

**ANALYSIS ON THE CAUSE OF PACKET LOSS IN WIRELESS  
NETWORKS**

**BY  
TERRY TEH**

**A REPORT  
SUBMITTED TO**

**Universiti Tunku Abdul Rahman  
in partial fulfilment of the requirements  
for the degree of**

**BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)  
COMMUNICATIONS AND NETWORKING**

**Faculty of Information and Communication Technology  
(Kampar Campus)**

**JAN 2022**

## REPORT STATUS DECLARATION FORM

**Title:** ANALYSIS ON THE CAUSE OF PACKET LOSS IN  
WIRELESS NETWORKS

**Academic Session:** 2022/01

**I** TERRY TEH  
**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



(Author's signature)

Verified by,



(Supervisor's signature)

**Address:**

8M-03-08 Tanjung Court,  
Lebuhraya Thean Teik,  
11500 Pulau Pinang

Vasaki Ponnusamy

Supervisor's name

**Date:** 21.04.2022

**Date:** 20 April 2022

<b>Universiti Tunku Abdul Rahman</b>			
Form Title : <b>Sample of Submission Sheet for FYP/Dissertation/Thesis</b>			
Form Number: <b>FM-IAD-004</b>	Rev No.: <b>0</b>	Effective Date: <b>21 JUNE 2011</b>	Page No.: <b>1 of 1</b>

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 21.04.2022

**SUBMISSION OF FINAL YEAR PROJECT**

It is hereby certified that **TERRY TEH** (ID No: **19ACB02136** ) has completed this final year project entitled “**ANALYSIS ON THE CAUSE OF PACKET LOSS IN WIRELESS NETWORKS**” under the supervision of **TS DR VASAKI A/P PONNUSAMY** (Supervisor) from the Department of **COMPUTER AND COMMUNICATION TECHNOLOGY**, Faculty of **INFORMATION AND COMMUNICATION TECHNOLOGY**.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,





---

(TERRY TEH)

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**ANALYSIS ON THE CAUSE OF PACKET LOSS IN WIRELESS NETWORKS**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  \_\_\_\_\_

Name : TERRY TEH

Date : 21.04.2022

## **ACKNOWLEDGEMENTS**

I would like to express thanks and appreciation to my supervisor, Ts Dr Vasaki a/p Ponnusamy and my moderator, Dr Robithoh Annur who have given me a golden opportunity to involve on the Wireless Security field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

## **ABSTRACT**

WSN (Wireless Sensor Network) is a new technology that has a lot of potential for both civilian and military uses in the future. With the wide adoption of WSN, ensuring data accuracy is a must for decision making in the sensing area of the system. However, the combination of sensor technologies, computational power, and wireless connectivity makes it lucrative to be exploited in large quantities. Every packet contains the information sensed by the sensor node in the sensing system and deliver to the Base Station (BS) as a statistic for the end-users to further analyse. In WSN, intermediate-quality links frequently generate vulnerable connectivity, but packet losses induced by such volatile links are difficult to track. A packet loss implies that the information sent to the end-users will result in an inaccurate state. Thus, it might affect the end-users decision making on the sensing system. In previous studies, packet loss rate in WSN is a very common issue that every expert tried to minimize. Notwithstanding, very few studies or none of them are discussing identifying the actual reason for the packet loss. Without identifying the actual reason for the packet loss, the effectiveness of packet loss responses can be harmed, one might continue suffering from the problem of packet loss because of being unable to troubleshoot it. This project intended to propose a lightweight analysing scheme to identify the reason packet loss is due to network issues or malicious discard. The analysing scheme may also be used to enhance the current Intrusion Detection System (IDS) as nowadays most of them are only capable of detecting packet loss. Lastly, having an analysing scheme to identify the reason for packet loss can improve the effectiveness of packet loss responses, thus one will not be continuing to receive inaccurate data from the sensing nodes. This project will verify the proposed analysing scheme by using a network simulation through the OMNeT++ network simulator and INET framework. A selective forwarding attack will be used to emulate the malicious discarding of the packet and a radio interferer will generate radio interference to emulate the network issues in the network model.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>REPORT STATUS DECLARATION FORM</b>	<b>ii</b>
<b>FYP THESIS SUBMISSION FORM</b>	<b>iii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>TABLE OF CONTENTS</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement and Motivation	1
1.2 Project Scope	3
1.3 Project Objectives	5
1.4 Contributions	6
1.5 Background Information	7
1.5.1 Introduction to Wireless Sensor Networks	7
1.5.2 Wi-Fi Standards	8
1.5.3 Definition of Packet Loss	10
1.6 Previous Work	11
1.7 Report Organization	12
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>13</b>
2.1 Previous work on Intrusion Detection System (IDS) in Wireless Sensor Networks (WSN)	13
2.2 Previous work on detection of selective forwarding attack	15
2.2.1 Lightweight multi-hop acknowledgement-based detection scheme	15
2.2.2 Multi-dataflow topologies (MDT) scheme	17

2.3	Previous work on packet loss analysis on Wireless Sensor Networks (WSN)	19
2.3.1	Impact of packet loss consideration on LEACH routing protocol	19
2.3.2	Performance analysis on AODV routing protocol	21
<b>CHAPTER 3 SYSTEM MODEL</b>		<b>25</b>
3.1	Network Design/Overview	25
3.2	Methodologies	26
3.3	Implementation Issues and Challenges	28
3.4	Gantt Chart	29
<b>CHAPTER 4 EXPERIMENT/SIMULATION</b>		<b>30</b>
4.1	Software Setup	30
4.1.1	OMNeT++ Installation	30
4.1.2	INET Framework Installation	35
4.2	Setting and Configuration	42
4.2.1	Network Description File (NED)	42
4.2.2	INI File	45
4.3	System Operation	49
4.4	Concluding Remark	55
<b>CHAPTER 5 NETWORK EVALUATION AND DISCUSSION</b>		<b>56</b>
5.1	Network Model Observation	56
5.1.1	Observation without Network Interference	56
5.1.2	Observation with Network Interference	61
5.2	Network Model Discussion	63
5.2.1	Discussion without Network Interference	64
5.2.2	Discussion with Network Interference	66
5.3	Project Challenges	67
5.4	Objectives Evaluation	68



<b>CHAPTER 6 CONCLUSION AND RECOMMENDATION</b>	<b>69</b>
6.1 Conclusion	69
6.2 Recommendation	71
<b>REFERENCES</b>	<b>72</b>
<b>APPENDIX</b>	<b>A-1</b>
Appendix A – Network Description File (NED)	A-1
Appendix B – INI File	B-1
<b>FINAL YEAR PROJECT WEEKLY REPORT</b>	
<b>POSTER</b>	
<b>PLAGIARISM CHECK RESULT</b>	
<b>FYP 2 CHECKLIST</b>	

# LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 1-2-1	Simulation model	4
Figure 1-5-1	WSN architecture	8
Figure 2-1-1	Number of spontaneous watchdogs	15
Figure 2-2-1	Deployment of malicious nodes	15
Figure 2-2-2	Example of multi-hop acknowledgement scheme	16
Figure 2-2-3	Alarm reliability with different channel error rate	17
Figure 2-2-4	Example of multi-dataflow topologies scheme	18
Figure 2-3-1	Average and maximum packet loss of each node	20
Figure 2-3-2	PDR as a function of reporting rate	23
Figure 2-3-3	Throughput as a function of reporting rate	23
Figure 2-3-4	Delay as a function of reporting rate	24
Figure 2-3-5	Routing overheads as a function of reporting rate	24
Figure 3-2-1	Methodology explanation figure	27
Figure 3-4-1	Gantt chart	29
Figure 4-1-1	Software Setup Guide 1	30
Figure 4-1-2	Software Setup Guide 2	31
Figure 4-1-3	Software Setup Guide 3	32
Figure 4-1-4	Software Setup Guide 4	32
Figure 4-1-5	Software Setup Guide 5	32
Figure 4-1-6	Software Setup Guide 6	33
Figure 4-1-7	Software Setup Guide 7	33
Figure 4-1-8	Software Setup Guide 8	34
Figure 4-1-9	Software Setup Guide 9	34
Figure 4-1-10	Software Setup Guide 10	34
Figure 4-1-11	Software Setup Guide 11	35
Figure 4-1-12	Software Setup Guide 12	35
Figure 4-1-13	Software Setup Guide 13	36
Figure 4-1-14	Software Setup Guide 14	36
Figure 4-1-15	Software Setup Guide 15	37

Figure 4-1-16	Software Setup Guide 16	37
Figure 4-1-17	Software Setup Guide 17	38
Figure 4-1-18	Software Setup Guide 18	38
Figure 4-1-19	Software Setup Guide 19	39
Figure 4-1-20	Software Setup Guide 20	39
Figure 4-1-21	Software Setup Guide 21	40
Figure 4-1-22	Software Setup Guide 22	40
Figure 4-1-23	Software Setup Guide 23	41
Figure 4-1-24	Software Setup Guide 24	41
Figure 4-3-1	System Operation Guide 1	49
Figure 4-3-2	System Operation Guide 2	50
Figure 4-3-3	System Operation Guide 3	50
Figure 4-3-4	System Operation Guide 4	51
Figure 4-3-5	System Operation Guide 5	52
Figure 4-3-6	System Operation Guide 6	53
Figure 4-3-7	System Operation Guide 7	54
Figure 4-3-8	System Operation Guide 8	55
Figure 5-1-1	Network Observation 1	56
Figure 5-1-2	Network Observation 2	57
Figure 5-1-3	Network Observation 3	58
Figure 5-1-4	Network Observation 4	59
Figure 5-1-5	Network Observation 5	60
Figure 5-1-6	Network Observation 6	61
Figure 5-1-7	Network Observation 7	62
Figure 5-1-8	Network Observation 8	63
Figure 5-2-1	Network Discussion 1	64
Figure 5-2-2	Network Discussion 2	65
Figure 5-2-3	Network Discussion 3	65
Figure 5-2-4	Network Discussion 4	66
Figure 5-2-5	Network Discussion 5	66
Figure 5-2-6	Network Discussion 6	67

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2-3-1	Detected packet loss variation with distance	21
Table 2-3-2	Network model parameter and details	22
Table 3-1-1	Simulation parameter and details	25

## LIST OF ABBREVIATIONS

<i>WLAN</i>	Wireless Local Area Networks
<i>WWAN</i>	Wireless Wide Area Networks
<i>WSN</i>	Wireless Sensor Networks
<i>RF</i>	Radiofrequency
<i>IDS</i>	Intrusion Detection System
<i>BS</i>	Base Station
<i>IoT</i>	Internet of Things
<i>IRS</i>	Incident Response Systems
<i>MIMO</i>	Multiple-Input Multiple-Output
<i>OFDM</i>	Orthogonal Frequency Division Multiplex
<i>OFDMA</i>	Orthogonal Frequency-Division Multiple Access
<i>MTU</i>	Maximum Transport Unit
<i>TCP</i>	Transmission Control Protocol
<i>UDP</i>	User Datagram Protocol
<i>DOS</i>	Denial of Service
<i>AODV</i>	Ad-hoc On-demand Distance Vector
<i>MDT</i>	Multi-dataflow Topologies
<i>LEACH</i>	Low-Energy Adaptive Clustering Hierarchy
<i>QoS</i>	Quality of Service
<i>CBR</i>	Constant Bit Rate
<i>PDR</i>	Packet Delivery Ratio
<i>NED</i>	Network Description File
<i>INI</i>	Network Initiator File

# CHAPTER 1

## Introduction

This chapter presents the problem statement and motivation for this project, other than that, also discusses the project scope, project objectives of this project. Furthermore, following by discussing the contribution of this project and some background information. This chapter will also present the previous work and a summary of the proposed approach. Lastly, the end of the chapter will highlight the organization of this report.

### 1.1 Problem Statement and Motivation

Nowadays, wireless networks or wireless devices had become commonplace in today's computing environment. Wireless technologies had found their application in different types of wireless networks, from local area networks to wide area networks, etc, such as Wireless Local Area Network (WLAN) and Wireless Wide Area Networks (WWAN). One of the most popular networks that establishing itself as a significant new layer in the IT ecosystem, as well as a burgeoning field of active research including hardware and system design, networking, distributed algorithms, programming models, data management, security, and social issues is Wireless Sensor Network (WSN). In addition, temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of specific objects or substances, mechanical stress levels on attached objects, and other attributes can all be monitored via WSN. According to [1], WSN is a new technology that has a lot of potential for both civilian and military uses in the future. Wireless technologies are established through the assistance of radio waves. Sometimes radio waves are also had been called radiofrequency (RF) signals. A radio wave is an electromagnetic signal that is used to transmit data across great distances over the air. Although radio waves had helped people in the mean of mobility without the restriction of wired, it contains a major disadvantage – security issues due to radio interference. While routing techniques and wireless sensor network modelling are receiving a lot of attention, security issues have yet to garner significant attention. That draws forth the main theme of this project – packet loss in wireless sensor networks.

In wireless sensor networks, intermediate-quality links frequently generate vulnerable connectivity, however, packet loss due to these vapour scent links is difficult to trace. The defective channel error rates, which are induced by those unpredictable noises and interference, can tamper down the robustness of an energy-constrained wireless sensor network. The size of sensors, and thus the processing power, memory, and type of duties required from the sensors pose the most significant obstacle for implementing any effective security mechanism in wireless sensor networks. Moreover, the unattended nature of wireless technologies such as radio frequency interference, weaker signals, distance or path loss, and multipath makes the wireless sensor networks susceptible to attacks compared to wired networks. In WSN, attacks on the nodes or the network issues both can cause packet losses. The effectiveness of packet loss responses can be harmed if the actual reason is not identified. Nowadays, most intrusion detection systems (IDS) are often only capable of sensing there is packet loss occurring however are incapable of pinpointing the reason for the losses, such as malicious discard or network issues packet losses. It is not enough for an IDS to only detect there is a packet loss happening in the current moment, it is also critical to get the right diagnosis regarding what is causing the losses because packet losses could be caused by malicious discard or network issues as mentioned. According to [2], this knowledge is essential for sensor systems to survive, both in terms of responding to attacks and in terms of recovery and debugging. For example, if a sensor node on the sensor network suffers from the problem of packet loss, one could not easily diagnose the causes of losses that will cause the whole sensing system's failure due to inaccurate data. Recent research by [3] states that while data are on their route to a base station that performs decision-making, data is generated at a large number of sensor node sources and processed in-network at intermediate hops. They further stated that the variety of data sources necessitates ensuring the data's reliability so that only reliable information is evaluated during the decision-making process. Furthermore, according to [4], sensor networks must become self-contained, with real-time reactivity and flexibility to evolving changes, without the intervention of a user or administrator. Therefore, this research project would like to enhance the current IDS by providing a lightweight analysing scheme for wireless sensor networks to differentiate the actual causes of the packet losses to have a quick incident response to packet losses intrusion.

## 1.2 Project Scope

This section further describes the scope of this project including the hardware and software that will be used to simulate the performance study, some assumptions, the network model and finally, what will be delivered at the end of this project.

First of all, start with discussing the hardware. The hardware that will be used is the ASUS X560U laptop, loaded with an Intel Core i7 8550U processor, 8GB of RAM, and NVIDIA GTX1050 graphic cards. Next, move on to the software that will be used. The software that will be used to complete this project is one of the most popular network simulator tools – ‘OMNeT++’. OMNeT++ is a C++ simulation toolkit and framework that is flexible, modular, and component-based and is mostly used to create network simulators [5]. It contains a lot of frameworks that can be used to build up the environment without the users manually adding in all the modules from scratch. Following the framework, this project will be using the most popular framework in OMNeT++, which is the INET Framework. For academics and students working with communication networks, it includes all protocols, agents, and other models. Furthermore, it does contain models for the Internet stack such as TCP, UDP, IPv4, IPv6, etc, wired and wireless link layer protocols such as Ethernet, IEEE 802.11, etc. Therefore, the combination of OMNeT++ and INET Framework is the most suitable software to be used in this project.

Secondly, will be discussing the network model that will be used for the simulation. The network model will consist of a Base Station (BS), a source node, and several intermediate nodes that act as forwarding nodes. This project does not consider which routing algorithm will be used as the routing protocol. Any routing protocol that utilizes the minimum hop-count from the source node to BS is applicable for this network model. Since this project objective is just to differentiate packet loss is due to malicious discard or network issues, therefore this project will purposely emulate the malicious discard and radio channel error. As for the packet loss to emulate the malicious discarding, this project will utilize the selective forwarding attacks, where one of the intermediate nodes that had been compromised will be forwarded only a few packets to the base station but not every packet. For the packet loss to emulate the



network issues, this project will generate a radio interference to cause channel error at one of the links. The figure below shows the network model:

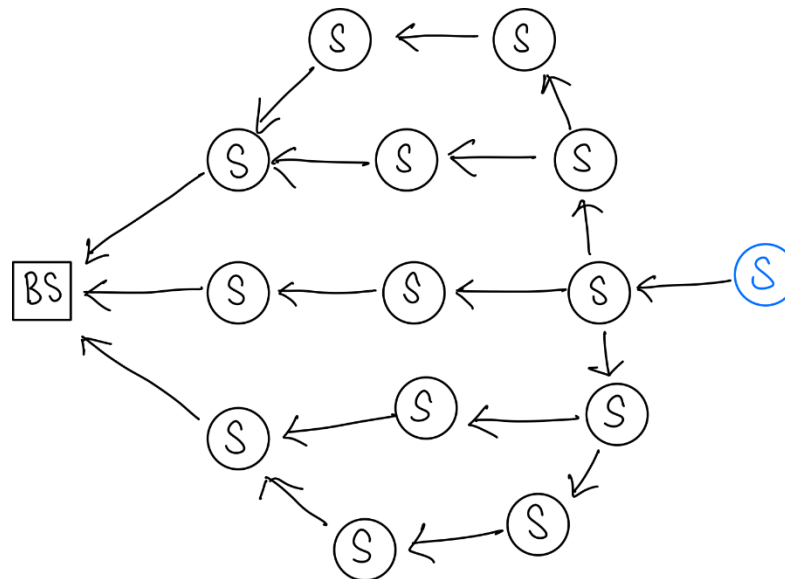


Figure 1-2-1

Next, will be discussing some of the assumptions of the network model used for simulation. The first assumption is that the network model build will not be considered to be used in which application of wireless sensor networks. As mentioned earlier, various areas had been adopted the application of wireless sensor networks, such as military applications, area monitoring, transportation, health applications, environmental sensing, etc. Second, this project assumes that the sensor nodes in the network will not be constrained by computational capability and energy power. According to empirical research, such as [6], they stated that sensor networks are generally defined by a low power supply, low bandwidth, tiny memory sizes, and low energy consumption. Third, this project assumes that packet loss will only be detected on Base Station (BS) by the IDS. All of the sensor nodes in the network will only be responsible for forwarding the data packet to BS from the source node. Fourth, the analysis of the cause of packet loss will only be done on BS, therefore the computational power, limited memory, and limited storage will not be a consideration for BS. BS will be a high scale node compared to other sensor nodes. Lastly, this project assumes that all the sensor nodes will be stationary whereas immobile after initial deployment.

Finally, will be discussing the final deliverables of this project. This project will be developing a Wireless Sensor Network simulation model alongside simulation results. The simulation results will consist of how accurate the base station will be able to differentiate the reason for packet loss induced by malicious discard or network issues. In addition, this project also seeks to enhance the IDS or algorithm with a method that will be able to differentiate the cause of packet loss. The algorithm will be used in the base station to analyse the reason for packet loss when it detected packet loss while receiving packets from all the sensor nodes.

### **1.3 Project Objectives**

This section will be discussing the project objectives to solve the problem statement listed in section 1.1. As mentioned in section 1.1, in WSN, attacks on the sensor nodes and network interference both will cause packet losses. If the real assault is not detected, the efficacy of attack responses may be hampered. Most intrusion detection systems (IDS) nowadays are only capable of sensing there is a packet loss occurring however are incapable of determining the reason for the losses, whether they are caused by humans or by network failures. It is not enough for an IDS to just detect that a packet loss is occurring right now; it is also important to obtain the correct diagnostic as to what is causing the losses, as packet losses might be caused by malicious activity or network breakdown. Therefore, this project aims to propose a solution to identify or differentiate the actual reason for packet losses whether it is caused by network interference or malicious discarding of the source nodes.

Be able to identify the actual reason for packet losses whether it is caused by network interference or malicious discarding on the source nodes, one will not need to worry about the data being unable to deliver to the end-users or the data received will be inaccurate due to partial packet losses. In WSN, data are crucial information for the end-users to perform some action on the system. End users will have to analyse the data that is captured by the sensor nodes to make some important decisions on the system. If the data are inaccurate or the data are unable to deliver to the end-users, it is hard to make some important decisions on the system. Therefore, it is not sufficient for the IDS to just detect that is a packet loss, because one cannot immediately identify the actual reason for the packet loss and fix it. So, this project proposes a method that can

immediately identify the reason for the packet loss, which will affect the time of troubleshooting the issue of packet loss is reduced.

In short, the objective of this project will propose a method that can identify the reason for packet losses induced by malicious discards or network issues. This project also aims to enhance the current IDS as it is only able to detect packet losses, however, unable to analyse the reason for packet losses. In addition, this project will not be covering the actual source nodes that are causing the problem, it will just focus on analysing the cause of packet losses.

#### **1.4 Contributions**

As mentioned in section 1.2, this project will deliver a simulation result that shows how accurate the base station will be able to differentiate the actual cause of packet losses in WSN. In addition, packet loss is one of the most popular topics in the wireless security area. Many of the researchers had done a vast number of studies on this topic. This research project will be able to benefit future researchers to continue work on this topic based on the foundation that this project had built. They do not have to start from scratch as it is time-consuming and needs a lot of setups. Future research can based on the result or parameters of this project as a reference, produce a much more efficient and intelligent method to further enhance the algorithm to be quicker to identify the cause of packet losses.

As mentioned earlier, WSN had been applied in many areas, from environmental monitoring and management to medical and health care services, as well as other aspects such as positioning and tracking, localization, logistic, etc [7]. Therefore, ensuring the data will be transmitted to the end-users without loss is a must. If packet losses occur, one must quickly identify the actual cause of packet losses and troubleshoot it as soon as possible, otherwise, it will cause the end-users frustrating in keep receiving incomplete data or inaccurate data and result in end-users being unable to make quick decisions and responses on the system. Thus, having a general algorithm that can identify or differentiates the cause of packet losses is a must nowadays. It will be benefiting to all the sensor systems as Industry 4.0 had emerged and the core component of Industry 4.0 is the Internet of Things (IoT), which is wireless sensor

networks. For example, in the military application of wireless sensor networks, once the sensor nodes detect opposing forces such as tank, the end-users must quickly send soldiers or missiles to counter it, therefore packet loss is not tolerable.

In addition to that, this new algorithm can be used to improve the capabilities of forensic works in the wireless security field. It is very helpful that if the forensic workers in wireless security could quickly pinpoint the reason for packet losses, it could save much time to start and get into the main context of the forensic works. Moreover, it can be utilised in real-time incident response systems (IRS) to be more accurate to identify the reason for packet losses on the incident responses.

## **1.5 Background Information**

### **1.5.1 Introduction of Wireless Sensor Networks**

Wireless Sensor Networks (WSN) are self-configured, infrastructure-free wireless networks that monitor physical or environmental conditions such as temperature, sound, vibration, pressure, motion, or pollutants and cooperatively pass their data through the network to a central location or sink where the data can be observed and analysed. A sink, also known as a base station, serves as a link between users and the network. By injecting queries and receiving results from the sink, one may get the necessary information from the network. The end users can use the data or results from the sink, to make some important decisions on the system. There are typically consists of hundreds to thousands of sensor nodes in a wireless sensor network. Radio transmissions allow the sensor nodes to communicate with one another. Sensing and processing devices, radio transceivers, and power components are all included in a wireless sensor node [8]. In addition, due to the limitation of resources implemented in the sensor nodes, such as computing speed, storage capacity and transmission speed, they are resource-constrained devices compared to others. After being installed, the sensor nodes are responsible for self-organizing a suitable network architecture, which frequently includes multi-hop communication. The inbuilt sensors then begin gathering data of relevance. Wireless sensor devices also respond to requests for particular instructions or sensing samples supplied from a "control site", which means end-users. The figure below shows a typical WSN architecture:

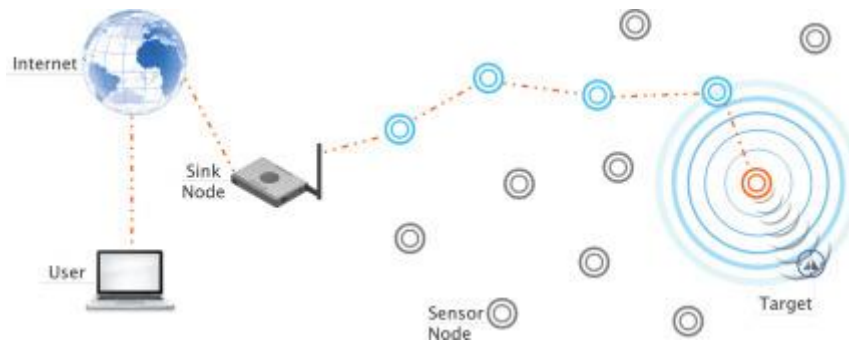


Figure 1-5-1

Due to numerous restrictions, wireless sensor networks (WSNs) allow novel applications and necessitate non-traditional protocol design paradigms. Due to the need for minimal device complexity and low energy consumption, a suitable balance between communication and signal/data processing capabilities must be implemented. Since the previous decade, this has motivated a massive effort in research, standardisation, and industry investments in this sector. Furthermore, WSN has grown in popularity as a result of its versatility in solving issues across a variety of application areas, and they have the potential to transform our lives in a variety of ways. WSNs have been effectively used in a variety of applications, including military applications, area monitoring, transportation, health applications, environmental sensing, etc.

