Implementation and Performance Analysis of Wireless Ad-hoc Mesh Networks

Thomas Mok Shao Chung

UNIVERSITI TUNKU ABDUL RAHMAN

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF WIRELESS AD-HOC MESH NETWORKS

THOMAS MOK SHAO CHUNG

**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Engineering (Honours)
Electronics (Computer Networking)**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

Oct 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Thomas Mok Shao Chung

ID No. : 1901593

Date : 2 October 2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"IMPLEMENTATION AND PERFORMANCE ANALYSIS OF WIRELESS AD-HOC MESH NETWORKS"** was prepared by **THOMAS MOK SHAO CHUNG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Electronics (Computer Networking) at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ____________________________________

Supervisor : Tham Mau Luen

Date : 3 Oct 2024

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Ir. Ts. Dr. Tham Mau Luen for his invaluable advice, guidance and his enormous patience throughout the development of the research.

I am incredibly thankful to my fellow classmates at UTAR, whose collaboration and insightful suggestions have been a great source of motivation throughout this journey. Especially, Mr. Mok Jia Cheng for his advice and support made the process more rewarding and interesting.

Finally, I wish to recognize the ongoing encouragement from my friends and family. Their unwavering belief in me has been a constant source of strength and motivation. Without their understanding and reassurance, the successful completion of this project would not have been possible. Once again, thank you to everyone who contributed to this achievement.

ABSTRACT

Wireless ad-hoc mesh network is an effective solution to establish resilient communications, particularly in urban and rural areas prone to emergency situations. Existing approaches often rely on such network to disseminate low-bandwidth text data, which could filter the most critical information of relevance to the situation. In contrast, this project aims to develop a multi-hop Wi-Fi mesh network for high-bandwidth video streaming applications. Specifically, a testbed is built using the Better Approach to Mobile Ad-hoc Networking- Advanced (BATMAN-adv) protocol and Raspberry Pi devices. The developed mesh platform is integrated with a real-time disaster detection system, which classifies different types of natural disasters and counts the number of victims in a streaming video. A wide and rapid dissemination of these analyzed disaster information is facilitated by a proposed database synchronization system. Experimental results show that the proposed solution achieves a real-time streaming protocol (RTSP) latency of approximately 5 seconds. Besides that, the output text and image data can be synchronized within 0.59 and 0.56 second for the single-hop route. Overall, the project contributes to the development of decentralized communication platforms, offering valuable insights for future applications in disaster response systems.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

MANET	Mobile Ad-hoc Network
BATMAN	Better Approach To Mobile Ad-hoc Network
TQ	Transmission Quality
WMNs	Wireless Mesh Networks
VPN	Virtual Private Network
AODV	Ad-hoc On-Demand Distance Vector
PDR	Packet Delivery Ratio
VoIP	Voice Over Internet Protocol (IP)
MTU	Maximum Transmission Unit
HTTP	Hypertext Transfer Protocol
OS	Operating System
RAI	Raspberry Pi Based Access Infrastructure
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IQR	Interquartile Range
ITU	International Telecommunication Union
QoS	Quality Of Service

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

This project contributes to the efforts aimed at improving communications in disaster-prone and remote areas through the development and analysis of a wireless ad-hoc mesh network. In this sense, it is relevant to Malaysia, as a country situated in a region with no major typhoon paths and outside of the Pacific Ring of Fire but affected by recurring natural and human-made disasters. The Malaysian Communications and Digital Ministry, represented by Deputy Minister Teo Nie Ching, recently called for an urgent development of the country's telecommunications sector, including providing services and infrastructure to rural and islander communities. The goals of this initiative are in line with the project of ensuring connectivity and access to those areas that are susceptible to isolation caused by emergencies.

The idea of this project is inspired by the experience of Malaysia and its proactive policy with regard to developing the national telecommunications. The practice of cooperating with the telecommunication companies and providing opportunities to extend the scope of service to the areas with no network is absolutely in line with the mission. To construct the network infrastructure that will be able to operate effectively in the time of natural disasters and calamities, thus ensuring the unrestricted communication for those needing immediate help or carrying out their usual activities. However, this project should be considered not only from a technically and practically useful perspective. The main focus is on the implications for the communities that faced the threat of complete isolation during disasters. The network is capable of operating on its own and does not need any immovable objects, and this makes it particularly important for the Orang Asli Settalements in Mersing that are often flooded and are not completely covered by the telecommunications.

Performance Analysis has shown, through applied methods and lessons, that Wireless Ad-hoc Mesh Networks can work to bridge the digital divide and ensure no community is without a connection to the wider world. This project is not only about functionality and innovation, but inclusivity and preparedness;

qualities espoused by the ministry of the vision for a connected, resilient Malaysia (Malay Mail, 2022).

1.2 Importance of the Study

The relevance of the research studying the performance of the implementation wireless mesh ad hoc networks cannot be underestimated. This is a significant exploration venture in unknown fields of digital user experience. At the same time, it provides valuable insights with regard to coverage holes worldwide, primarily in remote and underdeveloped areas. The purpose of this study is to identify the lack of robust coverage in certain territories and make particular recommendations to address it through new wireless network infrastructure. In that sense, the examination of non covered territories is beneficial for the advancement of telecommunications in the wireless sphere, as well as meeting the national agenda of more inclusive and widely available digital use of Internet and communication technologies. That is why the research on the performance of wireless mesh ad hoc networks is critical for the development of communications. The proposed research is focused on the abovementioned networks. It offers a strategic outlook for such issues as network congestion. Most important it provides guidance for consistent and reliable provision of high-speed internet in a variety of settings, including remote and underdeveloped areas. It focuses not only on improving the actual performance but also on supplying policymakers and service developers with strong insights, which can improve the inclusiveness of the digital landscape. In this way, the research will be able to make a significant contribution to the resolution of current issues in the telecommunication industry. The advent of innovative solutions is going to ensure that every layer of the population has access to high-quality digital instruments. As a result, the researched issue has a high degree of relevance for the furthering of socioeconomic frameworks and improving the level of digital literacy nationwide (TheEdgeMalaysia, 2023).

1.3 Problem Statement

In Malaysia, telecommunication networks are mainly arranged in a star topology. All user nodes are connected to a central node that provides an Internet connection. Such an organization of the network greatly simplifies control and reduces the cost of setting up the network initially, however, an unacceptable dependence arises – there is one small central node by which the entire network is connected. If it is suddenly disconnected due to a malfunction, periodic preventive maintenance, or other reasons, all other user nodes will lose internet access, which can be unacceptable for both private users and large companies. One way to get rid of these shortcomings would be to switch to a decentralized wireless mesh ad-hoc network with data routing according to the BATMAN-adv protocol, which operates in environment routing mode. This kind of network allows data to route between a number of nodes which means that even if one of the nodes no longer works, the network is still connected and, ergo, presents a more reliable and safer network. Making use of this topology in order to verify how it works and assess the credibility of a mesh network as a means of enhancing the safety and stability of local internet activity.

1.4 Aim and Objectives

The given study aims to introduce an experimental work related to the implementation and evaluation of a wireless mesh ad hoc network based on the BATMAN-advanced routing protocol. The main idea of the work is to enhance the level of connectivity in a dynamic and decentralized environment. Beginning from the System Setup, it is required to configure the hardware and software timely to get the opportunity to implement a wireless mesh network. The core of the work includes the configuration and setup of three Raspberry Pi nodes with the help of an Alfa AWUS036NHA wireless adapter to guarantee the proper functioning and, hence, the best performance of the batman-adv framework.

The purpose of the second phase is to configure the BATMAN-Advanced after the assembly of hardware. The principal target of the phase is to optimize the protocol settings to ensure that the mesh network demonstrates the best performance or, in other words, the most efficient mesh-wide packet forwarding via the adjustment of better route metrics. The phase also implies the optimization of the mesh topology procedures.

The third phase of the project is the deployment of the configured network, which consists of the strategic placement of three Raspberry Pi nodes to form a mesh network. The relevant details of deployment are defined in terms of the location of nodes depending on the best position and, hence, the proper network coverage and efficient channels of communication. The given phase is crucial for justifying the implementation of the Batman-ADV protocol in a real-world setting. The ultimate goal of the study is the performance evaluation of the mesh network through comprehensive testing, which includes several aspects, namely, throughput, latency, packet loss, jitter, scalability, and the ability to handle the changes in node dynamics. Finally, the study, as a whole, is aimed to demonstrate the path from the theoretical underpinnings of the batman-adv protocol to the creation of the wireless mesh network and justifying the need to implement the protocol in controlled conditions.

1.5 Scope and Limitation of the Study

This study is focused on the deployment of the wireless mesh ad-hoc network with the use of the BATMAN-Advanced protocol. It is aimed to explore the capacities and performance of the network in the course of a series of experiments in a controlled environment. Specifically, the study's limitations are determined by the use of three Raspberry Pi nodes. The devices are equipped with corresponding adapters, Alfa AWUS036NHA, for which the mesh point mode is supported. For this reason, the limitations of the study are primarily related to the hardware compatibility and budget restrictions.

It is essential to note that some attempts were made to utilize other wireless adapters with the same nodes to conduct the experiments. Specifically, it was expected that the Alfa AWUS036ACH or TP-Link TL-WN722N V3 could be used. However, it was identified that these adapters do not support mesh point mode, which is critical for the proper functioning of the BATMAN-Advanced protocol. The nodes would not be able to establish a network without a central point, which is in contradiction with the basic network topology characteristic for the mesh ad-hoc network. For this reason, other adapters could not be used, and the Alfa AWUS036NHA was chosen. This fact directly affects the study's limitations because the other options have not been tested, which could identify other opportunities for increasing the network's performance or exploring new options for other applications.

Approximately the same limitations refer to the budget. It is evident that acquiring other wireless adapters to proceed with the tests would lead to a severe increase in costs. Concurrently, for the sake of cost-efficiency, alternatives had to be limited to a single option for which the applicability and performance were assessed. It may have direct implications on the study’s scope because some recommendations made afterward could be different if the other adapters were acquired and assessed. Thus, the study’s limitations are defined by the hardware compatibility and budget-conscious choices.

1.6 Contribution of the Study

A four node BATMAN-adv mesh network was deployed for disaster monitoring across both urban and rural areas. The setup consists of one node functioning as an RTSP server with a camera, while another node serves as an RTSP receiver equipped with a Disaster Detection System. This receiver not only processes video streams but also acts as an emergency response system, enabling immediate action in disaster situations. The deployment spans urban areas, such as the UTAR Sungai Long campus library, and remote regions like the Pahang forest. Data is synchronized between nodes, ensuring efficient monitoring and real-time response without relying on centralized infrastructure. The outcome of this study offers valuable insights for future BATMAN-adv network deployments in disaster-prone areas.

1.7 Outline of the Report

Chapter 1 the Introduction outlines the project background, discussing the significance of wireless ad-hoc mesh networks for disaster monitoring in urban and rural areas.

Chapter 2 the Literature Review covers relevant research on wireless mesh networks, focusing on the BATMAN-adv protocol. It compares various network topologies and versions of the protocol, along with performance metrics such as throughput, latency, and packet loss. Additionally, it reviews wireless adapters and synchronization methods in mesh networks.

Chapter 3 the Methodology and Work Plan details the implementation process of the BATMAN-adv mesh network, including hardware selection, network configuration, and performance analysis. It also explains the integration of the Disaster Detection System and the trial-and-error approach used for wireless channel selection.

Chapter 4 Results and Discussion presents the findings from the performance analysis of the mesh network, evaluating metrics such as throughput, latency, jitter, and packet loss. It includes detailed discussions of the network's performance in various test environments.

Chapter 5 Conclusions and Recommendations summarizes the project outcomes, highlighting the efficiency of the BATMAN-adv protocol in disaster monitoring applications. It also offers recommendations for future improvements and potential real-world applications of the network.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the context of wireless communications, Wireless Mesh Networks have achieved significant progress to build reliable, scalable, and flexible network infrastructures. Mesh topology networks are characterized by a set of nodes that maintains an active connection among themselves at any given time, thereby ensuring the network’s adaptability to environmental changes and shifts in its topology. This feature makes them particularly suitable for ad-hoc networking, in which network nodes establish connections without involvement of a fixed infrastructure and quickly adapt to changes in the network.

The further development of these networks is due to the continuity of the shortcomings of traditional wireless networks. The evolution of protocols and algorithms to which WMNs operate is also noticeable. There are numerous studies on the subject of networking protocols designed for WMNs that help improve the efficiency of routing messages between nodes, ensure the reliability of data transmission, and protect the network from potential threats. It clearly represent challenges of deploying such networks in ad-hoc modes, where the absence of a sustained structure involves particular management and message routing (Akyildiz et al., 2005).

A thorough analysis of sources shows that the level of interest in network performance measurement is extremely high, including in cases with ad-hoc network mode. The analysis of network metrics such as throughput, latency, packet delivery rate, and scalability is fundamental and used in all studies. In many cases, researchers conduct research in parallel on how to increase the throughput of the network and the speed of its operation, and examples can be given on how it optimize routing protocols for intelligent resource utilization while increasing network reliability during dynamics its structure. Deployment of WMN in ad-hoc cases attracts special interest due to a diversified range of potential applications from disaster recovery communication systems to large sensor networks.

2.2 BATMAN-adv

The BATMAN protocol was developed as a new solution for routing in mobile ad-hoc networks. It works using a decentralized mechanism, that is, each node chooses the next hop towards the destination based on the information it has, rather than having complete routing. This approach gives the protocol significantly better scalability and adaptability in various network scenarios. The protocol has undergone three major evolutionary changes: The next generation of the BATMAN protocol was called BATMAN-III. This version was the first to introduce the concept of Originator Messages , which is a kind of “billboards of tell existence”. After a while the network eventually selects the best path to achieve good coverage from the available paths through the process of majority voting by originator messages. BATMAN-IV took the BATMAN-III route introducing lots of optimizations to the already working manet routing protocol. After each routing hop, the bidirectional links were checked to ensure the path’s reliability. The bidirectional verification was made possible by carefully tracking the packets being exchanged between each network couple. BATMAN-V also called BATMAN-adv was the next version developed. It proved to operate as a stand-alone, layer-2 routing protocol and a new wireguard VPN client integration. Several experiments have been conducted in comparing the different BATMAN versions amongst other versions. Different types of experiments compare the batman versions in aspects of throughput, latency, packet delivery ratio and scalability. The BATMAN-adv, which has a later-2 characteristics, has outperformed the previous versions when configured in a dense network environment and able to allow clients to roam seamlessly. This presents BATMAN-adv as the ideal candidate to be used in Mobile and Radionet applications, Community mesh networking, and outdoor events. The journey of BATMAN protocol, from BATMAN-III to BATMAN-adv, promises further possible optimization since, at the moment configuration, BATMAN-adv. Optimization will again be possible to adjust the network to best fit the manet situation. An ideal improvement should include adding functionality since efficiency is already achievable. The security issue should also need to be considered as a way of adding functionality. This follows the trend that soon; ad-hoc systems may rely on the LSDN network (Gurumoorthi et al., 2024).

2.3 Comparison between BATMAN-adv version IV and V

Titled “ Performance Evaluation of BATMAN-adv Wireless Mesh Network Routing Algorithms ”, the study by Liu, et al. investigates the performance of different generations of the BATMAN-adv protocol, specifically versions IV and V, in wireless mesh networks, or WMNs. The paper is relevant because, as Liu et al. explain, efficient routing protocols are critical for WMNs to have a reliable network performance across the highly variable environments. BATMAN-adv is based on the original BATMAN protocol, modified to operate at layer 2 of the network for a decentralized, self-organizing network resembling a virtual switch. This configuration is efficient in achieving more effective protocol operation while reducing overhead, which is crucially important for devices with limited resources. Liu et al. compare this latter version, called version V for the throughput-based routing concept, with a simpler version IV based on transmission quality, or TQ, whereas the throughput metric itself is not implemented. The study’s goal is to demonstrate the performance of the newest version in terms of delay, packet loss, and throughput across different performance measures and ascertain whether it is capable of outperforming the previous version. Overall, their experiment demonstrates that the more recent version does not consistently outperform the older across different metrics.

Despite the V version’s design to allow for more flexibility in routing by continuously adjusting decisions based on link conditions in real time, it is unable to consistently improve performance. Measuring the performance through packet loss and delay shows that, regardless of the conditions, the more recent version V usually has a higher packet loss and increased delays. The results of their performance metrics are relevant for future work. They further inform the development of more effective protocols by sharing the shortcomings of the initial version in which a potentially superior version did not exceed the older one. The study highlighted the necessity of finding future improvements for the designed protocol to be more effective. This is crucial for the study of the development of WMNs, especially considering the future work in which researchers will seek to test the performance of the protocols across different conditions. Accordingly, the study emphasizes the importance of feature development to compose new and improved algorithms to fulfill future deployment requirements (Liu et al., 2018).

2.4 Network Topologies

This study explores several common network topologies, especially those that are likely to have a significant impact on the design and performance of WMNs. It describes the most critical features and applications of commonly known Bus, Ring, Star, Tree, Mesh, and Hierarchical topologies and discusses how these topologies are utilized and expanded in the WMN environment (Jiang, 2015).

2.4.1 Star Topology

The star topology in WMNs encompasses a central node serving as a gateway or bridge that interconnects several surrounding wireless nodes. Innovations include a set of technologies that increase the capability and efficacy of the system in distributing and managing traffic in metropolitan networks. For example, Kukhta, et al. proposed a star-ring implementation that uses the functions of a central node for rapid routing of data and control of bandwidth (Kukhta et al., 2014).

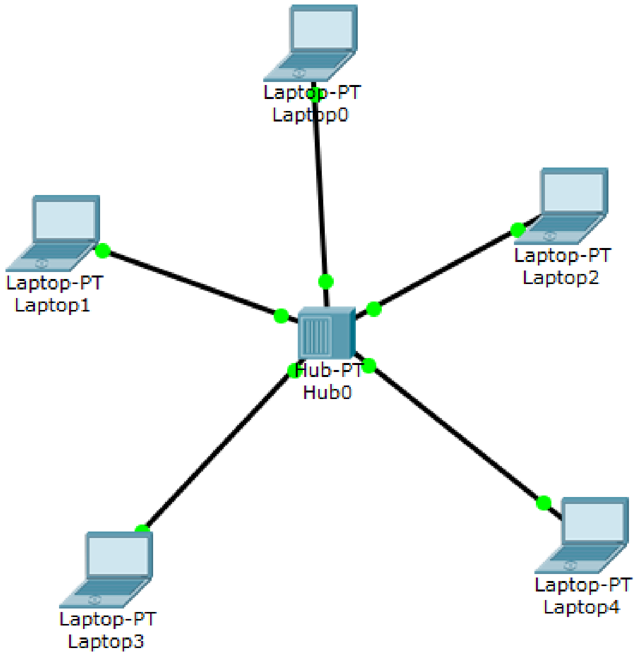


Figure 2.1: Star Topology

2.4.2 Ring Topology

In case of WMNs, the ring topology could improve the fault tolerance as well as ensure consistent service availability, responding to the failures by the development and subsequent securing of a closed-loop pathway for data transfer. Singh et al. extended the idea by creating a hybrid ring-tree-star topology for WMNs, thereby achieving a broader collection area and, therefore, greater amounts of data that can be handled effectively (Jiang, 2015).

