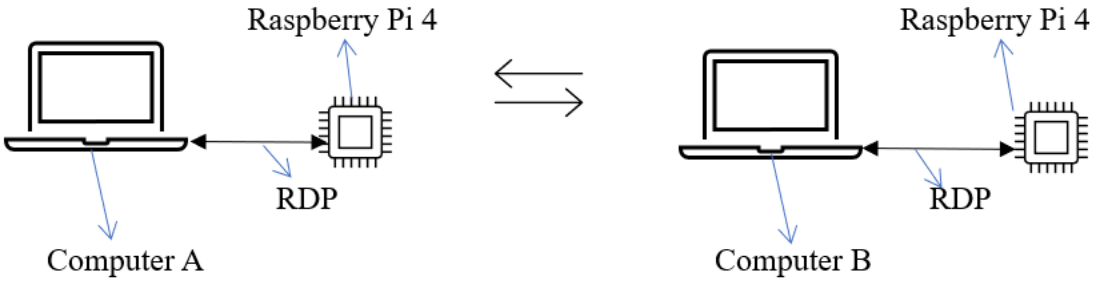

Figure 3.7: Bluetooth Network Used

The Functionality of The Scripts

The Sender Script

The script establishes a Bluetooth connection with a specific device and cyclically communicates data frames. It sends the device a series of frames and waits for a response. The script calculates the round-trip time (RTT) for each frame based on the number of sent frames, received frames, and missing frames. The communication statistics are written in a CSV file. If a frame is lost or if the received response does not match the sent frame, the details of the lost frame are noted. After the communication cycle has concluded, the script terminates the Bluetooth connection.

The Receiver Script

The script initializes a Bluetooth socket, assigns it to a specified port, and waits for a device connection request. The server acknowledges the connection after a device has established it and returns the Bluetooth socket object. The script includes a function responsible for transmitting data frames through the Bluetooth interface to facilitate communication. Encoding and transmitting the frames to the connected device.

The script's fundamental functionality consists of an infinite cycle that manages the communication process. It constantly receives frames from the connected device, transmits them back using the previously described function, and logs receive frames. The script also includes Bluetooth server configuration steps such as setting the socket timeout and specifying the server port. During the execution phase, the script initializes the Bluetooth server, manages the communication with the connected device, and terminates the Bluetooth socket gracefully when the task is complete.

Network Cycles

The network cycle is the repeated sequence of operations involved in transmitting and receiving data frames with the modules. The code establishes a connection and creates communication between the modules. It then transmits a specified number of frames with a fixed size to a receiver module at the destination. Following the transmission of a frame, the code awaits a response from the receiver.

Time Gaps

The time gap is the duration of the delay inserted between the transmission of consecutive data frames. This delay ensures synchronization and prevents data collisions or overlaps by permitting controlled timing between the transmission of each frame. By introducing a time gap, the code simulates a realistic communication scenario in which frames are transmitted and received at a specific rate, allowing for adequate processing and response time. The code's time gap can be modified as necessary to accomplish the intended temporal behavior during the communication cycle.

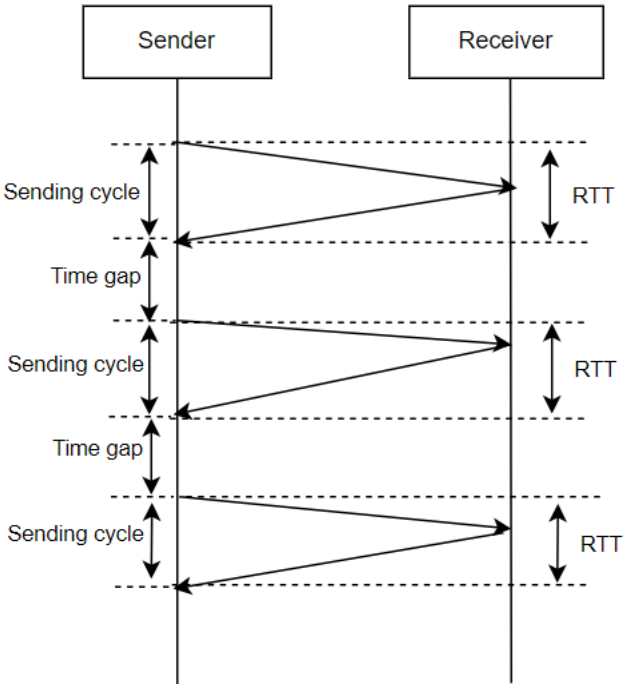

Figure 3.8: Time gaps and Sending cycles.

4. TESTS AND RESULTS

By manipulating frame size and time gap, the latency of Zigbee and Bluetooth protocols was investigated in the conducted test. In industrial communication systems, latency, which is the time required for data to travel from its source to its destination, plays a crucial function. To optimize the efficacy of these protocols in real-world applications, it is necessary to analyze the effects of frame size and time gap on latency.

The effect of data payload size on Zigbee and Bluetooth latency was investigated by varying the frame size, which represents the quantity of data transmitted in each communication packet. Due to the increased quantity of data being transmitted, larger frame sizes typically result in longer transmission times. In addition, the effect of various time gaps on latency was investigated. Time gaps are the durations during which a device is inactive or conserving power by adopting sleep or low-power modes. The purpose of this test was to determine, by varying these time gaps, how latency is affected when devices operate at various activity levels.

It is essential to remember that the specific latency values observed during the test depend on factors such as the implementation of the Zigbee and Bluetooth protocols, the utilized hardware, and the current network conditions. The distance between devices, signal quality, and the presence of obstructions or interference can have a significant impact on the latency experienced in a given environment. The results of this test and the analysis of Zigbee and Bluetooth latency with varying frame sizes and time gaps provide system designers and developers working with these protocols with valuable insights. Understanding the relationship between frame sizes, time gaps, and latency enables informed decision-making to optimize system performance based on the application's specific latency requirements.

4.1: Zigbee Protocol Tests and Results

This study intended to evaluate the characteristic and dependability of the ZigBee module Digi XBee3 802.15.4 TH in an industrial communication scenario. Industrial communication systems play an essential role in assuring the seamless and effective operation of a variety of industrial processes. Therefore, it is crucial to evaluate how various factors, such as time gaps and frame sizes, affect the efficacy of the system.

The objective of the first test was to determine the effect of varying time gaps and a fixed frame size of 60 bytes. The time gap is the interval between the transmission and reception of consecutive frames, whereas frame size is the quantity of data contained in each frame. By analyzing the effects of various time gaps, we can gain insight into the system's behavior and optimize its performance to meet particular communication requirements.

The second test expanded the investigation by contemplating 100-byte frame sizes while still altering the time gap. This evaluation aimed to evaluate the system's characteristics under various conditions, including larger data sizes. The round-trip time, which measures the time required for a frame to travel from the transmitter to the receiver and return, was used as a key metric to assess the latency and responsiveness of the system. In addition, the evaluations sought to identify any

instances of missing frames, which could indicate potential communication system limitations or problems.

The findings from these experiments can still be valuable in enhancing the efficiency and dependability of data transmission in industrial processes. By understanding how the Digi XBee3 module performs in wireless channels, you can evaluate its suitability for industrial applications that require robust and dependable communication. This analysis of the theoretical suitability of Zigbee can provide a foundation for further research and development efforts to adapt and optimize the module for industrial communication scenarios. It is crucial to note that while the experiments concentrate on the wireless channel and theoretical suitability, they can still contribute to enhancing industrial communication in the long run by informing decisions regarding technology selection and configuration.

4.1.1: Tests One with 9600 Baud Rate

To evaluate the characteristics of a Digi xbee3 802.15.4 TH ZigBee module that performs wireless channels under varying time gaps and frame sizes in an industrial communication scenario, the following experiments were conducted. The goal was to determine how varying time gap durations and frame sizes affect the round-trip time of frames and assess the system's ability to manage a variety of communication conditions. Data transmission must be effective and reliable in industrial communication systems for processes to run smoothly.

Time gap refers to the interval between the transmission and reception of consecutive frames, whereas frame size is the number of data contained within each frame. By analyzing the effects of varying time gap durations and frame sizes, one can obtain insight into the system's behavior and optimize its performance for particular needs.

Experiments were conducted with a 60-byte fixed frame size, which is the minimum size of an Ethernet frame for industrial communication protocols. The time gap ranged from 1 millisecond to 256 milliseconds, encompassing a wide variety of possible scenarios. 10,000 frames were transmitted in each experiment, and their round-trip times were measured. It is a crucial metric that reflects the latency and responsiveness of the entire communication system. A shorter round-trip time signifies quicker data transmission and fewer delays. In addition, the experiments aimed to identify instances of missing frames.

Table 4.1.1.1: 60 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	60 bytes	1	0.316	3160.29	No lost frame
2	10,000	60 bytes	2	0.3177	3177.52	No lost frame
3	10,000	60 bytes	4	0.3179	3179.89	No lost frame
4	10,000	60 bytes	16	0.3174	3174.94	No lost frame
5	10,000	60 bytes	256	0.328	3285.35	No lost frame

Performance Metrics of Data Transmission

These graphs are intended to provide a visual analysis of the data collected, showing the relationship between time gaps and average delays. By analyzing the graph's trends and patterns, we can draw insightful conclusions about the system's efficacy and make informed decisions regarding its optimization and improvement.

Results

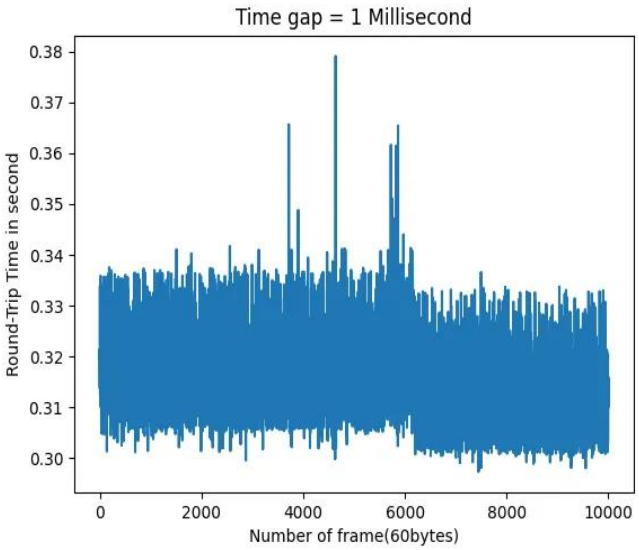

Figure 4.1.1.1: Time gap 1 millisecond (60b)

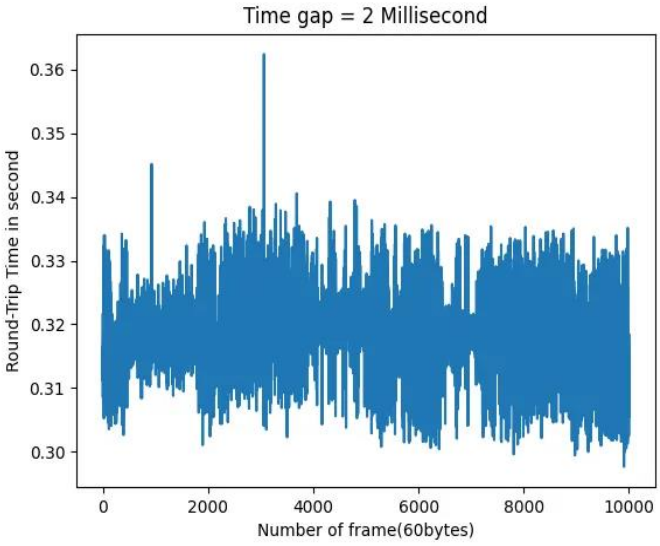

Figure 4.1.1.2: Time gap 2 milliseconds (60b)

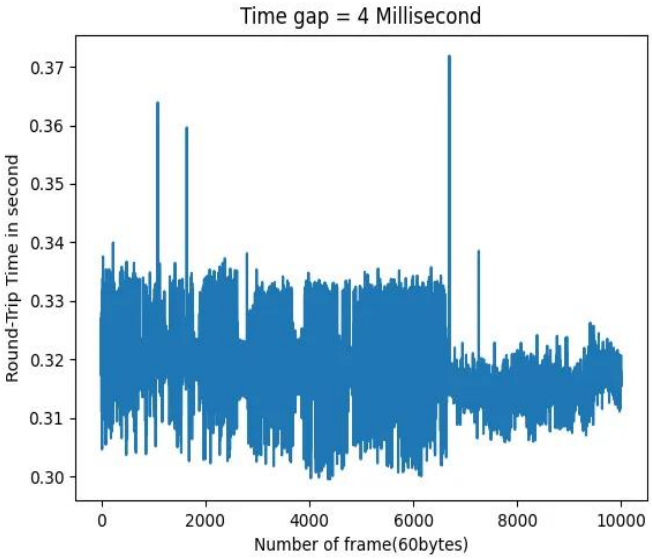

Figure 4.1.1.3: Time gap 4 milliseconds (60b)

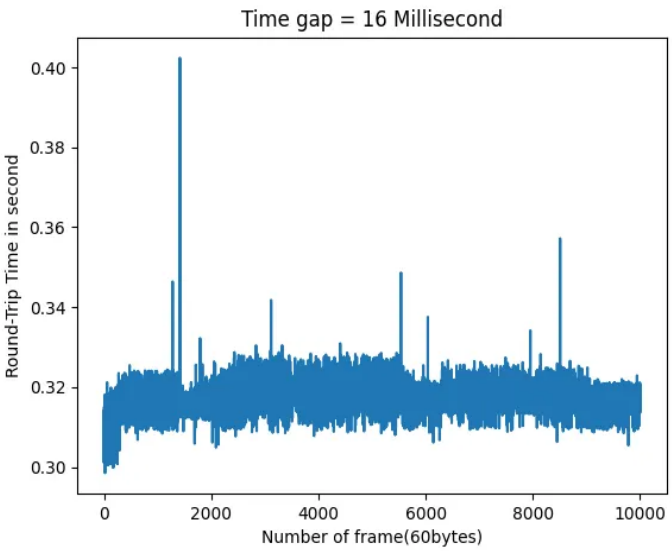

Figure 4.1.1.4: Time gap 16 milliseconds (60b)