### 1.5.2 Wi-Fi Standards

Nowadays, most wireless end devices equipment supports various levels of industry communication standards. The IEEE 802.11b/g is the most emerged standard throughout the wireless industry. It provides various support for device mobility to effectively serve most needs of the University. In this project, the communications between the sensor nodes and the base station, as well as the transmission of packets from sensor nodes to the base station, will also be utilizing Wi-Fi technology. Therefore, this section will provide a deep look into the standards of Wi-Fi to let readers have a better understanding of how Wi-Fi technologies work.

The name – 802.11b which is pronounced as “Eight-O-Two-Eleven-Bee”, actually is one of the wireless standards throughout the industry. The term ‘Wi-Fi’ is equivalent to wireless access in general, which is a trademark owned by the Wi-Fi Alliance. The Wi-Fi Alliance provides the certification that Wi-Fi products must meet

the IEEE's set of 802.11 wireless standards. The most popular questions from people are – What are Wi-Fi 6, Wi-Fi 5, and Wi-Fi 4? Is there any difference between them? The difference between them is actually in terms of their standards, speed, the frequency used, and the type of spread spectrum modulation used.

Wi-Fi 4 used the standards named IEEE 802.11n. It is standardized by the Wi-Fi Alliance, it was the next generation of the Wi-Fi standards after IEEE 802.11g, 802.11a, 802.11b. Wi-Fi 4 had improved the overall performance in terms of throughput in the access medium and increased security in the WLAN or WMAN. Wi-Fi 4 is the first standard to utilize the Multiple-Input Multiple-Output (MIMO) technology to enable the high efficiency of the communication with the modulation method Orthogonal Frequency Division Multiplex (OFDM), it would fully utilize the maximum bandwidth of the medium. Wi-Fi 4 uses 2.4GHz and 5GHz frequencies with a speed of up to 600Mbps.

IEEE 802.11ac standards are named Wi-Fi 5 by Wi-Fi Alliance. It is the standard that is mostly adopted by home wireless routers nowadays. It also adopts the previous Wi-Fi 4 technologies which are the MIMO, to enable multiple antennas on sending and receiving devices to reduce error and boost speed. It utilizes the frequency of 5GHz which can support data rates up to 3.46Gbps. However, some router vendors, still include the technologies to support the IEEE 802.11n, which also supports the 2.4GHz frequency for older devices, not only that, but it also provides additional bandwidth for improved data rates.

IEEE 802.11ax is the successor of IEEE 802.11ac, which is again labelled by the Wi-Fi Alliance as Wi-Fi 6. It is also well known as High-Efficiency WLAN. It is designed to operate under bands between 1 – 7.125GHz which also includes previous standards that support 2.4GHz and 5GHz. The key difference that differentiates Wi-Fi 6 is the use of orthogonal frequency-division multiple access (OFDMA), this technology is much equivalent to the cellular technology that utilize nowadays. It had better power control methods to avoid the interference of neighbouring nodes to prevent errors.

### 1.5.3 Definition of Packet Loss

This section will provide an overview of the definition of packet loss. A packet will be generated whenever two network devices were trying to communicate with each other. The packet will contain all the information that the receiver intended to receive, similarly, while the receiver was replying to the sender, the sender will opt to wait for the packet to arrive. However, sometimes the packet may be large, and it may not fit into the Maximum Transport Unit (MTU) of the router. The packet will be segmented into several small packets that transmit at different times. At the destination, the receiver will need to collect all the segmented packet and reassembles them. This transmitting process will cause packet loss, where some segmented packets are unable to reach the destination. Furthermore, the whole transmitting process was using a standard protocol - Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) to maintain the efficiency of the network. However, no matter which protocol was used, there is still a chance of packet loss happening.

As mentioned, packet loss simply refers to a packet or several packets that were lost or being dropped by the router during the transmission to the destination. In wireless networks, due to the nature of radio waves transmission, it is more frequent that a packet is lost because of radio wave interference. There are several types of packet loss – network interference packet loss, cyberthreats packet loss, etc. Regardless of which types of packet loss, they will lead to one problem – performance issues. For example, in videoconferencing, jitter is created as if several of the segmented small packets were unable to reach the destination, this situation also applies to audio communications where it creates frequent gaps in speech and jitter. Packet loss will also cause broken-up images and unintelligible speech. Finally, cyberthreats packet loss is the most harmful type of packet loss. This is a cyberthreat that which cybercriminals hack your routers and tampered them to drop the packets using certain commands. This is critical because cybercriminals can sniff your packet out and analyse the content inside the packet that might contain your credentials. Moreover, there is another well-known attack that might also lead to packet loss – The Denial of Service (DoS) attack. This situation is that the router is busy handling many illegitimate packets and will drop the legitimate packet, which leads to users can't access their emails, files, etc.

## 1.6 Recent Works

In previous work, the network simulation was carried out using the OMNeT++ network simulator and the INET framework. OMNeT++ is a C++ simulation toolkit and framework that is flexible, modular, and component-based and is mostly used to create network simulators [5]. It contains a lot of frameworks that can be used to build up the environment without the users manually adding in all the modules from scratch. The implemented work was according to chapter 3 network model and methodologies in project 1. There is a total of 15 nodes will be distributed in the network architecture of a 100x100 meters area. For the work on the previous OMNeT++, the network model was implemented at around 40%. Based on the proposed method discussed in the previous work, the source node will be sending UDP traffic with a packet size of 500 bytes to the Base Station at a random time interval using the exponential probability distribution. Furthermore, the routing protocol proposed is using the Ad-hoc On-Demand Distance Vector (AODV) routing protocol and it is successfully implemented into the network model on OMNeT++. Therefore, if the original link between the source node and the Base Station is down, it will immediately query other routes that utilize the least hop to the Base Station as the new route. Lastly, one of the main components of the network simulation is the network interferer had also been implemented on OMNeT++. It is placed behind the Base Station because of only intended to interfere with the links between the Base Station and the middle neighbour node of the Base Station to simulate the network issues packet loss. In short, the completed work is the source node can communicate with Base Station through the AODV routing protocol and the network interference is used to simulate the network issues packet loss. Project 1 also mentioned that the originally proposed method in chapter 3 of project 1 might be modified as the algorithm proposed might be too hard to be implemented into OMNeT++ as the library needed to be changed a lot. Based on what had been implemented, the source node can communicate with the base station successfully, and the radio interferer is also working very well on causing some packet loss in this implementation. The only concern is that there are a lot of changes that need to be done to the original INET framework library and its source code to produce the intended result. The base station is not only required to detect there is packet loss occurs, but it is also responsible for analysing the reason for packet loss. The workload of the base station might be overwhelming and required to modify all the statistics produced



by the base station on the INET framework and its source code. Finally, the algorithm of analysing the reason for packet loss involves many of the intermediate nodes as it requires different routes to the source node, therefore the proposed method might have slight changes in this project 2.

## **1.7 Report Organization**

This report is organized into 7 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Model, Chapter 4 Experiment/Simulation, Chapter 5 System Evaluation and Discussion, Chapter 6 Conclusion and Recommendation. The first chapter is the introduction of this project which includes the problem statement, project background and motivation, project scope, project objectives, project contribution, highlights of project achievements, and report organization. The second chapter is the literature review carried out on several existing intrusion detection systems (IDS) developed for Wireless Sensor Networks (WSN), the selective forwarding attack with detection and countermeasure scheme and some packet loss analysis in WSN. The third chapter is discussing the overall system model and specification that will be used as the network simulation model of this project. The fourth chapter is regarding the overall simulation run. The fifth chapter reports the simulation result such as the simulation result, the feasibility, etc. Lastly, the sixth chapter reports the conclusion of this simulation project and gives some recommendations for this project.

## CHAPTER 2

### Literature Review

This chapter will study some of the prior works to have a better understanding of the current trends. This project studied the current intrusion detection systems (IDS) developed for Wireless Sensor Networks (WSN), the selective forwarding attack that will be used in this project with some detection and countermeasure schemes and finally some of the packet loss analysis in WSN.

#### 2.1 Previous work on Intrusion Detection System (IDS) in Wireless Sensor Networks (WSN)

A wireless network is made up of nodes that can maintain a wireless communication channel amongst themselves without the use of any permanent infrastructure. This, among other things, distinguishes wireless networks from wired networks. According to [9], the IDS mechanism applied for wireless ad hoc networks cannot be implemented directly in wireless sensor networks. They had pointed out a few differences between wireless ad hoc networks and wireless sensor networks:

- Every node in an ad hoc network is generally kept and controlled by the end-user. In a sensor network, on the other hand, each node is completely self-contained, transmitting and receiving all the sensed data from the sensing node itself to the Base Station, which is often controlled by a human user.
- Sensor nodes have more limited computing resources and batteries than ad hoc nodes. For example, the MICA2 sensor node only contains an 8Mhz CPU. Furthermore, the programme flash memory and serial flash memory even only contain 128Kb and 512Kb respectively.
- Sensor networks have a very particular purpose: to monitor the surrounding environment information (such as temperature, humidity, etc). As a result, hardware components, as well as communication/configuration methods, are extremely specialised.
- Sensor networks have a greater node density than ad hoc networks. Sensor nodes, on the other hand, have a higher risk of failing and disappearing from the network owing to battery restrictions and inadequate physical security.

Among these differences, they had also concluded some reasons why the IDS mechanism developed for wireless ad hoc networks cannot be implemented directly in

wireless sensor networks. To begin with, having an active full-powered agent within each node is not possible. To notify the human user, an IDS for sensor networks must also transmit warnings to the base station. Finally, the IDS must be simple and highly specialised to respond to particular sensor network threats as well as the protocols that are utilised across the network.

[9] had proposed a general IDS architecture for wireless sensor networks. It consists of several entities. First, the local agents are located in every sensor node. Local agents' job is to find any assault or threat that might disrupt the sensor nodes' regular operation by evaluating just local sources of data. These sources include the node's current state, packets received and delivered by the node, environmental measures, and any accessible information about its neighbours. Second, the global agents that also located in every sensor node. Global agents oversee evaluating packets sent and received by their near neighbours. They can also act as watchdogs, receiving and processing packets relayed by next-hop nodes based on protocol-specific methods. Because global agents may receive packets from both neighbours and the next-hop (due to the broadcast nature of communications), they can be prepared to identify whether a specific node is discarding or altering packets by analysing them. However, studying the network would be an energy-intensive task if all global agents were engaged and listening to their neighbourhoods at the same time. As a result, only a fraction of nodes should activate their global agents to cover all communications in the sensor network. Third, the data structures are used to store all the information about every sensor node's surroundings to function properly. The information will be divided into two main categories: an alert database that contains information about alerts and suspicious nodes and a list of neighbours of the immediate neighbour's node. This will be done by using the Bloom filters data structure. This technique can reduce the size of the storage by 75% and at the cost of introducing a probability between 16% and 40% of false positives [9]. Lastly, a spontaneous watchdog. The spontaneous watchdog is activated through the global agents in the network. The global agents will become the spontaneous watchdog per packet circulating in the network. The spontaneous watchdog will monitor the whole traffic of the particular packet and save all the information.

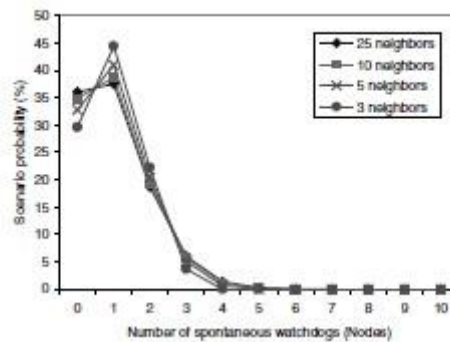


Figure 2-1-1

The figure above shows that every node has the same chance of activating its global agent becoming the spontaneous watchdog of the particular traffic, one out of every three packets in the network will travel unattended, and the majority of packets will be examined by one or two neighbours, regardless of network density.

## 2.2 Previous work on detection of selective forwarding attack

A selective forwarding attack is one or more malicious sensor nodes function similarly to regular sensor nodes, except that they selectively discard packets, thus will cause inaccurate data. If the malicious sensor nodes drop the packets containing critical information, the monitor systems' whole environment may be destroyed.

### 2.2.1 Lightweight multi-hop acknowledgement-based detection scheme

[10] had proposed a research paper that detects the selective forwarding attack using their proposed method which is a lightweight multi-hop acknowledgement-based detection scheme that lowers the communication overhead but increased detection accuracy. The figure below depicts the many types of malicious sensor node deployments on a single forwarding path:

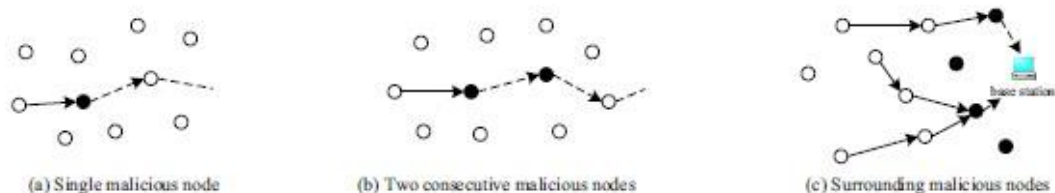


Figure 2-2-1

Figure 2-2-1(a) shows there is only a single malicious sensor node being implemented on the single forwarding path, however, figure 2-2-1(b) shows that two consecutive malicious sensor nodes are being deployed on a forwarding path. [10] stated that the deployment of figure 2-2-1(b) can make it more difficult to detect packet dropping. Lastly, figure 2-2-1(c) shows that the base station was surrounded by several malicious sensor nodes, which caused the base station to be deafened by refusing to forward any packets at all.

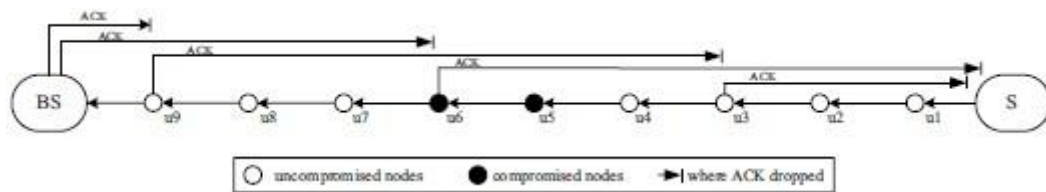


Figure 2-2-2

Figure 2-2-2 shows an example of the [10] detection scheme. The source sensor node will send its data packet to the base station. First, they pre-defined two variables:  $ACK\_SPAN$  and  $ACK\_TTL$ . According to figure 2-2-2, it sets  $ACK\_SPAN = 3$  and  $ACK\_TTL = 6$ . This implies that  $u_3$ ,  $u_6$  and  $u_9$  will be generating an ACK packet and transmitting it back to the source sensor node direction with a hop limit of 6 hops. Assume that  $u_5$  is the malicious sensor node, when it selective drops a packet, the remainder nodes ( $u_6$  to  $u_9$ ) will not receive any packet to be forwarded to the base station, therefore  $u_6$  and  $u_9$  will not generate ACK packets and send them back to the source sensor node. As a result,  $u_3$  and  $u_4$  will not be receiving any ACK packets, originally supposed to receive two ACK packets according to figure 2-2-2. Thus,  $u_3$  will set  $u_4$  as the suspicious node and raise an alarm packet and send it back to the source sensor node. Similar action will be taken by  $u_4$  as well. Source sensor node upon receiving the two-alarm packets from  $u_3$  and  $u_4$ , it will identify the malicious sensor node and use other paths to forward its data packet in the future.

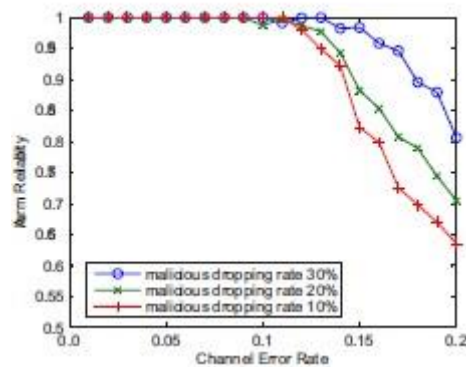


Figure 2-2-3

Figure 2-2-3 shows the simulation results of the [10] detection scheme. It shows that alarm reliability is close to 100 per cent when the channel error rate is less than 10%. Alarm reliability plummets as the channel error rate exceeds 10% since it's impossible to tell the difference between packet loss caused by intentional dropping and packet loss caused by bad radio circumstances. In short, even when the channel error rate is 15%, which is typically considered to be very severe radio circumstances, [10] detection system achieves over 80% alert reliability.

### 2.2.2 Multi-dataflow topologies (MDT) scheme

[11] proposed a multi-dataflow topologies (MDT) scheme that can protect the network topology against the selective forwarding attack instead of detecting it. They had pointed out that the previous work [10]'s detection scheme had several disadvantages. The primary disadvantage is inefficiency. As the intermediate sensor nodes need to be responsible for forwarding data packets from the source sensor node to the base station, generating ACK packets and detecting malicious sensor nodes at the same time, it causes communication overhead. The second disadvantages are some security concerns. The detection scheme cannot identify some of the attacks successfully in certain scenarios. A more crucial disadvantage is that if the detection scheme successfully identifies a malicious sensor node, the source sensor node needs to re-transmit the data packet that had been dropped using other routing paths, therefore it affects the base station cannot receive the data packet on time and perform quick response if it is sensitive events.

The multi-dataflow topologies scheme is described as follows. The base station separates sensor nodes into various categories before distributing them. A dataflow topology will be assigned to each set of sensor nodes. In addition, the routing information and identity of neighbour nodes will be stored individually in each sensor node. Then, the multi-dataflow topologies will be created among sensor nodes after they've been deployed. Each sensor node is assigned to a single dataflow topology and can interact exclusively with other sensor nodes in that topology. After the initial deployment, the surrounding environment conditions will begin to be sensed by each sensor node and transmitted to the sink (base station). The figure below shows an example of the multi-dataflow topologies:

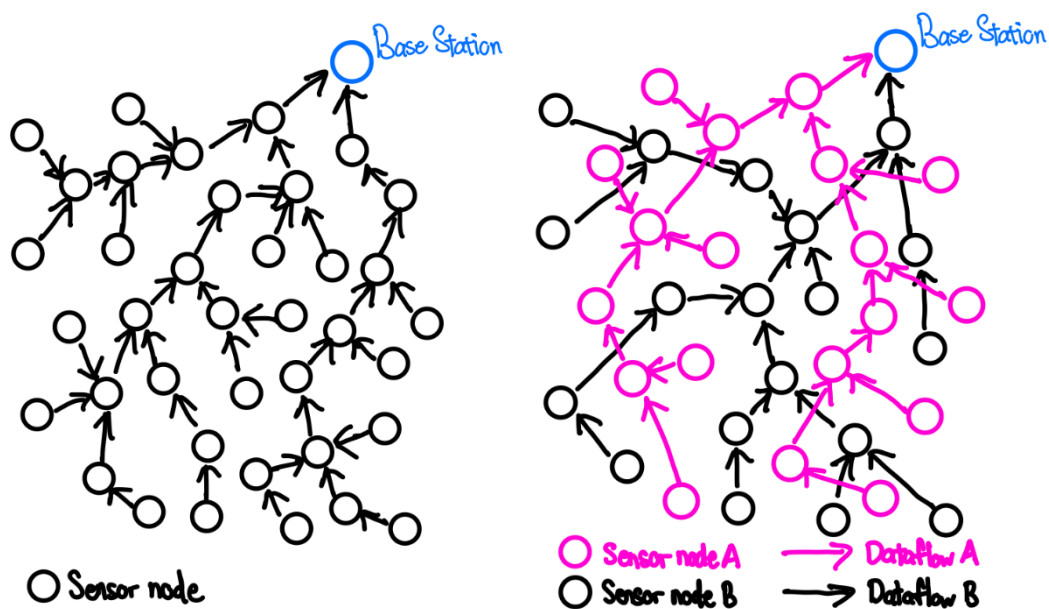


Figure 2-2-4

In the above figure, the base station will divide the original network topology into two dataflow topologies, noted as Topology A and Topology B. Both topologies can cover the whole monitored area as shown in the original network topology, therefore each of the topologies will transmit the same data to the base station, thus the base station will have the same copies of data from different each of them.

If a malicious sensor node (or nodes) engages in selective forwarding, some sensitive packets will be dropped, resulting in the sensed data by the source node will

not successfully transmit to the base station. Because the sensing region is overlapping, the base station can retrieve the lost data using alternative dataflow topologies. Take figure 2-2-4 as an example, if a malicious sensor node appears in topology A, the base station will not be able to receive the data packet sent from topology A, because of dropping packets. However, the same sensing information will still be able to receive from topology B because the sensing area of both topologies is overlapped [11].

The analysis performance of the multi-dataflow topologies scheme will be discussed as follows. First, the base station will be able to continually receive data packet which contains the information of the sensing area even though some malicious nodes are performing selective forwarding attack. In some sensitive situations, the base station cannot tolerate some data packets being delayed, however, using the multi-dataflow topologies can prevent this situation as the base station still can receive the sensitive information through the other topology. Second, the multi-dataflow topologies scheme is lightweight and simple. All the intermediate sensor nodes and source nodes are just doing their respective jobs like sensing the area and forwarding the packets to the base station. It does not require the intermediate sensor nodes to also be responsible for detecting the malicious sensor nodes and generating ACK packets compare to [10]'s detection scheme. Third, the data packet will not need to find another routing path to re-transmit the data packet to the base station, thus the base station can respond to the sensitive events immediately. Lastly, the multi-dataflow topologies scheme not only can counter selective forwarding attacks but also jamming attacks, mobile jamming attacks, sinkhole attacks, etc.

## **2.3 Previous work on packet loss analysis on Wireless Sensor Networks (WSN)**

### **2.3.1 Impact of packet loss consideration on LEACH routing protocol**

[12] had done empirical research on the impact of packet loss in wireless sensor networks using Low-Energy Adaptive Clustering Hierarchy (LEACH) routing protocols. LEACH is a dynamic cluster-based protocol that employs random cluster head rotation to provide equally distributed energy consumption throughout the network's sensors. The protocol works by separating each procedure into different rounds, in the beginning of each round, a cluster organisation set-up phase is executed



and immediately followed by a steady-state phase. The data will then be sent to the base station.

Before the simulation, they had utilised the information gathered on a test to construct a references table that will be utilised for assessing the impact of packet loss consideration on LEACH routing protocol. The information gathered on a test that was used to construct the look-up table is described as follows. [12] had utilised the Crossbow IRIS WSN to perform a series of tests. The network consists of 10 IRIS nodes in a straight line, 10 metres apart, in an open area. The test lasted three hours, while every ten seconds each of the nodes will communicate. The average packet loss and maximum packet loss for each link and the constructed table are shown below:

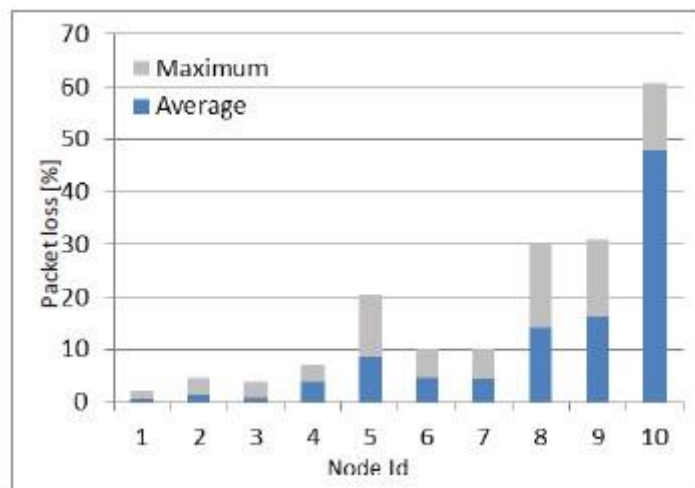


Figure 2-3-1

Distance (m)	Min (%)	Max (%)	Average (%)
10	0	2	0.57
20	0	4.59	1.19
30	0	3.75	0.72
40	1.38	7.14	3.73
50	2.09	20.37	8.60
60	2.54	10.18	4.52
70	1.86	10	4.29
80	2.77	30	14.13

90	5.90	30.95	16.08
100	23.33	60.74	47.89

Table 2-3-1

They had performed their simulations on a network model with a field that range from 50x50 meters to 150x150 meters. There are 100 nodes, and the simulation will increase the simulation area by 25x25 meters.