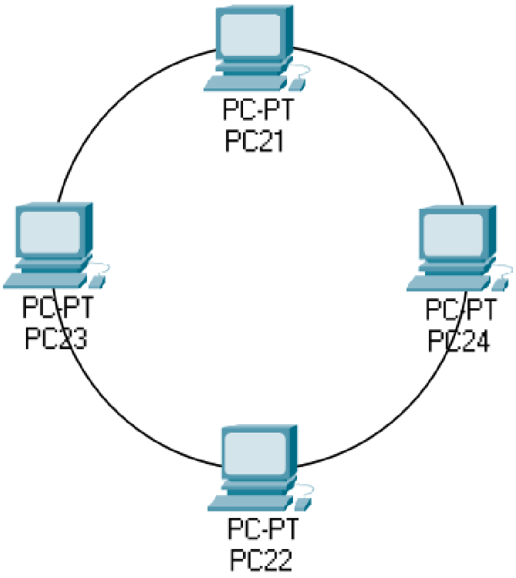


Figure 2.2: Ring Topology

2.4.3 Bus Topology

Bus topology, although not popular in wireless context because it requires only a horizontal line of communication, enhancements made to data transmission protocols are crucial to eliminating issues of delay and enhancing network utilization, an example being the study by Ramapriya et al. on double bus topology . In this context, dual bus topology aids in enhancing data handling in WMNs by improving channel access methodologies (Ramapriya et al., 1999).

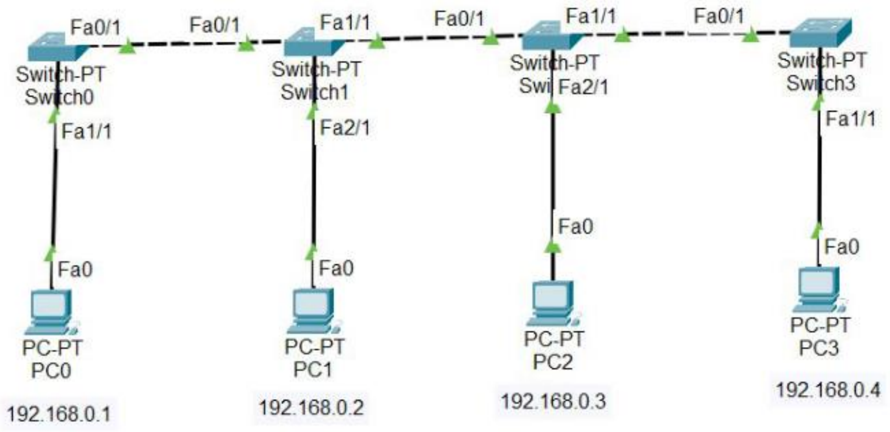


Figure 2.3: Bus Topology

2.4.4 Tree Topology

A Tree Topology. The use of tree topology becomes advantageous to WMNs because of the hierarchical design approach that helps cover a large area and simplifies the management of network resources flow. According to Ruiz et al. , tree topology can be used in mobile ad hoc networks and vehicular ad hoc networks such data dissemination and the network are achieved with increased scalability and better management capabilities (Ruiz et al., 2012).

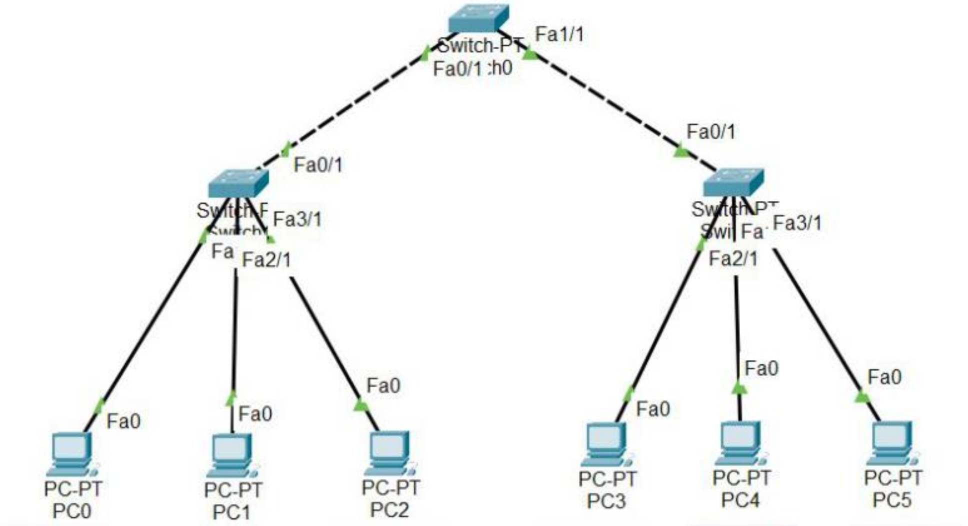


Figure 2.4: Tree Topology

2.4.5 Mesh Topology

A mesh topology is an approach that is particularly well-organized for WMNs. A feature of such a connection is multiple parallel interconnections between adjacent nodes. This topology has the property of ensuring redundancy, that is, multiple proper and faulty paths between any two nodes of the network. For example, Feng et al. presented a design of hybrid opto-electric components that provides fast data transfer between units and ensures that the network continues to function effectively in any environmental situation (Feng et al., 2013).

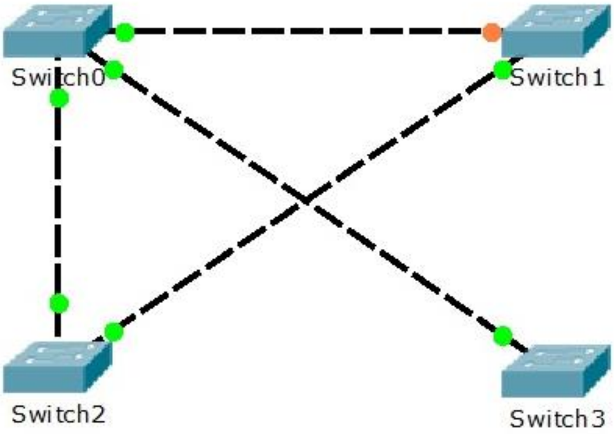


Figure 2.5: Mesh Topology

2.4.6 Hierarchical Topology

Hierarchical Topology: Like the tree topology, hierarchical topology has challenges in the context of the wireless mesh network since it uses several tiers of nodes to separate partition and separate the network traffic. Nonetheless, it leverages a hierarchical network organization approach as a positive dynamics, reducing the length of the routing processes. For large-scale mesh networks, the topology provides better network management results, allowing for the segmentation of the network into sections of a manageable size. With such a well-structured network, it is possible to improve the scalability of the overall WMN, achieving optimal results.

It should be noted that each of these topologies is designed for particular applications; moreover, with technological advancements, they may also become less generalized as their application potential grows. For this reason, research into these topologies is equally important for the future of the wireless mesh network since the modern versatile and scalable standard advantages will be used for more efficient and reliable WMNs.

2.5 Performance Metrics

The performance evaluation of the wireless mesh ad hoc networks using the Batman-adv protocol has focused on such metrics of the network quality and efficiency as latency, packet loss, throughput, packet delivery ratio and jitter.

2.5.1 Packet Loss

Packet loss in Batman-adv is lower in comparison to standard ad hoc routing protocols, as the default mode of operation of the protocol is proactive topology detection and the subsequent link failure avoidance. The algorithm supporting the process swiftly reacts to changes in the network topology, and matching results indicate significant improvement in this domain in comparison to AODV-derived algorithms. The proactive nature of the protocol tends to arrange packets better for transit and that results in higher likelihood for correct data delivery. This is beneficial in dynamically changing environments, where the topology of the network may change rapidly when nodes move (Perkins & Royer, 1999).

2.5.2 Throughput

Another aspect that is evaluated is the throughput of networks utilizing Batman-adv. A number of studies show that BATMAN-adv is capable of maintaining a higher throughput in most conditions. The main reason behind this is that the protocol utilizes efficient routing decisions and rapidly adapts to the change in the network conditions. As a result, the high throughput it can maintain makes Batman-adv suitable for high-bandwidth applications in mesh networks. Moreover, there are two types of throughput which are TCP throughput and UDP throughput. TCP throughput is measured by calculating the amount of data that has been successfully transmitted and acknowledged over a period of time. Since TCP is a connection-oriented protocol, it guarantees reliable data transfer by using acknowledgments and flow control mechanisms. The formula for TCP throughput is based on the total amount of data received (in bits) divided by the total time taken for the transmission (in seconds). TCP throughput is affected by factors such as congestion control, packet loss, and network latency, which can slow down the transmission rate.

On the other hand, UDP throughput is calculated similarly but with some key differences. UDP is a connectionless protocol, which means it does not guarantee reliable delivery or require acknowledgments for transmitted packets. As a result, UDP throughput is typically higher than TCP, as there is no overhead for acknowledgments or retransmissions. The formula for UDP throughput also divides the total amount of received data by the total time taken, but the focus is more on the raw speed of data transmission rather than reliability. UDP throughput is ideal for applications like real-time video or audio streaming where speed is prioritized over error correction.

2.5.3 Packet Delivery Ratio

Moreover, packet delivery ratio is one of the most important performance indicators where Batman-adv has the best results. This indicator measures the number of packets that arrived at their destination without loss. For example, researchers presented in the study by Johnson et al. claim that comparison of the four solutions in different scenarios shows the high effectiveness and performance if BATMAN-adv when the PDR is used (Robinson & Knightly, 2007).

2.5.4 Jitter

In applications where real-time data transmission is needed, such as VoIP or streaming services, jitter, or the variability in packet delay, is critical. Batman-adv has performed satisfactorily in terms of controlling jitter, with minor fluctuations in delays. This aspect is crucial for the service quality, as longer variations in packet delivery time could imply a degraded performance and thus a worsened user experience in real-time applications.

2.5.5 Latency

In mesh networks, latency is a critical performance metric that measures the time delay in data transmission from source to destination. This latency is influenced by several factors, including the number of network hops, the overall network topology, and the mobility of nodes. As the number of hops increases, latency tends to rise due to the additional time required for data to traverse through multiple nodes and potential delays from retransmissions. Network topology also plays a crucial role; denser networks with shorter paths typically experience lower latency compared to sparser networks. The batman-adv (Better Approach To Mobile Adhoc Networking - Advanced) protocol, designed to enhance routing in mesh networks, addresses some of these latency challenges. By utilizing a proactive routing approach and continuously exchanging neighbor information, batman-adv aims to reduce latency compared to traditional reactive protocols, which discover routes only on-demand. This protocol's optimizations help in minimizing delays and improving overall network performance.

2.6 Comparison Between Wireless Adapters

There are many wireless adapters available in the market. Thus, choosing a appropriate wireless adapter for this project is crucial. After considering the a few factors and requirements such as the price and support of 802.11s Mesh Mode, a few wireless adapters are shortlisted and comparison were made below.

2.6.1 Alfa AWUS036ACH

The AWUS036ACH is a dual-band adapter, meaning that it can work at both the 2.4 GHz and 5 GHz frequency range. The device is advantageous to the above-mentioned types of network environments because the 5 GHz band is much less crowded than 2.4 GHz, and there is ultimately less interference on it. Additionally, the device is able to take higher data rates and possibly have a higher range. The technology used in the AWUS036ACH is the Realtek RTL8812AU chipset. It represents an a b/g/n/ac device and is compatible with the newest wireless protocols. However, it is ultimately built for higher performance. High cost may be the issue to install the AWUS036ACH due to its setup as a device. Additionally, specialized drivers or any other configuration may be needed. Finally, dual-band device can support the maximum MTU size , which presents an unnecessary speed advantage in a mesh network setup. However, the 2.4-GHz band in this particular case scenario transmits on a greater distance, which is better for penetration, signal is lost much slower , meaning that range is better for simple coverage.

2.6.2 TP-Link TL-WN722N V3

The single-band TL-WN722N V3 adapter works on a 2.4 GHz frequency only. Nevertheless, it is a reliable piece of hardware and even comes with a high-gain antenna that allows the device to extend the range. The latter feature can be very helpful in the scenarios when it is critical to sustain a long-range communication. Since the adapter operates on the Realtek RTL8188EU chipset, the data rate is limited to 150 Mbps. For most basic networking activities, this capacity is sufficient, although it may not be enough for work intensive throughputs. It is fairly priced and easy to set up, which explains the popularity of this model for low budget projects. The single-band device is another factor that makes it less universal but more budget-friendly compared to the AWUS036ACH.

2.6.3 TP-Link TL-WN722N V1

The TP-Link TL-WN722N V1 operates on the 2.4 GHz band using the Atheros AR9271 chipset, offering a maximum data rate of 150 Mbps, similar to the V3 version. While sufficient for basic tasks, it lacks the throughput for intensive activities like large file transfers. However, the V1 stands out for its advanced features, such as support for monitoring mode, making it popular in network testing environments, especially with Linux. Compared to the AWUS036ACH, the V1 is budget-friendly and easier to set up but lacks dual-band capability and the higher performance of the AWUS036ACH. The V1 is a simpler, reliable choice for users seeking basic functionality, while the AWUS036ACH offers more versatility and speed for demanding environments.

2.6.4 Alfa AWUS036NHA

AWUS036NHA is especially suitable for mesh networks due to its compatibility and stability when used. Wireless mesh networks that operate on the B.A.T.M.A.N. advanced protocol can utilize this adapter. This adapter is one-band, having a frequency of 2.4 GHz and has Atheros AR9271 which, as mentioned above, has excellent stability. Also, Atheros AR9271 is widely compatible with different drivers and operating systems, out of which Linux has proven to be one of the most popular for research and development purposes.

According to the product’s specification, he AWUS036NHA has an essential feature for the creation of a mesh network – the Mesh Point mode. In the mode, devices within the network can communicate without focusing on the router, which is crucial for flexibility and reliability for implementations in the dynamic environments like drone communication and control or location-based signal detection.

Table 2.1: Comparison of The Wireless Adapters

Feature	Alfa AWUS036ACH	TP-Link TL-WN722N V3	Alfa AWUS036NHA	TP-Link TL-WN722N V1
Frequency Band	Dual-band (2.4 GHz & 5 GHz)	Single-band (2.4 GHz)	Single-band (2.4 GHz)	Single-band (2.4 GHz)
Wireless Standards	IEEE 802.11 a/b/g/n/ac	IEEE 802.11 b/g/n	IEEE 802.11 b/g/n/s	IEEE 802.11 b/g/n/s
Data Rate	Up to 300 Mbps (2.4 GHz)	Up to 150 Mbps (2.4 GHz)	Up to 150 Mbps (2.4 GHz)	Up to 150 Mbps (2.4 GHz)
	Up to 867 Mbps (5 GHz)			
Antenna	Dual detachable antennas	Detachable high gain antenna	Detachable high gain antenna	Detachable high gain antenna
Security	WEP 64/128-bit, WPA, WPA2, WPS	WEP 64/128-bit, WPA/WPA2, WPS	WEP 64/128-bit, WPA, WPA2, WPS	WEP 64/128-bit, WPA, WPA2, WPS
Chipset	Realtek RTL8812AU	Realtek RTL8188EU	Atheros AR9271	Atheros AR9271
Compatibility	Windows, Mac, Linux	Windows, Mac, Linux	Windows, Mac, Linux	Windows, Mac, Linux
Applications	High-performance networking	Basic networking,	Reliable throughput,	Advanced features for network testing

Feature	Alfa AWUS036ACH	TP-Link TL-WN722N V3	Alfa AWUS036NHA	TP-Link TL-WN722N V1
		Range extension	Compatibility focus	
Strengths	High throughput and dual-band flexibility	Cost-effective with good range	Excellent driver compatibility and stability	Excellent for monitoring mode, budget-friendly
Weaknesses	Higher cost, more complex setup	Limited to 2.4 GHz band	Older technology, less suitable for high-speed demands	Older, limited throughput
Price	RM 299.00	RM 52.00	RM 184.90	RM 49.00

On top of that, the AWUS036NHA has an MTU size that can be easily adjusted upward to exceed 1500 bytes. More significant MTU sizes are essential, especially in a mesh network using Batman-adv, because they may eliminate the need to disburb packets into smaller parts providing maximum platform efficiency and throughput. Moreover, it features a high-gain antenna that enhances its signal coverage and penetration, guaranteeing steady link quality even across extended distances with numerous physical obstacles between endpoints.

As shown in Table 2.1, the combination of a reliable chipset, Mesh Point capabilities, high MTU, and a strong antenna design makes Alfa AWUS036NHA a good fit for deployment in a Batman-adv protocol-based mesh network. It is a reasonable option for hardware for network builders who seek a balance between performance, compatibility, and cost. Alfa AWUS036NHA’s features also make it suitable for setups prioritizing network stability and coverage range while not requiring the high throughput levels available in more expensive dual-band adapters.

2.7 Database Synchronization Method

Decentralized database synchronization approaches play an important role in a B.A.T.M.A.N. advanced protocol-based wireless mesh ad-hoc networks as every node holds its own database and there is no central hub to regulate synchronization and replication. This literature review identifies several related methods suitable for the B.A.T.M.A.N. advanced protocol by considering several examples applicable in decentralized environments and discussing their pros and cons.

2.7.1 Peer-to-Peer Synchronization

One of the fundamental principles of decentralized networks in Peer-to-Peer Synchronization. Nodes are designed to be autonomous and serve as both clients and servers, sharing and updatings information with each other. The principle is based on the interaction of nodes that continuously update their state by passing the requested data; they do this by exchanging frequency updates data using the synchronization protocol that is based on timestamps and sequence numbers. This is essential to ensure effective data versioning, and in the event of conflicts, the use of strategies such as Last Write Wins or Multi-Version Concurrency Control should not be neglected to ensure data consistency, especially if the node topology is constantly changing.

2.7.2 Gossip Protocols

Another reliable synchronization method is the Gossip Protocols. Especially suitable for systems with unstable cluster configurations, gossip protocols significantly improve the data mirroring and fault tolerance. In general, gossip may work as a way of spreading information in a network where every node shares data with a random set of peers and receives updates constantly. Such a method ensures immediate distribution of the update but might sometimes happen inefficiently, which results in synchronization delays. As demonstrated in Figure 2.6, Gossip Protocol will periodically send data to other nodes that are within its range and for every repeated time interval (He et al., 2019).