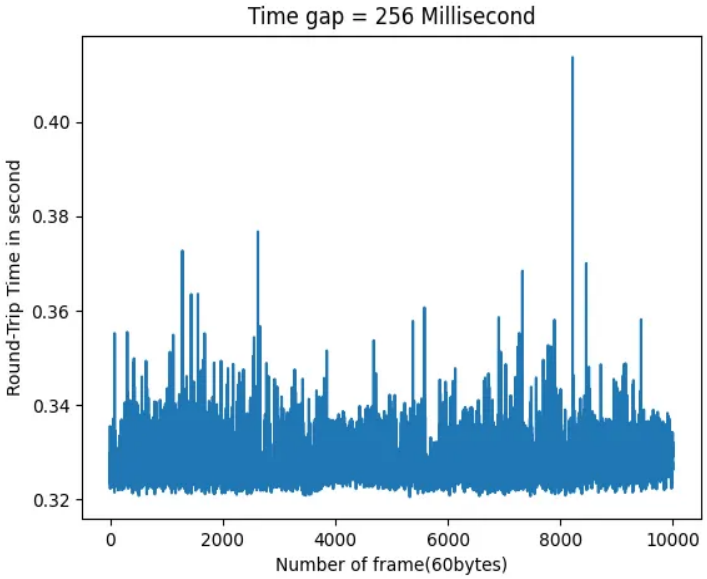

Figure 4.1.1.5: Time gap 256 milliseconds (60b)

The results illustrate the characteristics of a system at various time gaps. The experiments were conducted with a constant frame size of 60 bytes and variable time gaps extending from 1 millisecond to 256 milliseconds. Each examination included 10,000 frames. Transmission is synchronous, in which each frame is sent after the previous frame has been received, which is essential for understanding the experimental setup and the obtained results.

The average round-trip time was calculated in milliseconds after measuring the round-trip time for each frame. No frames were lost during any of the experiments, indicating successful transmission and reception of all frames.

Histogram Representation

Each data point on the graph represents the average round-trip time observed for a particular configuration of time gap and frame size. The graph enables a fast comparison of round-trip time performance across various time gaps. Analyzing the graph makes any substantial variations in round-trip time based on the selected time gaps and frame size easier to identify. It can also aid in making informed decisions regarding system configuration, such as selecting an appropriate time gap and optimizing frame size to accomplish the desired levels of performance. The graphs provide a visual representation of the experimental results.

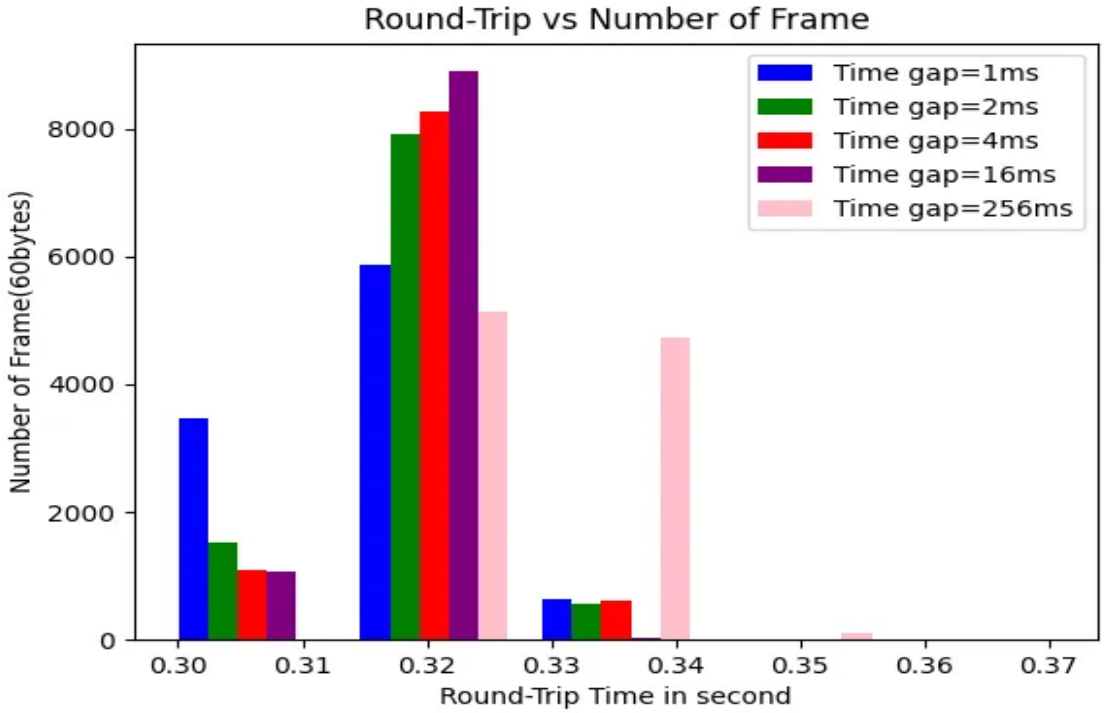

Figure 4.1.1.6: 60 bytes of the frame with a time gap from 1 millisecond to 256 milliseconds.

4.1.2: Tests Two With 9600 Baud Rate

The objective of the second test was to evaluate the characteristics of the Digi XBee3 802.15.4 TH ZigBee module performs wireless channels in an industrial communication scenario with variable time gaps and frame sizes. The objective was to determine how various time gap durations and larger frame sizes affect the round-trip time of frames and the system's ability to manage a variety of communication conditions. In industrial communication systems, data transmission must be efficient and dependable for process operations to run smoothly. By conducting these experiments, we intended to gain insight into the system's behavior and optimize its performance based on a set of predetermined criteria.

The second test concentrated on a frame size of 100 bytes, which is larger than the standard frame size of 60 bytes utilized by industrial communication protocols. The time gap ranged from 1 millisecond to 256 milliseconds, representing a wide variety of potential scenarios. Similar to the initial experiment, 10,000 frames were transmitted, and their round-trip times were measured.

The round-trip time represents the entire amount of time required for a frame to travel from the transmitter to the receiver and return. It is a crucial metric that reflects the latency and responsiveness of the system.

A shortened round-trip time indicates more rapid data transmission with fewer delays. In addition to measuring round-trip durations, the second test sought to identify any lost frames, which could indicate potential communication system issues or limitations. By conducting these experiments with varying frame sizes and time gaps, we can obtain a deeper understanding of the system's performance characteristics and optimize its configuration for specific industrial communication requirements.

Table 4.1.2.1: 100 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	100 bytes	1	0.477	477.12	No lost frame
2	10,000	100 bytes	2	0.480	4801.13	No lost frame
3	10,000	100 bytes	4	0.480	4801.08	No lost frame
4	10,000	100 bytes	16	0.480	4806.26	No lost frame
5	10,000	100 bytes	256	0.4797	4797.573	No lost frame

Results

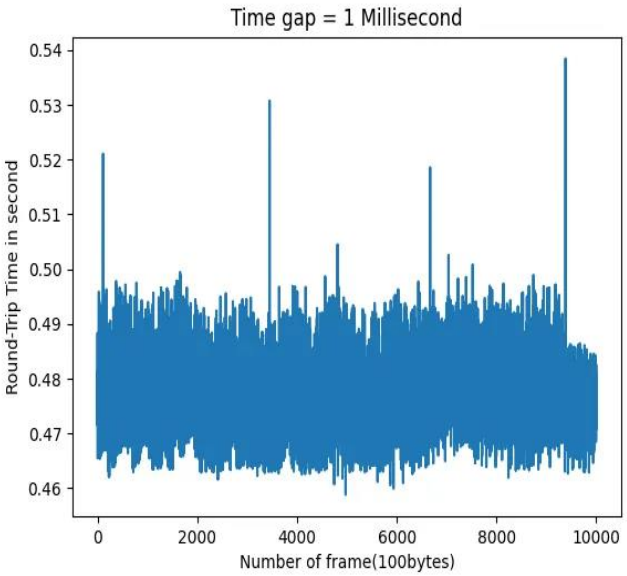

Figure 4.1.2.1: Time gap 1 millisecond (100b)

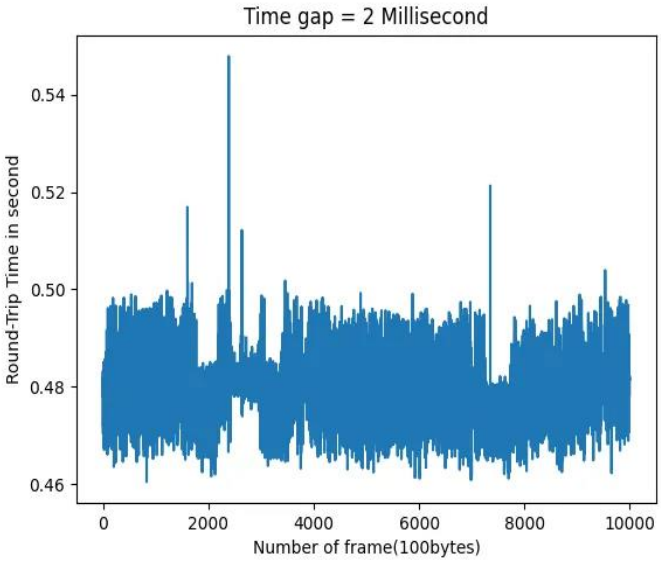

Figure 4.1.2.2: Time gap 2 milliseconds (100b)

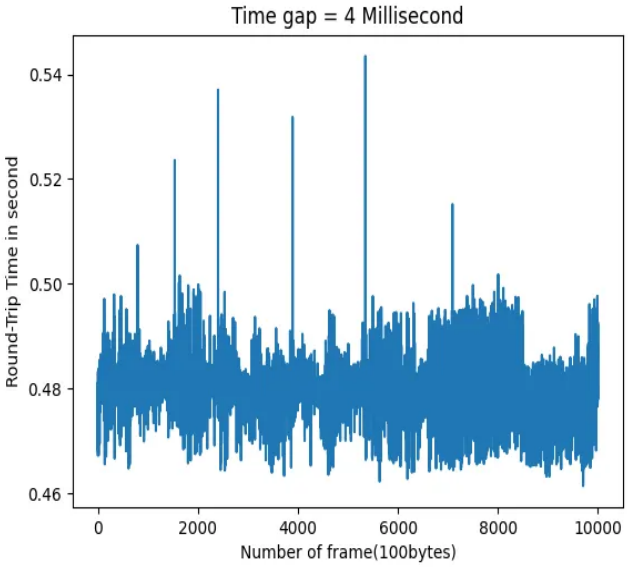

Figure 4.1.2.3: Time gap 4 milliseconds (100b)

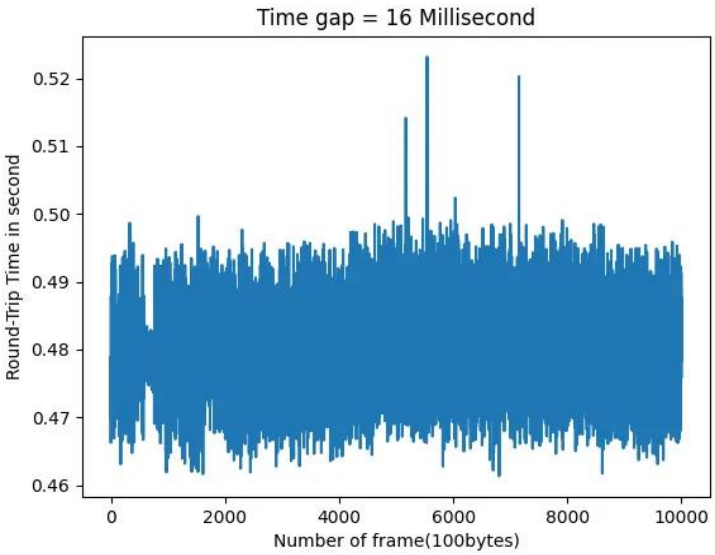

Figure 4.1.2.4: Time gap 16 milliseconds (100b)

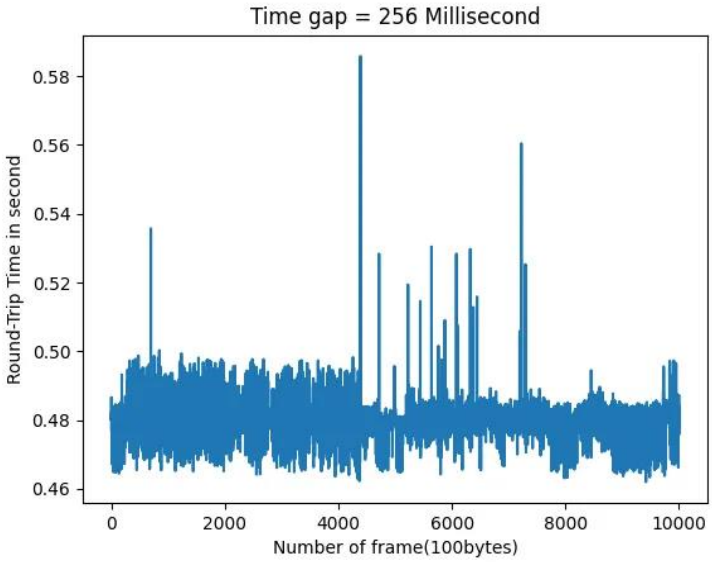

Figure 4.1.2.5: Time gap 256 milliseconds (100b)