They begin by examining the impact of considering packet loss on network lifespan. Their findings reveal that a network's lifetime may be increased by a factor ranging from 60% for a 50x50m network to 15% for a 150x150m network. They then look at the network's throughput. The average delivery rate of successful packets, given in packets per round, is the network throughput measure. Their simulation findings showed a 52 per cent improvement in successful packet delivery for the 50x50 metres network and an 11 per cent increase for the 150x150 metres network. Finally, they looked at the effects of packet loss on energy usage. Unfortunately, their modelling results revealed that the energy loss of each node for the 150x150 metre network every round can reach up to 30%.

### 2.3.2 Performance analysis on AODV routing protocol

[13] had also done empirical research on performance analysis of wireless sensor networks by varying the reporting rate in the AODV routing protocol. Ad-hoc On-Demand Distance Vector routing protocol (AODV) constructs a route to a destination on demand, as the name suggests. It checks for up-to-date routing information and avoids routing loops by using standard routing tables with one entry per destination and sequence numbers. Route discovery is based on query and reply to cycles, and route information is maintained in the form of route table entries in all intermediary nodes along the route.

In their study, network simulator NS-2 is used to investigate the performance of Quality of Service (QoS) settings in a wireless sensor network by varying the reporting rate. In a 1000x1000 meters area, 50 nodes are deployed. All of the 50 nodes deployed are static, with one of them being the sink node, 3 UDP nodes that generate CBR

(Constant Bit Rate) traffic to be sent to the sink node and the remaining 46 relay nodes for forwarding purposes. The table below lists all of the network topology's simulation settings and node configurations:

No.	Parameter	Details
1	Channel Type	Wireless Channel
2	Radio propagation model	Two Ray Ground
3	Network interface type	Wireless
4	MAC type	Mac/802_11
5	Interface queue type	Queue/Drop Tail/PriQueue
6	Link layer type	LL
7	Antenna model	Antenna/Omni Antenna
8	Routing protocol	AODV
9	X dimension of topography	2521
10	Y dimension of topography	100
11	Time of simulation end	10.0
12	Initial energy in Joules	5
13	Traffic Type	CBR
14	Topology of Network	Random

Table 2-3-2

They studied the performance of QoS parameters such as packet delivery ratio, throughput, delay, routing overheads, average energy consumed and average residual energy. Only the packet delivery ratio, throughput, delay and routing overheads performance metric will be discussed as follows. First, when the number of packets in the network is low, the Packet Delivery Ratio (PDR) increases somewhat because no packets are dropped, and the PDR is highest at a reporting rate of 20pps. The figure below shows the PDR as a function of reporting rate:

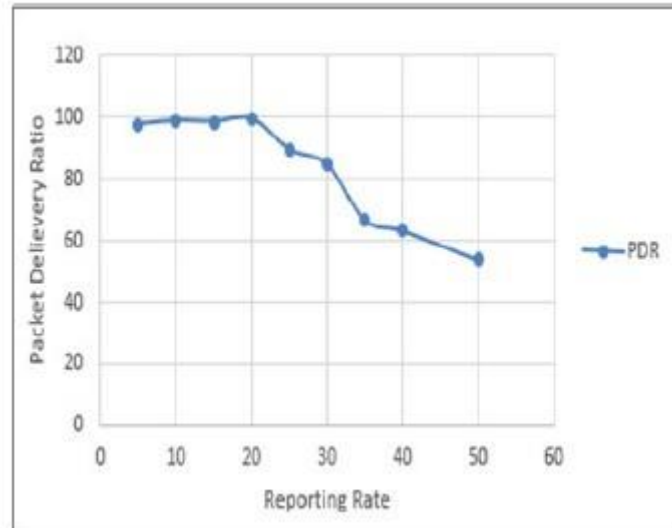


Figure 2-3-2

Second, as the number of packets in the network grows, so does the number of packets handled by each node per unit time, resulting in increased throughput. The figure below shows the throughput is maximum at reporting rate of 50pps and minimum at 5pps:

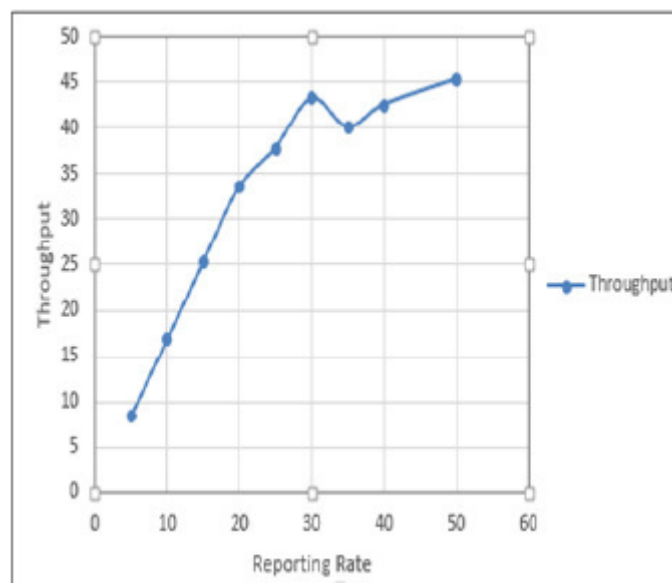


Figure 2-3-3

Third, the delay will be increased following the reporting rate increases accordingly. Delay is minimal when the quantity of packets in the network is low. However, when the network's traffic grows, the reporting rate delay is found to grow as well. The figure below shows the delay as a function of reporting rate:

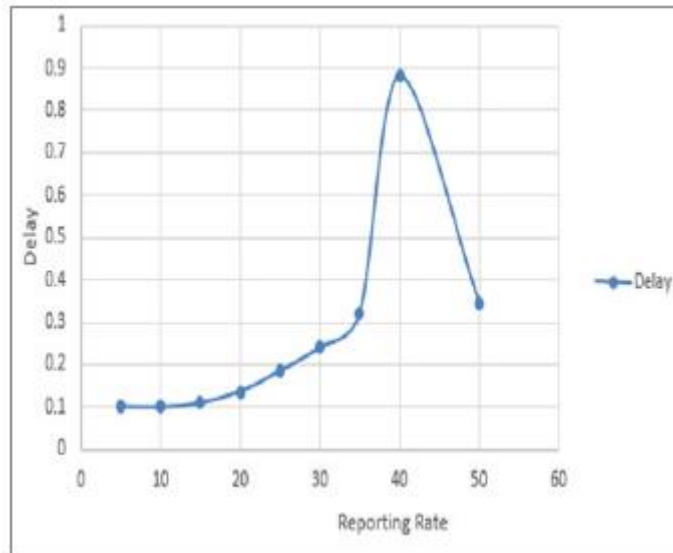


Figure 2-3-4

Finally, the routing overhead does not rise to the network's maximum reporting rate. When the quantity of packets in the network grows, the routing overhead grows as well. For a reporting rate of 50pps, the routing overhead is at its highest, minimum at reporting rate of 20pps. As a result, a reporting rate of 20 might be thought of as optimum. The figure below shows the routing overheads as a function of reporting rate:

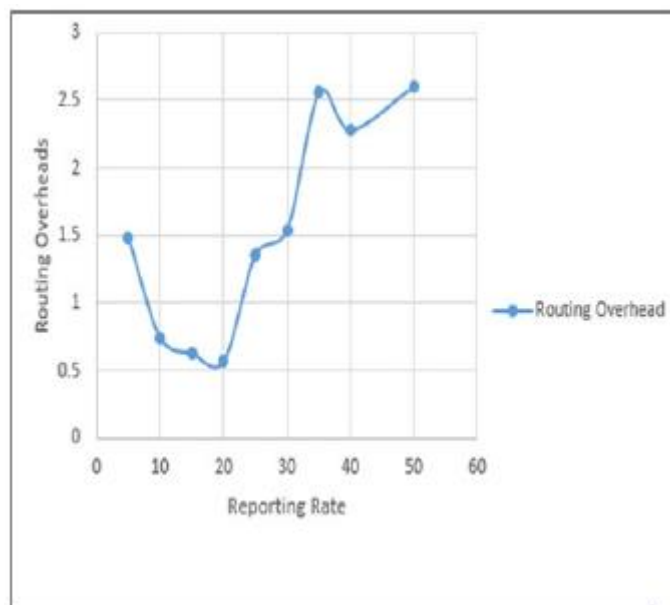


Figure 2-3-5

## CHAPTER 3

### System Model

This chapter describes the general approach to how the project's final deliverable and its outputs will be realized. It will be described in detail the methodologies, the detail of the network simulation model that will be used in simulation to get the simulation result and finally some of the implementation issues and challenges.

#### 3.1 Network Model/Overview

There is a brief discussion of this project network model design in the previous chapter 1.2 project scope. This section will be more discussed in detail the network design of this project that is used for simulation. First, there will be a total of 15 nodes distributed in the network architecture of 100x100 meters area. Two of them will be the Base Station (BS) and the source node respectively, two of these nodes will be placed at the one end of each other in a straight line. The remaining will be the forwarding nodes that sit between BS and the source node as the intermediate nodes. The source node will be generating UDP traffic, and the destination is targeted at the BS. The UDP traffic will be transmitted at a random time interval using the exponential probability distribution, and the packet size will be 500 bytes. In addition, the routing algorithm that will be used is the Ad-hoc On-Demand Distance Vector routing protocol (AODV), which is the minimum hop counts routing algorithm, and the nodes are communicated through the 802.11 wireless technology which is Wi-Fi. Lastly, the simulation will last 20 seconds by using the OMNeT++ network simulator and INET framework. The table below shows the general details of the network model:

Parameter	Details
Channel Type	Wireless Channel
MAC Type	802.11
Interface Queue Type	Drop Tail Queue
Routing Protocol	AODV
Time of simulation	20.0s
Traffic Type	UDP

Table 3-1-1

Among the 13 remaining nodes as the intermediate nodes, one of them will be the malicious node that will drop the packet without forwarding it to the BS. The attack that will be used in this simulation is the selective forwarding attack, which had been researched in the chapter 2 literature review. As the word selective forwarding means, it will just drop some of the packets but not all of the packets it receives from its neighbour's nodes. In this simulation, it will randomly drop a packet at a random time interval using the exponential probability distribution. This will be used to emulate the malicious discarding of the packet in the network.

Other than that, [14] proposed a method to generate precise packet loss patterns by intentional network interference. In this project, will be utilizing their method for intentionally creating the network interference to emulate the network issue for the reason of packet loss in the network. The radio interferer will be placed between the Base Station (BS) and one of the neighbour nodes of the BS. Therefore, the network interference will most likely happen in the link between BS and one of its neighbours.

### **3.2 Methodologies**

This project objective is to differentiate Wireless Sensor Networks' packet loss is due to malicious discarding or network issues. To propose a methodology to have the capabilities to fulfil the objective, some assumptions need to be made. Although the assumptions had been discussed in Chapter 1.2, however, this section will provide a clearer assumption about the simulation model. First, the malicious node will be configured at one of the neighbour nodes of the Base Station (BS). The malicious node will not be changed during the entire simulation time. Second, the radio interferer will only affect the link between BS and one of its neighbour nodes which is also the malicious node. Therefore, packet loss will happen in the link between BS and one of its neighbour nodes, packet loss will not happen in other links between intermediate nodes or source nodes. Third, the BS will detect there is a packet loss, it is not within the scope of this project to configure the BS on how it is going to detect whether there is a packet loss occurs. Fourth, all the nodes will not be constrained by resources such as memory, battery and computational power. Fifth, the network model is fixed, it will not be changed during this project and the simulation run.

Next, will be discussing the methodology to differentiate the cause of packet loss. According to [10], they had proposed a method of detecting the selective forwarding attack, where some of the intermediate nodes will generate an ACK packet and send it back towards the source node indicating they had received the data packet generated by the source node and successfully forwarded the packet to the next-hop node. If the node is malicious, it will not do the operation. This project had done some modifications to the detection scheme, where instead of some of the intermediate nodes will generate an ACK packet, all the nodes will be generating an ACK packet that contains its node id and sending it back to the source node. The figure below shows the details of the scheme.

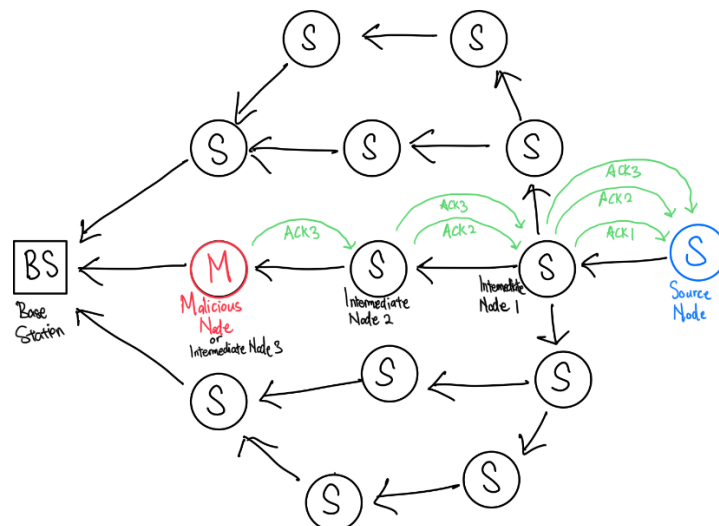


Figure 3-2-1

According to figure 3-2-1, one can observe that once the source node generates the data packet and send it to Base Station (BS), it will travel across intermediate node 1 (node 1), intermediate node 2 (node 2) and intermediate node 3 (node 3) and reach BS, it is because this simulation using the AODV routing protocol which will utilize the minimum hop count that reaches the destination. First, assume that node 3 has not yet been compromised, the packet will successfully be forwarded by node 3. Therefore, all three immediate nodes will generate an ACK packet with its node id and send it back to the source node. According to figure 3-2-1, node 2 will receive one ACK packet generated by node 3, node 1 will receive two ACK packets generated by node 2 and node 3 respectively. Lastly, the source node will receive a total of three ACK packets



generated by three intermediate nodes. This is the successful packet delivery to the BS scenario. On the other hand, assume that node 3 had been compromised and become the malicious node. Therefore, it will not forward the data packet to the BS and will not generate the ACK packet. As a result, node 2 will not be receiving any ACK packet from the malicious node (node 3), node 1 will only be receiving one ACK packet that is from node 2 and the source node will also be receiving only two ACK packets that are generated by node 1 and node 2 only. In this case, the source node will generate an alarm packet indicating the suspicious node is node 3 in the alarm packet because it does not receive the ACK packet with node id 3 and sends it to the BS using another path than the original one. Once the BS realise that there is a packet loss, it will check whether it receives an alarm packet from the source node using another path. If it receives an alarm packet, one can conclude that there is a malicious discard of the packet, otherwise is a network issue induced packet loss. It is because if the packet loss is due to network issues, node 3 will forward the packet and generate an ACK packet, therefore the source node will receive three ACK packets as a normal scenario.

### 3.3 Implementation Issues and Challenges

This project will be using the network simulator OMNeT++ to simulate the proposed method discussed in section 3.2. Furthermore, the entire simulation model and all the assumptions as discussed in section 3.1 will be implemented into OMNeT++. The implementation issue that had been encountered is that there is no specific way to directly configure a node into the malicious state that will emulate the selective forwarding attack. One must configure some of the parameters manually to make it into not dropping every packet but some by tuning the forwarding state on and off using a probability distribution. Second, the radio interferer that this project references is different scope. In prior works, they had used the physical radio interferer to generate the interference, however, this project is using the simulation tool to produce it. The network simulator tool must use another radio interferer to generate the interference. Nevertheless, the interference generated is still similar to what this project had reference to, it still produces the result intended. Finally, the challenge for implementing this project is the OMNeT++ is a software that utilises the Network Description (NED) language, which is similar to C++, however, it still took quite some time to learn the whole new language and the library required to implement this project.

### 3.4 Timeline

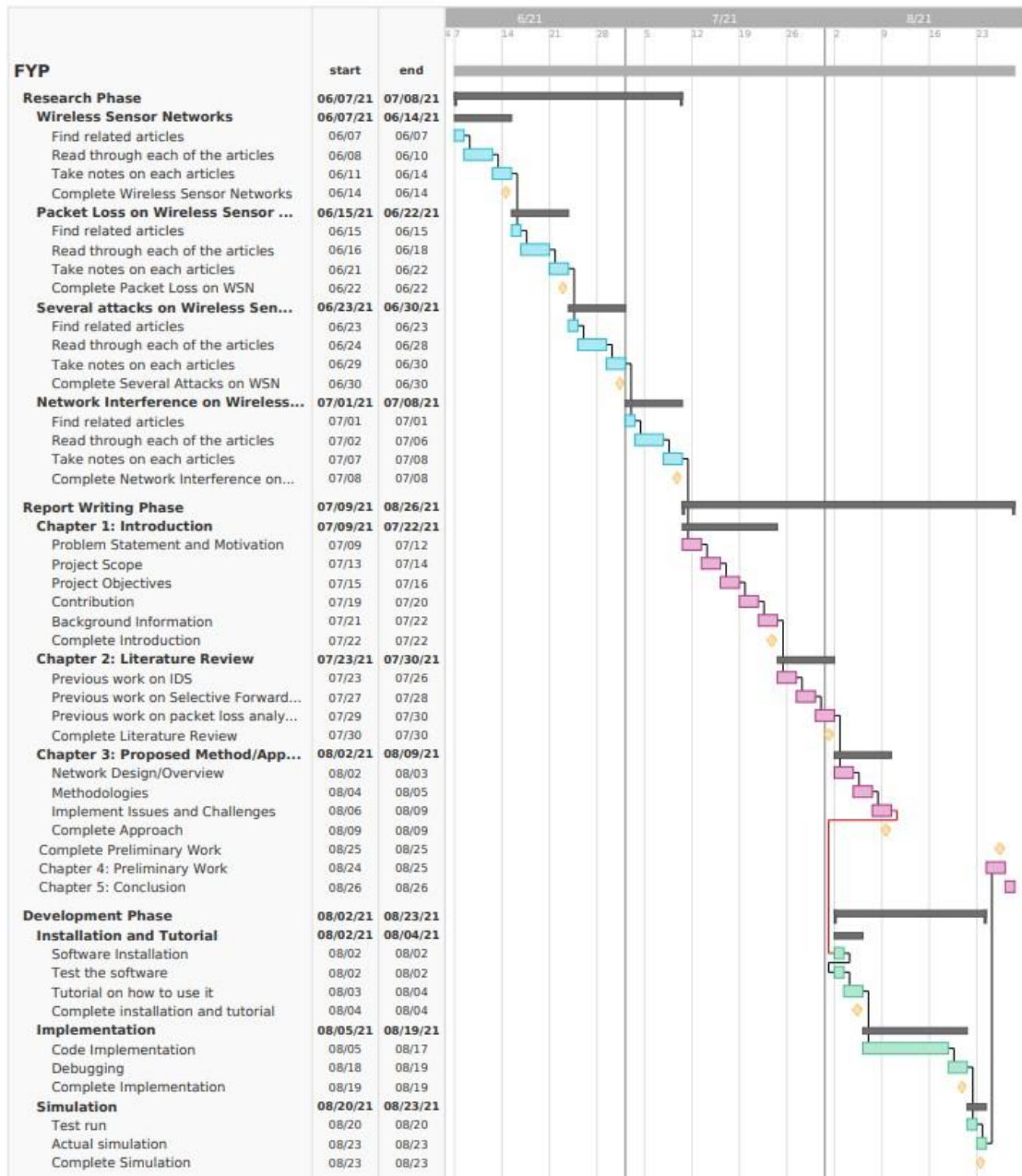


Figure 3-4-1

## CHAPTER 4

### Experiment/Simulation

This chapter will present the overall simulation setup step by step, some details settings and configuration of the simulation parameter, the network simulation process with some screenshot examples. Finally, this chapter will have a short conclusion of all the simulation settings as a remark.

#### 4.1 Software Setup

The first section will be discussing the software setup process guidelines. As mentioned in the previous chapter, this project will be utilizing the OMNeT++ network simulator and the INET framework to simulate the network model situation. Therefore, this sub-section will discuss the OMNeT++ network simulator and the INET framework setup process.

##### 4.1.1 OMNeT++ Installation

1. Download the OMNeT++ from <https://omnetpp.org/download/>, and select the Windows Versions to download. A folder named omnetpp-5.6.1-src-windows.zip will be downloaded.

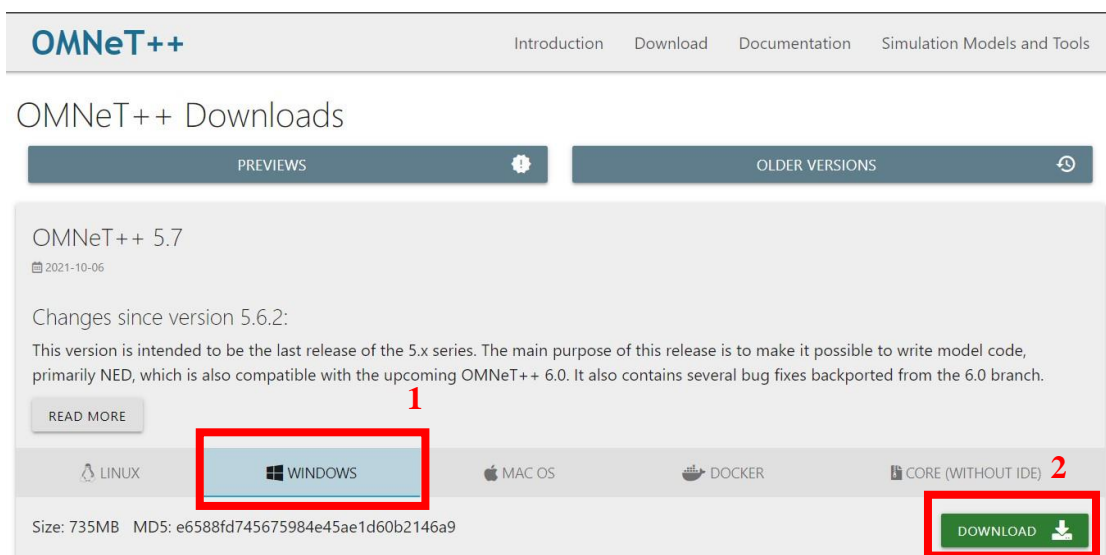


Figure 4-1-1

2. After downloading the folder, move it to the root C:/ drive for better configuration. Extract the zip file by right-clicking on it in Windows Explorer and selecting Extract Here from the menu, one can also use external programs like Winzip or 7zip to extract the folder. After extracting, one should be able to see a new omnetpp-5.6.1 folder created in C:/ drive.

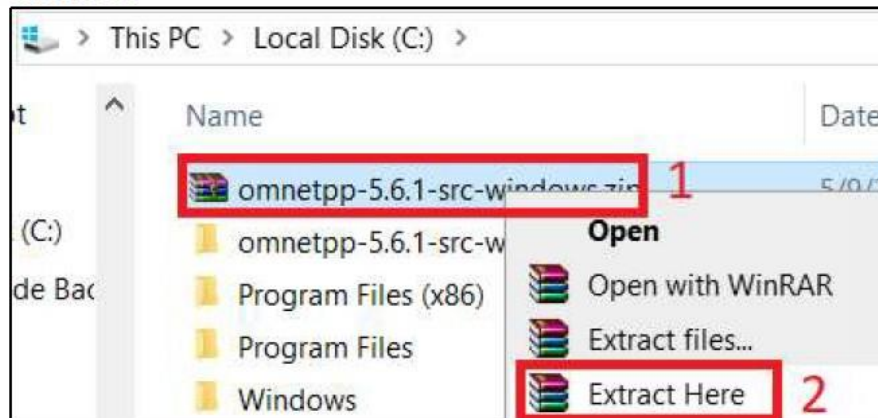
**Before extract:****After extract:**

Figure 4-1-2

3. Double-clicking on the omnetpp-5.6.1 folder, one should see a file named mingwenv.cmd. Start it by double-clicking it. It will bring up a console with the MSYS bash shell.