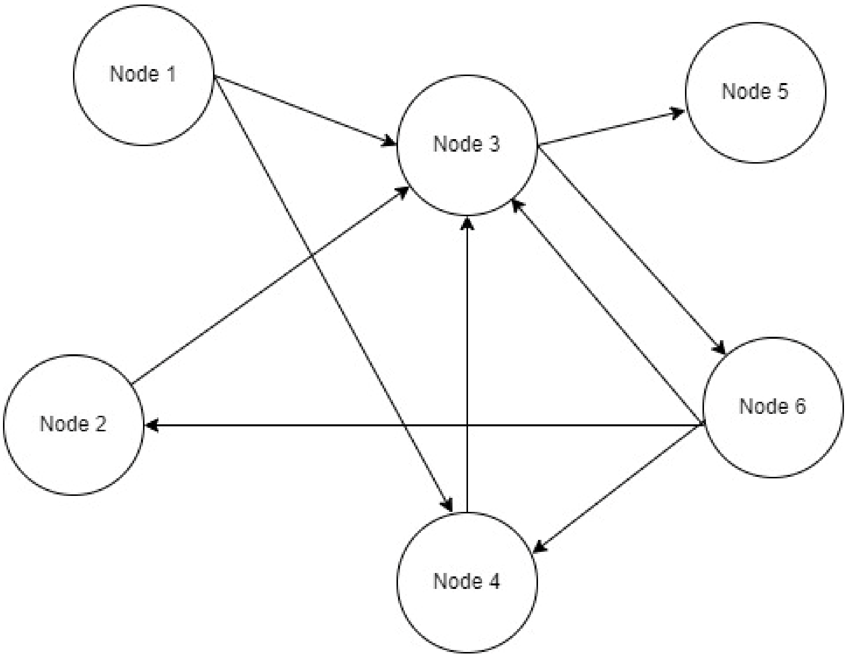


Figure 2.6: Gossip Protocol

2.7.3 Distributed Hash Tables

In decentralized systems, Distributed Hash Tables (DHT) allow to structure the storage and retrieval of data. It uses a function that sends data to a specific node depending on the hash function. Thus, the primary purpose of DHT is to reduce the search time of data and keep the load on the nodes balanced. However, it also implicates that the hash function and the node must be closely monitored to avoid data leaks and unbalanced weapons (Tao Qian et al., 2014).

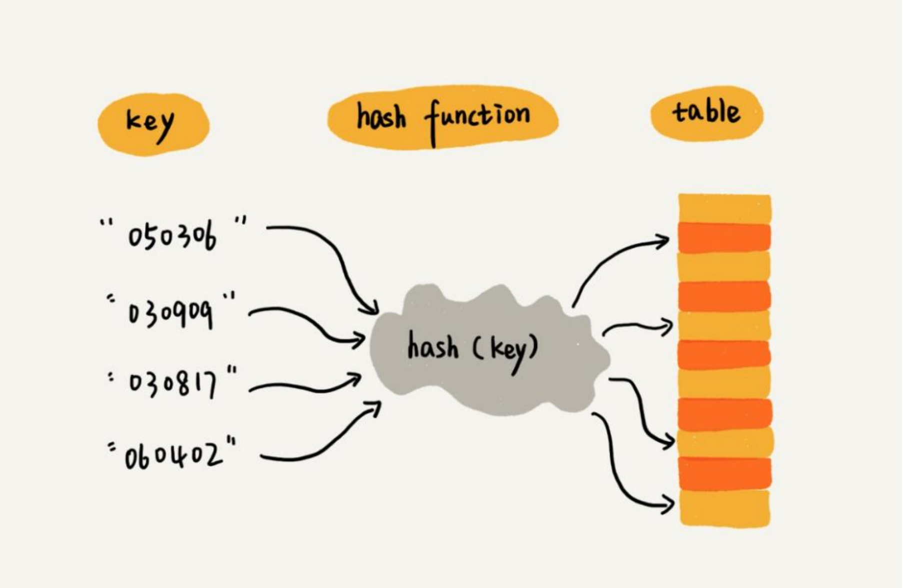


Figure 2.7: Distributed Hash Table (DHT) (Luis, 2013)

2.7.4 Blockchain Technology (Consensus Mechanism)

Lastly, due to the advantages of immutability and high verifiability of record-keeping, Blockchain Technology can be considered a remedy to the discussed issue. Applying consensus mechanisms, such as Proof of Work or Proof of Stake , Blockchain adheres to such a principle that once the data is recorded for the first time, the further change is impossible without reaching the agreement of more than half of all nodes. In such a way, the integrity and security of the data are extremely high. Still, at the same time, in the networks where the nodes are powered by low-resistant devices, the computational cost and time taken by such mechanisms can be irrational (Hussein et al., 2023).

To sum up, all synchronization methods are characterized by their advantages and challenges for deployment in decentralized mesh networks. The optimal selection of synchronization options should be based on the performance that is required, which includes scalability, data sensitivity, node capabilities, and the dynamic nature of the network. It is recommended to combine the methods in hybrid form to use their benefits and compensate the drawbacks. This would help to achieve stable and reliable data synchronization in parallel with decentralized data exchanges via wireless mesh networks.

2.8 Existing Mesh Networking Solutions

In disaster-prone and remote areas, enabling reliable communication remains a challenge due to the lack of traditional network infrastructures or fixed network technologies. Thus, various existing mesh networking solutions were explored.

2.8.1 Hybird Mesh Networking Solution using WiFi and LoRa

In previous studies, researchers have explored various architectures for enhancing disaster response communication. One such work proposed a drone-based communication infrastructure that forms a wireless mesh network but is constrained by the transmission range limitations of Wi-Fi technology. Another approach developed a system for distributing content synchronously through Wi-Fi mesh networking, though this system is still bound by the inherent limitations of Wi-Fi. To improve IoT coverage, some researchers implemented a Device-to-Device (D2D) solution using LoRaMAC for data dissemination.

However, this approach mainly addressed the transmission of generic data types rather than multimedia content such as text or images.

Further analysis has focused on unmanned aerial vehicles (UAV) with LoRa networks for disaster management, particularly through ns-3 simulations, demonstrating the potential of LoRa in such applications. A hybrid ad-hoc network that combines Wi-Fi and LoRa was also developed, leveraging smartphones and IoT devices as mesh network nodes. Despite the innovation, the data distribution in this setup is limited to plain text but not multimedia data. In contrast, this project, focusing on Wireless Ad-hoc Mesh Networks using BATMAN-adv, aims to address these limitations by facilitating the synchronization of both text and images among nodes. This added capability is crucial for effective decision-making in disaster scenarios, allowing response teams to utilize a broader range of data for critical assessments (Tham et al., 2023).

2.8.2 Drone-Based Mesh Networks for Disaster Response

In the study conducted by Chand et al., the authors developed a drone-based wireless mesh network that utilizes Wi-Fi technology to establish rapid and flexible communication infrastructures in disaster zones. This mesh network which facilitated by drones, supports real-time video transmission and internet access and achieving transmission speeds of up to 160 Mbps in line-of-sight conditions. The study emphasizes the importance of using drones for low-cost, quick deployment in disaster situations, as opposed to traditional methods that depend on higher altitude aircraft or balloons, which are more expensive and take longer to implement.

The approach outlined by Chand et al. shares similarities with this project in terms of focusing on decentralized, autonomous communication networks. However, this project specifically employs the BATMAN-adv protocol for routing within the mesh network. While their Wi-Fi-based solution is effective, it suffers from limitations related to range and potential interference, particularly in congested environments. The implementation of BATMAN-adv provides greater reliability through its ability to dynamically adjust routes and retransmit packets, making it more suitable for scenarios with dense interference or rapidly changing network conditions.

This review of existing literature reveals that while Wi-Fi-based mesh networks are a common solution for disaster response, the BATMAN-adv protocol used in this project offers key advantages in terms of routing efficiency, scalability, and network resilience. These features are particularly beneficial in disaster situations where traditional communication infrastructures are often unreliable and unavailable. (Chand et al., 2018).

2.8.3 Real Time based Mesh Networks for Shrimp Farm Monitoring

In the paper "Implementation of mobile ad-hoc network using BATMAN routing protocol for salinity monitoring in Vaname shrimp farms", the authors proposed a solution to address the challenges of manually monitoring salinity levels in shrimp farms. The traditional method of manually visiting each pond plot to check salinity levels was inefficient and time-consuming. To solve this, they designed a Mobile Ad-Hoc Network (MANET) utilizing the BATMAN-adv protocol, which creates a self-sustaining wireless communication network for real-time data transmission across nodes.

The methodology involved configuring a set of nodes, consisting of laptops and Orange Pi Zero devices, with each node connected to a salinity sensor. The nodes were placed across different pond plots and established a mesh network using BATMAN-adv. It enables the transmission of salinity data without the need for a fixed network infrastructure. Performance metrics such as delay, jitter, packet loss, CPU and memory usage, and throughput were collected using tools like Wireshark to assess the network's performance.

The study found that the BATMAN-adv protocol performed effectively in providing communication services in an area without network infrastructure. The network was reliable, with low delay and jitter values, minimal packet loss, and efficient CPU and memory usage. The study concluded that the BATMAN-adv MANET is a effective solution for environments requiring decentralized communication, particularly in areas with minimal infrastructure, such as shrimp farms. This aligns with this final year project, which also involves implementing a wireless ad-hoc mesh network using BATMAN-adv for disaster monitoring. Both projects highlight the effectiveness of the BATMAN protocol in creating robust and efficient communication networks in scenarios where traditional infrastructure is unavailable or unreliable (Larasati et al., 2023).

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

A wireless mesh ad-hoc network established through the BATMAN-adv protocol is at the center of the implemented approach to promoting resilient communication capabilities in the decentralized environments where traditional network infrastructures are not applicable. The approach can be thoroughly divided into individual stages, where each phase targets a key aspect of the network's operation.

3.1.1 Network Design and Implementation

The first phase consists in creating a mesh network structure. Three Raspberry Pi nodes were backed with an Alfa AWUS036NHA wireless adapter and carefully set up to cooperatively build the mesh topology using the BATMAN-adv protocol version IV. This step is crucial for providing the nodes with the capacity to independently build a mesh topology, control links, and convey data packets throughout the network without involvement.

3.1.2 Data Synchronization Strategy

Moreover, the study highlights the implementation of a Gossip Protocol that enables the synchronization of data distributed over the mesh. The nodes are set to conduct frequent and random data exchanges amongst their neighbors makes sure each one of them upholds an updated copy of the dataset. This is customized to effectively facilitate the spread of text and image data throughout the network.

3.1.3 Performance Analysis

A performance analysis is carried out as part of efficiency assessment for the network. Throughput, latency or delay, packet loss, Packet Delivery Ratio (PDR), Glass-to-Glass latency and database synchronization latency are analyzed to determine the network's overall operational performance across multiple conditions. The outcomes also become critical in guiding optimization speculations that tailor the network to accommodate the needs of dynamic networking comprehensively.

3.1.4 Integration of AI IoT Applications

A Disaster Detection System will be integrated into the BATMAN-adv Mesh Network to showcase the potential and effectiveness of the Mesh Network in real scenario such as rural and urban areas. Why choosing Disaster Detection System? When a disaster occurred, the traditional networks that utilize star topology might failed due to the central node is down whereas the Mesh Network would not face this issue. Thus, it is a wise choice to integrate Disaster Detection with the BATMAN-adv Wireless Ad-hoc Mesh Network.

3.2 Proposed System Design

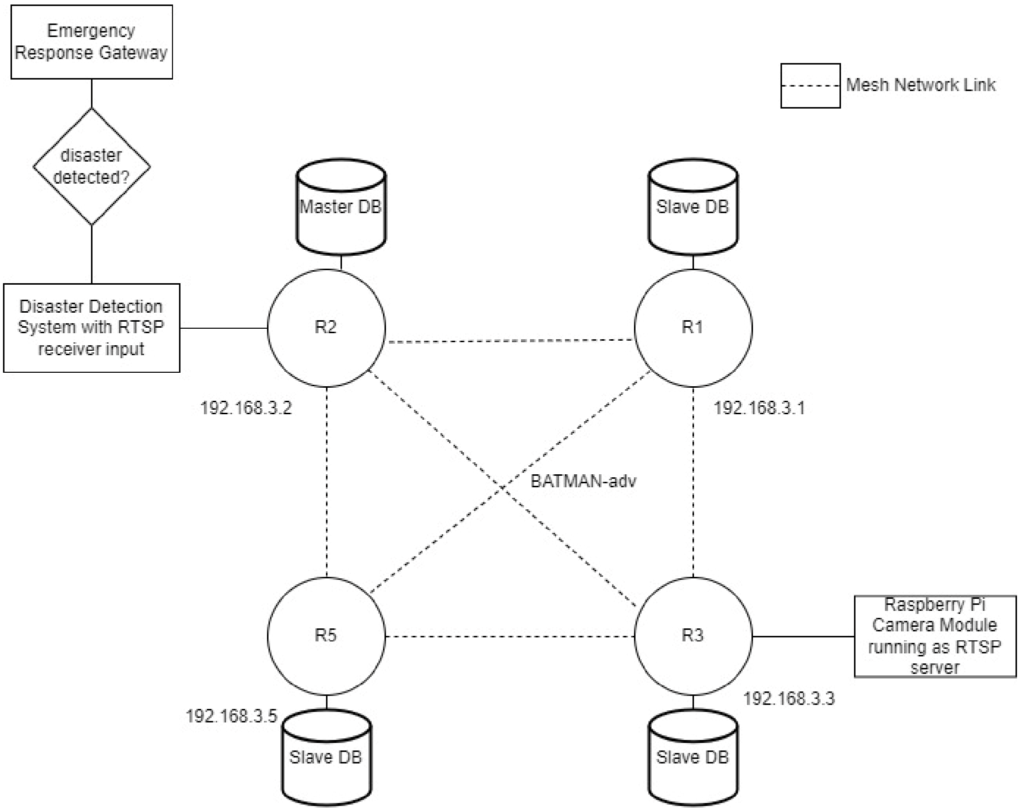


Figure 3.1: System Design

3.2.1 Hardware Selection

i. 4 units of Raspberry Pi 4 Model B+

ii. 4 units of Alfa AWUS036NHA with Atheros AR9271 chipset

iii. 4 units of microSD cards

iv. 1 units of Raspberry Pi Camera Module
v. 4 units of Power Banks/ Power Supply Raspberry Pi
vi. 6 units of ALFA 3 dBi Directional Antenna
vii. 2 units of RP-SMA Male to 2 Female Antenna Splitter

3.2.2 Network Configuration

In the proposed architecture, each Raspberry Pi node, which constitutes the basic unit of the mesh network, is equipped with the Alfa AWUS036NHA wireless adapter to be compatible with the mesh networking protocol, BATMAN adv version IV. These nodes are connected to each other forming a reliable mesh topology, which does not depend on a central node and can route data dynamically. The gateway, in turn, serves as a transitional point from the local mesh network to the internet for emergency response.

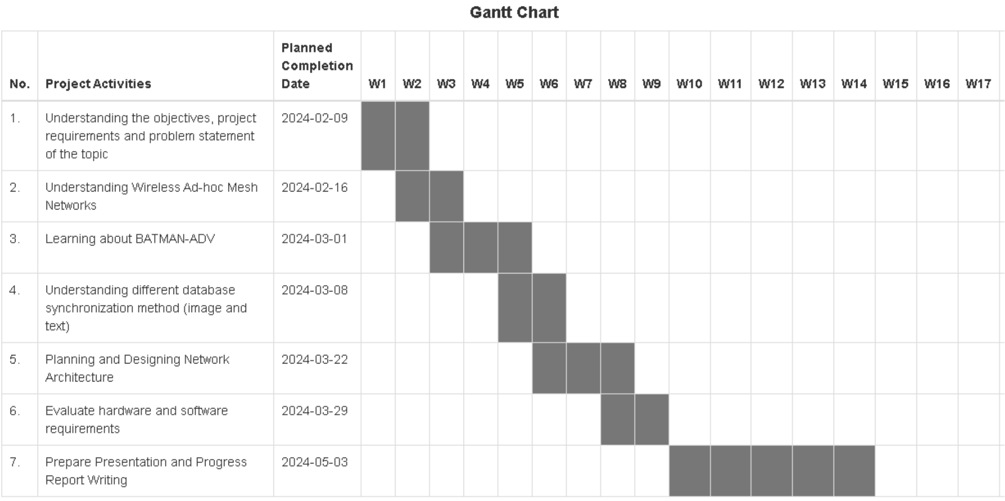
3.2.3 Work Plan

Firstly, The Gantt Chart of FYP I and FYP II were drafted as shown in Figure 3.2 and 3.3 for the workflow planning of the entire project. As shown in Figure 3.4, this project begins with a comprehensive literature review on Mesh Networks, with a specific focus on BATMAN-adv (Better Approach To Mobile Adhoc Networking) and related studies. This phase involves researching existing mesh network architectures, protocols, and relevant case studies to establish a solid foundation for the project. Simultaneously, hardware options are evaluated to ensure compatibility with the mesh network design. After the literature review, the next step is the network design and implementation phase. Here, the overall architecture of the mesh network is developed, and appropriate hardware is selected. This hardware is then purchased and tested for suitability in the context of the mesh network. If the chosen hardware meets the project requirements, the process moves forward; otherwise, a loop is followed to re-evaluate and select alternative hardware. Once the hardware is confirmed, the mesh network is set up, and the BATMAN-adv module is installed to enable communication between network nodes.

Parallel to this, a literature review on data synchronization methods is conducted. This research focuses on identifying various synchronization techniques and assessing their effectiveness for this specific network setup.

Multiple synchronization methods are then tested to determine the best-suited method for the project.

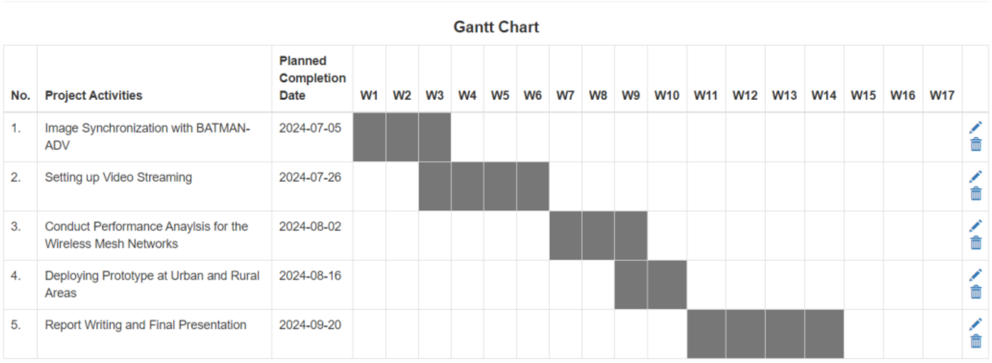
After the network is established, the next step involves tuning the BATMAN-adv parameters to optimize the network performance. Following this, the performance analysis of the mesh network is conducted, where various parameters like latency, throughput, and reliability are analyzed to ensure the network is operating efficiently. Once the network is optimized, it is integrated with an AI IoT application, such as disaster detection systems, to extend its functionality. This integration marks the final technical step before the project is considered complete. The entire project concludes with a performance evaluation and successful implementation of the AI IoT solution on the mesh network.