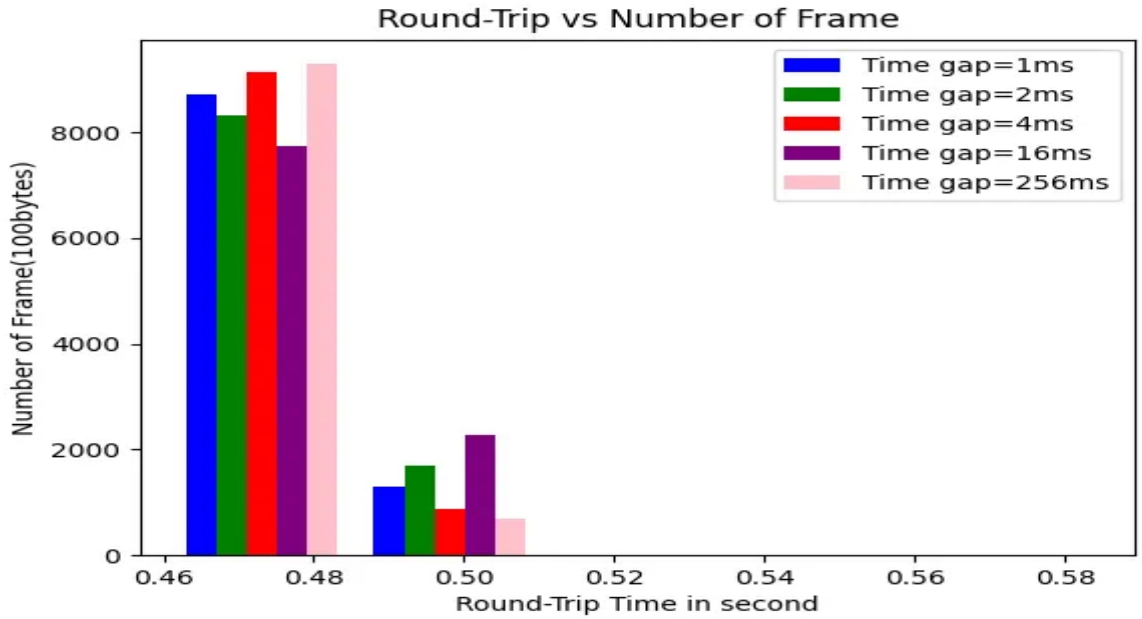

Figure 4.1.2.6: 100 bytes of the frame with a time gap from 1 millisecond to 256 milliseconds.

Throughout all testing, the typical frame transmission latency was roughly 0.480 seconds. A 1-millisecond gap required 477.12 seconds of transmission time, a 2-millisecond gap required 4801.13 seconds, a 4-millisecond gap required 4801.08 seconds, a 16-millisecond gap required 4806.26 seconds, and a 256-millisecond gap required 4797.573 seconds. Notably, in all of the tests, there were no lost frames. In conclusion, the tests showed a constant average delay in frame transmission and highlighted that the overall transmission time was influenced by the different time gaps, while having no lost frames underlined the consistency of the transmission process in all time gap settings.

4.1.3: Tests Three With 115200 Baud Rate

This experiment focused on frame sizes of 60 bytes and time gaps ranging from 1 millisecond to 256 milliseconds. In each measurement, a total of 10,000 frames with a 60-byte frame size were transmitted using bound rate of 115200. The results of this experiment are summarized below.

Table 4.1.3.1: 60 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	60 bytes	1	0.0520	520.219	No lost frame
2	10,000	60 bytes	2	0.0507	507.260	No lost frame
3	10,000	60 bytes	4	0.0527	527.644	No lost frame
4	10,000	60 bytes	16	0.0502	502.898	No lost frame
5	10,000	60 bytes	256	0.265	2658.836	No lost frame

Results

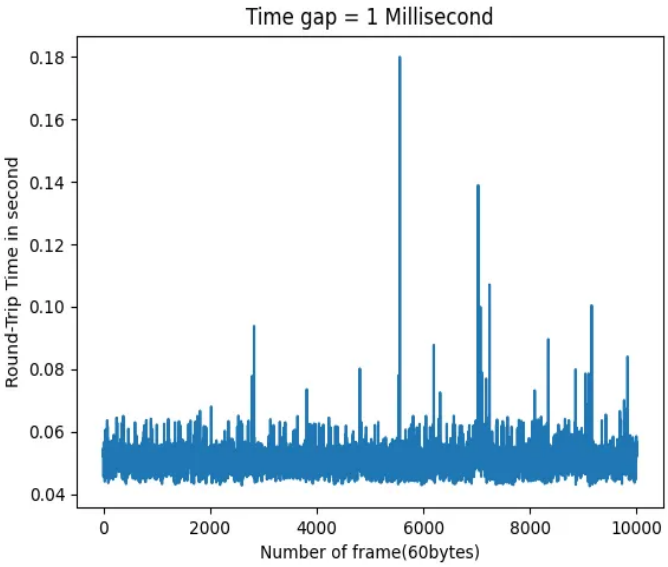

Figure 4.1.3.1: Time gap 1 millisecond (60b)

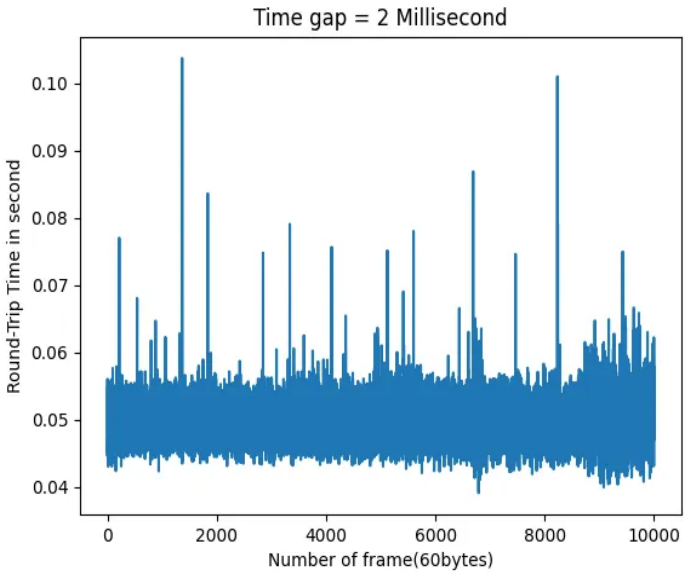

Figure 4.1.3.2: Time gap 2 milliseconds (60b)

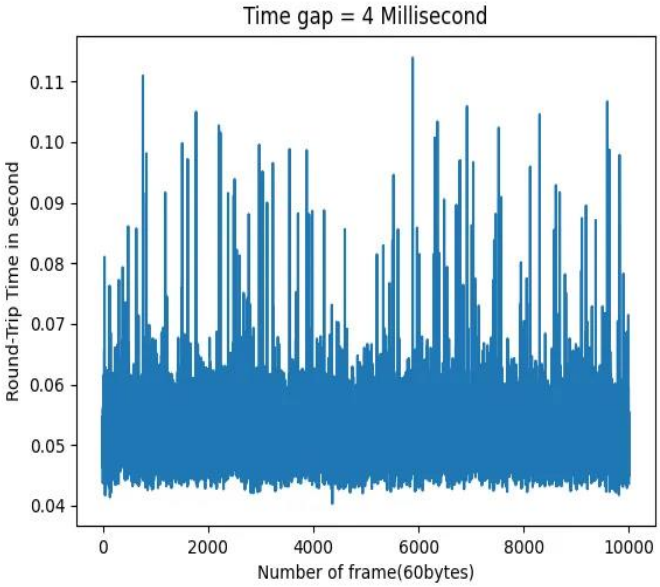

Figure 4.1.3.3: Time gap 4 milliseconds (60b)

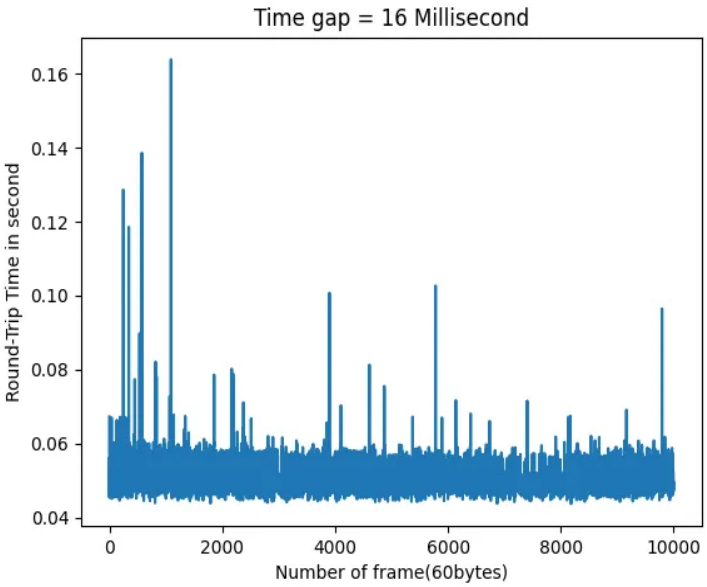

Figure 4.1.3.4: Time gap 16 milliseconds (60b)

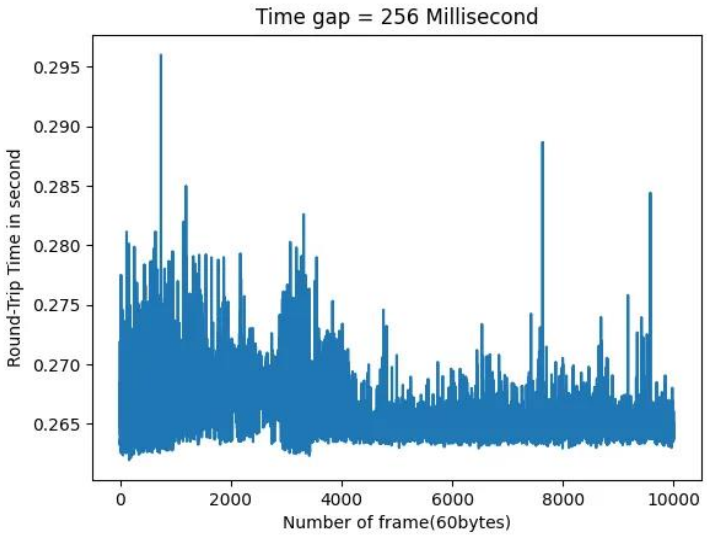

Figure 4.1.3.5: Time gap 256 milliseconds (60b)

No frames were lost throughout any test, which is a sign of dependable data transfer. The time gaps between frames ranged in length from 50.2 to 52.7 milliseconds on average, with the lowest delay being 16 milliseconds and the largest being 4 milliseconds. Each test took a different amount of time to complete; the 16-millisecond test took the lowest amount of time (502.898 seconds), while the 256-millisecond test took the largest amount of time (2,658.836 seconds).

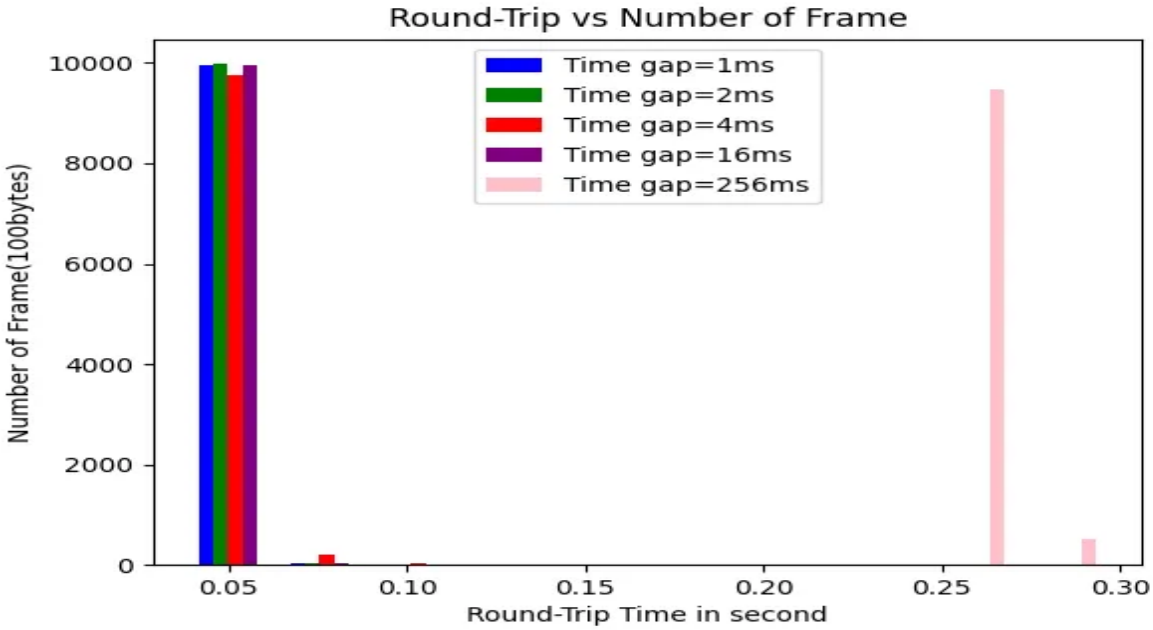

Figure 4.1.3.6: 100 bytes of the frame with a time gap from 1 millisecond to 256 milliseconds.