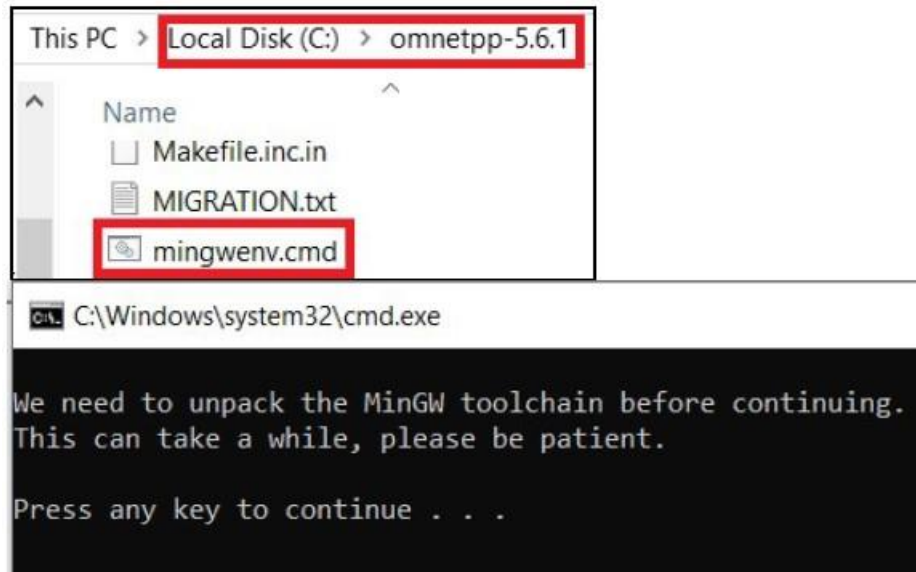


Figure 4-1-3

4. Press any key in the command prompt. It will take a while to extract all the files.

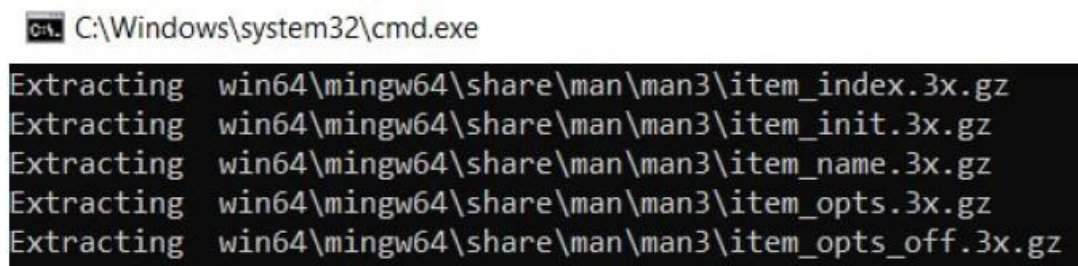


Figure 4-1-4

5. Once the extraction is completed, enter the following command in the command prompt. It will start to configure the simulation library:

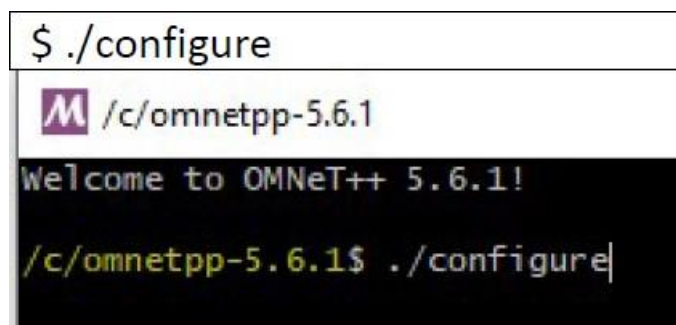
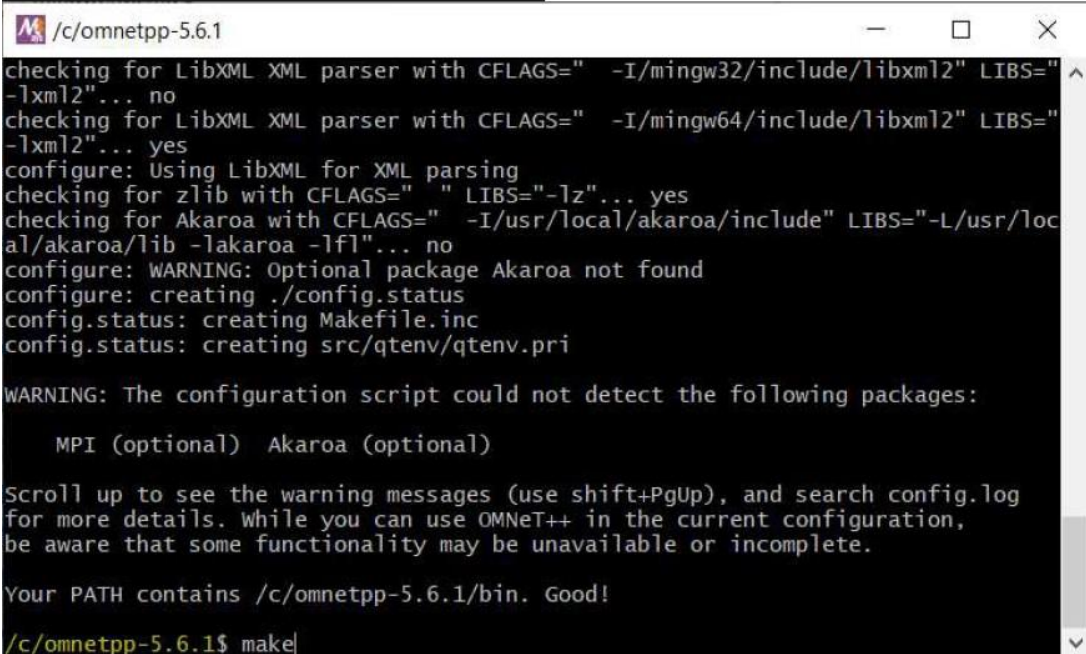


Figure 4-1-5

6. Once the configuration is completed, enter the following command in the command prompt. It will start to make the simulation library:

```
$ make
```



```
checking for LibXML XML parser with CFLAGS="-I/mingw32/include/libxml2" LIBS="-lxml2"... no
checking for LibXML XML parser with CFLAGS="-I/mingw64/include/libxml2" LIBS="-lxml2"... yes
configure: Using LibXML for XML parsing
checking for zlib with CFLAGS="" LIBS="-lz"... yes
checking for Akaroa with CFLAGS="-I/usr/local/akaroa/include" LIBS="-L/usr/local/akaroa/lib -lakaroa -lf1"... no
configure: WARNING: Optional package Akaroa not found
configure: creating ./config.status
config.status: creating Makefile.inc
config.status: creating src/qtenv/qtenv.pri

WARNING: The configuration script could not detect the following packages:

  MPI (optional)  Akaroa (optional)

Scroll up to see the warning messages (use shift+PgUp), and search config.log
for more details. While you can use OMNeT++ in the current configuration,
be aware that some functionality may be unavailable or incomplete.

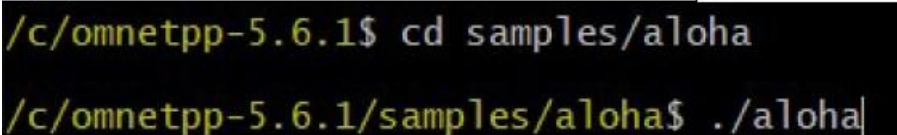
Your PATH contains /c/omnetpp-5.6.1/bin. Good!

/c/omnetpp-5.6.1$ make|
```

Figure 4-1-6

7. After the making process is completed, enter the following commands to test if the OMNeT++ is installed correctly before proceeding to the next section.

```
$ cd samples/aloha
$ ./aloha
```



```
/c/omnetpp-5.6.1$ cd samples/aloha
/c/omnetpp-5.6.1/samples/aloha$ ./aloha|
```

Figure 4-1-7

8. If it is working correctly, the samples will run using the graphical Qtenv environment. One should see GUI windows and dialogues. Click OK(1) followed by RUN(2). The simulation will be started. To stop the simulation, click STOP(3). Close the GUI.

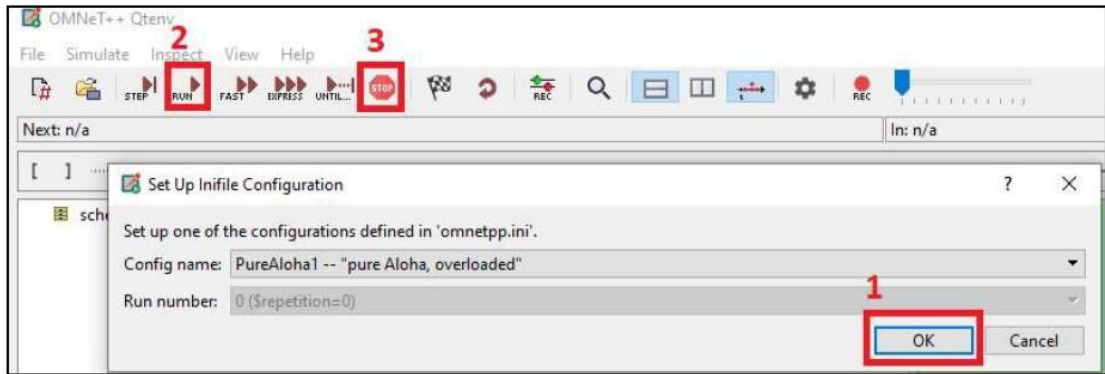


Figure 4-1-8

When the simulation is running:

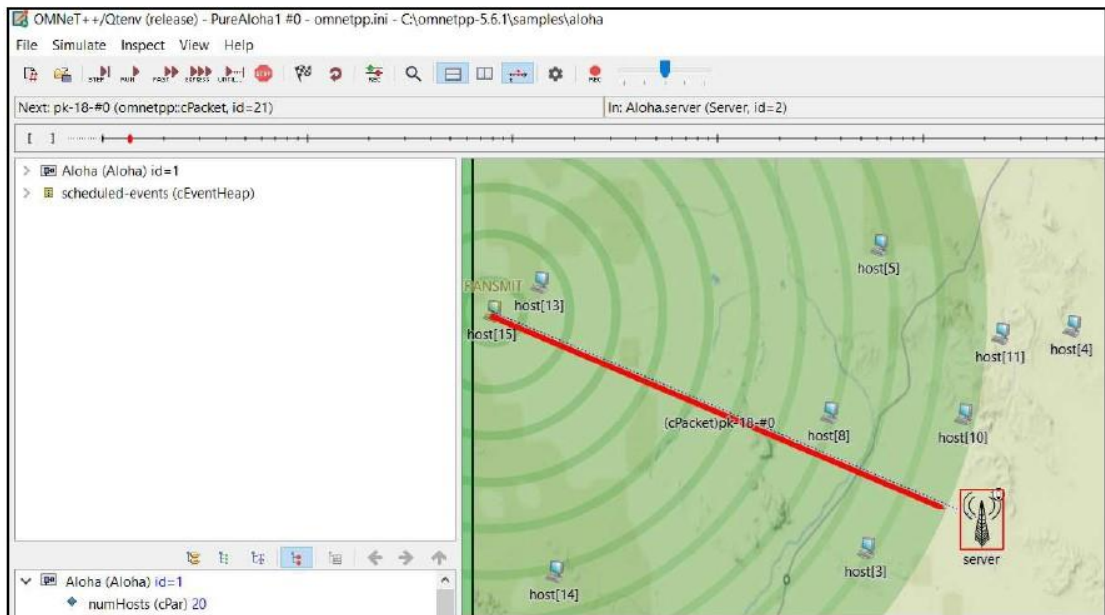


Figure 4-1-9

To close the simulation GUI:

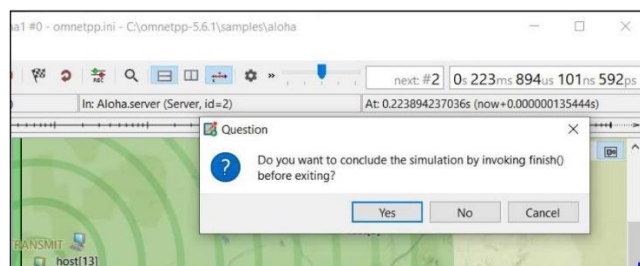


Figure 4-1-10

### 4.1.2 INET Framework Installation

1. Access [inet.omnetpp.org/Download.html](http://inet.omnetpp.org/Download.html) to download the INET framework folder.

2. Clicking on Download Latest Stable Version. A folder named inet-4.2.0-src.tgz will be downloaded.



Figure 4-1-11

3. Move the recently downloaded inet-4.2.0-src.tgz to the OMNeT++ samples folder: C:\omnetpp-5.6.1\samples



Figure 4-1-12



4. Right-clicking on inet-4.2.0-src.tgz and click Extract here. Inet4 folder is created. Rename it into inet.

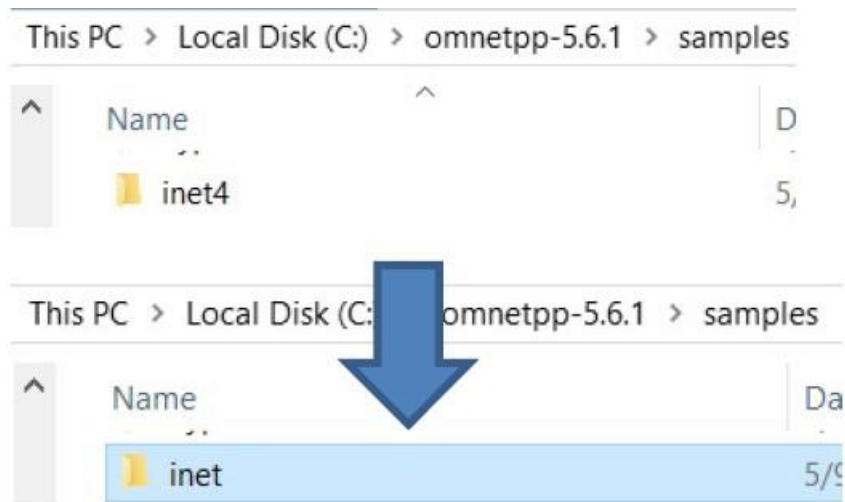


Figure 4-1-13

5. Go to the OMNeT++ folder open the file named mingwenv.cmd and type in the following command.

```
$ omnetpp
```

```
/c/omnetpp-5.6.1$ omnetpp
Starting the OMNeT++ IDE...
```

Figure 4-1-14

6. Tick Use this as the default and do not ask again. Follows by clicking Launch.

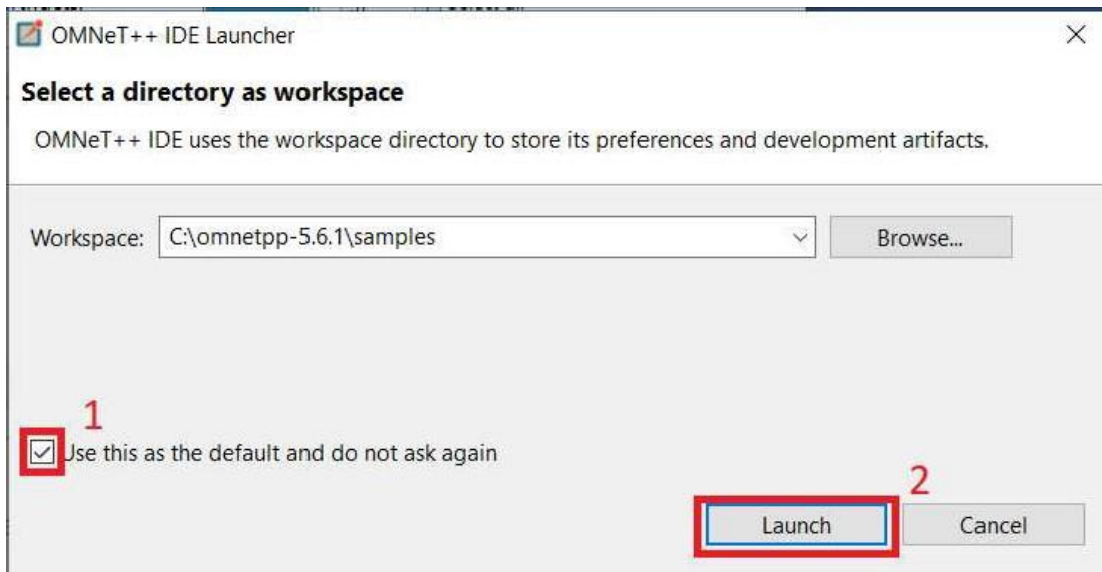


Figure 4-1-15

7. Close the Welcome page.



Figure 4-1-16

8. Untick Install INET Framework. It is because the inet is downloaded in the local drive earlier. If you download the inet by using OMNeT++ IDE, it will take a few hours to download the framework.

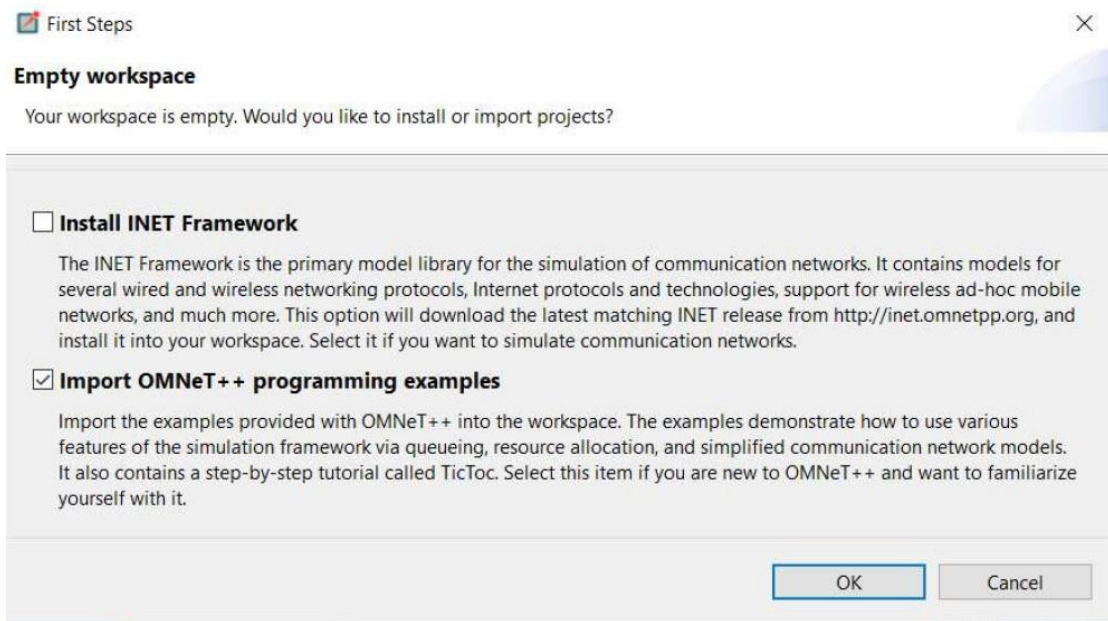


Figure 4-1-17

9. You may close all the default pages on the workspace. You should be able to see the inet folder in the Project Explorer.

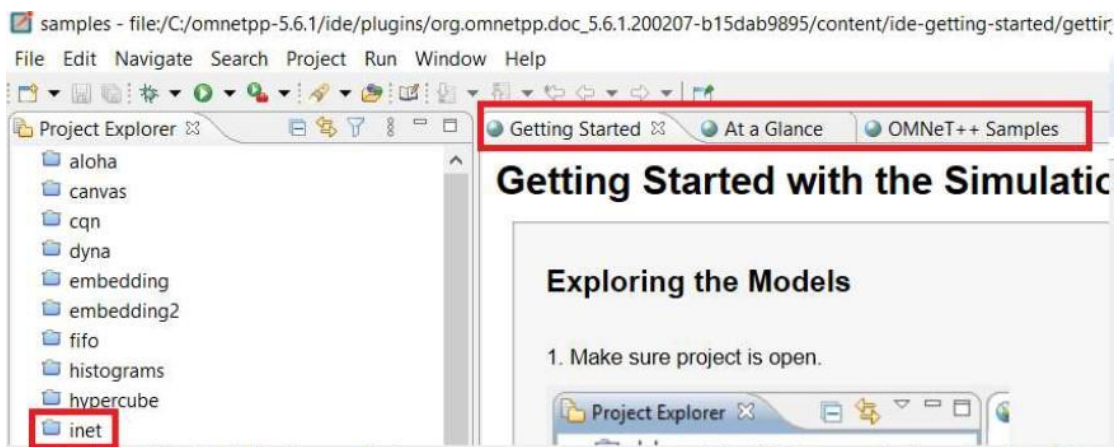


Figure 4-1-18

10. Double-clicking on inet. Expand inet → Example → adhoc → ieee80211 → right-clicking omnetpp.ini

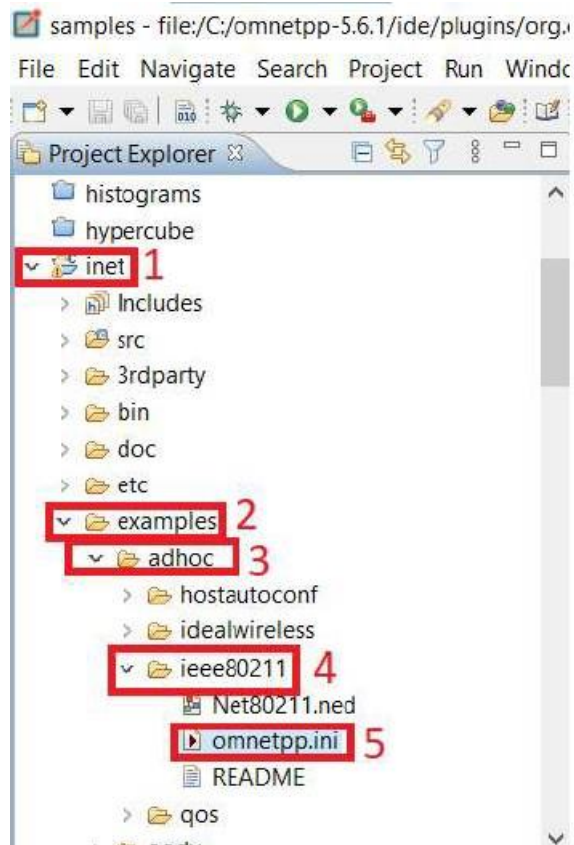


Figure 4-1-19

11. Select Run As → OMNeT++ Simulation

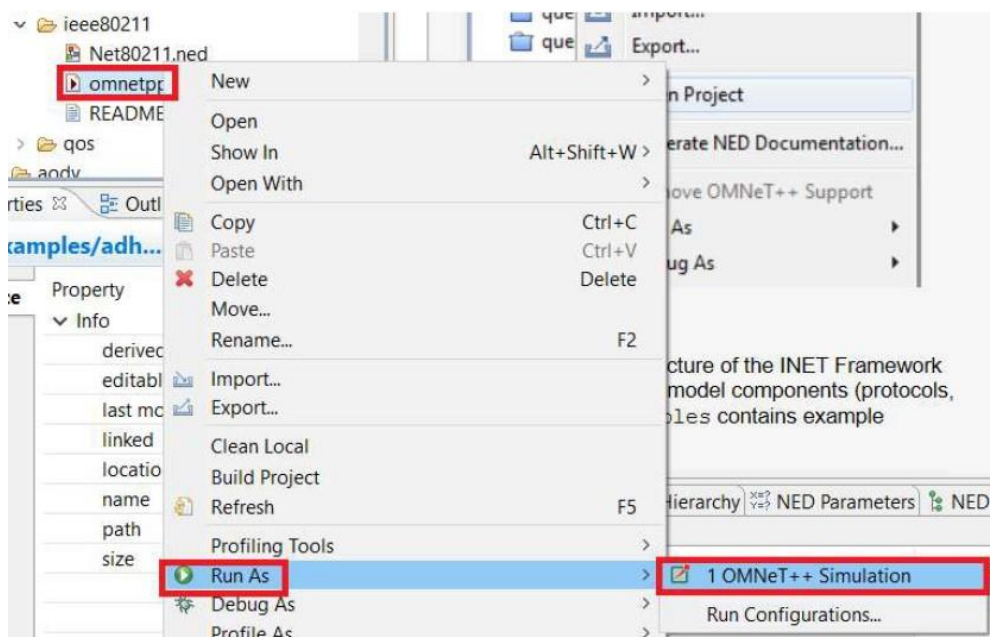


Figure 4-1-20

12. Click OK followed by Yes. For the first time, it will take a while to build the inet in OMNeT++. Once the process is completed, a GUI will appear.