Gantt Chart

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17
1.	Understanding the objectives, project requirements and problem statement of the topic	2024-02-09	■	■															
2.	Understanding Wireless Ad-hoc Mesh Networks	2024-02-16		■	■														
3.	Learning about BATMAN-ADV	2024-03-01			■	■	■												
4.	Understanding different database synchronization method (image and text)	2024-03-08					■	■											
5.	Planning and Designing Network Architecture	2024-03-22						■	■	■									
6.	Evaluate hardware and software requirements	2024-03-29								■	■								
7.	Prepare Presentation and Progress Report Writing	2024-05-03										■	■	■	■	■			

Figure 3.2: Gantt Chart for FYP I



Gantt Chart

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	
1.	Image Synchronization with BATMAN-ADV	2024-07-05	■	■	■															✎ 🗑
2.	Setting up Video Streaming	2024-07-26			■	■	■	■												✎ 🗑
3.	Conduct Performance Anaylsis for the Wireless Mesh Networks	2024-08-02							■	■	■									✎ 🗑
4.	Deploying Prototype at Urban and Rural Areas	2024-08-16									■	■								✎ 🗑
5.	Report Writing and Final Presentation	2024-09-20											■	■	■	■				✎ 🗑

Figure 3.3: Gantt Chart for FYP II

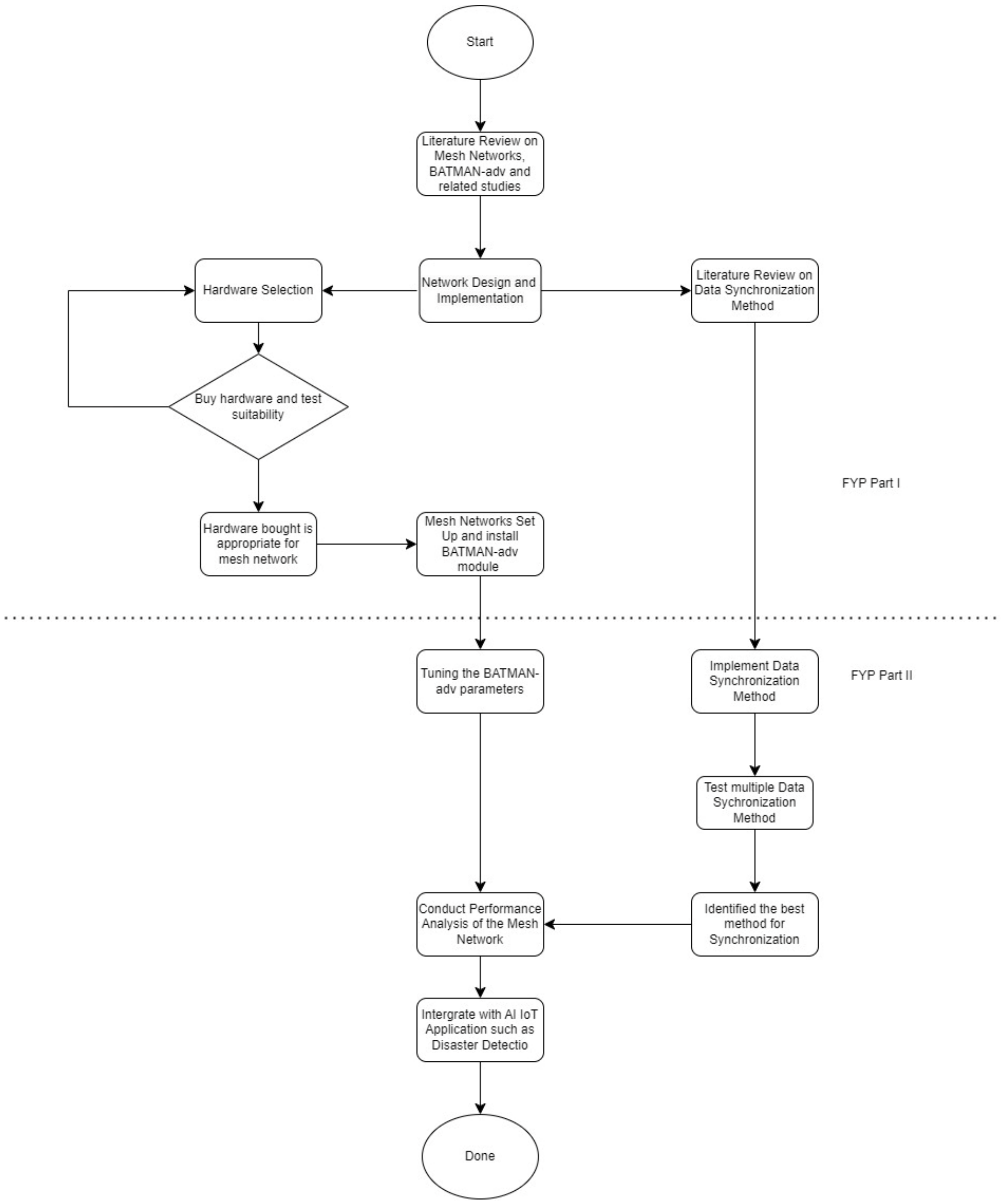


Figure 3.4: Flowchart of the Work Plan

3.3 Network Design and Implementation

There are several steps to implement a mesh network using Raspberry Pi devices with BATMAN-adv modules and WiFi adapters. Each step must be performed systematically since the proper interaction between all components is of paramount importance for the entire task’s feasibility.

Initially, there is a need to prepare Raspberry Pi devices. The selected operating system, such as Raspberry Pi OS in this case, should be flashed onto the SD cards that devices will boot from. After booting, several initial configuration tasks should be implemented, including the setup of necessary locales, the expansion of the filesystem, and the basic network configuration that will allow the device to be reached. Considering that individual nodes need to communicate, the following step of installing the required modules and dependencies should be directed at establishing this connectivity. It is also necessary to pay particular attention to ensuring that it is possible to connect to wireless networks. Subsequently, the installation and configuration of the most essential part of the task, the BATMAN-adv protocol, follows. The protocol is essential because it constitutes the wireless mesh networking operation of Raspberry Pis. The installation should be implemented by downloading and compiling BATMAN-adv software, which is supported by a variety of official sources. The final step in the process is the configuration of the protocol to adjust the network settings for the entire group.

Finally, considering the goal of proving that the nodes interact properly, WiFi adapters should be installed on the Raspberry Pi devices. All adapters must be compatible with the required communication standards and modes, which accommodate the configuration of the BATMAN-adv protocol. Within the research, the Alfa AWUS036ACH, the TP-Link TL-WN722N V1 and V3, and the Alfa AWUS036NHA were selected in the process of evaluating the best adapter.

Once the Alfa AWUS036NHA was identified as the better-performing option in the given comparison such as its capabilities of supporting 802.11s Mesh Mode and MTU Size of more than 1532 bytes, a series of activities focused on implementing and configuring it to create a mesh network in the case of the Raspberry Pi was required. Therefore, the follow-up activity will consist of adding an Alfa AWUS036NHA adapter physically to each RAI unit and configuring it to fit the Linux platform used for the target mesh network. Specifically, the configuration of the AWUS036NHA will revolve around the installation of the drivers needed for merging the former with both the operating system and the BATMAN advanced protocol. With the device drivers installed, the network interface setting will be adjusted towards maximum efficiency, which will include the further integration of the AWUS036NHA into the BATMAN advanced protocol setting. As a result, a series of batches will have to be run, during which the settings of the Raspberry Pi and the network transmission between the nodes will be analyzed for potential errors.

Specifically, the following activities will have to be conducted: ping and batctl o running, with further analysis and possible adjustments made to enhance network stability and performance. Thus, in the end, the described series of activities will lead to the transformation of each RAI into a node of a mesh network, which will significantly improve their efficiency in operating within a decentralized networking arrangement such as one designed.

3.3.1 Mesh Network Wireless Channel

During the implementation of the mesh network, due to the limitation of the ALFA wireless adapter of only having the 2.4 Ghz band, the selection of wireless channels (1-14) is critical to optimizing network performance, especially when interference from other devices such as Bluetooth or wireless access points is present. Since a spectrum analyzer is unavailable, a trial-and-error approach is used to identify the least congested channels. This involves manually setting the mesh network to different wireless channels and observing the network performance on each channel over time. Metrics such as latency, packet loss, and throughput are recorded for each channel to assess congestion levels. Based on the results, the channel with the best performance and minimal interference is selected for the mesh network. Periodic testing is also conducted to ensure the network remains stable as surrounding wireless conditions change. After a few testing during the implementation of the network, it can found that overlapping channel such as channel 2 and 3 have less congested traffic than non-overlapping channel such as channel 1,6 and 11.

3.3.2 Insertion Loss of Antenna Splitter

One point to be aware of during the implementation of the mesh network, an antenna splitter was used (highlighted in red) with a directional antenna as shown in Figure 3.5 to ensure that signals could cover and reach other nodes effectively. To evaluate the impact of the splitter on network performance, TCP throughput from iperf3 was used as the primary metric. Initially, the network was configured using the BATMAN-adv protocol, and baseline throughput measurements without the splitter showed a rate of 11.6 Mbits/s in Figure 3.6. After integrating the antenna splitter to distribute the signal across multiple antennas, TCP throughput dropped to 10.5 Mbits/s in Figure 3.7. This decrease in performance highlights the trade-off between broader signal coverage and reduced data transfer efficiency when using a splitter for directional antennas in a mesh network. The analysis indicates that while the splitter extends the signal's reach, it also introduces some performance degradation.

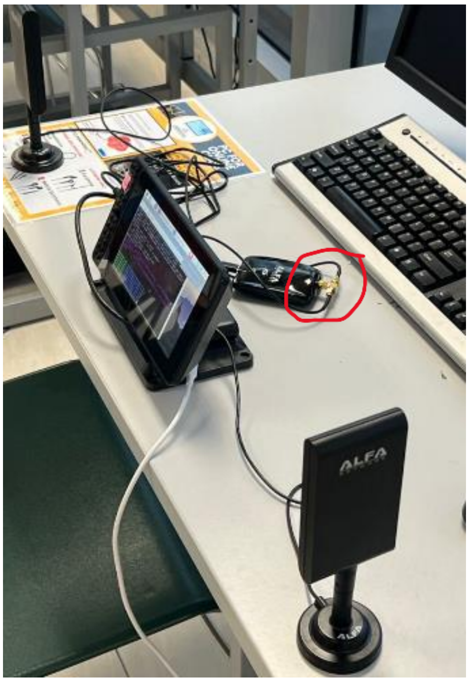
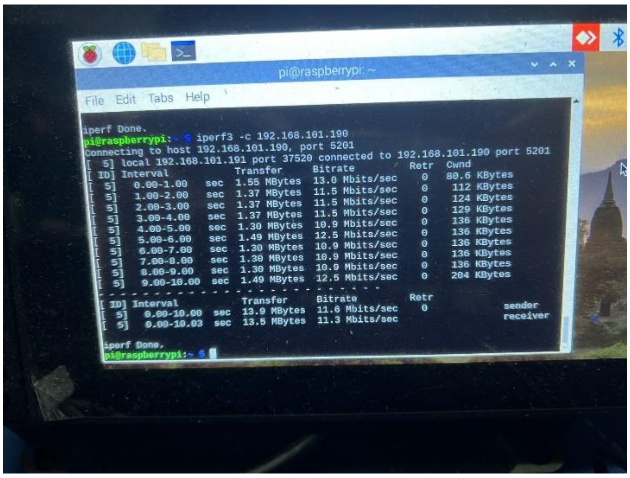


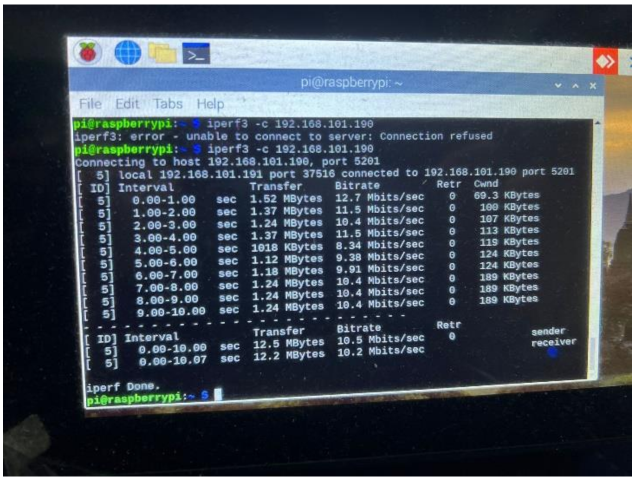
Figure 3.5: Antenna Splitter Setup



```
iperf Done.
pi@raspberrypi:~ $ iperf3 -c 192.168.101.190
Connecting to host 192.168.101.190, port 5201
[  5] local 192.168.101.191 port 37520 connected to 192.168.101.190 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.55 MBytes  13.0 Mbits/sec    0   80.6 KBytes
[  5]   1.00-2.00   sec  1.37 MBytes  11.5 Mbits/sec    0    112 KBytes
[  5]   2.00-3.00   sec  1.37 MBytes  11.5 Mbits/sec    0    124 KBytes
[  5]   3.00-4.00   sec  1.37 MBytes  11.5 Mbits/sec    0    129 KBytes
[  5]   4.00-5.00   sec  1.30 MBytes  10.9 Mbits/sec    0    136 KBytes
[  5]   5.00-6.00   sec  1.49 MBytes  12.5 Mbits/sec    0    136 KBytes
[  5]   6.00-7.00   sec  1.30 MBytes  10.9 Mbits/sec    0    136 KBytes
[  5]   7.00-8.00   sec  1.30 MBytes  10.9 Mbits/sec    0    136 KBytes
[  5]   8.00-9.00   sec  1.30 MBytes  10.9 Mbits/sec    0    136 KBytes
[  5]   9.00-10.00  sec  1.49 MBytes  12.5 Mbits/sec    0    204 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  13.9 MBytes  11.6 Mbits/sec    0             sender
[  5]   0.00-10.03  sec  13.5 MBytes  11.3 Mbits/sec                  receiver

iperf Done.
pi@raspberrypi:~ $
```

Figure 3.6: TCP Throughput without Splitter



```
pi@raspberrypi:~ $ iperf3 -c 192.168.101.190
iperf3: error - unable to connect to server: Connection refused
pi@raspberrypi:~ $ iperf3 -c 192.168.101.190
Connecting to host 192.168.101.190, port 5201
[  5] local 192.168.101.191 port 37516 connected to 192.168.101.190 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.52 MBytes  12.7 Mbits/sec    0   69.3 KBytes
[  5]   1.00-2.00   sec  1.37 MBytes  11.5 Mbits/sec    0    100 KBytes
[  5]   2.00-3.00   sec  1.24 MBytes  10.4 Mbits/sec    0    107 KBytes
[  5]   3.00-4.00   sec  1.37 MBytes  11.5 Mbits/sec    0    113 KBytes
[  5]   4.00-5.00   sec  1018 KBytes  8.34 Mbits/sec    0    119 KBytes
[  5]   5.00-6.00   sec  1.12 MBytes  9.38 Mbits/sec    0    124 KBytes
[  5]   6.00-7.00   sec  1.18 MBytes  9.91 Mbits/sec    0    124 KBytes
[  5]   7.00-8.00   sec  1.24 MBytes  10.4 Mbits/sec    0    189 KBytes
[  5]   8.00-9.00   sec  1.24 MBytes  10.4 Mbits/sec    0    189 KBytes
[  5]   9.00-10.00  sec  1.24 MBytes  10.4 Mbits/sec    0    189 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  12.5 MBytes  10.5 Mbits/sec    0             sender
[  5]   0.00-10.07  sec  12.2 MBytes  10.2 Mbits/sec                  receiver

iperf Done.
pi@raspberrypi:~ $
```

Figure 3.7 TCP Throughput with Splitter

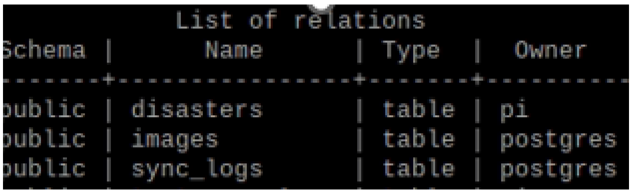
3.4 Data Synchronization

Firstly, the data synchronization method follows a master-slave approach for handling critical disaster detection data. In this setup, the RTSP receiver is responsible for capturing and storing key information such as the disaster type, number of victims, and images. To optimize storage and transmission efficiency, the images are first converted into hexadecimal format before being stored in the database and the hexadecimal format of the image is saved in BLOB (Binary Large Object) due to it’s large bytes size. This conversion helps secure and streamline the process, ensuring that the data remains consistent and intact during transmission. When the data is synchronized with another database, the hex format is converted back into JPG, allowing for easy access and retrieval of image data for further analysis.

In addition to the master-slave system, a Gossip protocol is used for periodic synchronization across the network. This protocol operates at intervals of 5 to 10 seconds, ensuring that all nodes regularly exchange the most recent data. Each node connects with a randomly selected neighboring node and shares the latest data packet until every node in the network is updated. This decentralized synchronization ensures that data remains consistent across the entire network, even when there are frequent updates or changes.

By combining the master-slave approach for critical data with the Gossip protocol for regular updates, the system achieves both reliability and scalability. The master-slave method handles essential disaster-related data efficiently, while the periodic updates through the Gossip protocol maintain synchronization across the mesh network, ensuring all nodes stay current.

As shown in Figure 3.8, four relations was created in the node database. The disasters relation was used to store text information such as disaster type and number of victims, images relation was used to store images in hexadecimal format and sync_logs relation are used to record the text and image synchronization start time and stop time.



```
           List of relations
Schema |      Name       | Type  |  Owner
-------+-----------------+-------+----------
public | disasters       | table | pi
public | images          | table | postgres
public | sync_logs       | table | postgres
```

Figure 3.8: Node Database Relations

3.5 Database Selection

The database was changed from SQLite3 to PostgreSQL to address the issue of conflicting database copies that occurred during periodic synchronization. This adjustment was necessary to ensure smoother and more reliable synchronization. In Figure 3.9 shows the logo of PostgreSQL, it offers better support for concurrent access and complex synchronization processes, reducing the chances of conflicts that arise in distributed systems like the mesh network. PostgreSQL's robustness and scalability made it a more suitable choice for handling the periodic synchronization tasks within the mesh network.

Figure 3.9 PostgreSQL

3.6 Performance Analysis

Performance Analysis involves setting up the network with predetermined numbers of nodes and systematically applying the mentioned tests with the mesh network. The performance of the mesh network were measured and tested using multiple metrics such as latency, throughput, jitter and packet loss. This approach ensures a comprehensive understanding of network capabilities and areas that require enhancements.

Overall, the performance analysis methodology uses these metrics and calculations to thoroughly evaluate the wireless mesh network's performance, providing valuable insights into its operational capabilities and limitations under different scenarios.