These results indicate that, even at shorter time intervals between frames, the communication channel was able to manage the data transfer without any frame losses. It is crucial to remember that the longer time gaps led to longer test durations overall and greater average delays.

4.1.4: Tests Four With 115200 Baud Rate

This experiment focused on frame sizes of 100 bytes and time gaps ranging from 1 millisecond to 256 milliseconds. In each measurement, a total of 10,000 frames with a 60-byte frame size were transmitted using bound rate of 115200. The results of this experiment are summarized below.

Table 4.1.4.1: 60 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	100 bytes	1	0.0697	697.085	No lost frame
2	10,000	100 bytes	2	0.0697	697.775	No lost frame
3	10,000	100 bytes	4	0.0667	667.887	No lost frame
4	10,000	100 bytes	16	0.0668	661.102	No lost frame
5	10,000	100 bytes	256	0.268	2680.10	No lost frame

Results

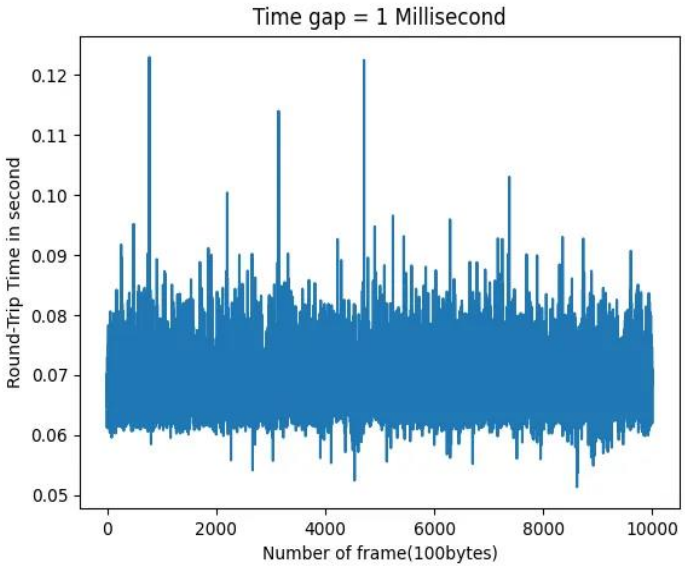

Figure 4.1.4.1: Time gap 1 millisecond (100b)

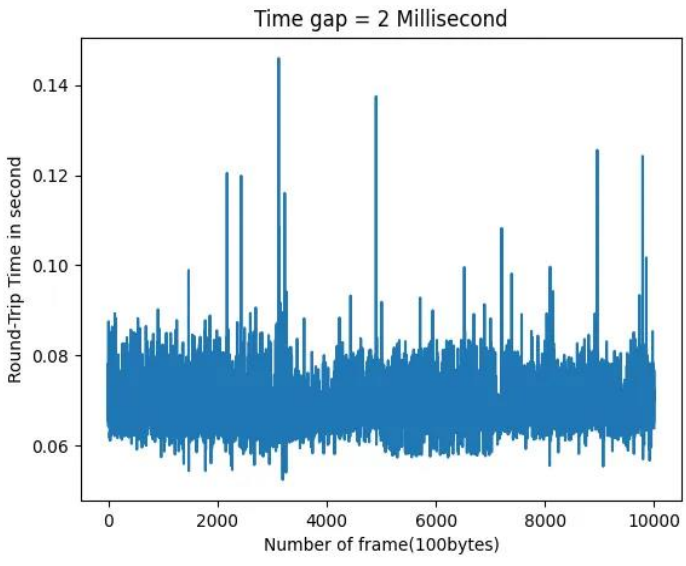

Figure 4.1.4.2: Time gap 2 milliseconds (100b)

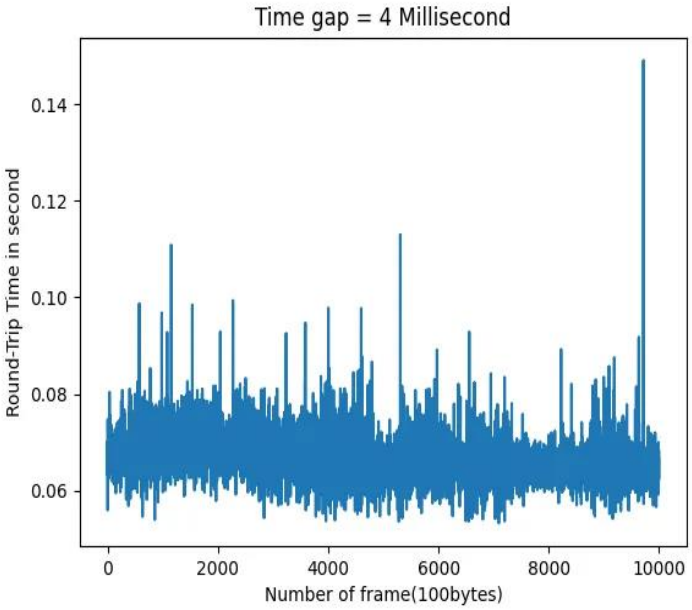

Figure 4.1.4.3: Time gap 4 milliseconds (100b)

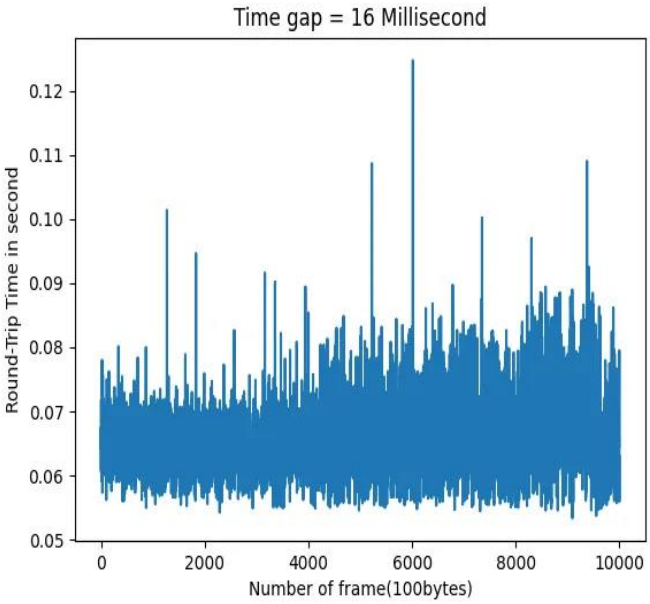

Figure 4.1.4.4: Time gap 16 milliseconds (100b)

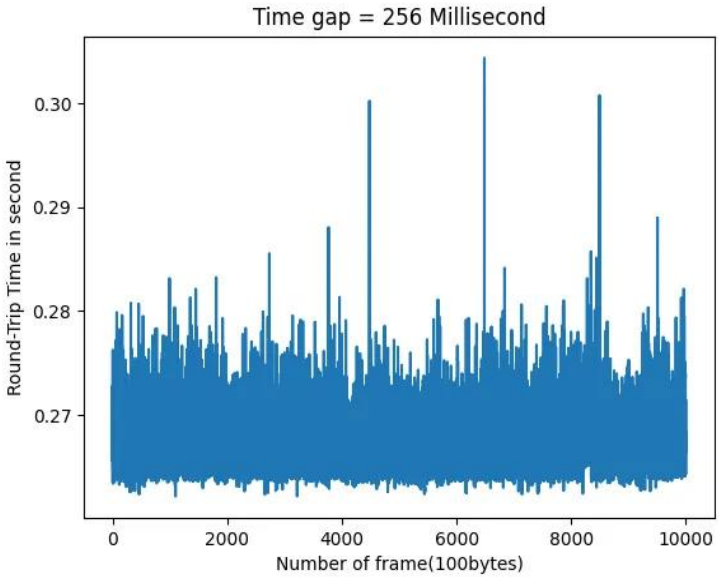

Figure 4.1.4.5: Time gap 256 milliseconds (100b)

It is shown that when the time gap between frame transmissions increases, the average delay and total time taken likewise increase, although missing frames do not occur in any of the tests. Test 1 takes 697.085 seconds to transmit all of the frames, with a time gap of 1 millisecond, resulting in an average delay of 0.0697 seconds. The findings of Test 2, which has a time difference of 2 milliseconds, are nearly similar, with an average delay of 0.0697 seconds and a total transmission time of 697.775 seconds. A 4 milliseconds time gap in Test 3 results in an average delay that is 0.0667 seconds less on average and a transmission time that is 667.887 seconds overall. With a time gap of 16 milliseconds in Test 4, the delay is much shorter, averaging 0.0668 seconds with a total transmission duration of 661.102 seconds. A major influence on delay seems to occur when the time gap widens in Test 5, which has a time gap of 256 milliseconds. This test exhibits a much greater average delay of 0.268 seconds and a total transmission time of 2680.10 seconds.

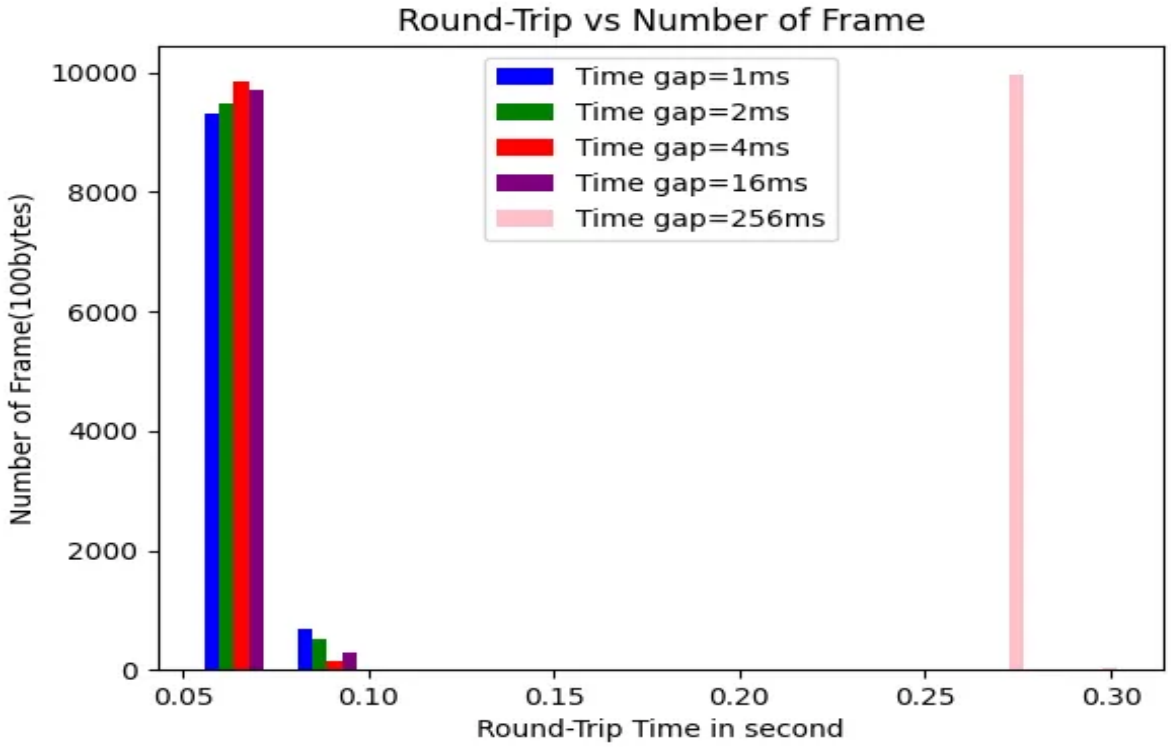

Figure 4.1.4.6: 100 bytes of the frame with a time gap from 1 millisecond to 256 milliseconds.

The Digi XBee3 802.15.4 TH ZigBee module supports a maximum payload size of 110 bytes. This indicates that the module is capable of transmitting and receiving data packets up to 110 bytes in size. A payload is the data contained within a communication packet, excluding any overhead or control data. In the case of the ZigBee module, the payload size indicates the quantity of user data that can be transmitted or received in a single frame.

The Digi XBee3 module supports a maximum payload size of 110 bytes, allowing for the transmission of larger quantities of data within a single communication packet. This can be advantageous in applications requiring the exchange of large amounts of data, such as industrial monitoring and control systems, sensor networks, and home automation. The quantity of data that

can be transmitted in a single frame is constrained by the utmost payload capacity. If the data to be transmitted exceeds the maximum payload size, it must be fragmented into multiple frames or divided into chunks before being transmitted. This fragmentation process can introduce extra overhead and may necessitate additional processing time and resources on both the transmitter and receiver end.

Additionally, larger payload capacities may affect the overall efficacy and throughput of the communication system. As the quantity of the payload increases, so does the transmission duration for each frame. Especially when transmitting large quantities of data, this can result in increased latency and lengthier round-trip durations. It is essential to carefully balance the payload capacity with the application's intended performance requirements.

4.2: Bluetooth Protocol Tests and Results

Bluetooth is a widely adopted wireless communication standard that enables devices to exchange information over brief distances. Using a Raspberry Pi 4 as the test platform, a series of experiments were conducted to evaluate the characteristics and dependability of the Bluetooth protocol. The purpose of these experiments was to evaluate the Bluetooth protocol's behavior under various conditions, concentrating on frame sizes and time gaps. This evaluation tested frame sizes of 60 bytes, 124 bytes, 252 bytes, and 508 bytes. The quantity of data that can be transmitted in a single Bluetooth transmission is referred to as the frame size. We can observe how the protocol manages varying data loading and determine if there are any limitations or performance differences based on frame size by evaluating multiple frame sizes.