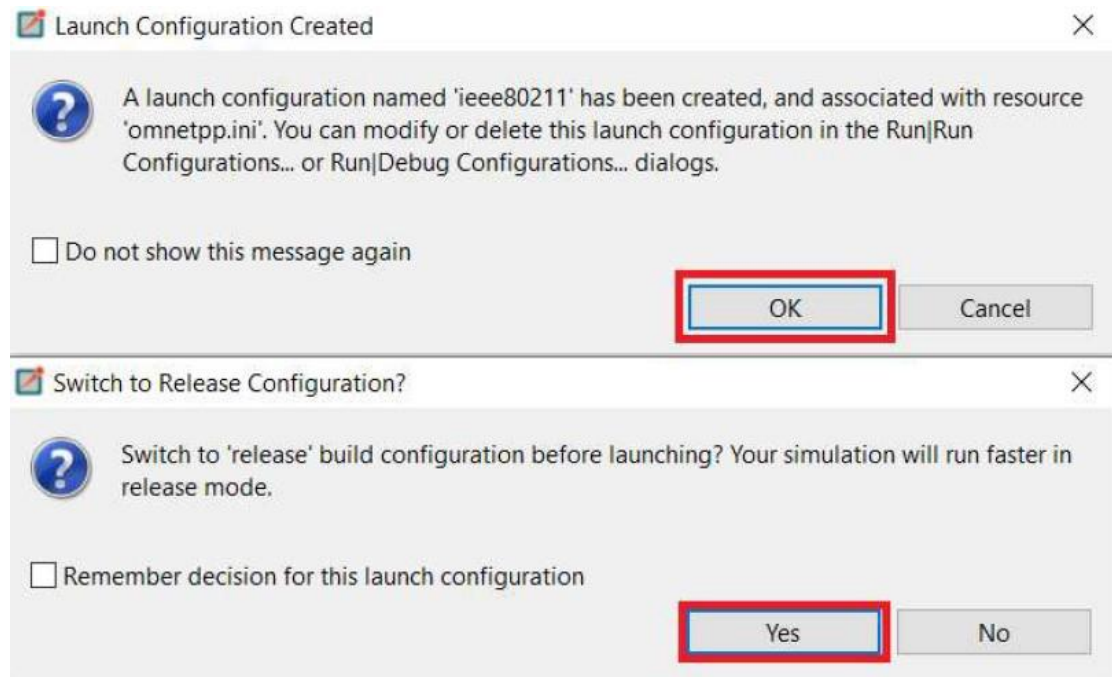


Figure 4-1-21

13. If you received the error message below, click OK so OMNeT++ will fix the error.

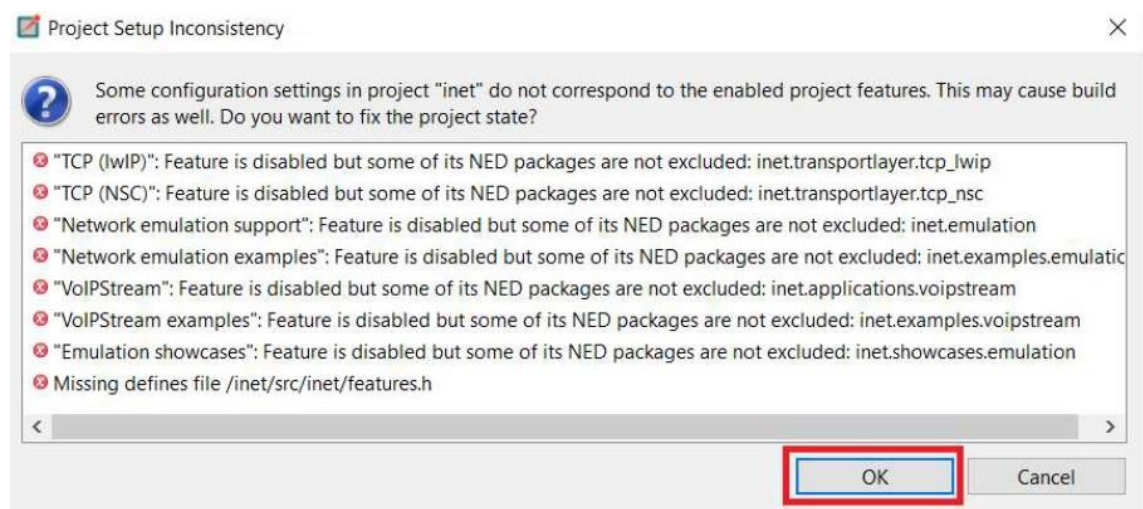


Figure 4-1-22

14. Click OK when the GUI popped.

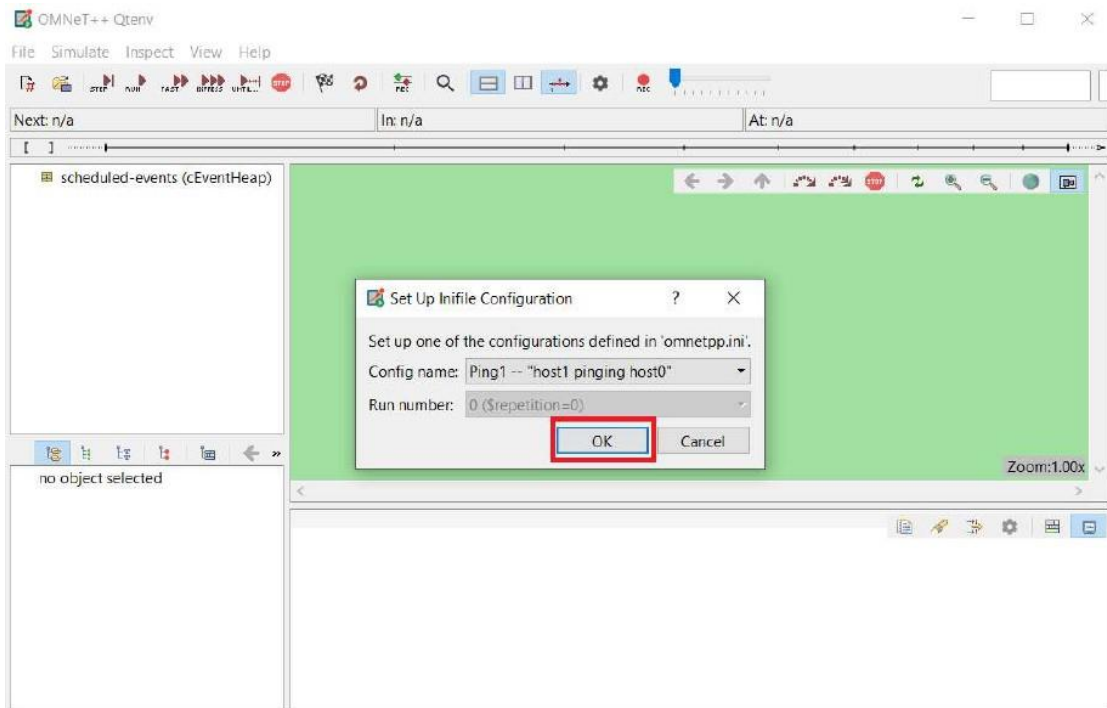


Figure 4-1-23

15. Click Run to start the simulation. If the INET installation is successful, one should be able to see that host0 is moving.

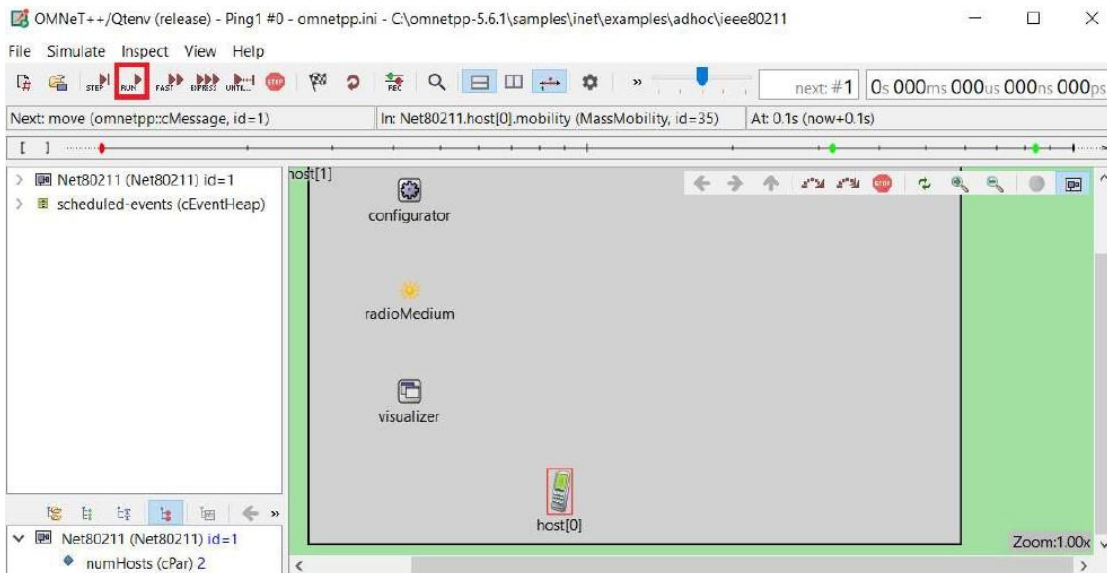


Figure 4-1-24

## 4.2 Setting and Configuration

This sub-section will be discussing the settings and configuration of the network simulation setup. As discussed earlier, this project will be utilizing the OMNeT++ network simulator and INET framework for the library. All nodes and radio mediums will be based on the INET framework library. The settings and configuration include which type of nodes were being used, which radio medium was being used, routing settings, etc, with some code snippets screenshots. The full source code will be attached in the appendix section A – Network Description File and appendix section B – INI file.

#### 4.2.1 Network Description File (NED)

First of all, in the network description file (NED), one must include the necessary library from the INET framework to configure the nodes, radio medium, some visualizer and network layer configuration. The code snippets below show the process of importing some of the necessary libraries:

```
import inet.node.inet.INetworkNode;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.visualizer.integrated.IntegratedCanvasVisualizer;
import inet.physicallayer.unitdisk.UnitDiskRadioMedium;
```

One can observe that all the libraries used were directly imported from the INET framework itself. The `INetworkNode` library is an interface for all the network nodes, such as normal PC, wireless host and sensor node. The second library is `Ipv4NetworkConfigurator`, this library is essential for all the network simulation as it setup all the network interface automatically and it is being used to configurator various IPv4 network parameter. The third library being imported is `IntegratedCanvasVisualizer`, it is a library that assists in visualizing all the network nodes, the transmit range, interference range, the packet travel path, etc. Finally, the last library to be imported is the `UnitDiskRadioMedium`, this library is being used to set up the wireless radio medium to let the wireless nodes to be communicated with each other.

Next, will be configuring the some of the parameters for the network. This includes the statistics, and the network title to be shown. The code snippets below show the network parameter configuration:

```

@display("bgb=1300,1500;bgg=100,1,greY95");
@figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

@figure[recvPkText](type=indicatorText; pos=1000,50; anchor=w; font=,18;
textFormat="packets received: %g"; initialValue=0);
@statiStic[packetReceived](source=baseStation.app[0].packetReceived;
record=figure(count); targetFigure=recvPkText);

```

The first line of the code snippets is `@display` is to set the overall network size of the field, it is set to 1300x1500, and the background colour is grey. The second line of code is `@figure`, it is to set the network model title to the desired position and set the text colour to dark blue. The third line of the code also is a `@figure`, however, this line is to set the word “packet received:”, and set the text position, etc. The last line of the code is the `@statiStic`, it is to get the statistic from a node source from a base station and record the packet received by it. The calculated statistic will be set to a targeted figure which is the figure set from the third line just now to display it.

Finally, is the last configuration in the network description file, which is the submodules configuration. In the submodules, the configuration will be setting all the nodes, the visualizer, and the wireless radio medium. The code snippets below show all the nodes configuration, configurator, visualizer and radio medium:

```

baseStation: <default("WirelessHost")> like INetworkNode {
    @display("p=50,750;i=misc/sensor2");
}
sourceNode: <default("SensorNode")> like INetworkNode {
    @display("p=1250,750;i=misc/sensor2");
}
node1: <default("SensorNode")> like INetworkNode {
    @display("p=950,750;i=misc/sensor2");
}
node2: <default("SensorNode")> like INetworkNode {
    @display("p=650,750;i=misc/sensor2");
}
node3: <default("SensorNode")> like INetworkNode {
    @display("p=350,750;i=misc/sensor2");
}
node4: <default("SensorNode")> like INetworkNode {
    @display("p=950,450;i=misc/sensor2");
}
node5: <default("SensorNode")> like INetworkNode {
    @display("p=650,450;i=misc/sensor2");
}
node6: <default("SensorNode")> like INetworkNode {
    @display("p=350,450;i=misc/sensor2");
}
node7: <default("SensorNode")> like INetworkNode {

```



```

        @display("p=950,1050;i=misc/sensor2");
    }
    node8: <default("SensorNode")> like INetworkNode {
        @display("p=650,1050;i=misc/sensor2");
    }
    node9: <default("SensorNode")> like INetworkNode {
        @display("p=350,1050;i=misc/sensor2");
    }
    node10: <default("SensorNode")> like INetworkNode {
        @display("p=800,150;i=misc/sensor2");
    }
    node11: <default("SensorNode")> like INetworkNode {
        @display("p=500,150;i=misc/sensor2");
    }
    node12: <default("SensorNode")> like INetworkNode {
        @display("p=800,1350;i=misc/sensor2");
    }
    node13: <default("SensorNode")> like INetworkNode {
        @display("p=500,1350;i=misc/sensor2");
    }
    radioInterferer: <default("WirelessHost")> like INetworkNode {
        @display("p=50,850");
    }
    configurator: Ipv4NetworkConfigurator {
        @display("p=50,50");
    }
    visualizer: IntegratedCanvasVisualizer {
        @display("p=150,50");
    }
    radioMedium: UnitDiskRadioMedium {
        @display("p=250,50");
    }
}

```

One can observe that the first line and second line of code are to set the base station and the source sensor node. The difference between these two nodes is that the base station is configured into a normal wireless host, and the source node is set to the sensor node. Both of them must implement the library `INetworkNode` to make the node into the desired one. Next, from node 1 to node 13 is the intermediate nodes that sit between the base station and source node, these 13 nodes also implement the `INetworkNode` library to configure them into a sensor node. There is also a radio interferer to be configured as a normal wireless host that also implements the `INetworkNode` library, this radio interferer is to simulate the network interference of the network. Lastly, there is a configurator, visualizer and radio medium to be configured. There respectively utilized the `Ipv4NetworkConfigurator`, `IntegratedCanvasVisualizer`, and `UnitDiskRadioMedium` library.

## 4.2.2 INI File

In this particular section, will be discussing the INI file. First of all, start with the general configuration of the network. The code snippets below show the general configuration of the network model:

```
[General]
network = FYP2
sim-time-limit = 20s

*.node*.ipv4.arp.typename = "GlobalArp"
*.sourceNode.ipv4.arp.typename = "GlobalArp"
*.baseStation.ipv4.arp.typename = "GlobalArp"
*.radioInterferer.ipv4.arp.typename = "GlobalArp"

*.sourceNode.numApps = 1
*.sourceNode.app[0].typename = "UdpBasicApp"
*.sourceNode.app[0].destAddresses = "baseStation"
*.sourceNode.app[0].destPort = 5000
*.sourceNode.app[0].messageLength = 500B
*.sourceNode.app[0].sendInterval = exponential(15ms)
*.sourceNode.app[0].packetName = "UDPData"

*.baseStation.numApps = 1
*.baseStation.app[0].typename = "UdpSink"
*.baseStation.app[0].localPort = 5000

*.node*.wlan[0].typename = "AckingWirelessInterface"
*.node*.wlan[0].radio.transmitter.communicationRange = 400m
*.node*.wlan[0].radio.receiver.ignoreInterference = true
*.node*.wlan[0].mac.headerLength = 23B

*.sourceNode.wlan[0].typename = "AckingWirelessInterface"
*.sourceNode.wlan[0].radio.transmitter.communicationRange = 400m
*.sourceNode.wlan[0].radio.receiver.ignoreInterference = true
*.sourceNode.wlan[0].mac.headerLength = 23B

*.baseStation.wlan[0].typename = "AckingWirelessInterface"
*.baseStation.wlan[0].radio.transmitter.communicationRange = 400m
*.baseStation.wlan[0].radio.receiver.ignoreInterference = true
*.baseStation.wlan[0].mac.headerLength = 23B

*.radioInterferer.wlan[0].typename = "AckingWirelessInterface"
*.radioInterferer.wlan[0].radio.transmitter.communicationRange = 150m
*.radioInterferer.wlan[0].radio.transmitter.interferenceRange = 150m

*.node**.bitrate = 1Mbps
*.sourceNode**.bitrate = 1Mbps
*.baseStation**.bitrate = 1Mbps
*.radioInterferer**.bitrate = 1Mbps
```

The code above shows the general configuration of the network model. One can observe that the “General” keyword is written in the square bracket, this implies that all the configuration below is under the “General” category. The network name was

being set to “FYP2” and the simulation time was also being set to 20 seconds. The next four lines of the code are setting all the nodes including source node, base station, radio interferer and intermediate nodes ARP to “GlobalArp”. This setting is to set all the nodes' IP addresses automatically. Next, the following 7 lines of code are configuring source node settings. The source node is configured into having one application. Furthermore, for the application set, it is an “UdpBasicApp”, and the destination is being configured to the base station. Other configurations like destination port number, message length, send interval and packet name are also being configured. The base station is being configured next. The base station was also being set the number of applications to 1 and it is a “UdpSink”, and the port number is 5000 to be matched with the source node sent port number. All the intermediate nodes will configure their wireless interface into “AckingWirelessInterface”. All the intermediate nodes will also configure the transmitter range and ignore interference to 400 meters and true respectively. The transmitter range and ignore interference are being configured is because to avoid overlapping of intermediate nodes. The same configuration is also being applied to the source node and base station. However, the radio interferer is configured the communication range and interference range to 150 meters respectively. This is to simulate the network interference to the base station. Finally, a bit rate of 1Mbps is being configured for all the nodes.

Secondly, will be configuring some of the animations to make the network simulation clearer to observe the result. The code snippets below show the configuration:

```
[Config Animations]
description = "Setting up some animations"

#Configurator
*.configurator.dumpAddresses = true
*.configurator.dumpTopology = true
*.configurator.dumpLinks = true
*.configurator.dumpRoutes = true

*.visualizer.dataLinkVisualizer.displayLinks = true
*.visualizer.dataLinkVisualizer.packetFilter = "UDPData*"
*.visualizer.networkRouteVisualizer.displayRoutes = true
*.visualizer.networkRouteVisualizer.packetFilter = "UDPData*"
*.visualizer.interfaceTableVisualizer.displayInterfaceTables = true
```

The square bracket is being set to “Animations”. It is just a title for this current configuration. The description is being set for a better understanding of this certain configuration. The configurator is configuring some parameters to a true value such as “dumpAddresses”, “dumpTopology”, “dumpLinks” and “dumpRoutes”. All of these configurations' purpose is to show the IP address of each node, to show the path that packets will be travelled, to show the links between each node if it is present and to show the routing path. The visualizer is being configured next. Same as the configurator, it is just configured some of the parameters to display out the packets and the links such as the packet filter parameter is being configured into “UDPData\*”, it is because in the network there have many packets being communicated, however this simulation only interested on the UDP packet that is being sent by the source node.

Thirdly, the routing protocol is being configured in each of the nodes to let the packet travel from the source node to the base station. The code snippet below shows the configuration of routing:

```
[Config AODVRouting]
description = "AODV routing"
extends = Animations

*.configurator.optimizeRoutes = false
*.configurator.addStaticRoutes = false

*.node*.ipv4.routingTable.netmaskRoutes = ""
*.sourceNode.ipv4.routingTable.netmaskRoutes = ""
*.baseStation.ipv4.routingTable.netmaskRoutes = ""

*.sourceNode.typename = "AodvRouter"
*.baseStation.typename = "AodvRouter"
*.node*.typename = "AodvRouter"

*.visualizer.dataLinkVisualizer.packetFilter = "aodv* or UDPData*"
*.visualizer.routingTableVisualizer.displayRoutingTables = true
*.visualizer.routingTableVisualizer.destinationFilter = "baseStation"
```

The configuration name is being configured to “AODVRouting”. It also extends the configuration “Animations”, it is because this configuration wants to implement the configuration in the “Animations”. To achieve the AODV routing in the network, the optimal route and add static routes parameters in the configurator must be set to a false value. Furthermore, the source node, base station, and all the intermediate nodes' routing tables must be set to blank. The source node, base station and all the

intermediate nodes are also being changed to an “AodvRouter”. This changes the nodes to implement the AODV routing scheme. Finally, the visualizer also being configured to visualize the routing path of the AODV packets including the packet exchange phase, etc.

Finally, the interference of the radio interferer is being configured. The code snippet below shows the configuration of the network interference.

```
[Config Interference]
description = "Adding Interference"
extends = AODVRouting

*.node*.wlan[0].radio.transmitter.interferenceRange = 0m
*.sourceNode.wlan[0].radio.transmitter.interferenceRange = 0m
*.baseStation.wlan[0].radio.transmitter.interferenceRange = 0m

*.baseStation.wlan[0].radio.receiver.ignoreInterference = false

*.radioInterferer.numApps = 1
*.radioInterferer.app[0].typename = "UdpBasicApp"
*.radioInterferer.app[0].destAddresses = "baseStation"
*.radioInterferer.app[0].destPort = 6000
*.radioInterferer.app[0].messageLength = 500B
*.radioInterferer.app[0].sendInterval = exponential(15ms)
*.radioInterferer.app[0].startTime = 1.5s
*.radioInterferer.wlan[0].radio.displayInterferenceRange = true

*.baseStation.numApps = 2
*.baseStation.app[1].typename = "UdpSink"
*.baseStation.app[1].localPort = 6000
```

The configuration name is being set to “Interference”. It also extends the “AODVRouting” configuration to implement all the AODV routing protocols from the previous configuration. The first configuration in this step is to configure all the nodes, the source node and the base station of its transmitter interference range to 0 meters. Furthermore, the base station receiver interface ignores interference value is set to false. This is because the network model is purposely only wanting to interfere with the base station but no other intermediate nodes. The radio interferer number of applications is also being configured to 1 and it is also an “UdpBasicApp”. The destination addresses are also being configured into the base station. However, the port number used is different from the source node UDP application port number. This is to differentiate both the UDP packets. Other settings like message length and send interval are also being configured. The display interference range value also being set to true to display

the range of interference to have a better visualization. Finally, the base station also has a second UDP application that receives the UDP packet from port number 6000 which is from the radio interferer.

### 4.3 System Operation

This section will show the overall network simulation operation from the beginning to the end of the simulation. The process includes the AODV routing protocol set-up process, the packet travel process, and the final packet received by the base station.

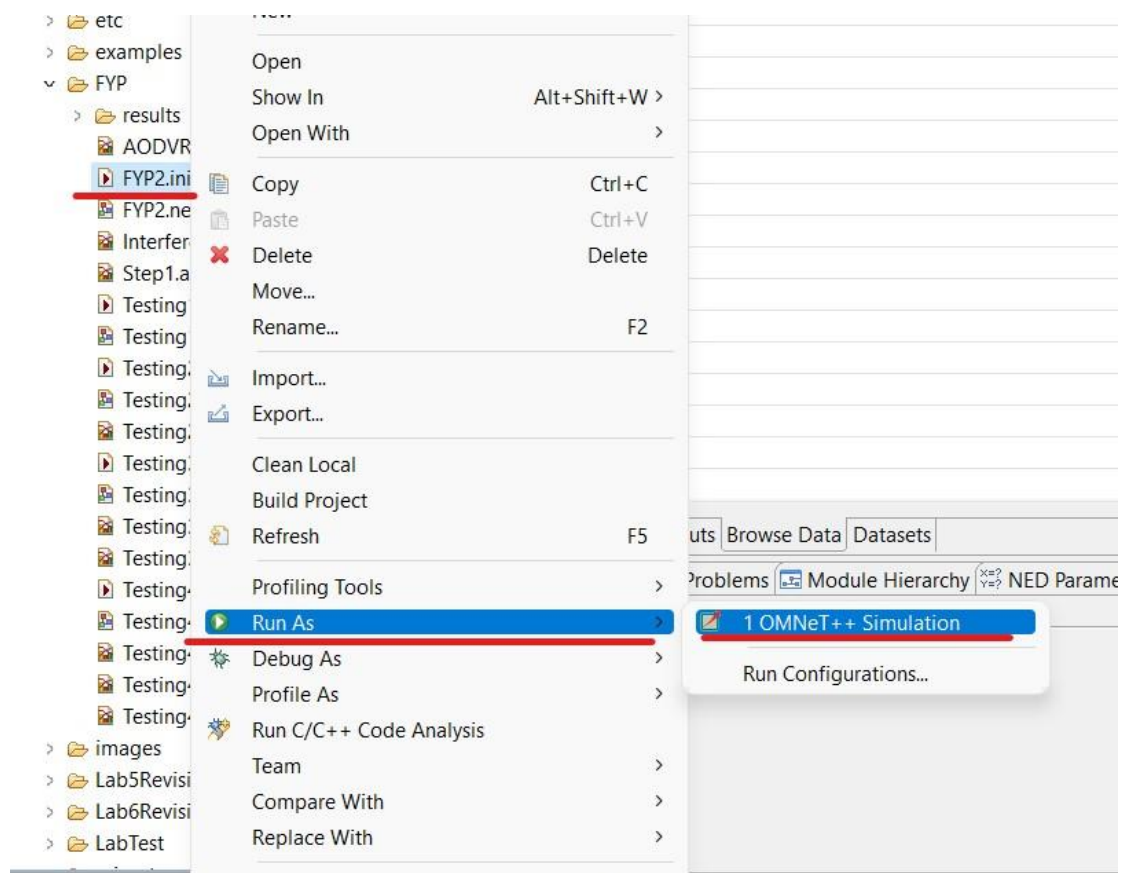


Figure 4-3-1

Figure 4-3-1 shows the step to run the network simulation. Users have to right-click on the FYP2.ini file, select Run As, and finally select the first option to run the network simulation.