3.6.1 Throughput

Throughput is the quantity of data received and successfully sent to the system in the unit time By the division of the amount of to data received (total bits) by the time to get that data (total seconds), we get it. In the TCP and UDP throughput, giving preference to the UDP Throughput is excellent as it is connection less nature with faster speed, that makes for performance cases where rapid data transmission matter than guaranteed delivery. UDP is not connection-oriented, and it does not provide any acknowledgment or retransmissions capabilities of TCP, which means lower latency and better throughput than that offered by the previous protocol. This establishes UDP as the appropriate transmission protocol for real-time applications video streaming, disaster detection, mesh topology networks that require fast and consistent data delivery. ThroughputThe formula for Throughput is as follows :

$$\text{Throughput} = \frac{\text{Total Data Received}}{\text{Total Time}} \tag{3.1}$$

3.6.2 Latency

Latency is defined as the time taken by one packet of data to go the source to the destination. Latency can be designed by both the time it was dispatched and the duration for which it was obtained and can be calculated as

$$\text{Latency} = \text{Time at Destination} - \text{Time at Source} \tag{3.2}$$

3.6.3 Packet Loss

Packet loss refers to the number of data packets undelivered between sender and receiver as a percentage, quantifying this problematic phenomenon helps diagnose network issues. To derive packet loss by subtracting received packets from those transmitted, dividing the difference by packets sent, then multiplying by one hundred to yield a percent. Specifically, the computation is: take the transmitted packet total, subtract those reached, divide this by all dispatched, and finally multiply the result by one hundred to get the packet loss percentage.

$$\text{Packet Loss Percentage} = \frac{\text{Total Packets Sent} - \text{Total Packets Received}}{\text{Total Packets Sent}} \text{x}100\% \tag{3.3}$$

3.6.4 Packet Delivery Ratio (PDR)

The Packet Delivery Ratio is calculated by dividing the number of packets successfully received at the destination by the number of packets that were sent from the source, then converting this ratio into a percentage. The formula for the Packet Delivery Ratio is:

$$\text{Packet Delivery Ratio} = \frac{\text{Total Data Received}}{\text{Total Packets Sent}} \text{ x } 100\% \tag{3.4}$$

3.6.5 Jitters

Jitter refers to the variation in packet arrival times, which is a critical performance metric, especially in real-time communication systems such as real time video streaming (RTSP) or voice over IP (VoIP). High jitter can result in delayed or out-of-order packet delivery, negatively affecting the quality of the communication. To calculate jitter, the difference in packet delay (latency) between consecutive packets is measured, and the average of these differences is taken to quantify jitter.

Specifically, the computation involves subtracting the latency of the previous packet from the latency of the current packet for each pair of consecutive packets. These differences are then averaged over the total number of packets to determine the jitter value.

$$\text{Jitter} = \frac{\text{Sum of }|\text{Latency of Packet N} - \text{Latency of Packet N} - 1|}{\text{Number of Packets} - 1} \tag{3.5}$$

3.6.6 Glass-to-Glass Latency

Glass-to-glass latency is the time it takes for one video frame to travel from the camera as soon as it is captured, until it reaches on a screen and is visible. This latency is very important in real-time systems eg disaster detection where video feedback can take more than a few seconds and this delay cannot be afforded during the crisis phase for taking decision. Glass-to-glass latency is measured by streaming video from a camera to a display using Real-Time Streaming Protocol (RTSP) over the mesh network. BATMAN-adv protocol is configured to create the mesh network and it allows multicasting of video over multiple hops from one node to another.

This latency is best measured by sending timestamps with the camera (source) and the display (destination). Facial expression recognition using camera timestamp for captured image and display timestamp for the same frame reunited. It calculates difference between the time two timestamps and gives total glass-to-glass latency. This method gives latency measurement, reflecting the real-time network performance in considering transmission delays, processing times and other condition that may take place on the network.

3.6.7 Database Synchronization Latency

The data synchronization latency is the duration when data moved from master database to slave database in the context of mesh network. Understanding this latency is important to analyse how fast the essential information, like disaster data, is synchronized over the network. This is measured by some process, which starts when the master database pushes data (disaster type or image files) to the slave database. A timestamp is taken right on the moment, when the data from the master database is forwarded.

As soon as the slave database gets the data, it makes another timestamp. The data synchronization latency is the difference between the two timestamps. This approach guarantees exact time taken for information to cross the network and update the slave database. This time difference can reconstruct a measure to evaluate the synchronization performance for mesh network namely timely data consistency and thereby system responsiveness.

3.6.8 Network Performance Evaluation

As shown in Table 3.1, the metrics were used to evaluate the mesh network performance and the tools used are stated below.

Table 3.1: Network Performance Metrics and Tools

Metrics	**Tools**
Latency	ICMP Ping (64 Bytes)
Packet Loss	ICMP Ping (64 Bytes)
TCP Throughput	iperf3 TCP
UDP Throughput	iperf3 UDP
Jitter	iperf3 UDP
Glass-to-Glass Latency	Gstreamer timestamp, Frame timestamp
Database Synchronization Latency	postgreSQL

3.6.9 Network Performance Evaluation Phase

The Evaluation Phase of the Mesh Network Performance starts with a single hop then two hops and ends with three hops as shown below in Table 3.2.

Table 3.2: Network Performance Evaluation Phase

Phase	**Route**
Single Hop	i. Node 1 > Node 2 ii. Node 2 > Node 3 iii. Node 3 > Node 4 iv. Node 4 > Node 3 v. Node 3 > Node 2 vi. Node 2 > Node 1
Two Hops	i. Node 1 > Node 2 > Node 3 ii. Node 2 > Node 3 > Node 4

	iii. Node 4 > Node 3 > Node 2 iv. Node 3 > Node 2 > Node 1
Three Hops	i. Node 1 > Node 2 > Node 3 > Node 4 ii. Node 4 > Node 3 > Node 2 > Node 1

3.7 Integration with AI IoT application

The integration of the Disaster Detection System into the mesh network involves configuring specific nodes to handle distinct tasks. Node 1 is designated as the RTSP receiver node, responsible for receiving video streams from other nodes, particularly the camera node (Node 3). Upon receiving the video frames, Node 1 processes them using the disaster detection system to identify potential incidents in real time. Once the system detects a disaster, the relevant data such as the type of disaster and the number of victims is stored in the master database. This master database is responsible for synchronizing disaster data with other slave databases across the mesh network, ensuring all nodes maintain updated information.

On the other hand, Node 3 functions as the camera node, operating as an RTSP server. GStreamer is used to set up Node 3 for video streaming, ensuring it can reliably send video frames to Node 1 for disaster detection. This setup leverages the capabilities of the mesh network to ensure that video and detection data can be efficiently transmitted and synchronized, enabling prompt disaster response and coordination across all nodes.

3.8 Summary

In conclusion, this project focuses on the integration of a wireless mesh network using the BATMAN-adv protocol, combined with AI-based disaster detection and real-time data synchronization. The network is built using Raspberry Pi nodes equipped with Alfa wireless adapters, allowing for decentralized communication in environments where traditional network infrastructures are inadequate.

The network design begins with the installation and configuration of the mesh network using Raspberry Pi nodes, which are set up to autonomously form a reliable and adaptive mesh topology. The data synchronization strategy employs a master-slave approach, where critical disaster detection data such as video streams and images are captured, processed, and stored in the master database. A Gossip protocol is used for periodic updates across nodes to ensure data consistency.

The performance analysis involves evaluating key metrics such as throughput, latency, jitter, packet loss, and database synchronization latency, with a focus on UDP throughput due to its high-speed, connectionless nature that is well-suited for real-time applications like disaster detection. Glass-to-glass latency is measured to assess the time taken for video data to travel from the camera to the display screen, and data synchronization latency is measured by tracking the time taken for data to transfer from the master database to the slave databases.

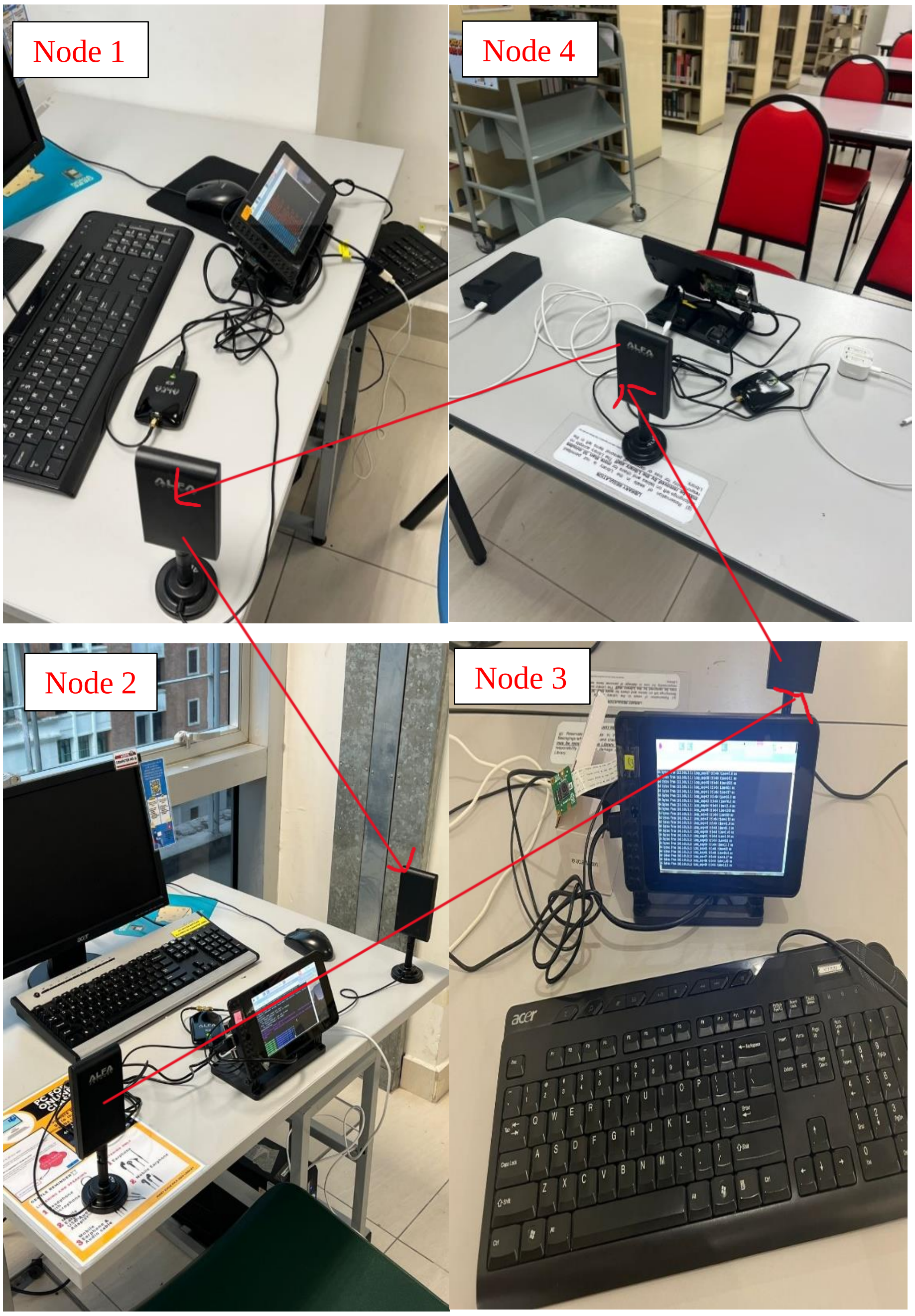
Finally, the disaster detection system is integrated into the mesh network. Node 1 functions as the RTSP receiver that processes video frames for real-time disaster detection and stores the data in the master database, while Node 3 operates as the RTSP server using GStreamer to stream video to Node 1. This integrated system ensures efficient data transmission, synchronization, and disaster response across the network.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

Figure 4.1 demonstrates the prototype of the Wireless Ad-hoc Mesh Network with BATMAN-adv testbed. The testbed was deployed at UTAR Sg. Long Campus Mary KUOK Pick Hoo Library and performance analysis of the Mesh Network was conducted.



RTSP Receiver Node with Disaster Detection

Camera Module with RTSP Streaming

Figure 4.1: Wireless Ad-hoc Mesh Network Prototype

The four nodes Mesh Network testbed were deployed as shown in Figure 4.1 and the red lines represent the signal propagation of the directional antenna.

Table 4.1: Distance between the four nodes

Route	**Distance (m)**
Node 1 - Node 2	24.83
Node 2 - Node 3	29.00
Node 3 - Node 4	19.06

4.2 Performance Analysis

In this section, the performance of the BATMAN-adv mesh network is evaluated using a variety of key metrics, including TCP throughput, UDP throughput, latency, jitter, packet loss, glass-to-glass latency, and database synchronization latency. These metrics are essential for understanding the network's overall efficiency and reliability, especially in real-time disaster monitoring applications. The analysis is conducted across different hop scenarios, including single hop, two hop, and three hop, to assess how the network performs as the number of intermediate nodes increases. By examining these metrics, we can gain insight into the network’s capacity to handle both data-intensive and real-time communication tasks effectively.

4.2.1 Single Hop Link Evaluation

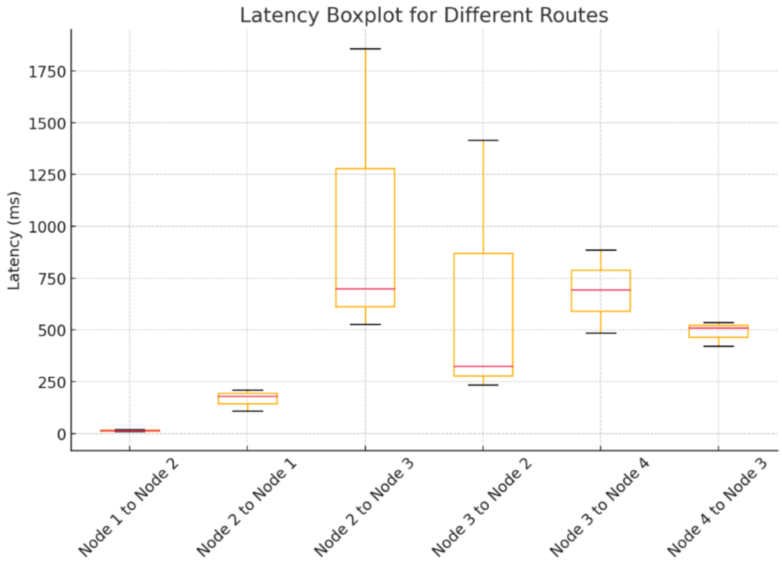


Figure 4.2: Average Latency Boxplot (Single Hop)

As shown in Figure 4.2, the latency boxplot represents the results from a series of three tests conducted for each route in the mesh network, using ICMP Ping with a 64-byte payload and 100 pings per test. The routes Node 1 to Node 2 and Node 2 to Node 1 show consistently low latency and minimal variability, reflecting stable performance. However, the routes involving Node 2 to Node 3 and Node 3 to Node 4 exhibit higher latency and greater variability. This increase in latency might be due to the antenna splitter used on Node 2 and Node 3, which could introduce signal degradation and delays in the transmission between nodes. The antenna splitter could affect the signal strength and thus contribute to the higher latency observed in these routes.

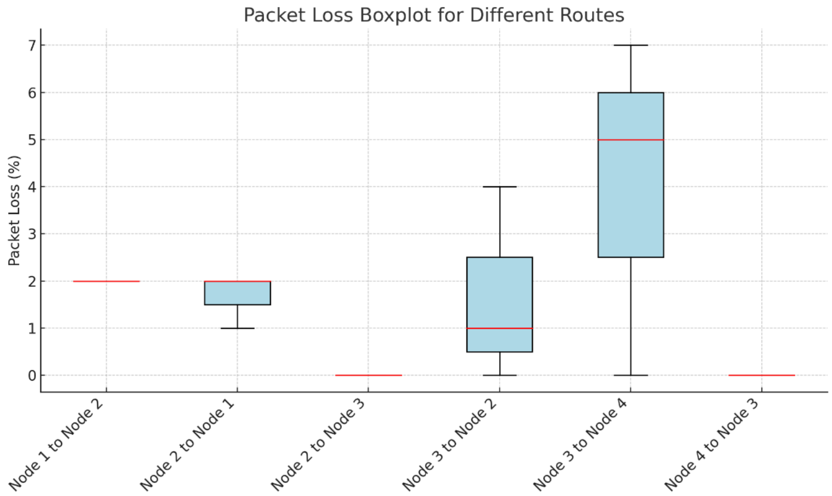


Figure 4.3: Packet Loss Percentage Boxplot (Single Hop)

As demonstrated in Figure 4.3, the boxplot illustrates the packet loss results for different routes within the mesh network, based on ICMP Ping tests using a 64-byte payload with 100 pings conducted for each route. The routes between Node 1 to Node 2 and Node 2 to Node 1 exhibit consistent packet loss of around 2%, indicating stable performance with minor losses. In contrast, Node 2 to Node 3 and Node 4 to Node 3 demonstrate 0% packet loss, highlighting highly reliable communication on these routes with no significant issues. However, the route from Node 3 to Node 2 shows moderate variability in packet loss, ranging from 0% to 4%, suggesting that this route experiences occasional packet delivery issues. The Node 3 to Node 4 route has the highest variability, with packet loss reaching up to 7%, indicating less reliable performance, possibly due to increased distance or interference between these nodes. These results highlight that certain routes perform better than others in terms of packet reliability.

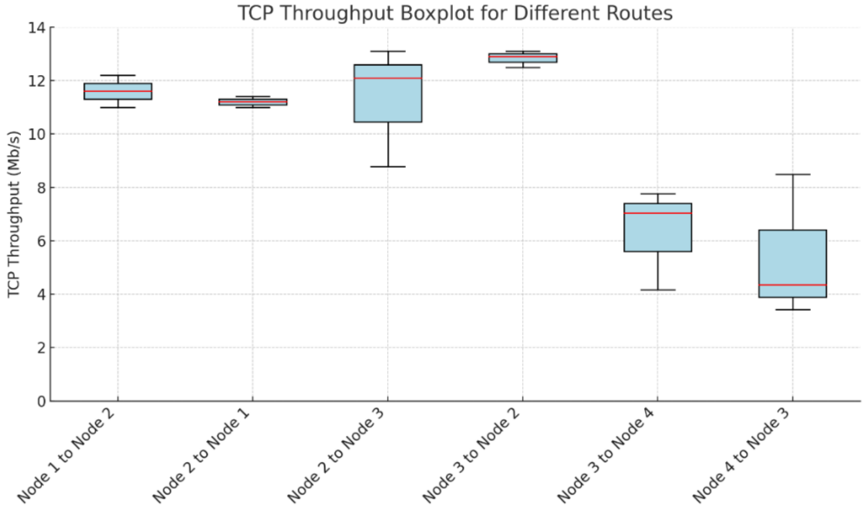


Figure 4.4: TCP Throughput Boxplot (Single Hop)

In Figure 4.4, the TCP Throughput boxplot shows the variation in throughput across different routes within the mesh network, with the y-axis starting from zero to accurately reflect performance. Routes like Node 1 to Node 2 and Node 2 to Node 1 demonstrate consistent and high throughput, ranging around 11 to 12 Mb/s, indicating stable data transmission over these routes. Similarly, Node 3 to Node 2 maintains high throughput with minimal variation, showing reliable network performance.