In addition to frame dimensions, the evaluations also considered the transmission interval between frames. The time gap refers to the interval between transmissions of Bluetooth packets. The objective of the experiments was to determine the impact of transmission delays on the overall performance and dependability of the Bluetooth protocol by varying the time gaps within a range of 1 millisecond to 256 milliseconds. As the testing platform, a Raspberry Pi 4, a renowned single-board computer, was utilized in the test configuration. The Raspberry Pi 4 offers a flexible and user-friendly environment for Bluetooth protocol evaluations, enabling precise control over test parameters and measurements. Throughout the evaluations, various performance metrics, including latency, and packet loss, were evaluated. These metrics shed light on the efficacy and dependability of the Bluetooth protocol under various conditions. By analyzing the results, we can gain a deeper comprehension of the Bluetooth protocol's capabilities and identify potential enhancement or optimization opportunities.

4.2.1: Test One

The Bluetooth protocol evaluation experiments were conducted on a Raspberry Pi 4 to assess the characteristics and dependability of Bluetooth communication under a variety of circumstances. This experiment focused on frame sizes of 60 bytes and time gaps ranging from 1 millisecond to 256 milliseconds. In each measurement, a total of 10,000 frames with a 60-byte frame size were transmitted.

Table 4.2.1.1: 60 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	60 bytes	1	0.0114	150.316	No lost frame
2	10,000	60 bytes	2	0.0131	165.313	No lost frame
3	10,000	60 bytes	4	0.0361	413.523	No lost frame
4	10,000	60 bytes	16	0.0265	437.868	No lost frame
5	10,000	60 bytes	256	0.0218	2793.39	No lost frame

Results

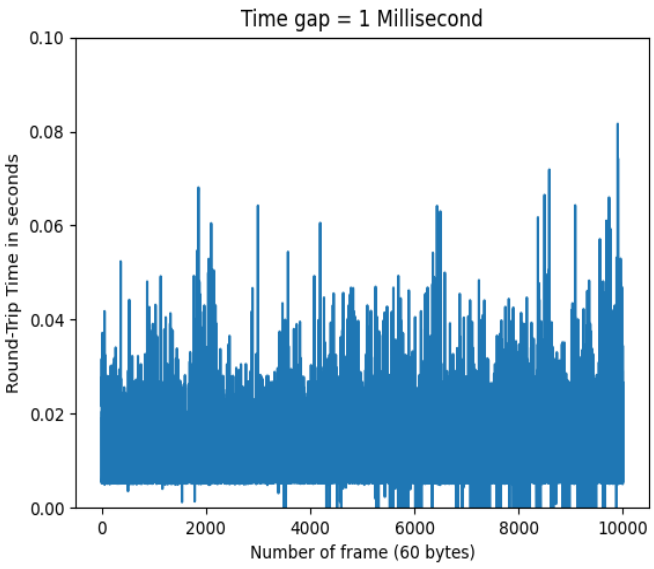

Figure 4.2.1.1: Time gap 1 millisecond (60b)

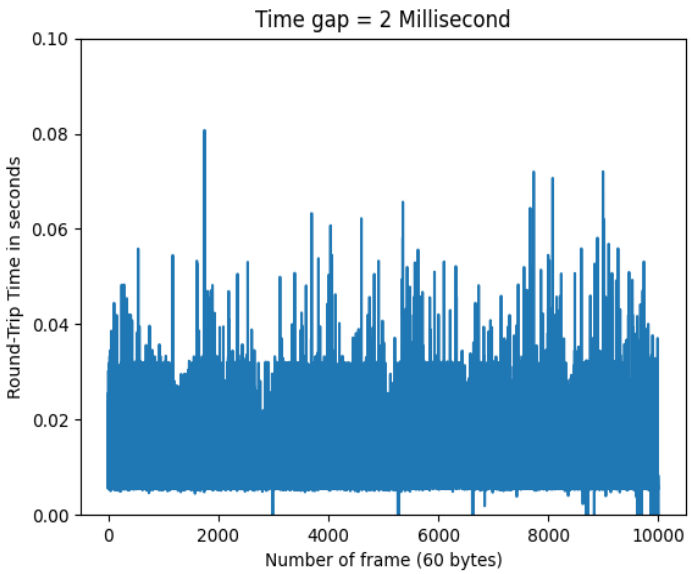

Figure 4.2.1.2: Time gap 2 milliseconds (60b)

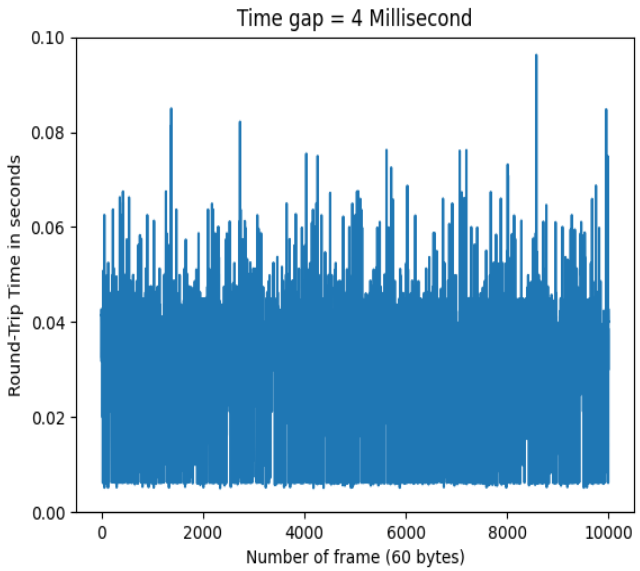

Figure 4.2.1.3: Time gap 4 milliseconds (60b)

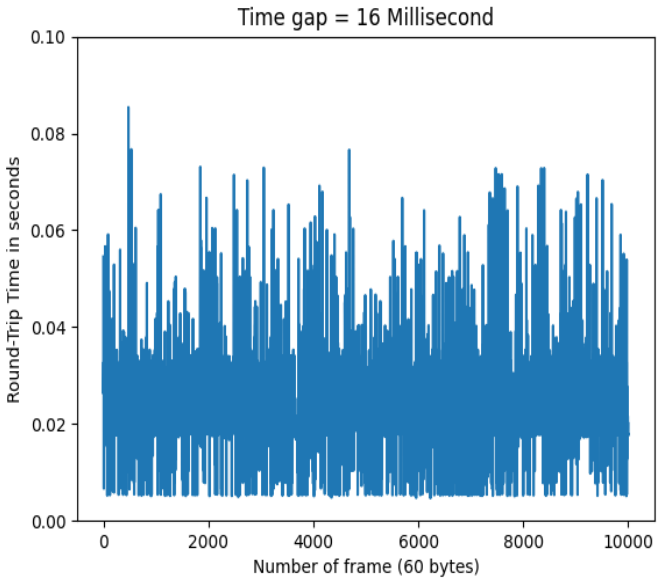

Figure 4.2.1.4: Time gap 16milliseconds (60b)

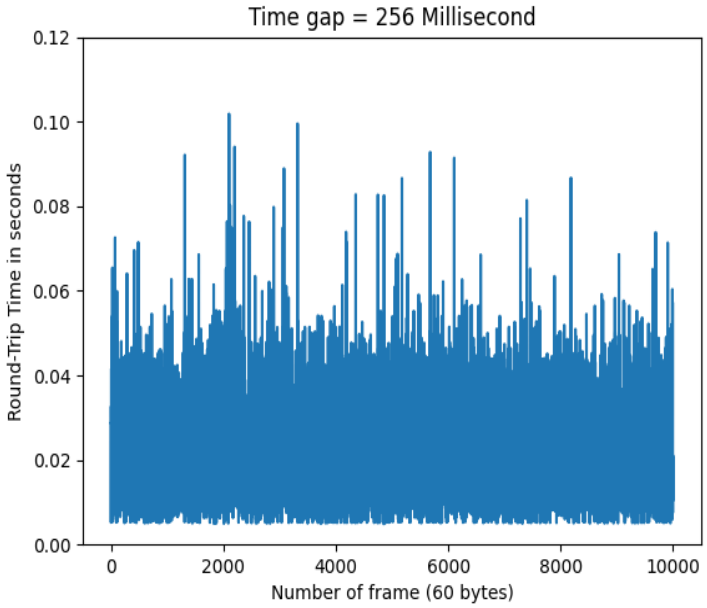

Figure 4.2.1.5: Time gap 256 milliseconds (60b)

Histogram Representation

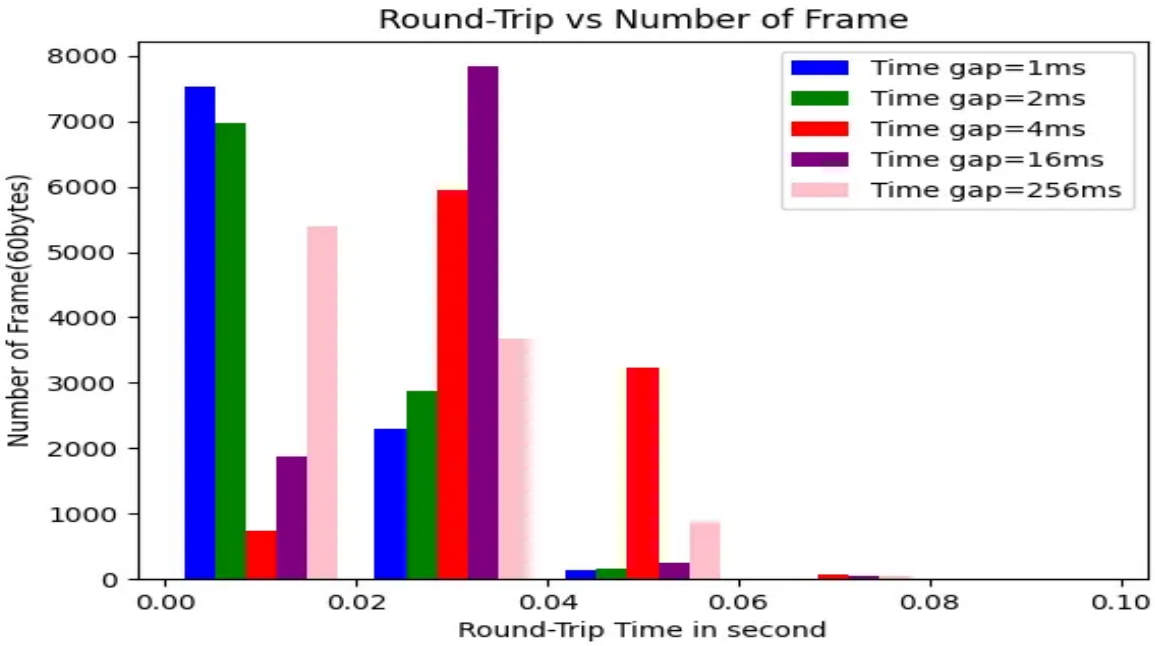

Figure 4.2.1.6: 60 bytes of Frame with a time gap from 1 millisecond to 256 milliseconds

Throughout the testing, there was variation in the average frame transmission delay, which ranged from 0.0114 seconds for a 1-millisecond gap to 0.0361 seconds for a 4-millisecond gap. With timings of 150.316 seconds, 165.313 seconds, 413.523 seconds, 437.868 seconds, and 2793.39 seconds reported for the various experiments, the overall transmission time likewise displayed differences. Importantly, in all of the tests, no frames got lost during transmission, demonstrating the dependability and consistency of the transmission process under a range of time gap settings.

In conclusion, the tests showed a range of average frame transmission delays and showed how different time gaps influenced the overall transmission duration. In spite of these changes, there were no lost frames in any of the experiments, proving the durability of the transmission mechanism no matter the used time gap.

4.2.2: Test Two

This experiment focused on frame sizes of 124 bytes and time gaps ranging from 1 millisecond to 256 milliseconds. In each measurement, a total of 10,000 frames with a 124-byte frame size were transmitted.