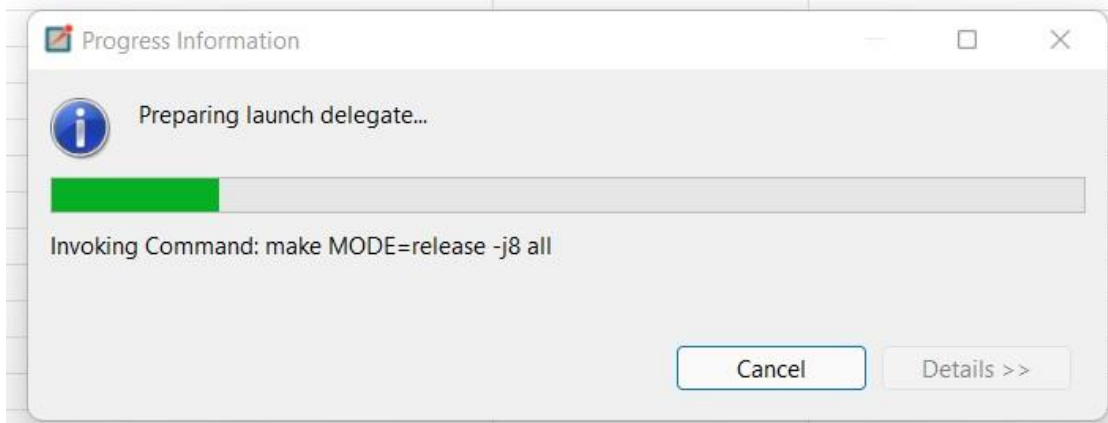


Figure 4-3-2

After users run the network simulation. A starting up process will begin, and it is shown in figure 4-3-2.

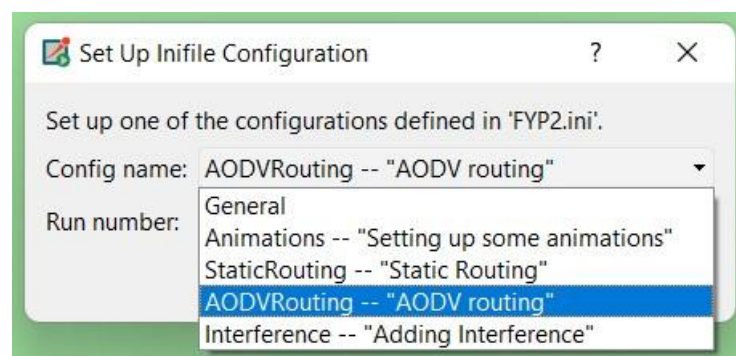


Figure 4-3-3

After the start-up process had completed, a new window will be prompted. As mentioned in the previous chapter, this project will be comparing two scenarios. The first scenario is the normal packet delivered from the source node to the base station without network interference. The second scenario is with network interference. Therefore, there are several options to choose from with the simulation. As shown in figure 4-3-3, there are “AODVRouting” and “Interference” options to choose from. In this project, it is recommended that the user selects the “AODVRouting” first, then the “interference” option to have a better comparison between these two options.

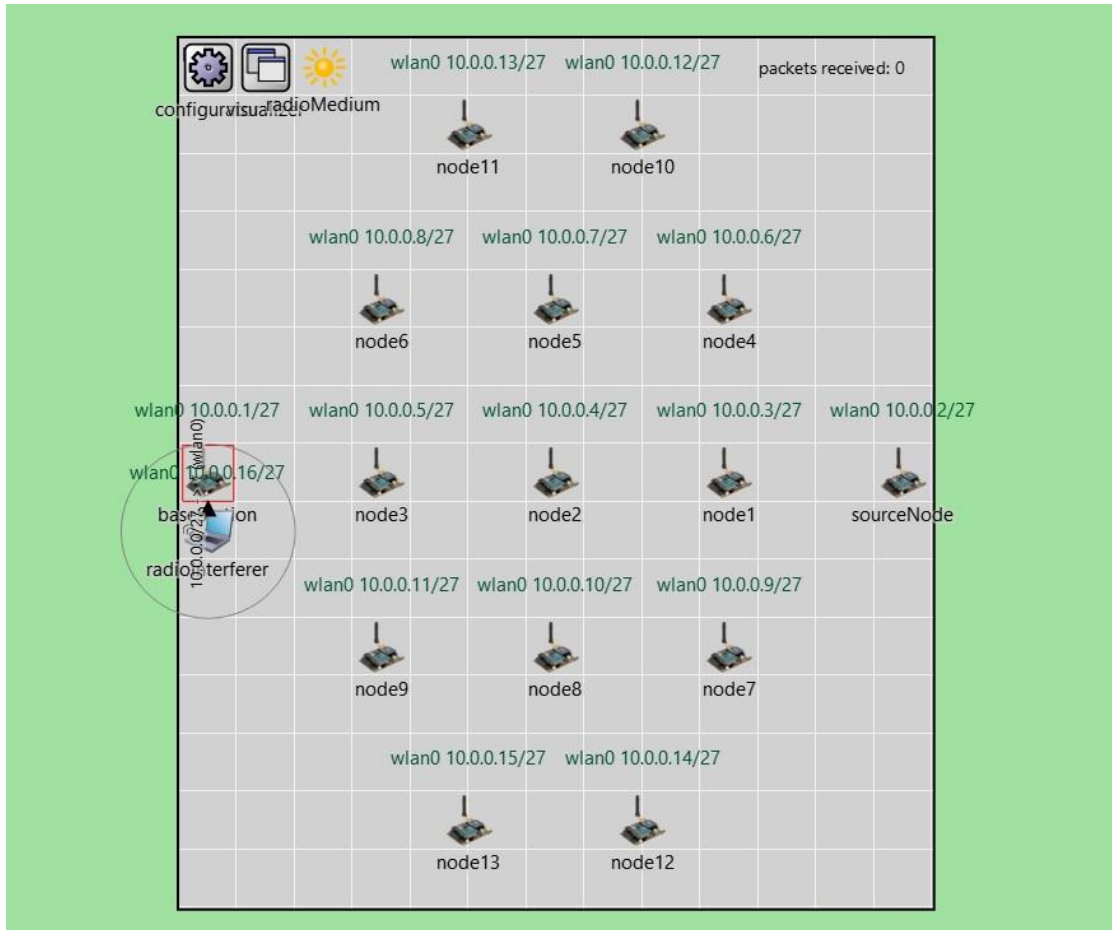


Figure 4-3-4

Figure 4-3-4 shows the initial network model configuration after the user selects one of the options from “AODVRouting” and “Interference”. One can observe that there is a total of 15 nodes including one source node and base station along with one radio interferer is being deployed in the network field. The source node and the base station are aligned in the middle line of the network. Node 1 to node 13 is the intermediate node that sits between the source node and the base station. There is also a radio interferer deployed just beside the base station to generate network interference and the interference range is shown.



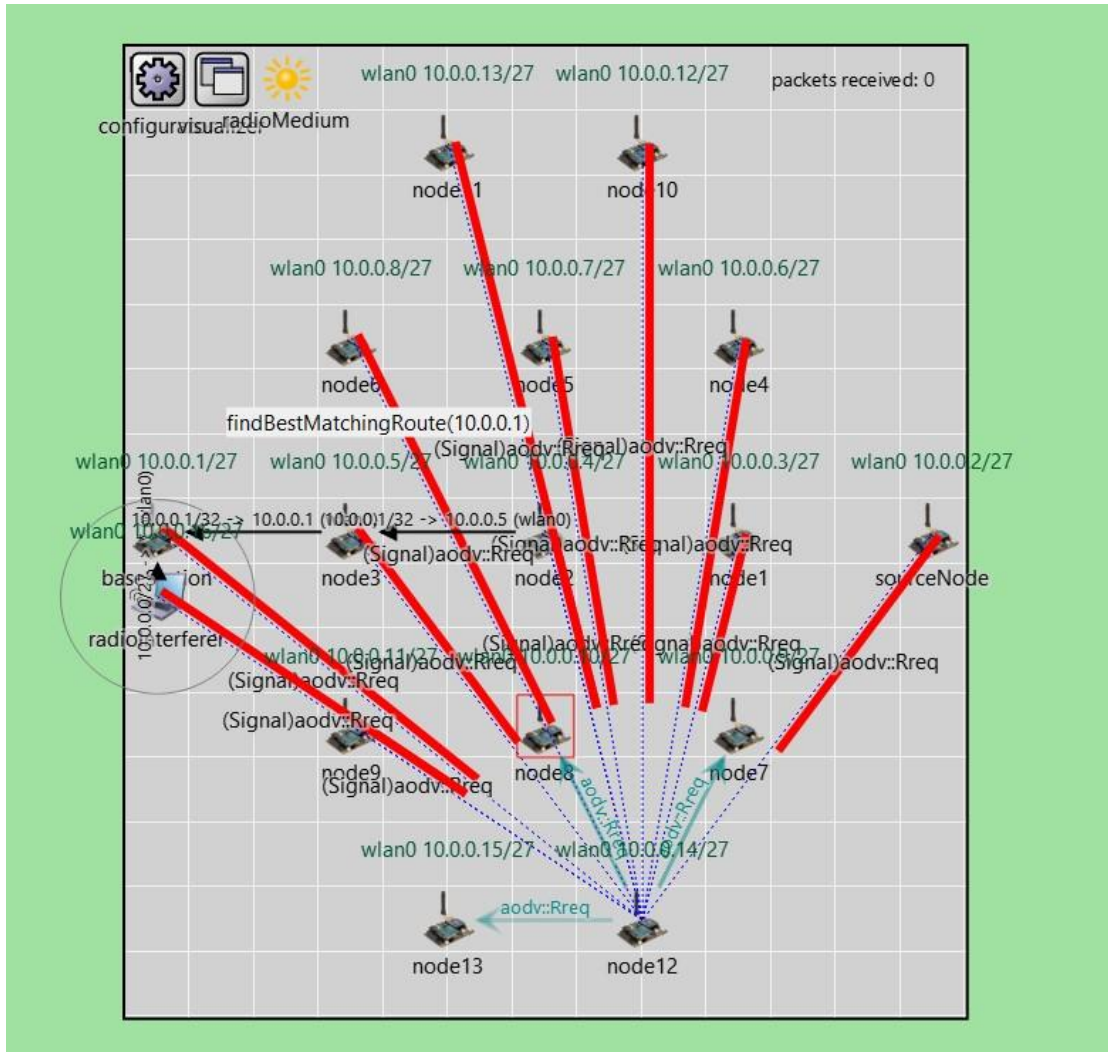


Figure 4-3-5

Figure 4-3-5 shows the AODV routing protocol is being set up no matter the users select which option as one is with network interference one is none. One can observe that some of the “aodv:Rreq” is being exchanged to find the least hop path from the source node to the base station. Some of the routes are also being discovered from node 3 to the base station and from node 2 to node 3.

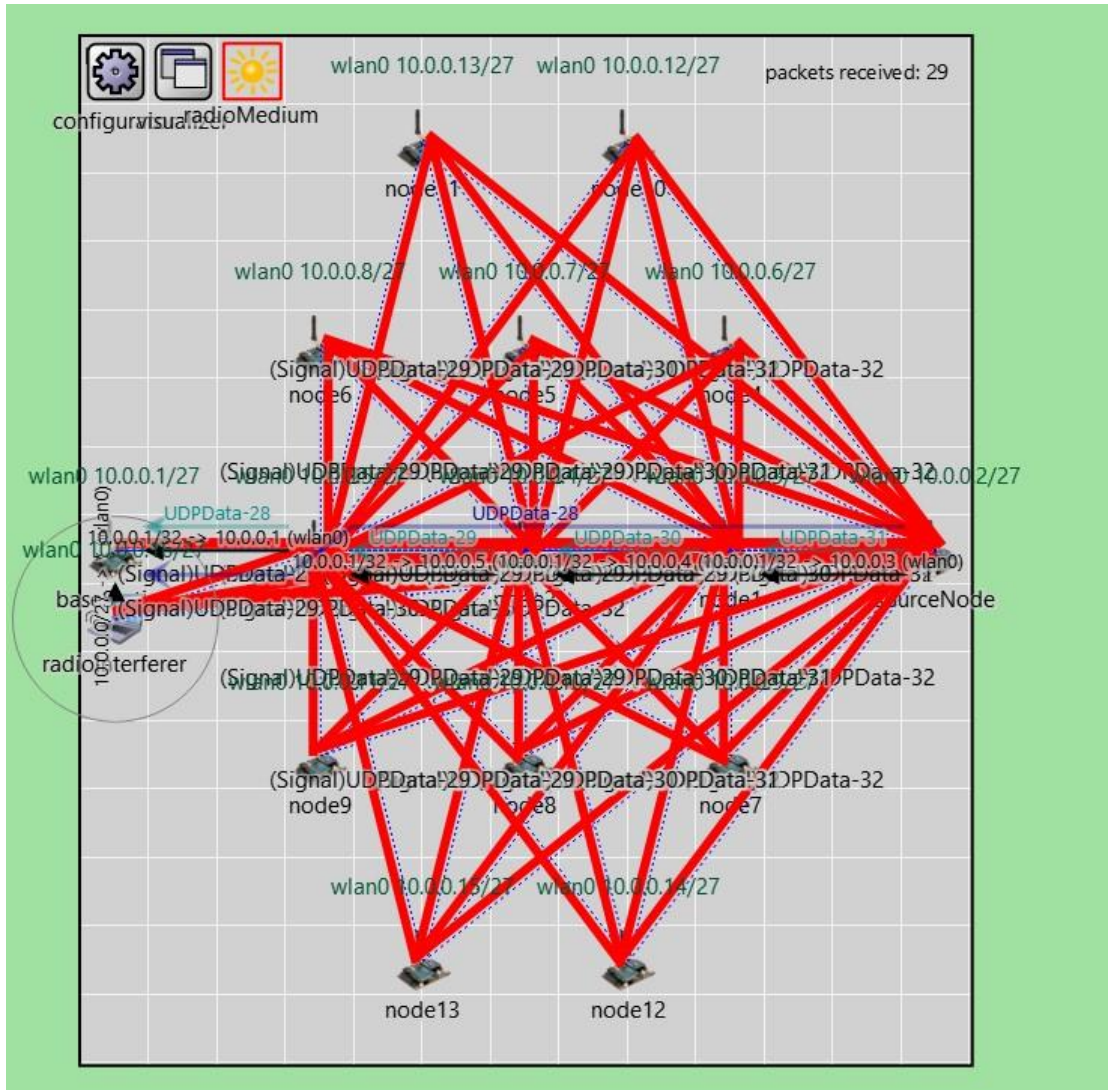


Figure 4-3-6

From figure 4-3-6, one can observe that the AODV routing protocol discovery process had been completed. The routing path from the source node to the base station has been fully discovered, which is the source node → node 1 → node 2 → node 3 → base station. The AODV routing protocol will choose this path because this is the least hop count from the source node to the base station. Since the routing path is being discovered, one can also observe that the UDP packet is being sent from the source node to the base station by utilizing the previously discovered path. Figure 4-3-3 shows the base station had received a total of 29 UDP packets.

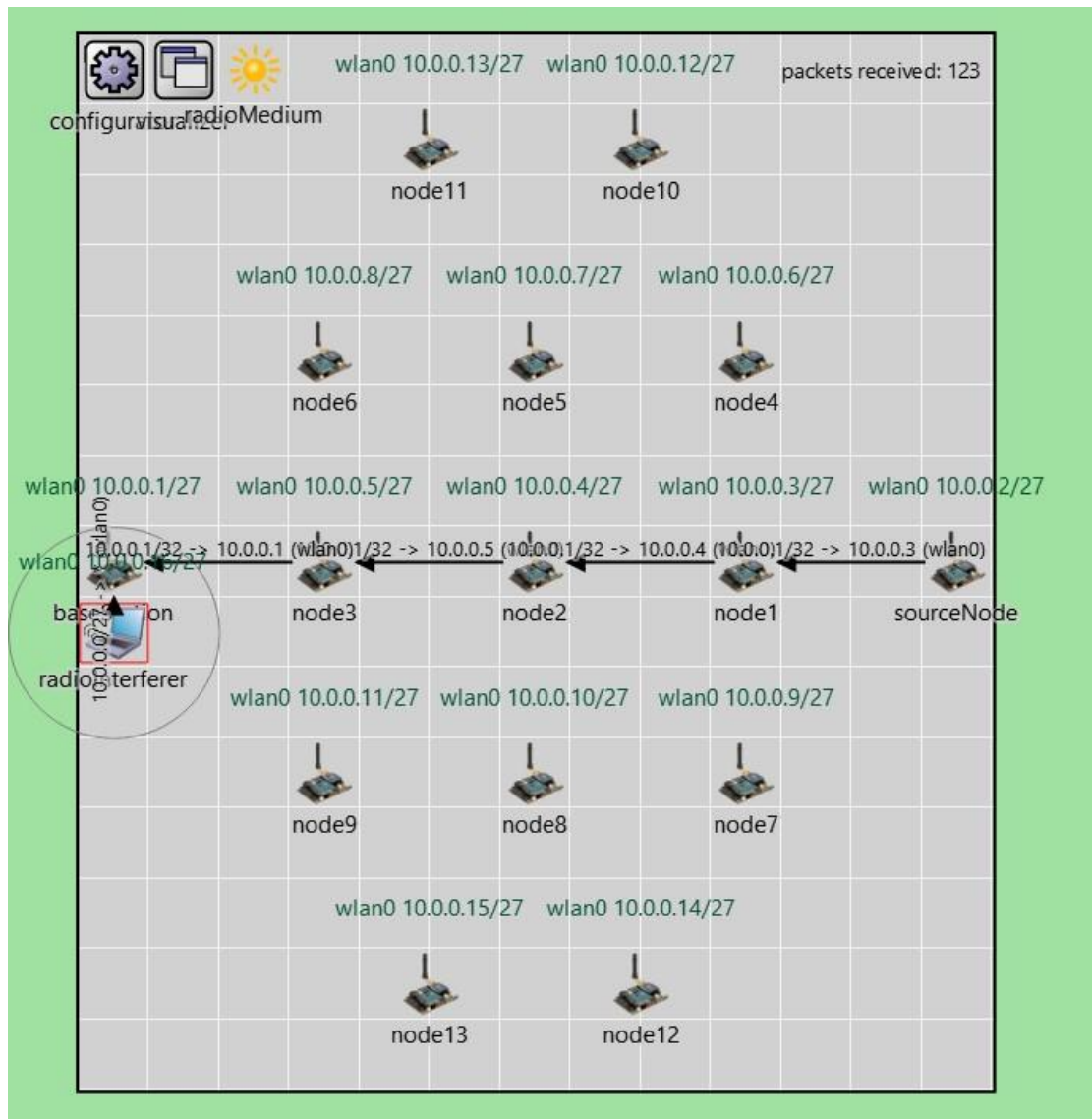


Figure 4-3-7

The simulation continues to run until it reaches the simulation time of 20 seconds. Figure 4-3-7 shows the simulation had reached the end of 20 seconds and stopped sending UDP packets from the source node to the base station. The total amount of packets received by the base station is 123 UDP packets that are sent from the source node. This number of packets received is the configuration with network interference. Figure 4-3-8 shows the number of packets received by the base station without network interference. The number of packets received by the base station without network interference is 1253 packets, it is way far more than the network with interference.

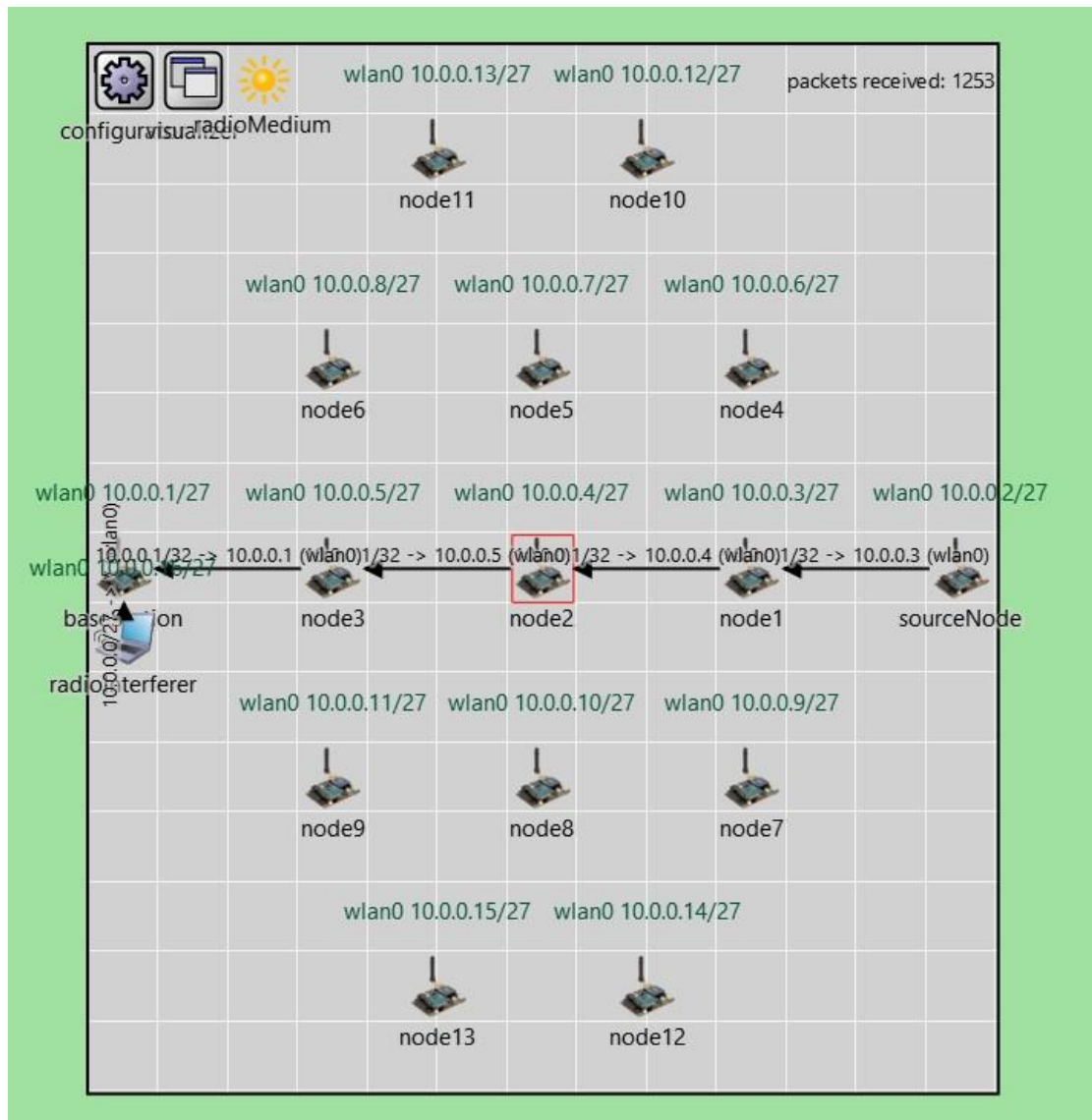


Figure 4-3-8

#### 4.4 Concluding Remark

In short, this chapter shows the process of using OMNeT++ including the installation process. Section 4.1, demonstrates the step-by-step installation guide of the network simulation software OMNeT++, which includes some figures to assist the installation process. Furthermore, also discusses the installation of the INET framework library into the OMNeT++ network simulator. In section 4.2, the overall network model configurations had been discussed including the network description file (NED) and INI file configuration. Finally, in section 4.3, this project network simulation run is explained. Chapter 5, will be discussing the evaluation of the network simulation, and some challenges of this project.