However, the routes Node 3 to Node 4 and Node 4 to Node 3 exhibit more significant variation, with lower throughput values ranging from around 3.43 to 8.48 Mb/s. This wider range of throughput values suggests potential performance issues, likely due to network interference or the use of an antenna splitter, which could be causing signal degradation on these routes. The lower and more variable throughput indicates that these routes may experience bottlenecks or signal disruptions, which could impact overall network performance.

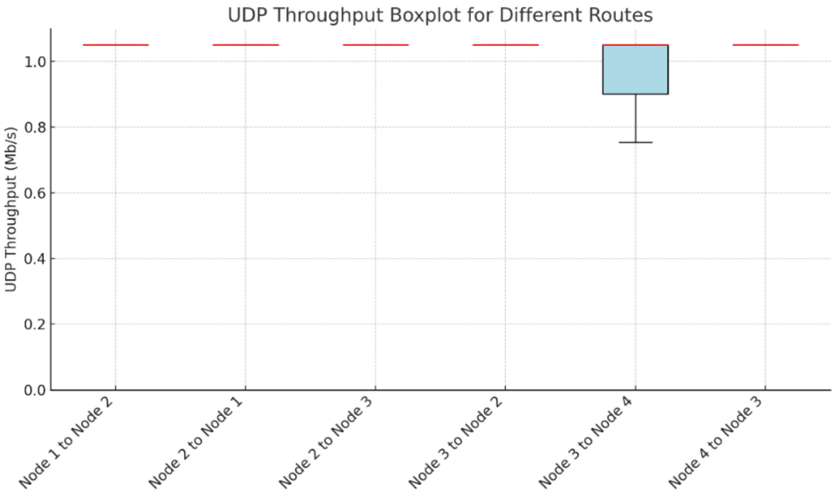


Figure 4.5: UDP Throughput Boxplot (Single Hop)

In Figure 4.5, the boxplot illustrates UDP throughput for different routes within the mesh network, measured in Mb/s. For most routes, including Node 1 to Node 2, Node 2 to Node 1, and Node 3 to Node 2, the throughput remains consistently at 1.05 Mb/s without any significant variation, indicating stable performance.

However, the route Node 3 to Node 4 shows a wider distribution, with a median of around 1.05 Mb/s but a notable drop in performance with values as low as 0.75 Mb/s. This variability suggests that there might be interference or network congestion affecting the throughput for this specific route. This behavior might be attributed to factors such as environmental interference or hardware limitations on specific routes, leading to lower throughput values.

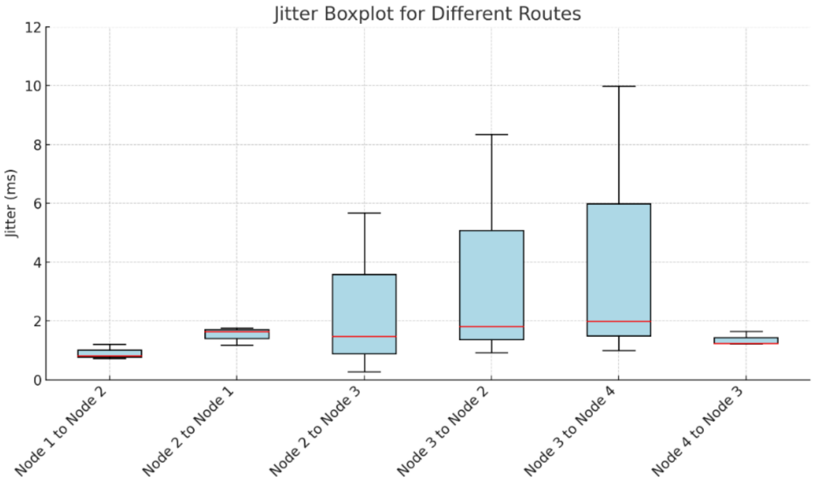


Figure 4.6: Jitter Boxplot (Single Hop)

In Figure 4.6, the jitter boxplot illustrates the variation in delay between data packets for different routes in the mesh network. Each boxplot represents jitter measurements taken across multiple test runs for different node-to-node connections. The routes such as "Node 1 to Node 2" and "Node 2 to Node 1" show relatively low and consistent jitter, which indicates that these routes have stable transmission times with minimal variation in packet delivery. In contrast, routes like "Node 3 to Node 4" and "Node 3 to Node 2" exhibit higher jitter values, with the former showing a larger range and variability. This suggests that these routes experienced more significant fluctuation in packet delivery times, potentially due to environmental factors, hardware configuration (such as the antenna splitter), or network congestion. The high jitter values in these routes could lead to performance issues in applications sensitive to timing, such as real-time video streaming or VoIP. Overall, the results indicate that some routes have stable performance while others may need optimization to reduce jitter.

4.2.2 Two-Hop Link Evaluation

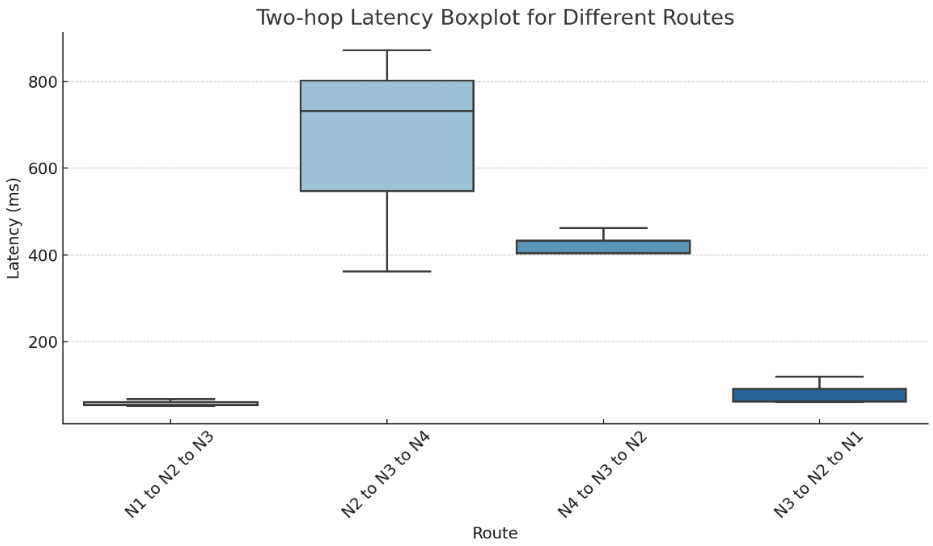


Figure 4.7: Latency Boxplot (Two Hop)

The boxplot above represents the two-hop latency measurements for different routes in the mesh network. The route from Node 2 to Node 3 to Node 4 shows the highest variability in latency, with a median value around 800 ms, indicating significant delays over this route. This could be due to network congestion or interference. On the other hand, the routes from Node 1 to Node 2 to Node 3 and Node 3 to Node 2 to Node 1 show much lower latency values, with medians close to 60 ms, demonstrating more stable and faster performance over these routes.

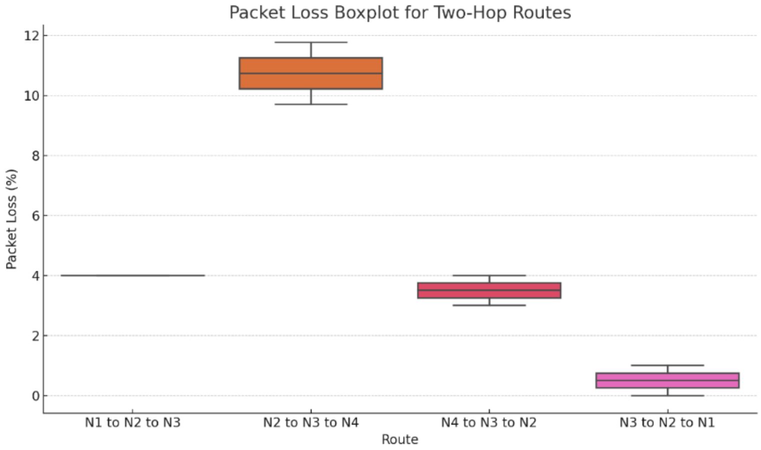


Figure 4.8: Packet Loss Percentage Boxplot (Two Hop)

In Figure 4.8, the packet loss boxplot for the two-hop routes shows distinct variations in packet loss across different routes. The route from Node 2 to Node 3 to Node 4 experiences the highest packet loss, with a median of around 10% and a range that extends above 11%. This indicates that this route is more prone to packet delivery issues, possibly due to longer distances or interference along the path. The routes from Node 1 to Node 2 to Node 3 and Node 4 to Node 3 to Node 2 exhibit relatively lower packet loss rates, with a more consistent packet loss around 4%, showing better performance. The route from Node 3 to Node 2 to Node 1 has the lowest packet loss, with values close to zero, demonstrating the most reliable data transmission in terms of packet delivery.

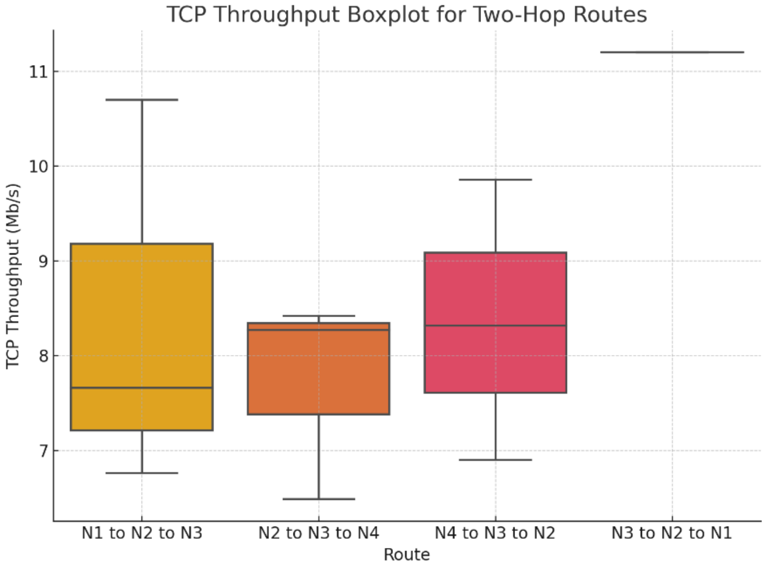


Figure 4.9: TCP Throughput Boxplot (Two Hop)

In Figure 4.9, the boxplot for the two-hop TCP throughput demonstrates variations in network performance across different routes. For the Node 1 to Node 2 to Node 3 route, the throughput shows noticeable fluctuations, with the median slightly lower compared to other routes, indicating some instability in data transmission. The ‘Node 2 to Node 3 to Node 4’ route shows a more consistent range but generally lower throughput values compared to the other routes, suggesting that this path experiences greater network congestion or interference. The Node 4 to Node 3 to Node 2 route performs the best, with a higher median throughput and relatively smaller variations, reflecting more stable network performance across this path.

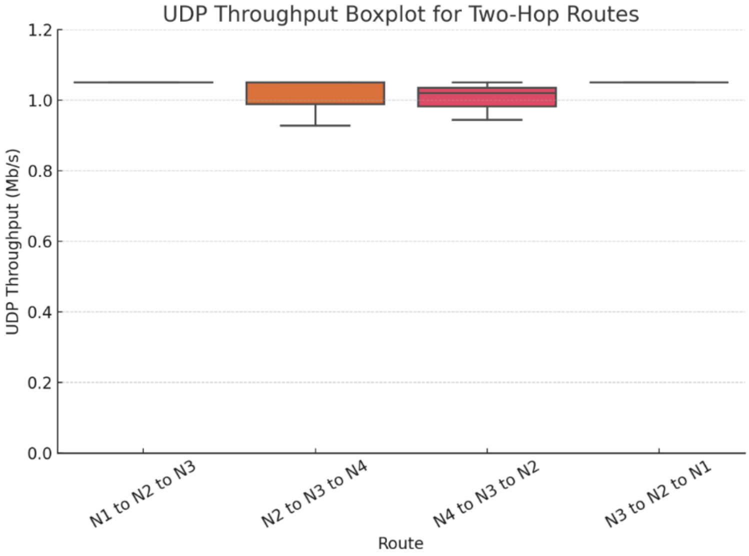


Figure 4.10: UDP Throughput Boxplot (Two Hop)

In Figure 4.10, the boxplot above illustrates the UDP throughput performance across different two-hop routes within the mesh network. The throughput measurements were fairly consistent across most routes, generally remaining around 1.05 Mb/s. Notably, Route "Node 2 to Node 3 to Node 4" and “Node 4 to Node 3 to Node 2” displayed slightly lower throughput values, with one instance dropping to 0.928 Mb/s. The consistency in throughput for the other routes demonstrates stable performance under UDP, where lower reliability is acceptable but speed is prioritized.

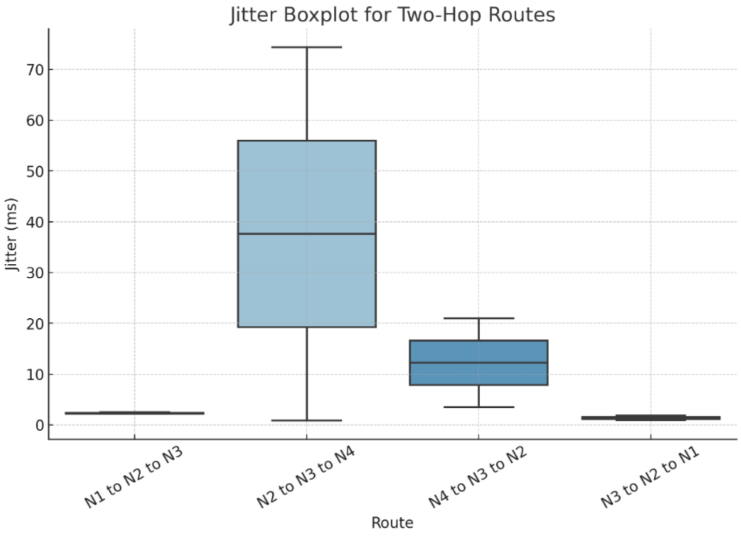


Figure 4.11: Jitter Boxplot (Two Hop)

In Figure 4.11, the boxplot illustrates the jitter measurements for different two-hop routes in the network. The route "Node 2 to Node 3 to Node 4" shows the most significant jitter variation, with values ranging from around 10 ms to over 70 ms, indicating high inconsistency in packet transmission timing, which may severely affect real-time applications. In contrast, the "Node 4 to Node 3 to Node 2" route has a relatively smaller jitter, with a tighter distribution, suggesting more stable transmission performance. The routes "Node 1 to Node 2 to Node 3" and "Node 3 to Node 2 to Node 1" display almost no variation, indicating consistent transmission with minimal jitter.

4.2.3 Three-Hop Link Evaluation

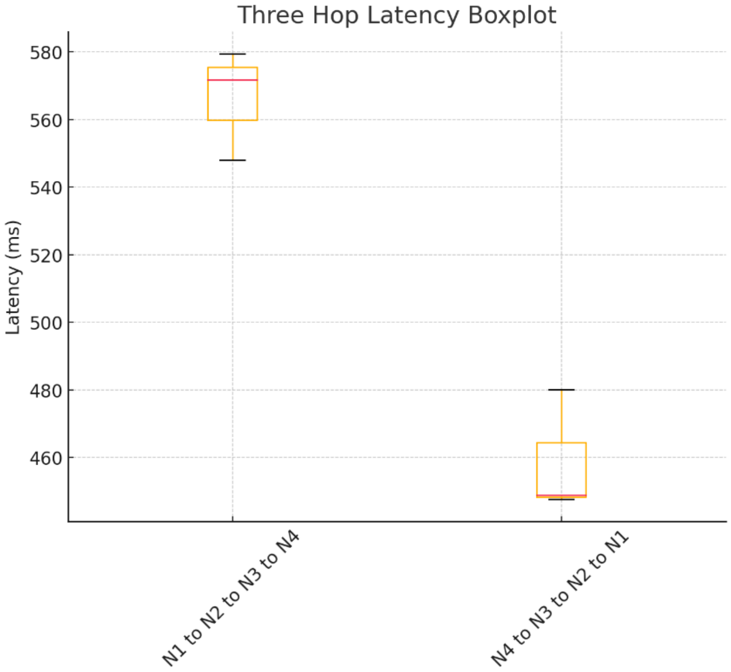


Figure 4.12: Latency Boxplot (Three Hop)

In Figure 4.12, the boxplot shows the latency distribution for two routes: "Node 1 to Node 2 to Node 3 to Node 4" and "Node 4 to Node 3 to Node 2 to Node 1." For the "Node 1 to Node 2 to Node 3 to Node 4" route, latency ranges between 548 ms and 579.4 ms, with a median of 571 ms, indicating a fairly consistent transmission delay across nodes. The narrow interquartile range (IQR) reflects little variability in the latency values, and the short whiskers show no extreme delays or outliers.

On the other hand, the "Node 4 to Node 3 to Node 2 to Node 1" route has a lower latency range, from 447.7 ms to 480.1 ms, with a median of 448.9 ms. This route demonstrates better latency performance with less delay. Both routes show consistent network behavior without significant outliers or fluctuations in latency, but "Node 4 to Node 3 to Node 2 to Node 1" generally offers quicker transmission times.

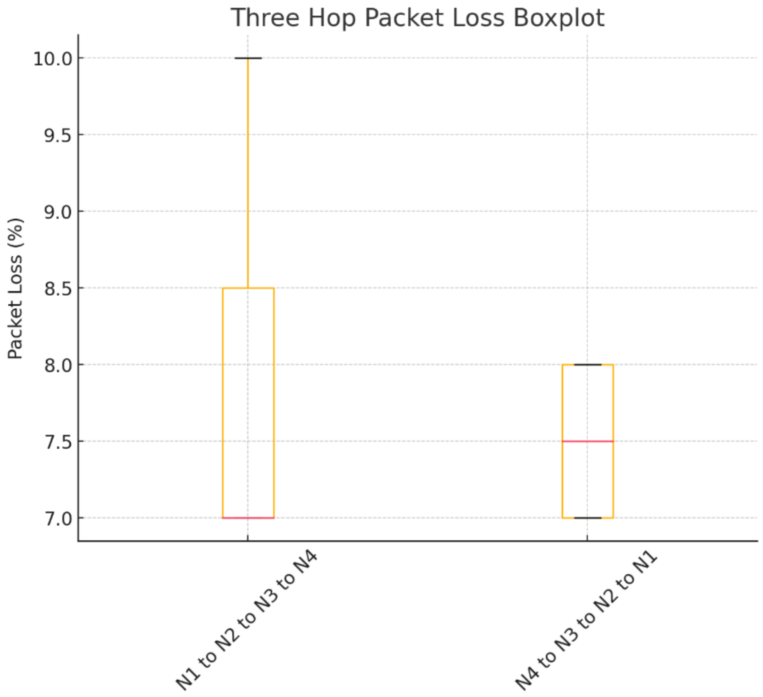


Figure 4.13: Packet Loss Percentage Boxplot (Three Hop)

As shown in Figure 4.13, the packet loss for "Node 1 to Node 2 to Node 3 to Node 4" shows more variability, ranging from 7% to 10%, with a median around 7%. This suggests that although the median is low, the route occasionally experiences higher packet loss. In contrast, the route "Node 4 to Node 3 to Node 2 to Node 1" has a more consistent packet loss performance, ranging from 7% to 8%, with a median closer to 7%. This reflects a more stable network performance with less fluctuation in packet loss for this route.