Table 4.2.2.1: 124 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	124 bytes	1	0.0189	207.659	No lost frame
2	10,000	124 bytes	2	0.0181	216.287	No lost frame
3	10,000	124 bytes	4	0.0237	289.513	No lost frame
4	10,000	124 bytes	16	0.0275	450.834	No lost frame
5	10,000	124 bytes	256	0.0232	2806.70	No lost frame

Results

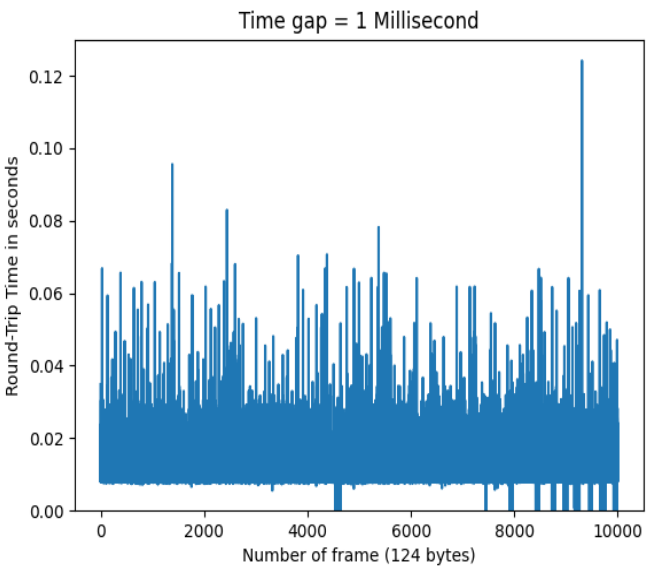

Figure 4.2.2.1: Time gap 1 millisecond (124b)

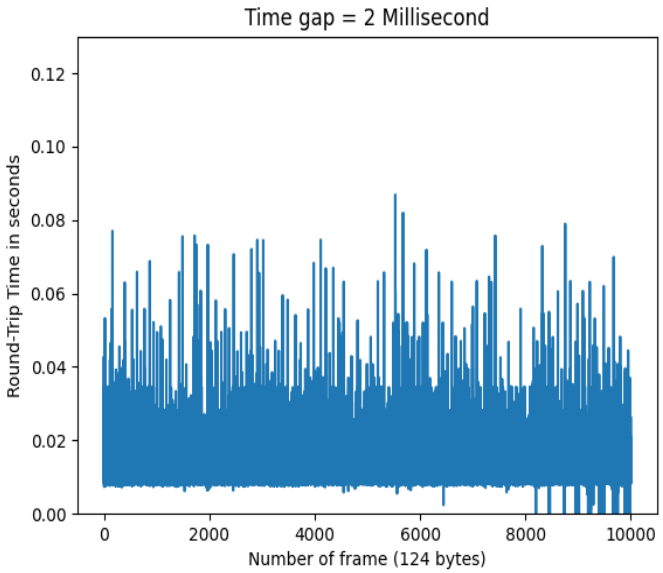

Figure 4.2.2.2: Time gap 2 milliseconds (124b)

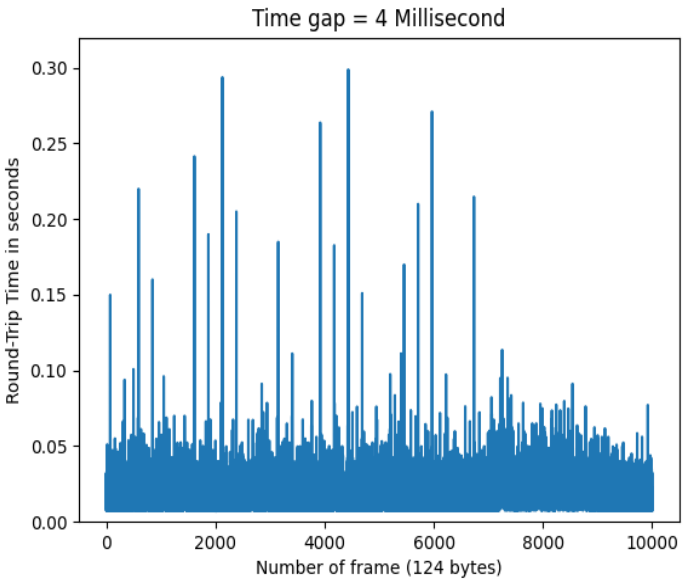

Figure 4.2.2.3: Time gap 4 milliseconds (124b)

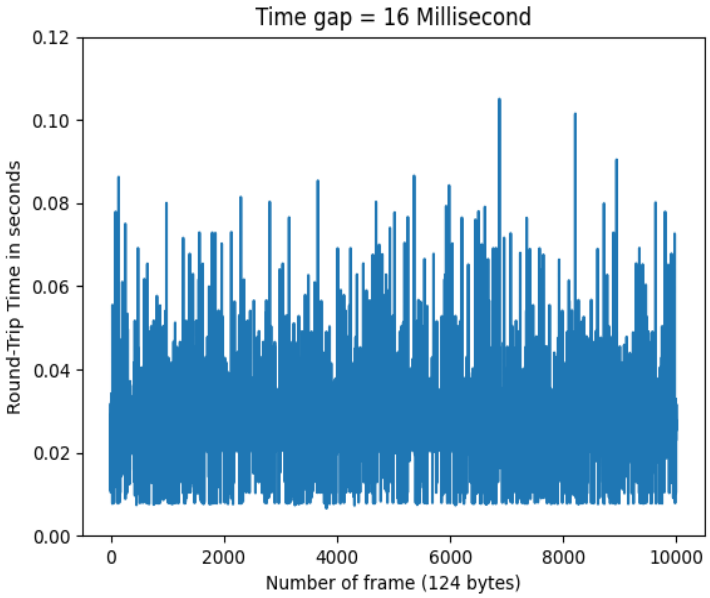

Figure 4.2.2.4: Time gap 16 milliseconds (124b)

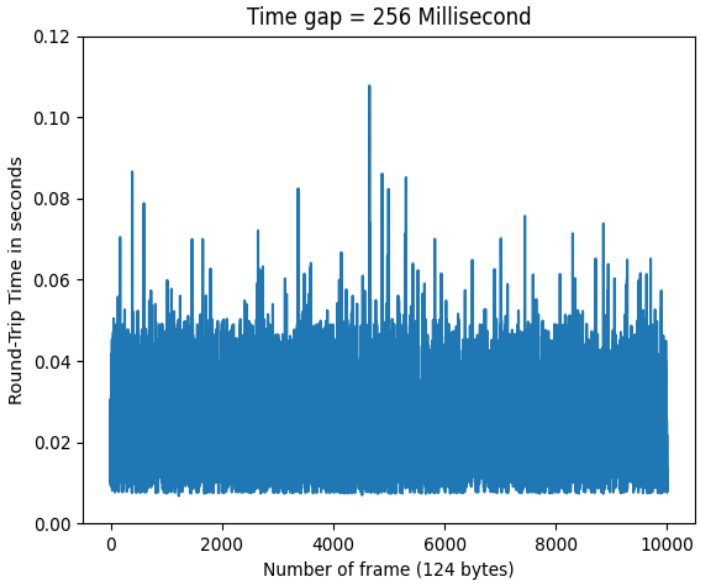

Figure 4.2.2.5: Time gap 256 milliseconds (124b)

Histogram Representation

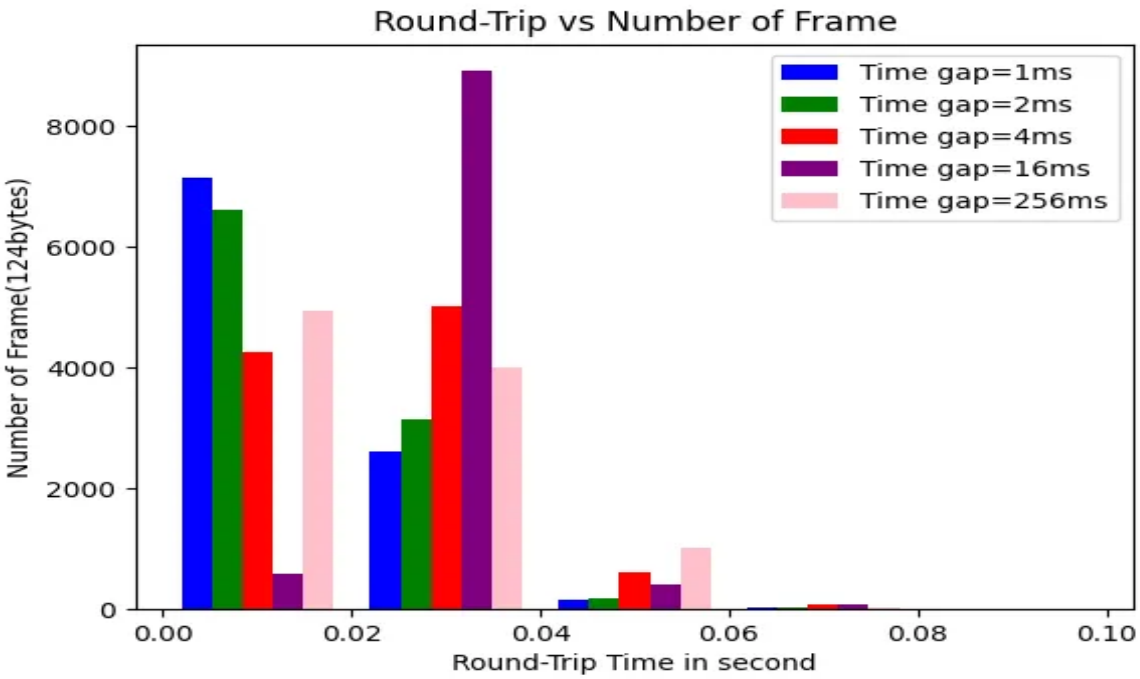

Figure 4.2.2.6: 124 bytes of Frame with a time gap of 1 millisecond to 256 milliseconds

Throughout the testing, the average frame transmission latency fluctuated subtly between 0.0181 seconds and 0.0275 seconds. The total time for entire transmission, which ranged from 207.659 seconds to 2806.70 seconds, similarly showed proportionate fluctuations. The non-occurrence of any frame loss during transmission was a critical constant throughout these tests, illustrating indisputably the transmission mechanism's unwavering dependability and durability across a variety of time gap scenarios. In conclusion, the testing showed changes in average transmission delays and total transmission durations that were related to the different time intervals used.

4.2.3: Test Three

This experiment focused on frame sizes of 252 bytes and time gaps ranging from 1 millisecond to 256 milliseconds. In each measurement, a total of 10,000 frames with a 252-byte frame size were transmitted.

Table 4.2.3.1: 252 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	252 bytes	1	0.0508	529.982	No lost frame
2	10,000	252 bytes	2	0.0519	551.409	No lost frame
3	10,000	252 bytes	4	0.0543	595.465	No lost frame
4	10,000	252 bytes	16	0.0434	610.716	No lost frame
5	10,000	252 bytes	256	0.0258	2834.679	No lost frame

Results

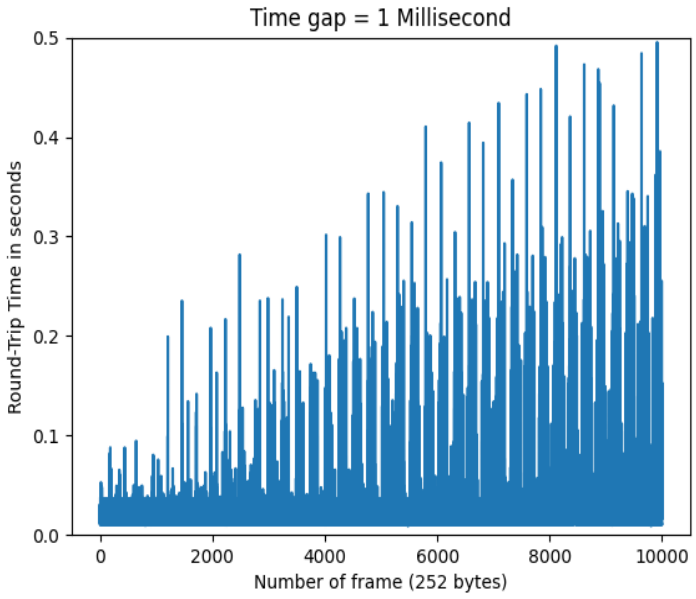

Figure 4.2.3.1: Time gap 1 millisecond (252b)

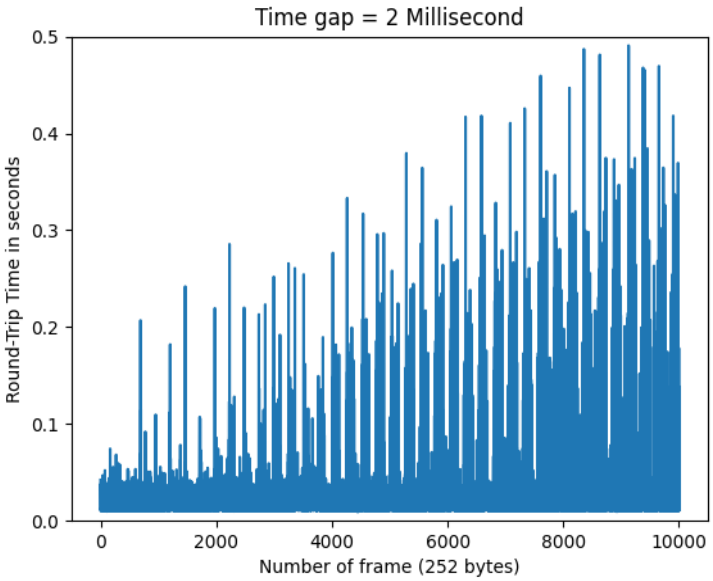

Figure 4.2.3.2: Time gap 2 milliseconds (252b)

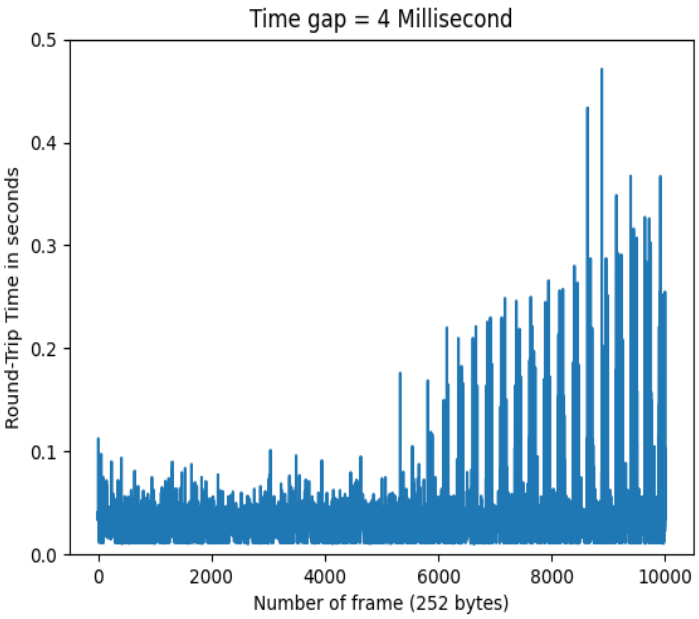

Figure 4.2.3.3: Time gap 4 milliseconds (252b)