## CHAPTER 5

### System Evaluation and Discussion

This chapter presents the overall network simulation model observation and discussion. The network model observation and discussion will mainly discuss the simulation outcome and result from evaluation and discussion. Furthermore, this chapter also discusses the objectives evaluation and project challenges.

#### 5.1 Network Model Observation

This section will discuss the network simulation observation. It will be done in two parts, which are the observation without the network interference with only normal AODV routing; the observation with the network interference generated by the radio interferer with normal AODV routing. Both scenarios will be having the same conditions where the source node will send UDP packets to the base station at a random time interval using the exponential probability of 15ms, packet size is 500 bytes, and the destination port is 5000 and destined to the base station.

##### 5.1.1 Observation without Network Interference

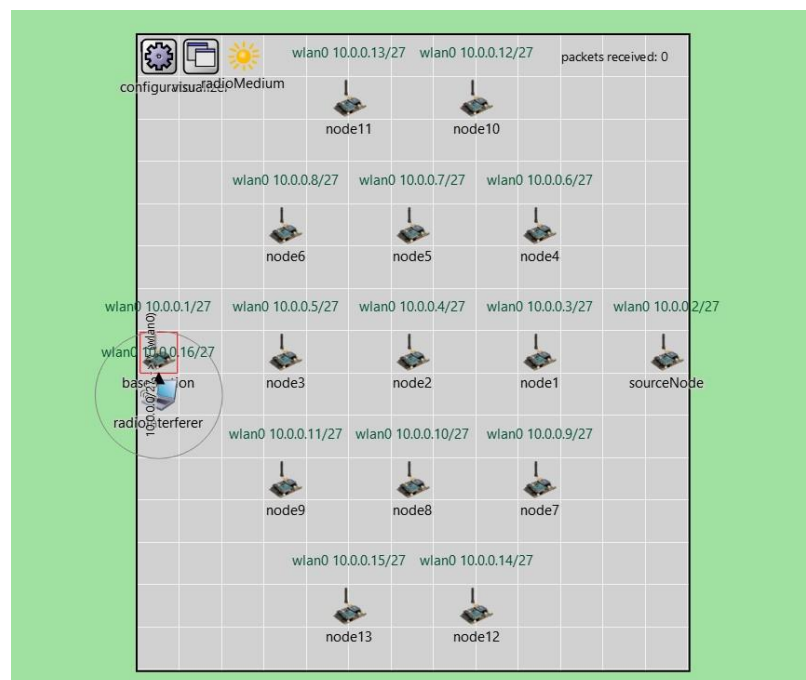


Figure 5-1-1

Figure 5-1-1 shows the initial network topology of the network simulation. There are 15 nodes in total and 1 radio interferer are deployed in the network. The network topology will not be changed as the nodes are immobile. In the total of 15 nodes, there is 1 source node that will be sending the packet, 1 base station that will keep receiving the packet, and 13 remaining nodes as the intermediate nodes between the source node and the base station. One can also observe that the statistics at the top right corner display 0 packets received initially.

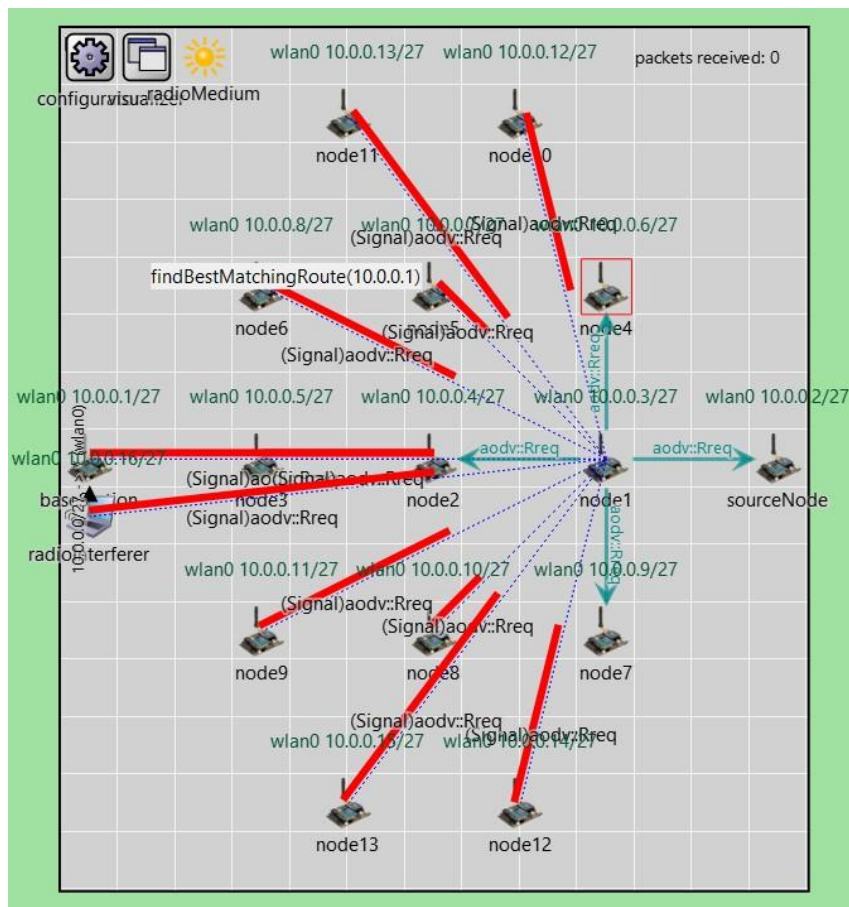


Figure 5-1-2

Figure 5-1-2 shows that after the beginning of network simulation. The AODV routing protocol starts to operate. It begins with sending the request packet to search for neighbour nodes to discover the route.

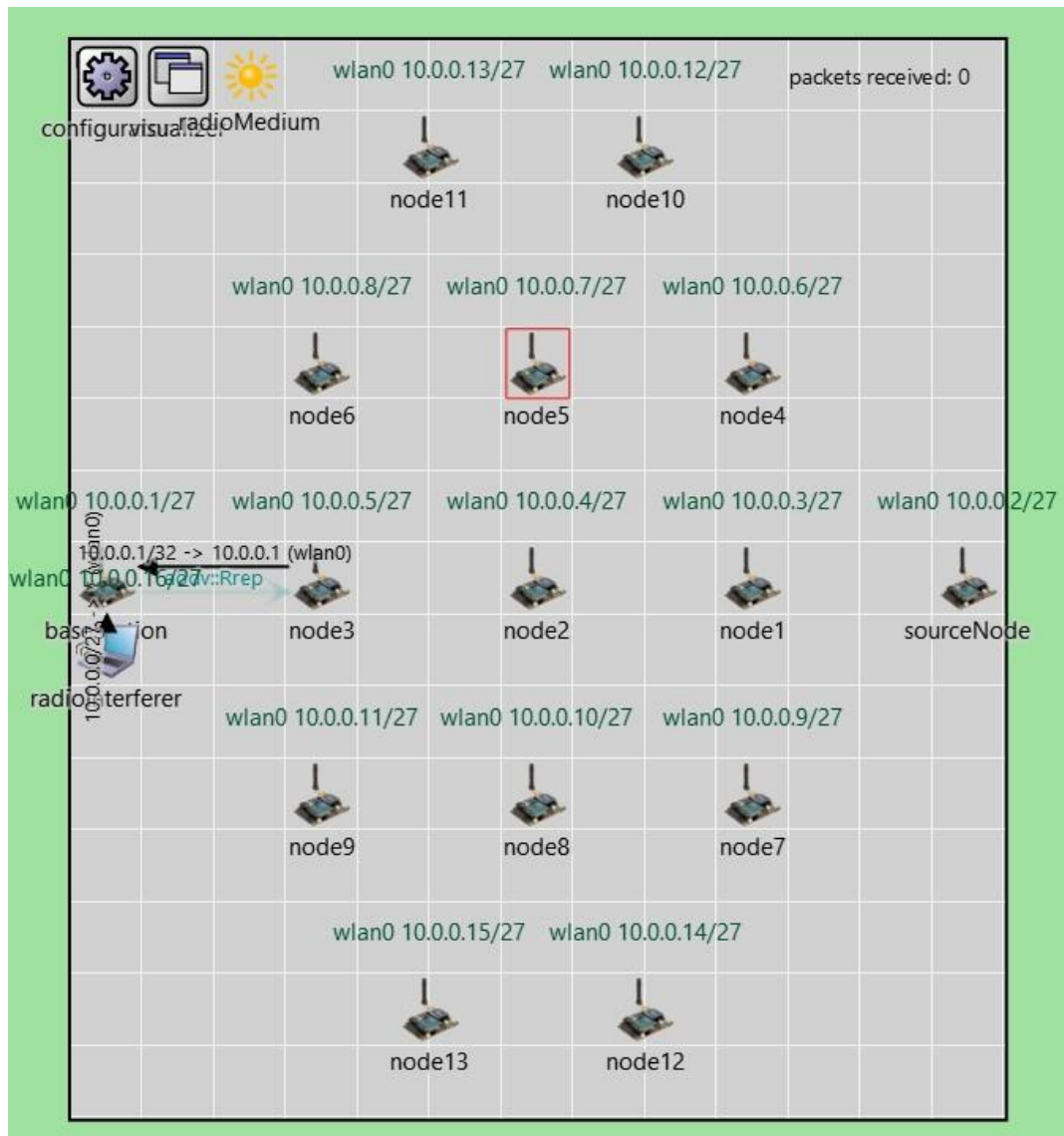


Figure 5-1-3

The route is starting to form after the AODV routing protocol packets exchange process as shown in figure 5-1-2. The first route that had been formed is shown in figure 5-1-3 where the route is from node 3 to the base station.

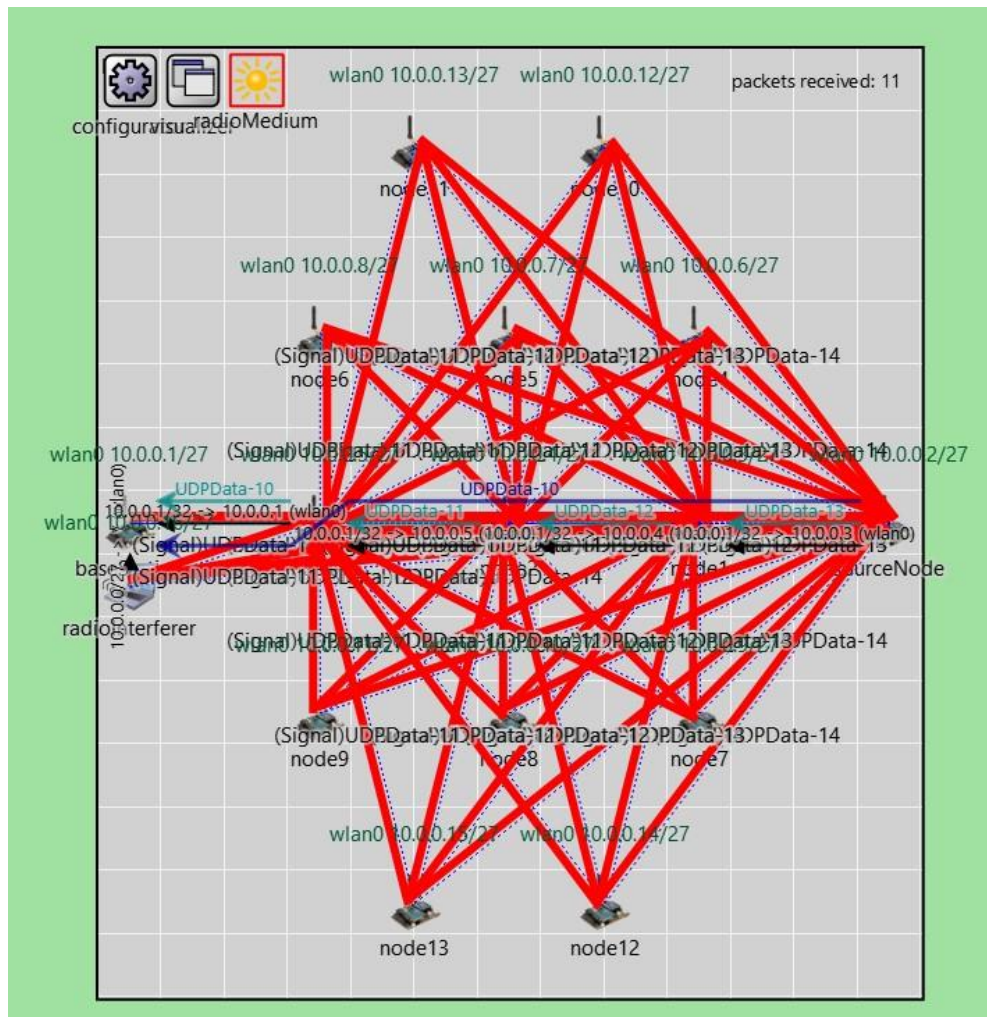


Figure 5-1-4

After some time, a route had been formed from the source node to the base station as shown in figure 5-1-4. The route is: sourceNode → node 1 → node 2 → node 3 → baseStation. The other observation from this figure 5-1-4 is that after the route had been formed successfully, the source node will start to send the UDP packet to the base station. The purple line indicates there is a packet named “UDPData-10” that had been traversed from the source node to the base station. One more observation is that the packet received statistic from the top right corner also increases as the base station receives the UDP packet.



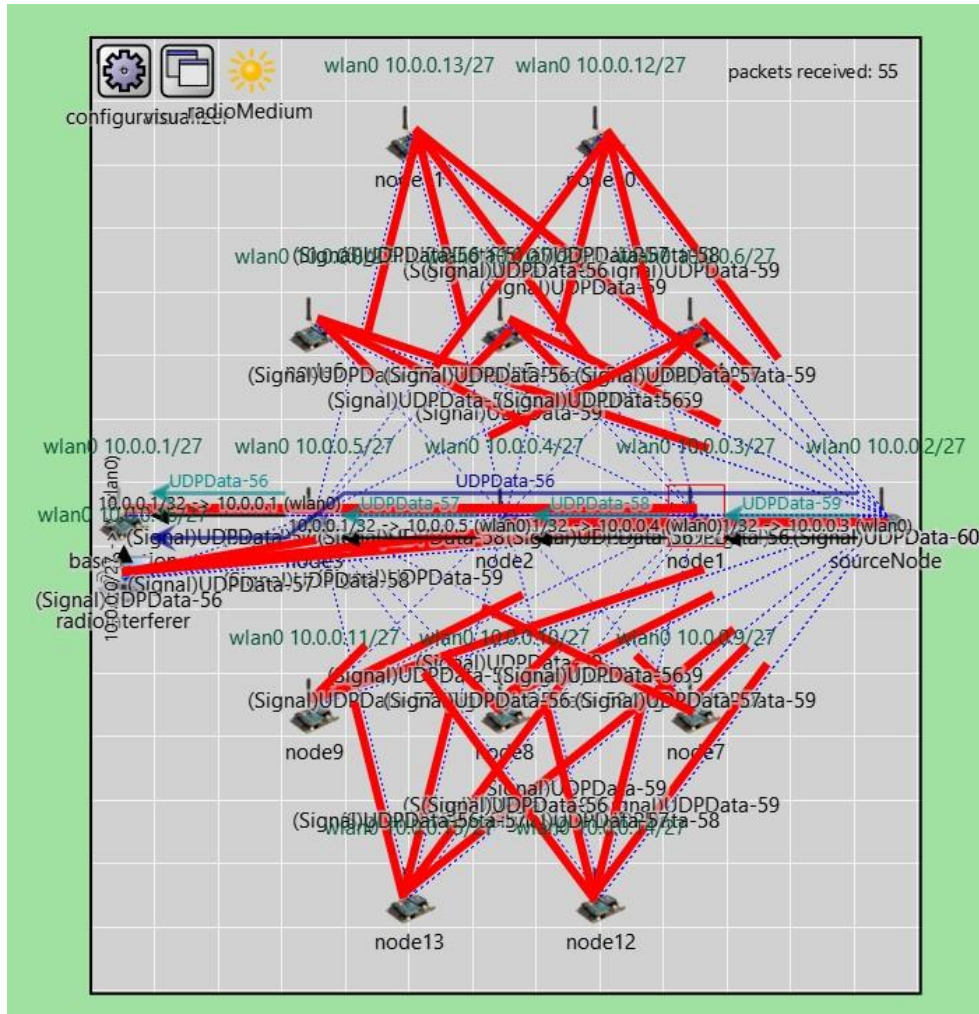


Figure 5-1-5

Figure 5-1-5 shows the simulation continues running. The UDP packet that had been received by the base station is currently 55 packets. Same as previous, the route that had been utilised is the same which is: sourceNode → node 1 → node 2 → node 3 → baseStation.

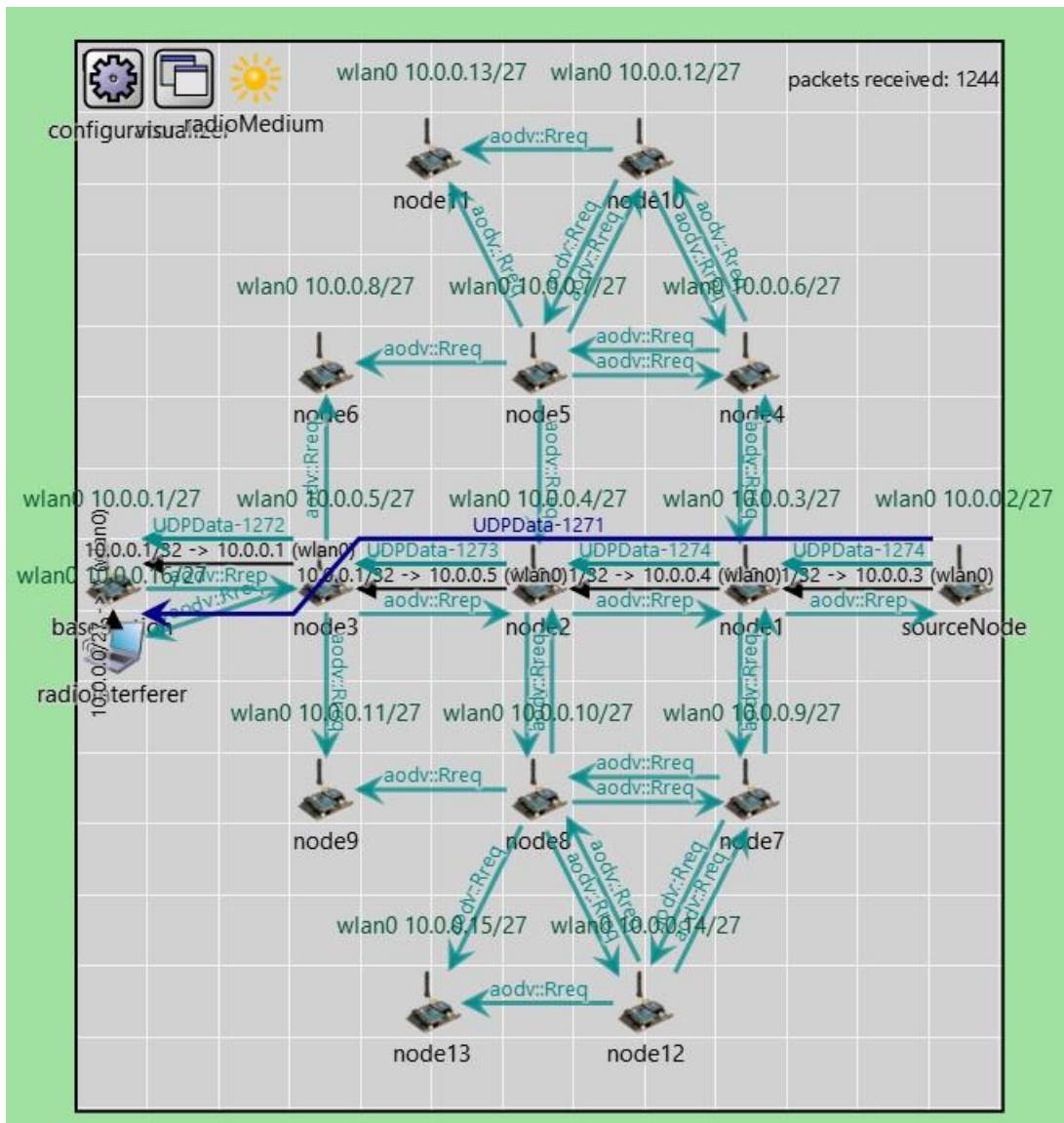


Figure 5-1-6

Finally, at the end of the simulation, figure 5-1-6 shows the final packet number that had been sent from the source node to the base station: “UDPData-1271”. The packet received statistics from the base station also increase to 1244 packets.

### 5.1.2 Observation with Network Interference

The observation with network interference is almost the same as at the beginning of the simulation. It is the same as from figure 5-1-1 until figure 5-1-3. The network simulation also will begin with the same process by setting up the routing path using the AODV routing protocol and exchanging the packets to find out the least hop count path. However, from figure 5-1-4 onward, there will be some slight difference.

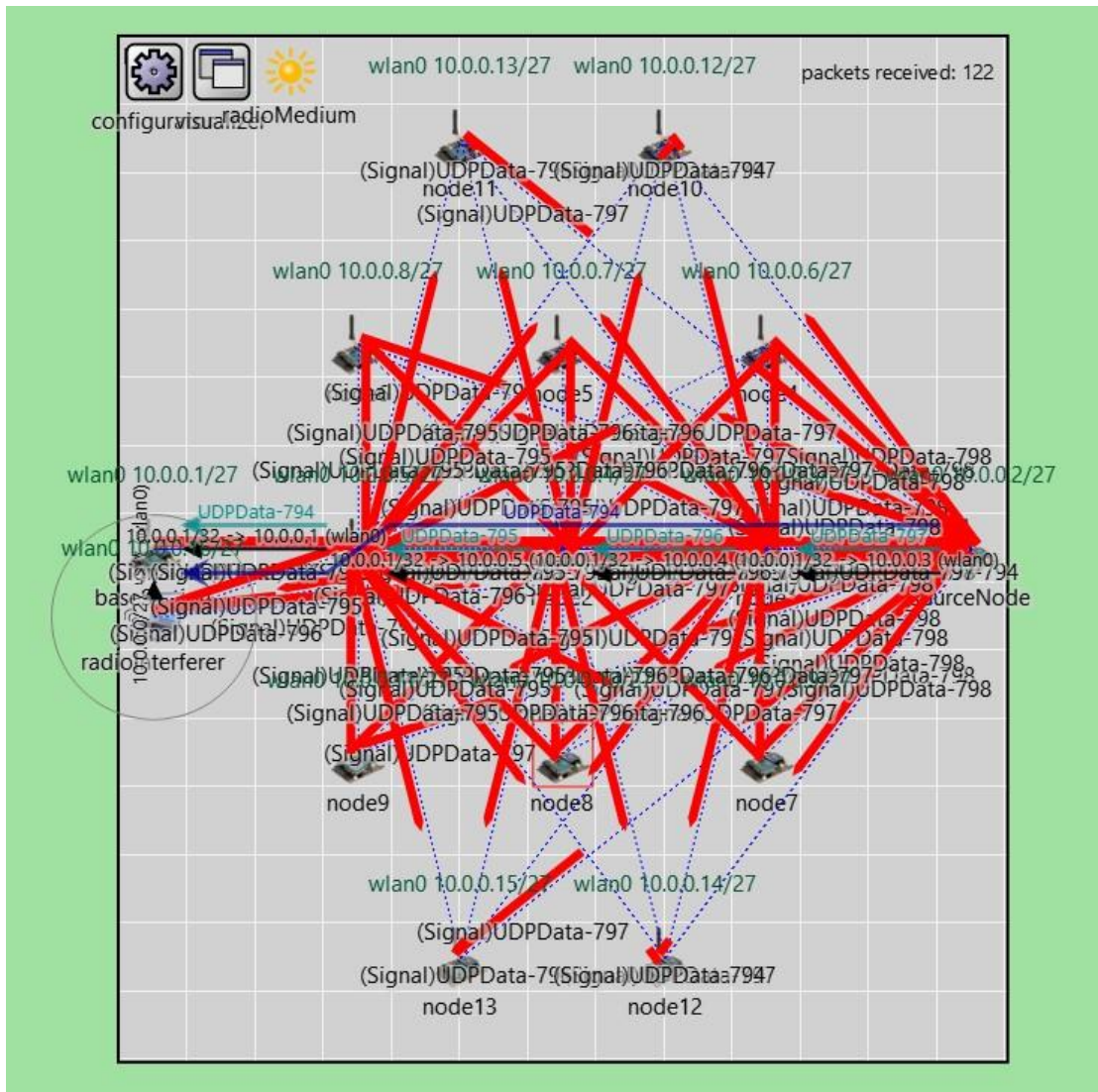


Figure 5-1-7

Figure 5-1-7 shows there is a difference between the network model with and without network interference. From the figure, one can observe that the source node already sending UDP packet number 794 (as purple line shown), however, the statistics packet received by the base station on the top right corner is only 122 packets received.