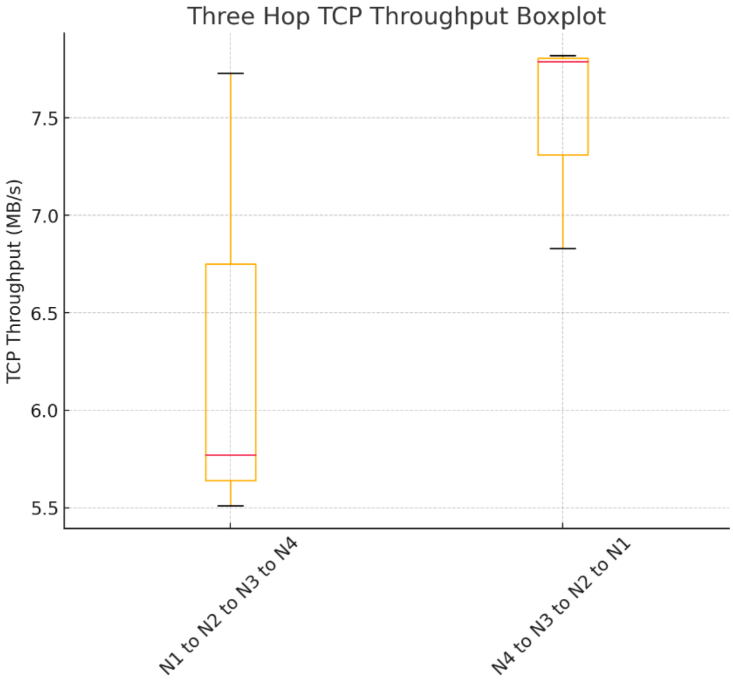


Figure 4.14: TCP Throughput Boxplot (Three Hop)

In Figure 4.14, the boxplot above shows the three hop TCP throughput distribution for the two routes, "Node 1 to Node 2 to Node 3 to Node 4" and "Node 4 to Node 3 to Node 2 to Node 1." For the "Node 1 to Node 2 to Node 3 to Node 4" route, the throughput values range from 5.5 Mb/s to 7.73 Mb/s. The median is around 6 Mb/s, with the throughput having a wider variability, as shown by the larger interquartile range (IQR). The lower throughput values indicate potential performance issues or bottlenecks in this route. In contrast, the "Node 4 to Node 3 to Node 2 to Node 1" route has higher and more consistent throughput, ranging from 6.83 Mb/s to 7.82 Mb/s, with a median closer to 7.8 Mb/s. This indicates a better and more stable TCP performance for this route compared to the first one.

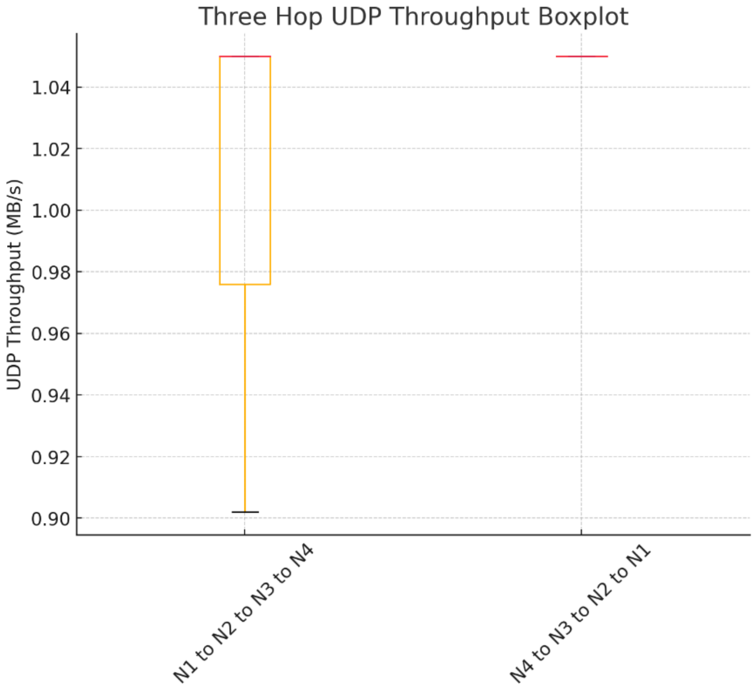


Figure 4.15: UDP Throughput Boxplot (Three Hop)

In Figure 4.15, the boxplot shows the three hop UDP throughput distribution for the two routes, "Node 1 to Node 2 to Node 3 to Node 4" and "Node 4 to Node 3 to Node 2 to Node 1." For the "Node 1 to Node 2 to Node 3 to Node 4" route, the throughput values range from 0.902 Mb/s to 1.05 Mb/s, with the median around 1.0 Mb/s. This indicates some variability in performance, with the lower throughput value suggesting a potential bottleneck or interference along this route. In contrast, the "Node 4 to Node 3 to Node 2 to Node 1" route has more consistent throughput, with all values clustering at 1.05 Mb/s. The lack of a box (since all values are identical) indicates no variability, showing that this route performs uniformly better in terms of UDP throughput.

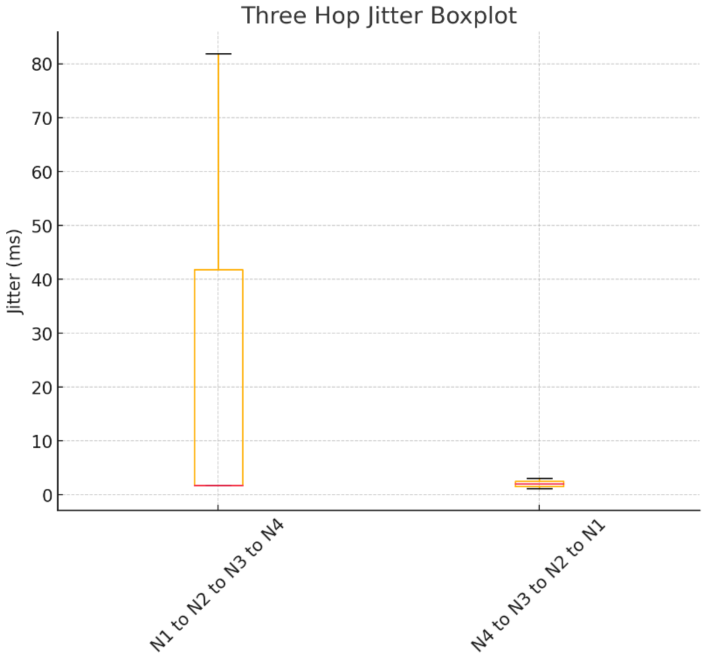


Figure 4.16: Jitter Boxplot (Three Hop)

As demonstrated in Figure 4.16, for the route "Node 1 to Node 2 to Node 3 to Node 4," the jitter values show a wide range, from 1.707 ms to 81.9 ms, indicating significant variability. This could cause issues with the stability and performance of real-time communications or data transfers across this route.

On the other hand, the "Node 4 to Node 3 to Node 2 to Node 1" route demonstrates much more consistent performance, with jitter values ranging from 1.188 ms to 3.1 ms, and the interquartile range (IQR) being much smaller. This indicates more stable and reliable network performance on this route, with fewer variations in jitter.

The comparison clearly shows that "Node 4 to Node 3 to Node 2 to Node 1" offers superior jitter performance with less fluctuation compared to the other route.

4.2.4 Hop Count Performance Summary

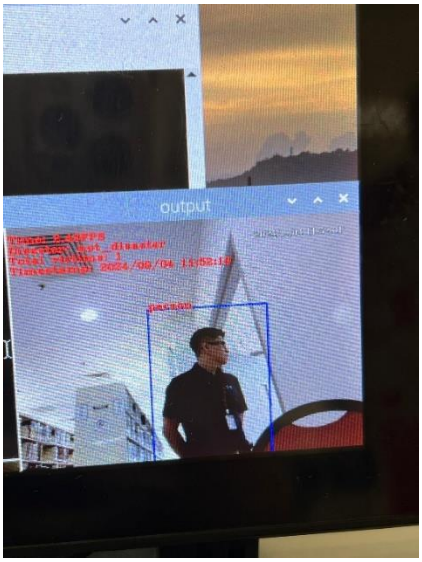
By comparing single hop, two hop, and three hop routes across key performance metrics such as latency, packet loss, TCP throughput, UDP throughput, and jitter. Firstly, single hop routes consistently exhibited the lowest latency, with values typically under 200 ms and minimal packet loss, making them the most reliable for fast and stable data transmission. These routes also maintained steady TCP and UDP throughput, with nearly no variability in performance.

As we move to two hop routes, the latency increased significantly, reaching around 500-700 ms. While packet loss remained manageable, it increased compared to single hop routes. Throughput also showed greater variability, indicating less stable data transmission, though still within acceptable limits for most applications.

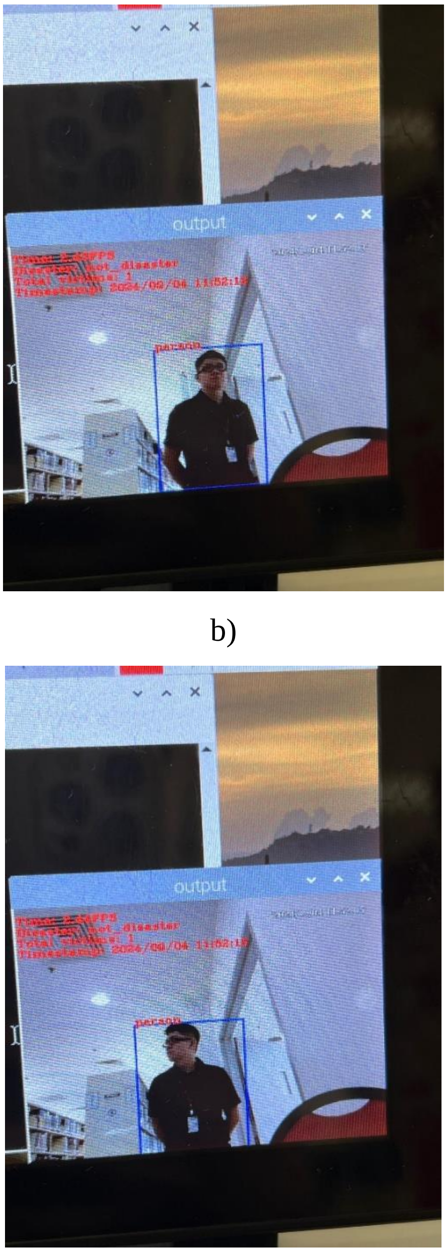
In contrast, three hop routes, such as "Node 1 to Node 2 to Node 3 to Node 4," demonstrated the highest latency, sometimes exceeding 1800 ms, along with more variable performance. Packet loss became more noticeable, and TCP throughput dropped compared to the simpler routes. Jitter, especially in the three hop routes, exhibited much higher variability, which could affect real-time communications. Overall, the addition of more hops introduced higher delays, greater packet loss, and less consistent throughput, showing that the complexity of routing through multiple nodes significantly impacts network performance.

4.3 Glass-to-Glass Latency

The glass-to-glass latency in the three RTSP receiver frames varies, reflecting differences in the time it takes for each frame to travel from the camera (GStreamer timestamp) to the display (receiver timestamp) from Node 1 to Node 3 (two hops).



a)



b)

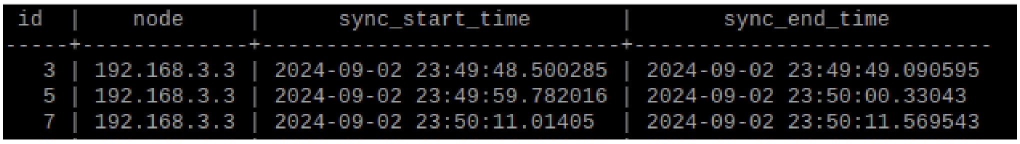
c)

Figure 4.17: RSTP Receiver Frame Node 1 (RTSP Server Node 3 Camera Source) (a) Test I (b) Test II (c) Test III.

In Frame 1, the GStreamer timestamp is 11:52:04, while the receiver displays it at 11:52:10, resulting in a latency of 6 seconds. For Frame 2, the GStreamer timestamp is 11:52:08, and the receiver timestamp is 11:52:13, leading to a latency of 5 seconds. Finally, Frame 3 has a GStreamer timestamp of 11:52:09 and a receiver timestamp of 11:52:15, with a latency of 6 seconds. These variations in latency across the frames suggest that the RTSP stream processing is influenced by several factors, such as network congestion, buffering, and encoding/decoding delays. The inconsistent latency results indicate the presence of jitter and also reduce in UDP Throughput which could impact the real-time performance of the video stream.

4.4 Database Synchronization Latency

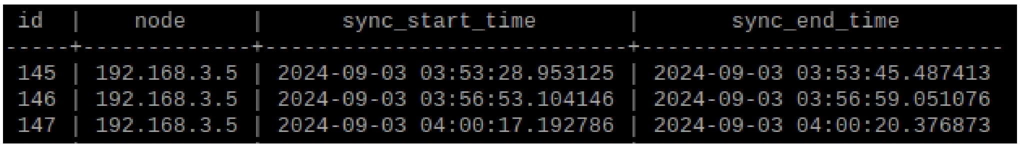
In the Mesh Network testbed, when Node 1 pushed updates to the other slave database (Node 2, Node 3, Node 4) for the text and image sychronization, a timestamp is recorded in the sync_logs relations as sync_start_time. After the Slave Database has acknowledged the updates, it will the timestamp as sync_end_time. The time difference between the two timestamp will be the database synchronization latency.



```
id  |     node      |       sync_start_time        |        sync_end_time
-----+-------------+----------------------------+----------------------------
   3 | 192.168.3.3 | 2024-09-02 23:49:48.500285 | 2024-09-02 23:49:49.090595
   5 | 192.168.3.3 | 2024-09-02 23:49:59.782016 | 2024-09-02 23:50:00.33043
   7 | 192.168.3.3 | 2024-09-02 23:50:11.01405  | 2024-09-02 23:50:11.569543
```

Figure 4.18: Sync Logs Node 1 to Node 2 (Single-Hop)

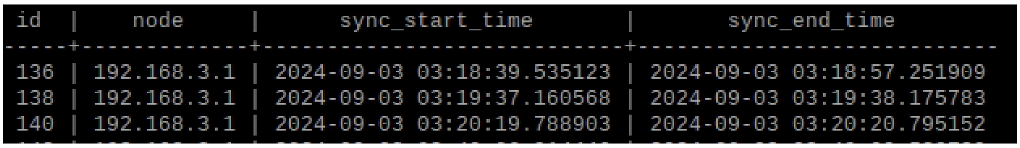
Figure 4.17 shows the sync logs of the synchronization process between Node 1 and Node 3, with three entries detailing the start and end times of each synchronization event. For the first synchronization (id = 3 image), the process began at 23:49:48.500285 and ended at 23:49:49.090595, resulting in a latency of approximately 0.59 second. The second synchronization (id = 5 text) started at 23:49:59.782016 and finished at 23:50:00.33043, with a latency of around 0.55 second. Lastly, the third synchronization (id = 7 image) began at 23:50:11.01405 and completed at 23:50:11.569543, resulting in a latency of 0.56 second. Overall, the synchronization latencies are consistent, each taking slightly over half a second to complete.



```
id  |     node      |       sync_start_time        |        sync_end_time
-----+-------------+----------------------------+----------------------------
 145 | 192.168.3.5 | 2024-09-03 03:53:28.953125 | 2024-09-03 03:53:45.487413
 146 | 192.168.3.5 | 2024-09-03 03:56:53.104146 | 2024-09-03 03:56:59.051076
 147 | 192.168.3.5 | 2024-09-03 04:00:17.192786 | 2024-09-03 04:00:20.376873
```

Figure 4.19: Sync Logs Node 1 to Node 3 (Two Hop)

As shown in Figure 4.18, the synchronization data from Node 1 to Node 2 reveals varying latencies across three recorded events. The first synchronization (id = 145 image) started at 03:53:28.953125 and ended at 03:53:45.487413, resulting in a relatively long latency of 16.53 seconds. The second synchronization (id = 146 text), which began at 03:56:53.104146 and finished at 03:56:59.051076, was significantly quicker, with a latency of 5.95 seconds. The third event (id = 147 image) started at 04:00:17.192786 and ended at 04:00:20.376873, showing the fastest synchronization time of 3.18 seconds.



```
id  |    node     |      sync_start_time       |       sync_end_time
-----+-------------+----------------------------+----------------------------
136 | 192.168.3.1 | 2024-09-03 03:18:39.535123 | 2024-09-03 03:18:57.251909
138 | 192.168.3.1 | 2024-09-03 03:19:37.160568 | 2024-09-03 03:19:38.175783
140 | 192.168.3.1 | 2024-09-03 03:20:19.788903 | 2024-09-03 03:20:20.795152
```

Figure 4.20: Sync Logs Node 1 to Node 4 (Three Hop)

As shown in Figure 4.18, the synchronization data between Node 1 and Node 4 reveals three synchronization events with varying latencies. The first synchronization (id = 136 image) started at 03:18:39.535123 and ended at 03:18:57.251909, resulting in a latency of 17.72 seconds. The second event (id = 138 text) began at 03:19:37.160568 and completed at 03:19:38.175783, showing a much shorter latency of 1.01 seconds. The third synchronization (id = 140 image) started at 03:20:19.788903 and finished at 03:20:20.795152, with the fastest latency of 1.01 seconds.

4.4.1 Hop Count Performance Summary

The image and text synchronization latency comparison between single hop, two hop, and three hop routes reveals distinct patterns in performance. For the single hop route (Node 1 to Node 2), the synchronization process is highly consistent for both text and image synchronization, with latencies slightly over half a second for each event which werer 0.59 seconds, 0.55 seconds, and 0.56 seconds indicating stable and efficient performance.

In contrast, the two hop route (Node 1 to Node 3) shows more variability in synchronization latencies. The first image synchronization event took 16.53 seconds, significantly longer than the single hop events, while the second and third events were quicker, with latencies of 5.95 seconds and 3.18

seconds, respectively. This suggests that two hop synchronizations may experience initial delays, but performance improves with subsequent attempts.