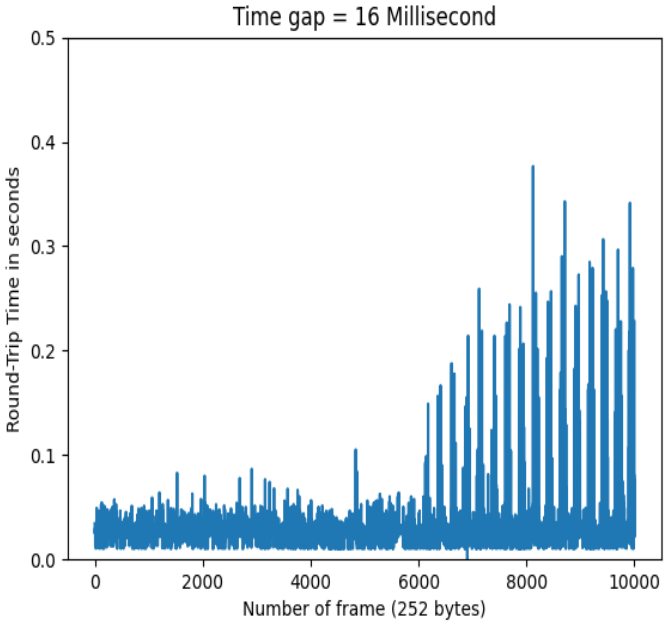

Figure 4.2.3.4: Time gap 16 milliseconds (252b)

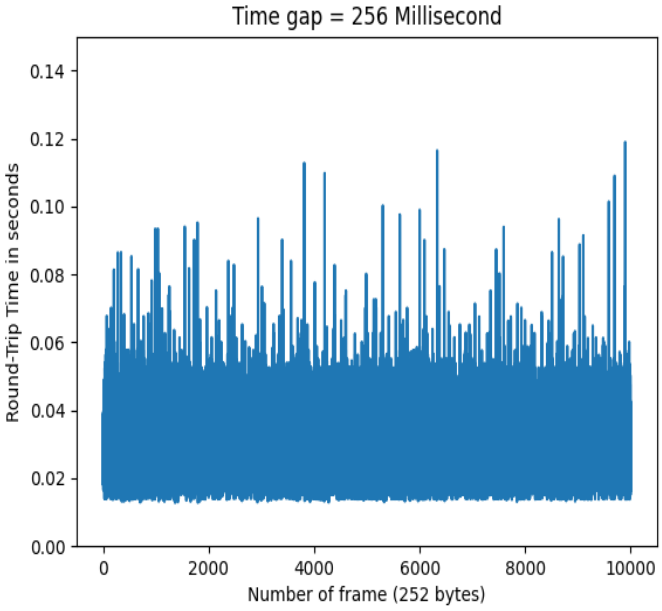

Figure 4.2.3.5: Time gap 256 milliseconds (252b)

Histogram Representation

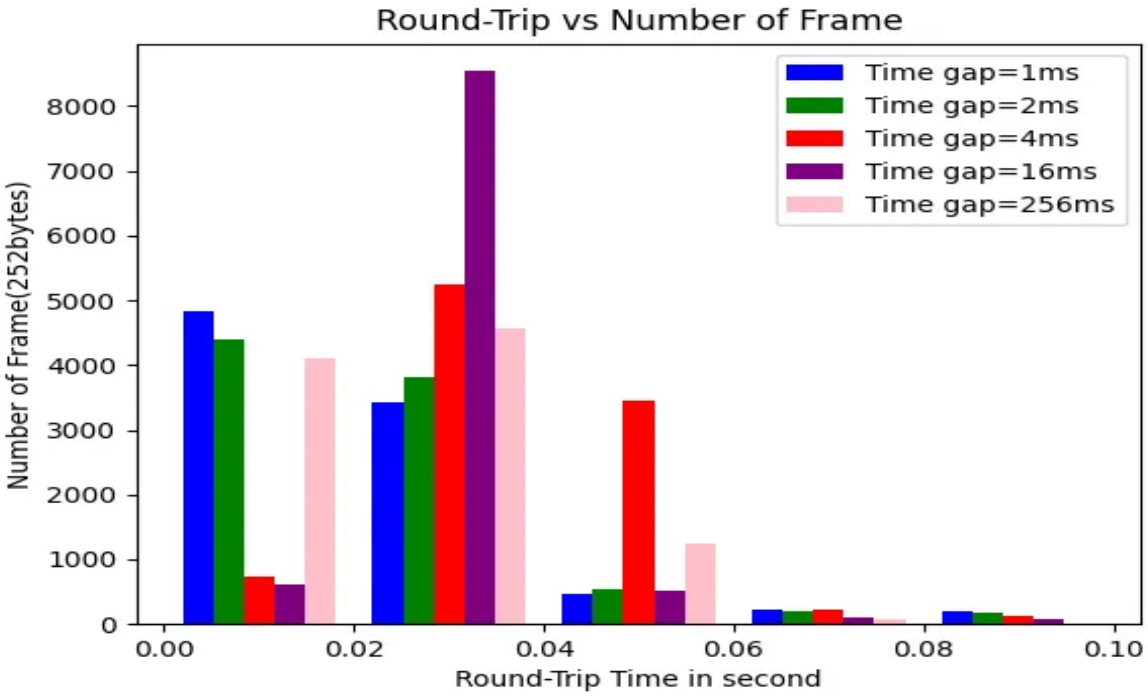

Figure 4.2.3.6: 252 bytes of Frame with a time gap from 1 millisecond to 256 milliseconds

The average delay in frame transmission exhibited incremental increases across the tests, ranging from 0.0508 seconds to 0.0543 seconds. Correspondingly, the overall time required for complete transmission displayed corresponding variations, spanning from 529.982 seconds to 2834.679 seconds. Importantly, there were no lost frames in any of the tests, reaffirming the consistency and reliability of the transmission process under diverse time gap conditions.

Overall, the testing revealed a clear trend in average transmission delays and the ensuing total transmission durations, which was strongly connected with the varying time intervals. An important finding was that, regardless of the temporal separation used for transmission, there was never any frame loss throughout all testing.

4.2.4: Test Four

This experiment focused on frame sizes of 508 bytes and time gaps ranging from 1 millisecond to 256 milliseconds. In each measurement, a total of 10,000 frames with a 508-byte frame size were transmitted.

Table 4.2.4.1: 508 bytes of Frames with a time gap from 1 millisecond to 256 milliseconds

Number of Tests	Number of Frames	Frame Size	Time gap in Milliseconds	The average delay in Seconds	Total Time in Seconds	Lost frames
1	10,000	508 bytes	1	0.1017	1038.673	No lost frame
2	10,000	508 bytes	2	0.0983	1013.663	No lost frame
3	10,000	508 bytes	4	0.0975	1024.474	No lost frame
4	10,000	508 bytes	16	0.0855	1027.925	No lost frame
5	10,000	508 bytes	256	0.0483	3059.2029	No lost frame

Results

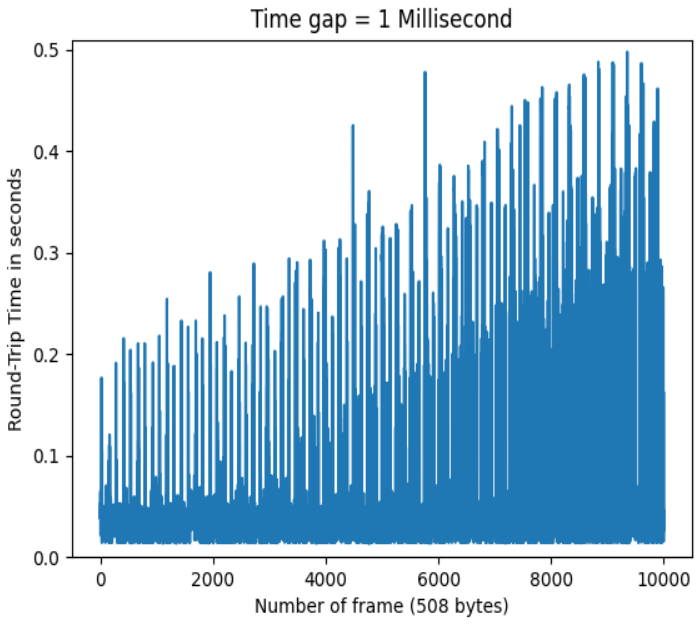

Figure 4.2.4.1: Time gap 1 millisecond (508b)

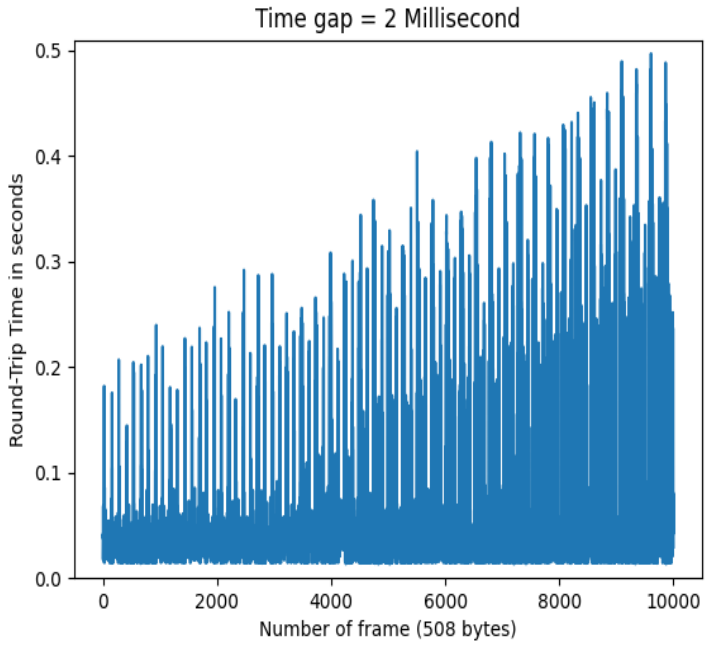

Figure 4.2.4.2: Time gap 2 milliseconds (508b)

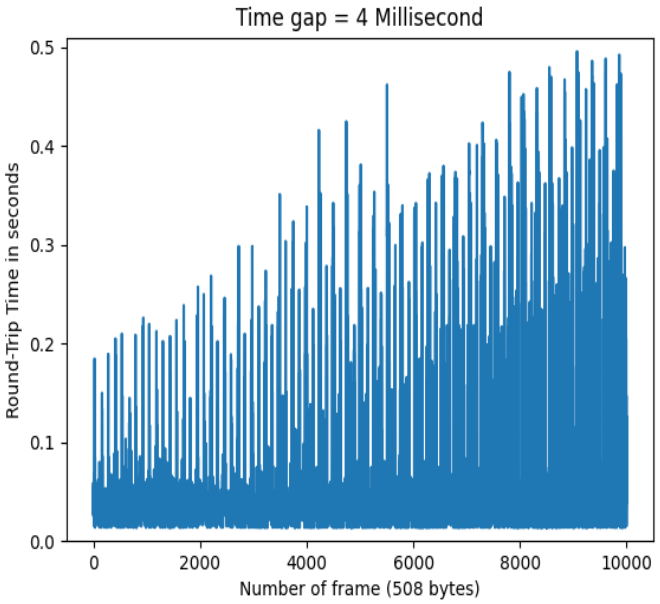

Figure 4.2.4.3: Time gap 4 milliseconds (508b)

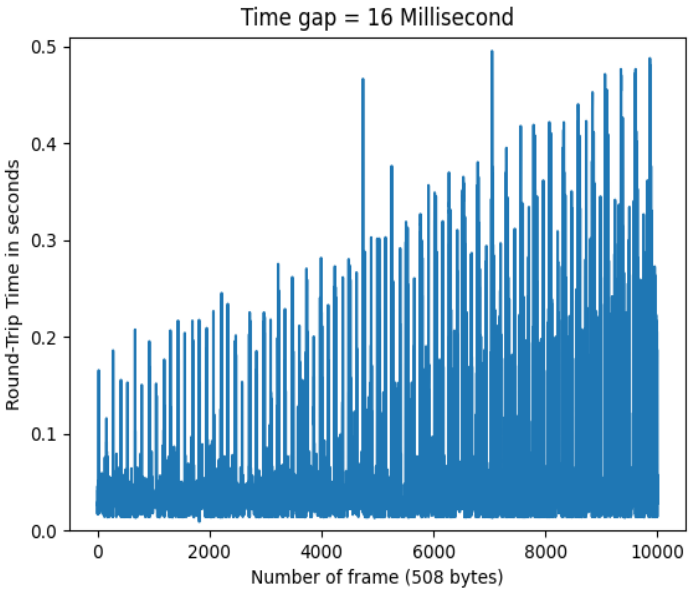

Figure 4.2.4.4: Time gap 16 milliseconds (508b)

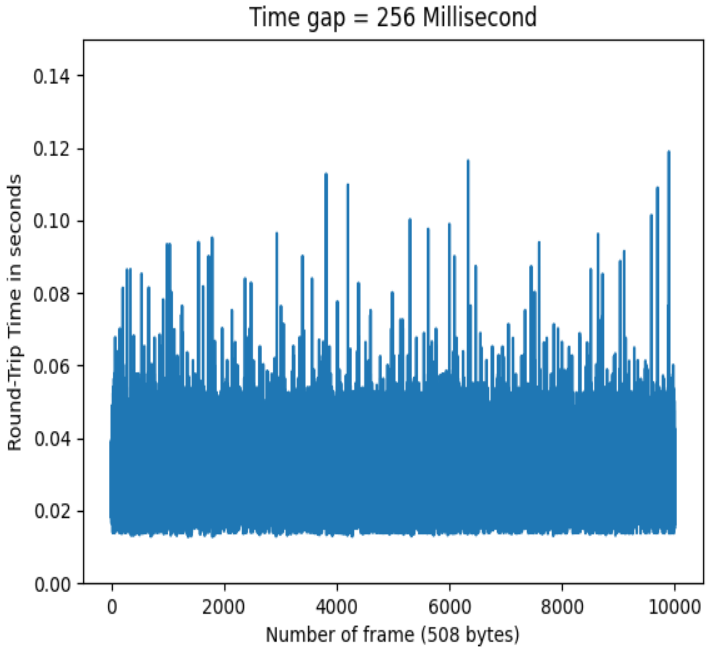

Figure 4.2.4.5: Time gap 256 milliseconds (508b)