For the three hop route (Node 1 to Node 4), the first image synchronization event had a latency of 17.72 seconds, the longest across all routes. However, the second and third events completed much faster, both at 1.01 seconds, showing that three hop routes can also stabilize over time, similar to the two hop route. In summary, single hop routes offer the most consistent and fastest synchronization, while two and three hop routes exhibit more variability.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, this Final Year Project successfully designed and deployed a disaster response using a wireless ad-hoc mesh network based on the BATMAN-adv protocol. The prototype's performance falls within acceptable parameters for regional disaster monitoring and response, providing practical insights for similar implementations. This project serves as a valuable reference for designing and deploying disaster response platforms utilizing wireless ad-hoc mesh networks, contributing to the advancement of resilient communication systems and distributed applications for disaster response and recovery.

5.2 Recommendations for future work

For future work, there are several promising enhancements that can be implemented to further optimize the wireless ad-hoc mesh network for disaster response and monitoring. One key improvement would be transitioning the network to operate on the 5GHz frequency band. This shift would significantly reduce wireless interference from commonly used devices like Bluetooth and 2.4GHz wireless access points, which often crowd the lower frequencies. The 5GHz band offers more channels and less interference, leading to enhanced network performance and reliability, particularly in dense environments.

Additionally, the use of channel bonding could further increase network throughput by combining multiple channels to create higher bandwidth, making it ideal for data-heavy applications such as real-time video streaming or large-scale data transfer during disaster response scenarios. Another promising direction is to integrate multi-radio capabilities into the network. With each node equipped with multiple radios operating on different frequencies, congestion can be minimized, and overall network efficiency can be boosted. For instance, one radio could handle 5GHz node-to-node communication while another manages sensor data collection on the 2.4GHz band (Gimenez-Guzman et al., 2022).

Security is another area for advancement. By implementing enhanced security features such as WPA3, the network can be fortified against potential threats, ensuring secure and reliable communication even in critical disaster situations. Additionally, the adoption of dynamic bandwidth allocation and Quality of Service (QoS) mechanisms could prioritize data, such as emergency alerts, over less critical traffic, ensuring timely and efficient data transmission when it matters most. These improvements, combined, would significantly elevate the mesh network's performance, making it a highly resilient and capable platform for disaster response and monitoring.

REFERENCES

Akyildiz, I.F., Wang, X. and Wang, W. (2005) ‘Wireless Mesh Networks: A survey’, *Computer Networks*, 47(4), pp. 445–487. doi:10.1016/j.comnet.2004.12.001.

Chand, G.S., Lee, M. and Shin, S.Y. (2018b) ‘Drone based Wireless Mesh Network for disaster/military environment’, *Journal of Computer and Communications*, 06(04), pp. 44–52. doi:10.4236/jcc.2018.64004.

Feng, K., Ye, Y. and Xu, J. (2013) ‘A formal study on topology and floorplan characteristics of mesh and torus-based optical networks-on-chip’, *Microprocessors and Microsystems*, 37(8), pp. 941–952. doi:10.1016/j.micpro.2012.06.010.

Gimenez-Guzman, J.M. *et al.* (2022) ‘On the benefits of channel bonding in dense, decentralized Wi-Fi 4 Networks’, *Wireless Communications and Mobile Computing*, 2022, pp. 1–11. doi:10.1155/2022/8497585.

Gurumoorthi, E. *et al.* (2024) ‘Trust and secured routing in Mobile ad hoc network using block chain’, *Lecture Notes in Networks and Systems*, pp. 349–360. doi:10.1007/978-981-99-8451-0_30.

He, X., Cui, Y. and Jiang, Y. (2019) ‘An improved gossip algorithm based on semi-distributed Blockchain Network’, *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* [Preprint]. doi:10.1109/cyberc.2019.00014.

Hussein, Z., Salama, M.A. and El-Rahman, S.A. (2023) *Evolution of blockchain consensus algorithms: A review on the latest milestones of Blockchain Consensus Algorithms - Cybersecurity*, *SpringerOpen*. Available at: https://cybersecurity.springeropen.com/articles/10.1186/s42400-023-00163-y (Accessed: 14 September 2024).

Jiang, R. (2015) ‘A review of Network topology’, *Proceedings of the 2015 4th International Conference on Computer, Mechatronics, Control and Electronic Engineering* [Preprint]. doi:10.2991/iccmcee-15.2015.222.

Kukhta, N.A. *et al.* (2014) ‘Effect of linking topology on the properties of star-shaped derivatives of triazine and fluorene’, *Synthetic Metals*, 195, pp. 266–275. doi:10.1016/j.synthmet.2014.06.019.

Larasati, F., Suharto, N. and Hariyadi, A. (2023) ‘Implementation of mobile ad-HOC network using Batman Routing Protocol (better approach to mobile ad-hoc network) for salinity monitoring in Vaname shrimp farms’, *Jurnal Jartel Jurnal Jaringan Telekomunikasi*, 13(2), pp. 201–208. doi:10.33795/jartel.v13i2.558.

Liu, L. *et al.* (2018) ‘Performance evaluation of batman-ADV wireless mesh network routing algorithms’, *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* [Preprint]. doi:10.1109/cscloud/edgecom.2018.00030.

Luis Soares, M.Sc. (2023) *Implementing a distributed hash table (DHT) in Golang*, *LinkedIn*. Available at: https://www.linkedin.com/pulse/implementing-distributed-hash-table-dht-ingolang-luis-soares-m-sc- (Accessed: 14 September 2024).

Malay Mail (2022) *Comms Ministry to identify rural areas, islands still lacking communication network, says deputy minister*, *Malay Mail*. Available at: https://www.malaymail.com/news/malaysia/2022/12/23/comms-ministry-to-identify-rural-areas-islands-still-lacking-communication-network-says-deputy-minister/46834#google_vignette (Accessed: 13 April 2024).

Neumann, A. *et al.* (no date) *Better approach to mobile ad-hoc networking (B.A.T.M.A.N.), IETF Datatracker*. Available at: https://datatracker.ietf.org/doc/html/draft-wunderlich-openmesh-manet-routing-00 (Accessed: 27 April 2024).

Tham, M.-L. *et al.* (2023) ‘Performance study of disaster-resilient mesh networking using NerveNet Wi-Fi and Lora’, *2023 6th Conference on Cloud and Internet of Things (CIoT)*, pp. 219–224. doi:10.1109/ciot57267.2023.10084890.

The Edge Malaysia (2023) *Communications ministry looking into solutions to Telcos Service Quality Issues* (2023) *The Edge Malaysia*. Available at: https://theedgemalaysia.com/node/661403 (Accessed: 10 April 2024).

Perkins, C.E. and Royer, E.M. (1999) ‘Ad-hoc on-demand distance vector routing’, *Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications* [Preprint]. doi:10.1109/mcsa.1999.749281.

Ramapriya, M.R. *et al.* (1999) ‘An end-connected dual bus topology for Metropolitan Area Networks’, *Computer Networks*, 31(5), pp. 445–456. doi:10.1016/s0169-7552(98)00277-3.

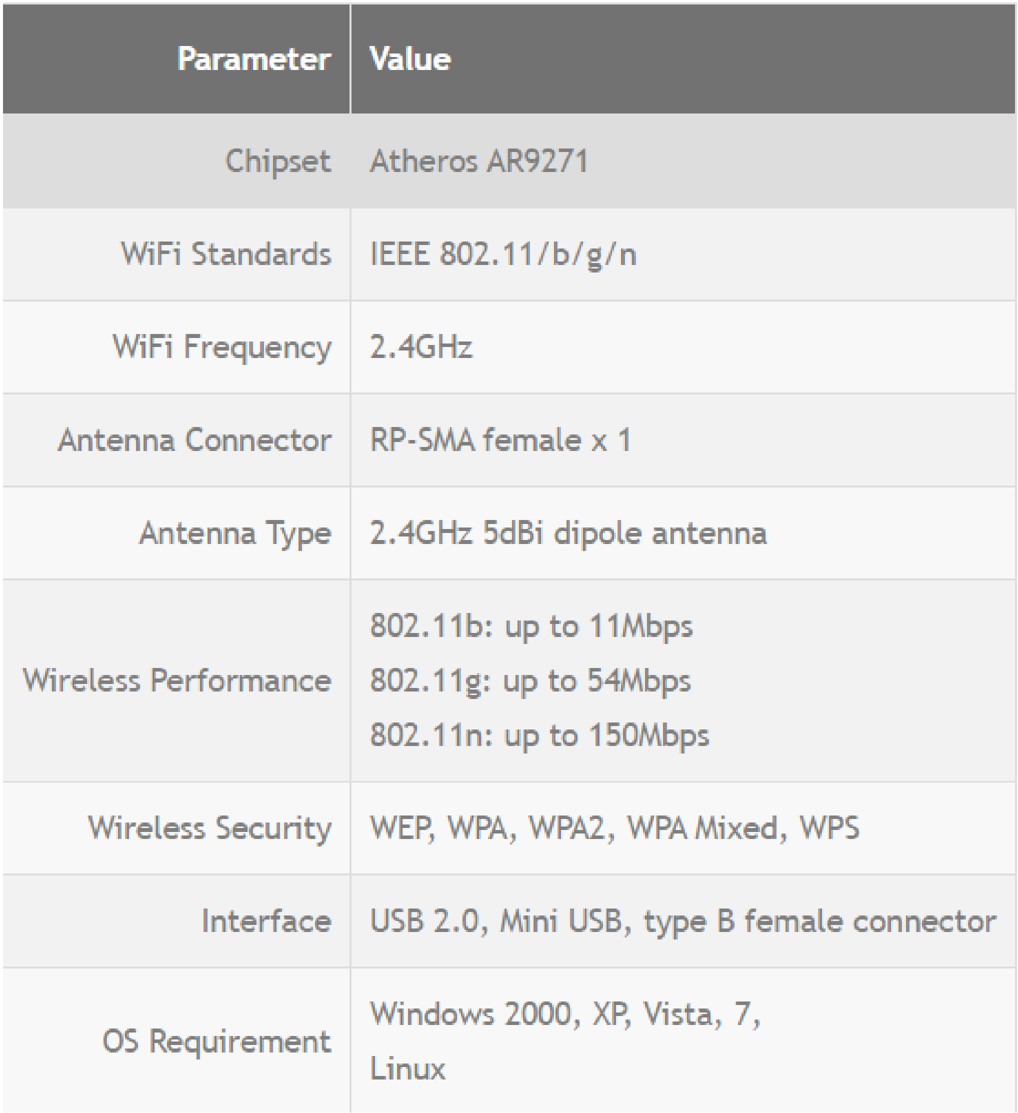
Robinson, J. and Knightly, E.W. (2007) ‘A performance study of deployment factors in wireless mesh networks’, *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications* [Preprint]. doi:10.1109/infcom.2007.238.

Ruiz, P. *et al.* (2012) ‘Information dissemination in VANETs based upon a tree topology’, *Ad Hoc Networks*, 10(1), pp. 111–127. doi:10.1016/j.adhoc.2011.06.005.

Tao Qian, Mueller, F. and Yufeng Xin (2014) ‘A real-time distributed hash table’, *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications* [Preprint].

APPENDICES

Appendix A: ALFA AWUS036NHA Wireless Adapter Specifications

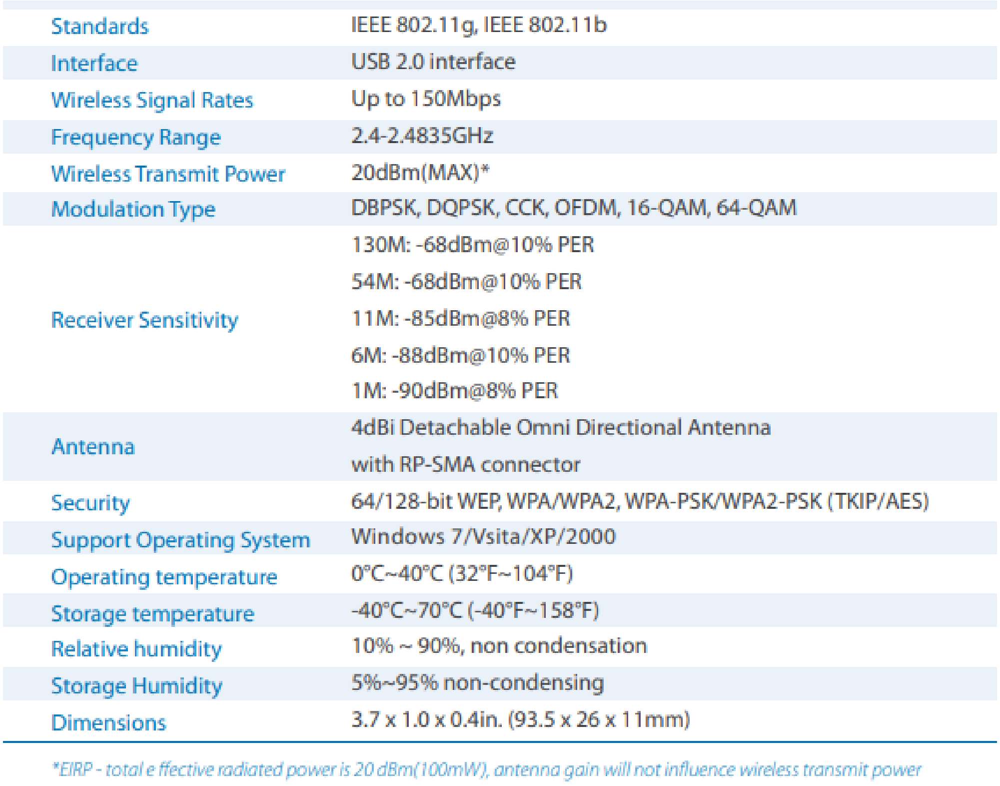


Parameter	Value
Chipset	Atheros AR9271
WiFi Standards	IEEE 802.11/b/g/n
WiFi Frequency	2.4GHz
Antenna Connector	RP-SMA female x 1
Antenna Type	2.4GHz 5dBi dipole antenna
Wireless Performance	802.11b: up to 11Mbps 802.11g: up to 54Mbps 802.11n: up to 150Mbps
Wireless Security	WEP, WPA, WPA2, WPA Mixed, WPS
Interface	USB 2.0, Mini USB, type B female connector
OS Requirement	Windows 2000, XP, Vista, 7, Linux

Appendix B: ALFA AWUS036ACH Wireless Adapter Specifications

SKU	AWUS036ACH-C
EAN	4718050307302
Chipset	RTL8812AU
Wi-Fi Standards	802.11b, 802.11g, 802.11n, 802.11a, 802.11ac
Wi-Fi Frequency	2.4 Ghz, 5 Ghz
2.4GHz Radio Rate	300Mbps
5Ghz Radio Rate	867Mbps
Wireless Security	WEP, WPA, WPA2
Antenna Type	Omni-Directional
Antenna Connector	RP-SMA Female
Antenna Gain	5dbi x 2
OS Requirement	Windows Vista/7/8/8.1/10. macOS 10.5-10.13, Linux, Kali Linux
Environment	Indoor
USB Ports	USB 3.0

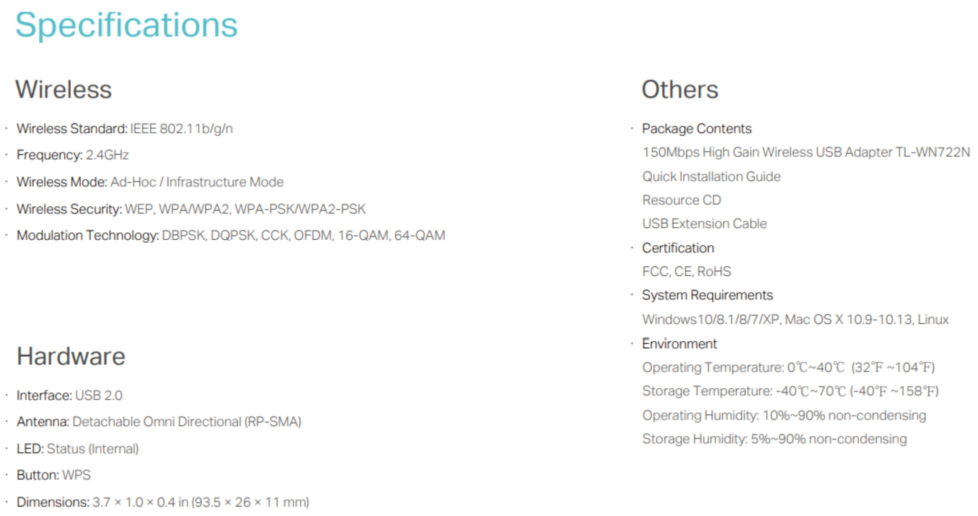
Appendix C: TP-Link TL722WN V1 Wireless Adapter Specifications



Standards	IEEE 802.11g, IEEE 802.11b
Interface	USB 2.0 interface
Wireless Signal Rates	Up to 150Mbps
Frequency Range	2.4-2.4835GHz
Wireless Transmit Power	20dBm(MAX)*
Modulation Type	DBPSK, DQPSK, CCK, OFDM, 16-QAM, 64-QAM
Receiver Sensitivity	130M: -68dBm@10% PER 54M: -68dBm@10% PER 11M: -85dBm@8% PER 6M: -88dBm@10% PER 1M: -90dBm@8% PER
Antenna	4dBi Detachable Omni Directional Antenna with RP-SMA connector
Security	64/128-bit WEP, WPA/WPA2, WPA-PSK/WPA2-PSK (TKIP/AES)
Support Operating System	Windows 7/Vsita/XP/2000
Operating temperature	0℃~40℃ (32℉~104℉)
Storage temperature	-40℃~70℃ (-40℉~158℉)
Relative humidity	10% ~ 90%, non condensation
Storage Humidity	5%~95% non-condensing
Dimensions	3.7 x 1.0 x 0.4in. (93.5 x 26 x 11mm)

*EIRP - total effective radiated power is 20 dBm(100mW), antenna gain will not influence wireless transmit power

Appendix D: TP-Link TL722WN V3 Wireless Adapter Specifications



Specifications

Wireless

- **Wireless Standard:** IEEE 802.11b/g/n
- **Frequency:** 2.4GHz
- **Wireless Mode:** Ad-Hoc / Infrastructure Mode
- **Wireless Security:** WEP, WPA/WPA2, WPA-PSK/WPA2-PSK
- **Modulation Technology:** DBPSK, DQPSK, CCK, OFDM, 16-QAM, 64-QAM

Hardware

- **Interface:** USB 2.0
- **Antenna:** Detachable Omni Directional (RP-SMA)
- **LED:** Status (Internal)
- **Button:** WPS
- **Dimensions:** 3.7 × 1.0 × 0.4 in (93.5 × 26 × 11 mm)

Others

- **Package Contents**
  150Mbps High Gain Wireless USB Adapter TL-WN722N
  Quick Installation Guide
  Resource CD
  USB Extension Cable
- **Certification**
  FCC, CE, RoHS
- **System Requirements**
  Windows10/8.1/8/7/XP, Mac OS X 10.9-10.13, Linux
- **Environment**
  Operating Temperature: 0℃~40℃ (32℉ ~104℉)
  Storage Temperature: -40℃~70℃ (-40℉ ~158℉)
  Operating Humidity: 10%~90% non-condensing
  Storage Humidity: 5%~90% non-condensing