Histogram Representation

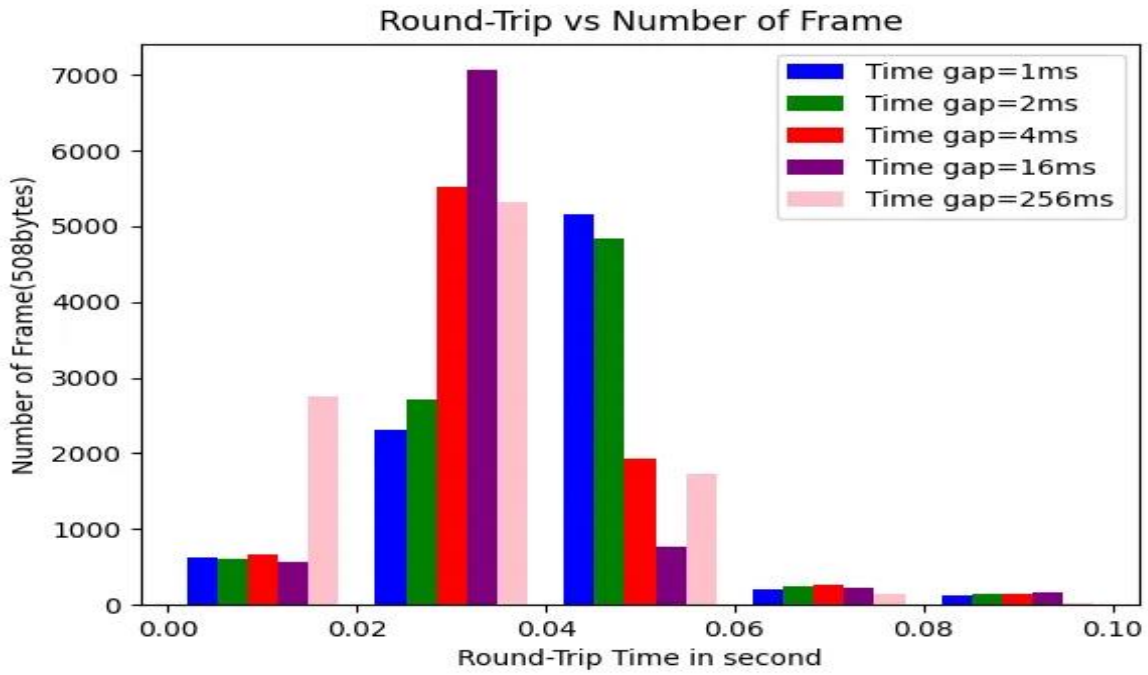

Figure 4.2.4.6: 508 bytes of Frame with a time gap from 1 millisecond to 256 milliseconds

Notable conclusions were drawn from five thorough experiments that each involved the transmission of 10,000 frames, each of which had a frame size of 508 bytes. The tests also used different transmission intervals (1, 2, 4, 16, and 256 milliseconds). Intriguing patterns could be seen in the experiments' average frame transmission latency, which varied from 0.0855 seconds to 0.1017 seconds. The total amount of time required for the complete transmission, which varied from 1013.663 seconds to 3059.2029 seconds, followed suit. The fact that there were no missed frames in any of the tests is extremely important since it clearly establishes the consistency and reliability of the transmission process under a variety of time-gap scenarios.

Regardless of frame size and time gap, the Bluetooth protocol performed dependably in all experiments conducted on the Raspberry Pi 4. None of the experiments reported any missing frames. The average latency and total time varied based on the parameters for frame size and time gap. Larger frame sizes and extended time gaps generally resulted in significantly longer average delays and total transmission times. However, these latencies remained acceptable for Bluetooth communication.

5. CONCLUSIONS

In conclusion, the experiments performed to analyze the latency of Zigbee and Bluetooth protocols with varying frame sizes and time gaps provided valuable insights into the behavior and characteristics of these protocols in an industrial communication scenario. Both Zigbee and Bluetooth protocols showed reliable and efficient data transmission under varying conditions, according to the findings. By comprehending the relationship between frame sizes, time gaps, and latency, system designers and developers can optimize the performance of these protocols based on specific industrial communication needs.

Experiments revealed that for the Zigbee protocol, varying time gaps and a fixed frame size of 60 bytes did not highly affect the round-trip time or result in any lost frames. This demonstrates the effectiveness and dependability of the Digi XBee3 802.15.4 TH Zigbee module to perform wireless channels in a scenario involving industrial communication. The findings indicate that the Zigbee protocol can provide robust and reliable wireless communication, making it suitable for a variety of industrial applications requiring seamless data transmission.

Similarly, the Bluetooth protocol experiments conducted on a Raspberry Pi 4 platform demonstrated consistent performance and no lost frames across various frame sizes and time gaps. The Bluetooth protocol demonstrated its capacity to manage frame sizes of up to 508 bytes without affecting performance. This adaptability enables the exchange of vast quantities of data in industrial monitoring and control systems, sensor networks, and residential automation.

Zigbee and Bluetooth protocols performed well in terms of reliability and frame transmission, as no lost frames were reported in any of the experiments. Both protocols demonstrated the ability to transmit frames without losing any frame, which is essential for assuring reliable device-to-device communication. There are, however, notable differences between the two protocols. A significant difference is the average frame transmission latency. In the Zigbee tests, the average delay ranged from 0.316 to 0.328 seconds, whereas in the Bluetooth tests, the average delay ranged from 0.0114 to 0.0361 seconds, which is a significant decrease. This indicates that Bluetooth typically provides lower latency and quicker frame transmission than Zigbee.

Another distinction is the frame size and its effect on transmission time. Bluetooth consistently outperformed Zigbee when it came to transmission speed, even though frame sizes varied. This indicates that Bluetooth is more effective than Zigbee at managing larger frame sizes and maintaining relatively minimal delays.

Due to their distinct advantages and applications, Zigbee and Bluetooth protocols are important in industrial settings. Zigbee, with its low power consumption and mesh network capabilities, is well-suited for applications such as industrial automation, building management systems, and smart grids where network scalability and energy efficiency are crucial. It offers dependable and robust communication in industrial contexts, ensuring uninterrupted data transmission and system dependability.

Bluetooth, on the other hand, provides seamless connectivity, usability, and compatibility with a vast array of devices. It is suitable for industrial applications requiring real-time monitoring, wireless sensor networks, and asset tracking due to its ability to accommodate larger frame sizes and provide reliable data transmission. Bluetooth enables efficient communication between nearby devices, allowing for the seamless integration and control of industrial processes.

Zigbee and Bluetooth protocols play crucial roles in industrial settings where effective and dependable communication is essential for efficient operations. They facilitate seamless data exchange, efficient control and monitoring, and the integration of diverse devices and systems.

In addition, the scalability of Zigbee and Bluetooth protocols makes them ideal for industrial scenarios. The mesh network architecture of Zigbee enables network expansion through the seamless addition of new devices. This scalability is advantageous for industrial applications where the number of devices might change or grow over time, such as manufacturing plants and warehouse automation systems. Similarly, Bluetooth's ability to connect multiple devices in a piconet facilitates the integration of diverse apparatus and systems, enhancing interoperability and adaptability in industrial environments.

The experiments conducted on Zigbee and Bluetooth protocols demonstrate their utility and suitability for industrial applications. Zigbee is a good option for industrial automation due to its minimal power consumption, mesh networking capabilities, and security features. Bluetooth's seamless connectivity, compatibility, and scalability facilitate the development of effective communication solutions for industrial monitoring and control systems, wireless sensor networks, and asset tracking. By utilizing these protocols effectively and optimizing their performance based on specific industrial requirements, organizations can ensure reliable and secure data transmission, improve operational efficacy, and increase industrial productivity.

5.1 Limitations and Suggestions

The maximum payload capacity of the Zigbee module is restricted to 110 bytes, as observed in the experiments. When dealing with larger data sets or when transmitting information that exceeds this size limit, this restriction may cause difficulties. This limitation can inhibit the effectiveness of the communication process in industrial settings where complex data, such as sensor readings or control commands, must be transmitted. To overcome this limitation, it is suggested that a data chunking mechanism be implemented, in which larger data sets are divided into smaller portions that can fit within the Zigbee module's payload capacity. By partitioning the data into manageable chunks, it is possible to transmit larger quantities of data while maintaining compatibility with the Zigbee protocol.

Bluetooth on the other hand, with its maximal payload size of 1024 bytes, offers a greater data transmission capacity than Zigbee. In certain industrial scenarios involving exceedingly large data sets or high-resolution data streams, this limitation may continue to be a limitation. To get over this limitation, the same suggestion can be implemented by dividing the data into smaller segments that suit the payload size of the Bluetooth module. By partitioning the data into smaller segments, the communication process can proceed efficiently, and the entire data set can be reconstructed on the receiving end. Implementing a chunking mechanism enables the efficient transmission of large volumes of data over the Bluetooth protocol and ensures that industrial processes that rely on this communication standard can effectively manage and transmit data.

Even though Zigbee and Bluetooth protocols offer valuable features and benefits for industrial scenarios, their maximal payload sizes are limited. The suggestion is to implement a data chunking mechanism that divides larger data sets into smaller, more manageable portions that can be transmitted within the respective payload size limits. This strategy ensures that industrial processes utilizing these protocols can transmit data effectively without sacrificing compatibility or efficiency. By implementing these recommendations, industries can optimize their communication systems and guarantee seamless data transmission, even when dealing with larger or more complex data sets in industrial environments.

Bibliography

[1] Chune TANGa, Wenjuan LIb and Zhen WANGaa Communication Quality Indicator System of Industrial Communication Network and Related Diagnostic Evaluation Dan LIUa,1, Instrumentation Technology & Economy Institute, ChinabBeijing DS Fieldbus Technology Co.,Ltd, China

[2] H. Yu, P. Zeng and C. Xu Industrial Wireless Control Networks: From WIA to the Future Engineering 8 (2022) 18–24

[3] Stefano Scanzio, Lukasz Wisniewski, Piotr Gaj, Heterogeneous and dependable networks in industry – A survey, Computers in Industry, Volume 125, 2021, 103388, ISSN 0166-3615,

[4] Federico Montori, Luca Bedogni, Marco Di Felice, Luciano Bononi, Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues, Pervasive and Mobile Computing, Volume 50, 2018,

[5] Ahmad Hani El Fawal, Machine-to-machine communication congestion mechanism. Networking and Internet Architecture [cs.NI]. ENSTA Bretagne - École nationale supérieure de techniques avancées Bretagne, 2018. English. ffNNT : 2018ENTA0010ff. fftel-02374816f

[6] Introduction to ZigBee Technology Ankur Tomar– Global Technology Centre Volume 1, July 2011

[7] Paolo baronti, prashant pillai, vince chook , stefano chessa , alberto gotta, y. fun hu, Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards paolo baronti, prashant pillai, vince chook , stefano chessa , alberto gotta, y. fun hu.

[8] Elli Kartsakli 1,*, Aris S. Lalos 1, Angelos Antonopoulos 1,2, Stefano Tennina 3, A Survey on M2M Systems for mHealth: A WirelessCommunications Perspective

[9] Chanakya Kumar, Rajeev Paulus, A prospective towards M2M Communication

[10] H. labiod,h. afifi,c. de santis "wi-fitm, Bluetooth, zig bee and WiMAX"

[11] Pooja D. Pandit, Study of Bluetooth protocol and applications, Mumbai University, India

[12] Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology

[13] P. P. Parikh, M. G. Kanabar and T. S. Sidhu, Opportunities and Challenges of Wireless Communication Technologies for Smart Grid Applications P. P. Parikh, M. G. Kanabar and T. S. Sidhu, "Opportunities and challenges of wireless communication technologies for smart grid applications," IEEE PES General Meeting, Minneapolis, MN, USA, 2010, pp. 1-7, doi: 10.1109/PES.2010.5589988.

[14] S. Subhani, H. Shi and J. F. G. Cobben, "A survey of technical challenges in wireless machine-to-machine communication for smart grids," 2015 50th International Universities Power Engineering Conference (UPEC), Stoke on Trent, UK, 2015, pp. 1-6, doi: 10.1109/UPEC.2015.7339879

List of abbreviations

API--Application Programming Interface

BLE--Bluetooth Low Energy

DCSs--Distributed Control Systems

FDMA --Frequency Division Multiple Access

HVAC--Heating, Ventilation, and Air conditioning

IEC--Electrotechnical Commission

ISO--International Organization for Standardization

LoRaWAN--Long Range Wide Area Network

LPWAN--Low-Power Wide-Area Networks

M2M--Machine to Machine

MAC--Media Access Control

OFDMA --Orthogonal Frequency Division Multiple Access

OSI--Open System Interconnection

PAN ID--Personal Area Network ID

PHY--Physical Layer

RDP--Remote Desktop Protocol

RF--Radio Frequency

RTT--Round-Trip Time

TDMA--Time Division Multiple Access

USB--Universal Serial Bus

XCTU--Xbee Configuration and Test Utility

List of Figures

List of Tables