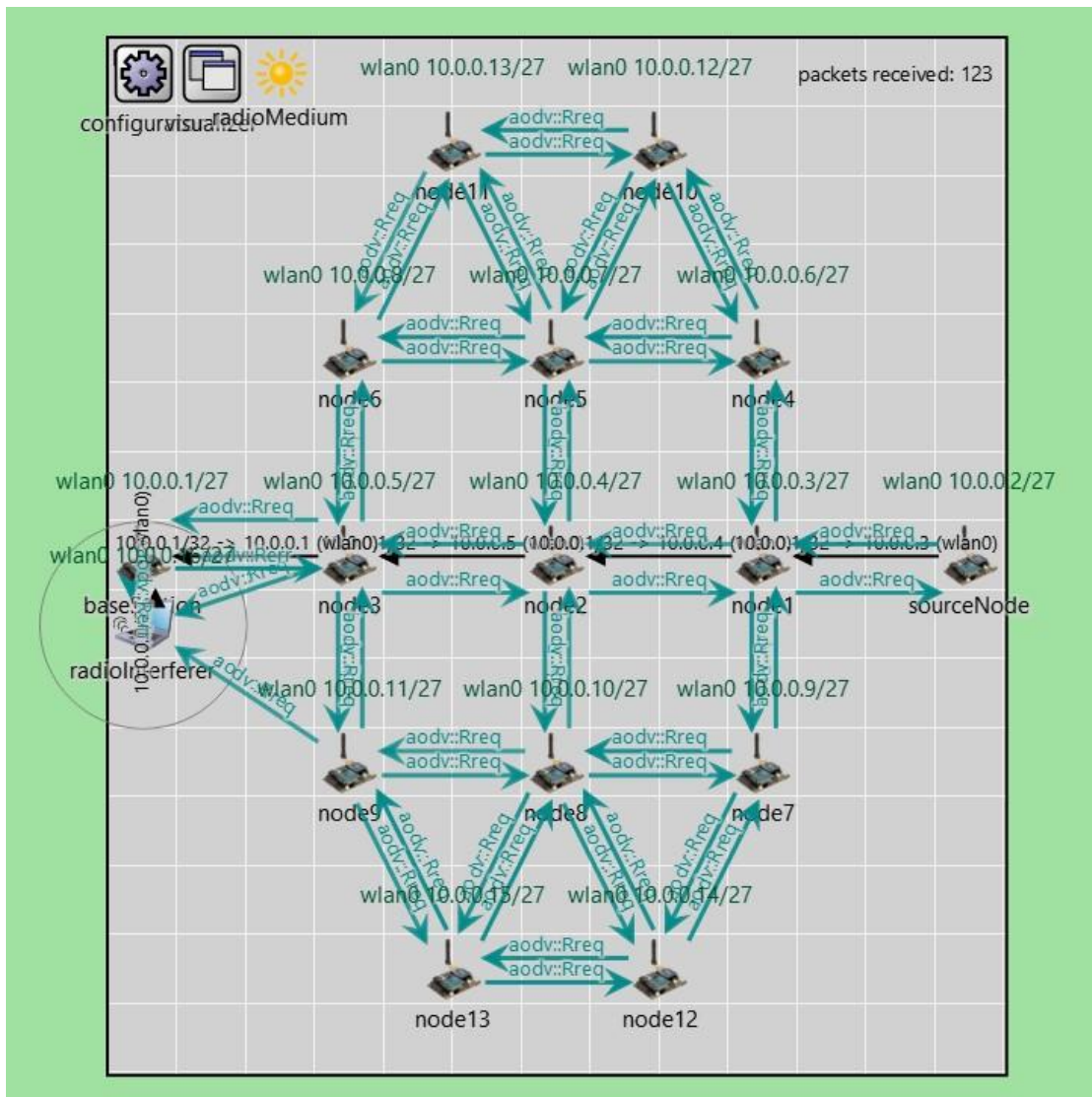


Figure 5-1-8

Figure 5-1-8 shows the end of the network simulation. The figure shows a huge difference compared to figure 5-1-6. One can observe that if the network interferes, the base station only will receive 123 packets from the source node. It is shown in the statistics from the top right corner of figure 5-1-8.

## 5.2 Network Model Discussion

This section will present the network simulation discussion. The discussion will carry out by discussing the result using some of the graphs and statistics to provide some useful information. Similar to chapter 5.1, the discussion will be separated into two parts, which are the discussion without the network interference, and the discussion

with network interference. Both scenarios also will be utilizing the same conditions such as the packet type, packet size and random time interval probability.

### 5.2.1 Discussion without Network Interference

This section will discuss the simulation result of the network without interference. From the previous observation, the network simulation will begin with the AODV routing protocol discovery. The AODV routing protocol must be set up in the beginning due to it must have a route from the source node to the base station to send the UDP packet. Once the AODV routing protocol is done discovering the least hop count path from the source node to the base station, the source node will start to send the UDP packet to the base station. The simulation will last 20 seconds, during this time, the source node will continuously send the UDP packet to the base station using a random time interval. The figure below shows the graph of the source node total sent packets (Figure 5-2-1) and base station total received packets (Figure 5-2-2).

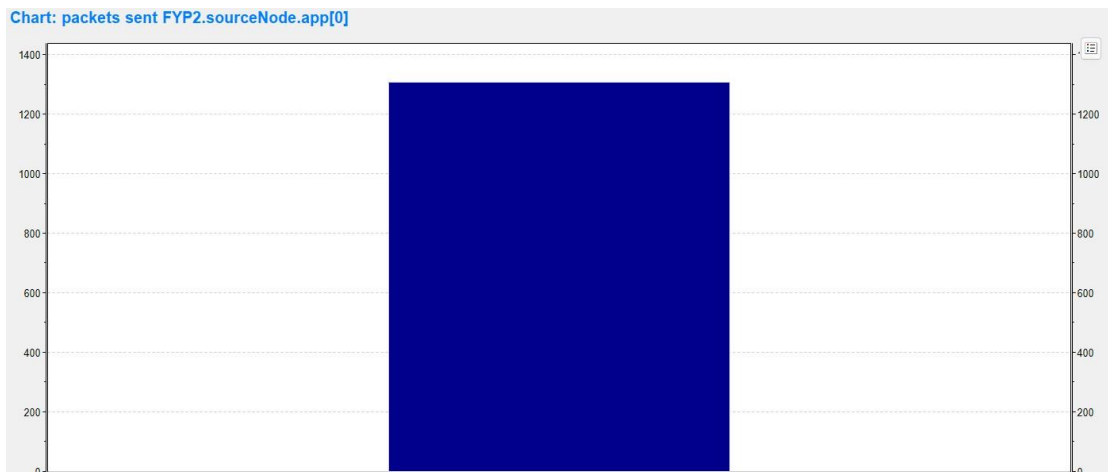


Figure 5-2-1

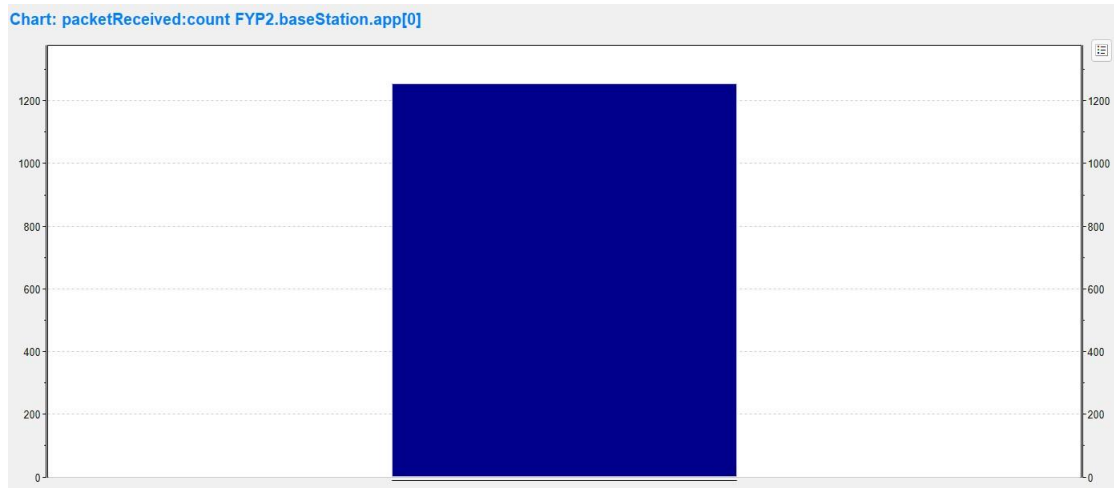


Figure 5-2-2

From both figures above, one can observe that the number of packets sent by the source node and the number of packets received by the base station is almost the same. This is because, in this situation, the network is without any interference from any source. All the packets sent by the source node are received successfully by the base station. Figure 5-2-3 also shows the number of packets dropped by the base station is 0, this again proves that all the packets sent are successfully received by the base station.

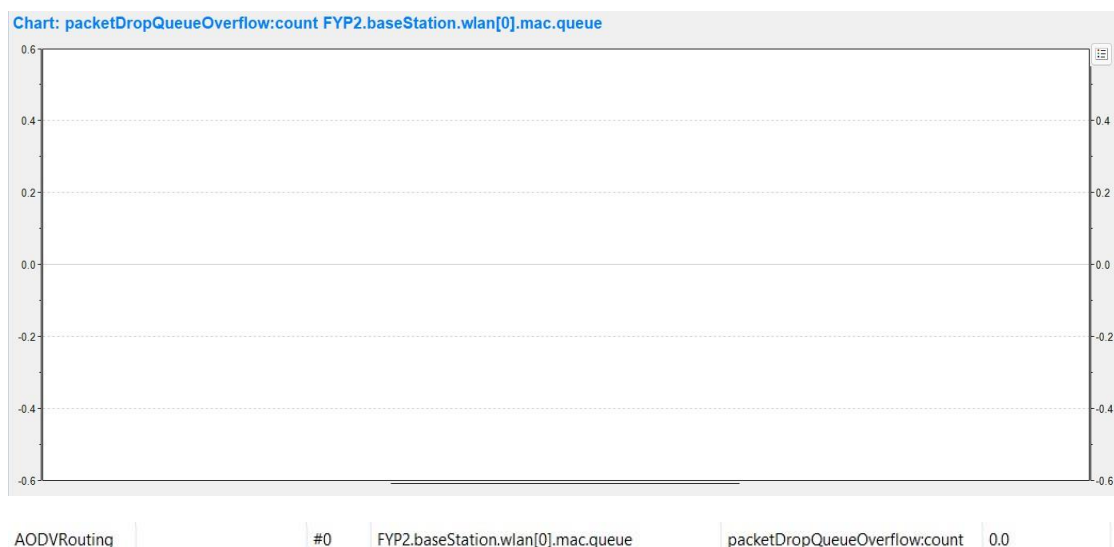


Figure 5-2-3

The network simulation without the network interference is considered successful as the result is as expected, which is no packet loss because of no interference. The next section will be discussing the network with interference.

### 5.2.2 Discussion with Network Interference

This section will be discussing the network model simulation with the network interference. It is the same as the previous section, the AODV routing protocol will first be set up as the source node must have a route to the base station to deliver the packet. The source node will also start sending the UDP packet to the base station once the AODV routing protocol is done. The radio interferer in this network model will also start working to interfere with the network to purposely cause some packet loss in the network. The figure below shows the graph of the source node total sent packets (Figure 5-2-4) and base station total received packets (Figure 5-2-5).



Figure 5-2-4

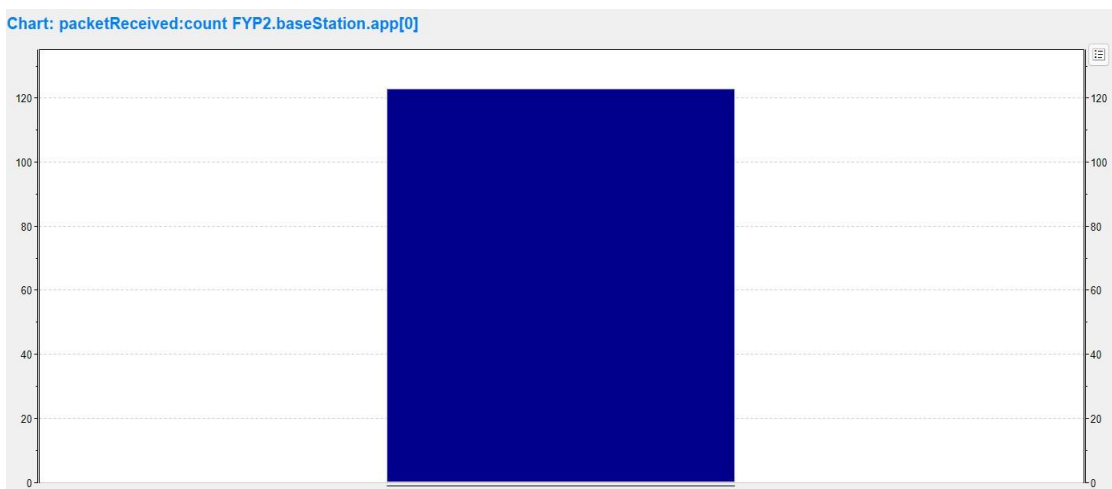


Figure 5-2-5

From both figures above, one can observe that the number of packets sent by the source node and the number of packets received by the base station is different, which in this case it is also different from the previous scenario where the network is without interference. This is because, in this situation, the network is interfered with by a radio interferer on the network. The radio interferer also sends a ton of packets towards the base station, which in this scenario the medium will be congested, and some packets might be dropped. The packet sent by the source node is around 1300 packets and the packet received by the base station is only around 120 packets. There are around 1000 packets dropped by the base station. Figure 5-2-6 show the number of packets dropped by the base station (around 14000 bytes); this shows why the packets received by the base station are that little.

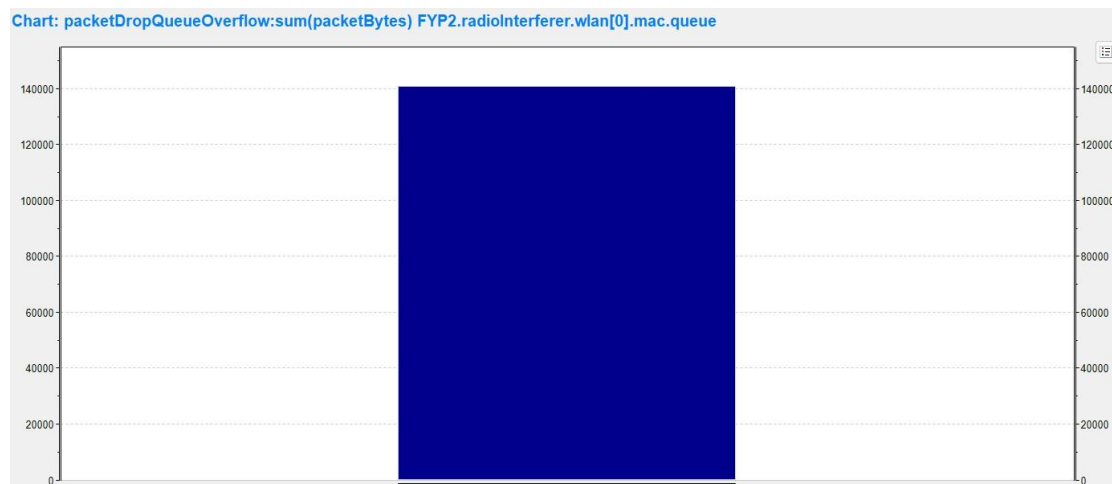


Figure 5-2-6

### 5.3 Objectives Evaluation

This project's objective was to propose a solution to identify or differentiate the actual reason for packet losses whether it is caused by network interference or malicious discarding of the source nodes. In chapter 3.2, a methodology had been proposed to achieve the objective. A short description of the proposed methodology is a methodology that had done some modifications to [10] detection of selective forwarding attack scheme. All the intermediate nodes between the source node and base station in the network model will generate an ACK packet and send it back to the source node indicating they had received the data packet generated by the source node and successfully forwarded the packet to the next-hop node. If the node is malicious, it will



not do the operation. Unfortunately, this project had only achieved only partial objectives. It was due to the difficulty of the implementation of the network simulation. The OMNeT++ network simulator and INET framework have a complex source code library, it is hard to modify the original library to modify the packet structure, packet receive or send operation and the generation of acknowledgement packet from the intermediate nodes. However, this project can generate the network interference on purpose to simulate the effect of natural network interference packet loss on the network. In short, this project is unable to achieve its objective due to the difficulty of complex source code and library however it can simulate the network interference to purposely generate network interference cause packet loss.

#### **5.4 Project Challenges**

This research project does consist of several challenges that hinder its progress of this project. The first challenge is that during the proposal stage of this project, the research of the useful material is difficult to find as there is not much relevant research work done on differentiating the packet loss due to network interference or malicious discard in wireless networks. However, there are also many related works had been found such as [6], [10], [12], [14], [15]. This research project utilized their works of them as a reference to come out with a possible solution to be able to differentiate the reason for packet loss is network interference or malicious discard. One of the works that had been contributing a lot to this project is the work of [10]. This project's proposed method is by utilizing the acknowledgement packet generated from the intermediate nodes when receiving a data packet from the source node and successfully forwarding it to the next-hop node. The source node will be based on this acknowledgement to determine whether there is a malicious node in the network. This project's main challenges are the implementation of the network simulation model. As mentioned, many times in the previous chapter, the network simulation will be using the OMNeT++ network simulator and INET framework library. Both of these tools have a complex library and source code to modify. It is hard to modify the source code to change the packet structure, packet receive or send operation, and the generation of acknowledgement packet from the intermediate nodes. However, the generation of the network interference by a radio interferer to purposely generate the packet loss had been done.

## CHAPTER 6

### Conclusion and Recommendation

This chapter presents the conclusion of this project, including what had been done during project 1 and project 2, the overall proposed method functionality and the objective achievement. The second sub-section will also discuss some of the recommendations of this project

#### 6.1 Conclusion

In a conclusion, this project studies the cause of packet loss that occurs in wireless sensor networks whether it is due to malicious discarding or network interference. Wireless Sensor Network (WSN) is a new technology that has a lot of potential for both civilian and military uses in the future. In WSN, the data packet generated by the source nodes is very important, it might contain some sensitive event information as in military applications the sensor nodes sense the movement of the opposing force and need to report to the base station immediately. Therefore, packet loss is not tolerable in WSN. However, the effectiveness of responses might be harmed if the actual reason is not identified. Furthermore, most intrusion detection systems (IDS) nowadays are only applicable for detecting packet loss but are unable to identify the reason for the loss. In short, this project seeks to come out with an analysing scheme that can identify the cause of packet loss in WSN due to malicious discarding or network interference. Furthermore, this project also aims to enhance the current IDS to have the capabilities to identify the reason for packet loss instead of detecting it.

In chapter 2, this project researched a few prior works to have a better understanding of the current trends on packet loss and IDS. [9] had proposed a general IDS architecture for WSN that consists of several entities including local agent, global agent, etc. Furthermore, both [10] and [11] had proposed their detection and countermeasure for selective forwarding attack in WSN which is a lightweight multi-hop acknowledgement-based detection scheme and multi-dataflow topologies (MDT) scheme respectively. Finally, [12] and [13] had done the considering packet loss in a WSN using the LEACH routing protocol as an influence. and QoS parameters performance analysis in WSN using AODV routing protocol respectively. They had shown very detailed simulation results based on their studies.

Besides that, chapter 3 discussed the proposed method and general work procedures of this project. First, the network simulation model had been discussed such as the traffic, nodes configuration, network setup, etc. Second, the methodologies that were used to achieve the objective of this project had been discussed. This project used the detection scheme of [10] as a reference to produce the methodologies such as the generation of the ACK packet. In short, the methodology proposed is by using the ACK packet and the generation of alarm packet to differentiate whether the packet loss is due to malicious discard or network issues. Next, the implementation issues and challenges had been discussed. The issues such as the malicious node and radio interferer configuration as the network simulation tool capabilities are limited. The challenges will be the OMNeT++ network simulator is using a different language which is the Network Description (NED) language, therefore it took some time to familiarise with it and its library required to build this project. Finally, the Gantt chart had been shown that reflects the workflow of this project.

Chapter 4 discussed the simulation process. The first section discusses the software setup. The software setup is about the OMNeT++ network simulator installation guidelines which is a C++ simulation toolkit and framework that is flexible, modular, and component-based and is mostly used to create network simulators [5]. It contains a lot of frameworks that can be used to build up the environment without the users manually adding in all the modules from scratch. Furthermore, also discusses the INET framework to be installed into the OMNeT++ network simulator as this project requires the INET framework library. Next, the network model configurations are discussed. It contains the network description file (NED) and the INI file configuration to set up the network simulation. Finally, the whole network simulation operation is shown to guide the users of the whole simulation process. The network simulation process includes the configuration with the network interference or without it.

Chapter 5 presents the simulation outcome discussion and evaluation of the result. The network model discussion includes the network model observation and network model discussion of the simulation result. Furthermore, this chapter also discusses the objective achievement and the project challenges and obstacles.

Unfortunately, this research project was unable to achieve its objective due to the difficulty of the implementation of the network model. However, this project had proposed a possible methodology that can achieve the project objective theoretically. Finally, this project consists of several challenges such as there is a lack of information that is relevant to the topic of this project, and everything must start from scratch. However, there are still some of the works that were able to use as a reference to propose the methodology to achieve the project objective.

## **6.2 Recommendation**

This project does consist of several challenges while doing the research work and implementing the entire network model. Some recommendations that can be given to provide an easier way to work on this topic in future work is that one can utilize a different network simulator to build the network model. It is because as mentioned many times in the previous chapter, the current network simulator OMNeT++ and its INET framework library have a strong backbone source code. It is hard to modify the source code to change the packet structure such as by adding some fields to the existing packet format. It is also difficult for someone to modify the packet sent or receive operation as the intermediate nodes in the proposed methodology must forward the data packet to its next-hop node and generate the acknowledgement packet to be sent back to the source node. One can also modify the methodology to make it more simplified, therefore the implementation process can be slightly easier. For now, the proposed methodology requires all the intermediate nodes to acknowledge the data packet sent from the source node. This action might be overwhelming the nodes as it needs to have a large processing power to complete the task as the intermediate nodes also need to forward the data packet to its next-hop node. One final recommendation to provide is that for the network interference, one can use a natural environment to simulate network interference. In this project, the network interference is generated by using a wireless host that also keeps sending packets to the destination base station to interrupt the medium to cause the base station to overwhelm packets. The natural network interference can be rain or microwave that can cause the signal to multipath, reflect, diffraction or scatter. The natural network interference can be easier to be implemented in the network simulation application as it does not require much configuration.

**REFERENCES**

- [1] A. S. K. Pathan, H. W. Lee, and C. S. Hong, "Security in Wireless Sensor Networks: Issues and challenges," in *8th International Conference Advanced Communication Technology, ICACT 2006 - Proceedings*, 2006, vol. 2, pp. 1043–1048. doi: 10.1109/icact.2006.206151.
- [2] B. Shebaro, D. Midi, and E. Bertino, "Fine-grained analysis of packet loss symptoms in wireless sensor networks," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems - SenSys '13*, 2013, pp. 1–2. doi: 10.1145/2517351.2517408.
- [3] S. Sultana, G. Ghinita, E. Bertino, and M. Shehab, "A lightweight secure provenance scheme for wireless sensor networks," in *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, 2012, pp. 101–108. doi: 10.1109/ICPADS.2012.24.
- [4] I. Krontiris, T. Dimitriou, and F. C. Freiling, "Towards Intrusion Detection in Wireless Sensor Networks," 2007.
- [5] "What is OMNeT++?," *OpenSim Ltd.*, 2019. <https://omnetpp.org/intro/> (accessed Apr. 05, 2022).
- [6] C. Tumrongwittayapak and R. Varakulsiripunth, "Detecting sinkhole attack and selective forwarding attack in wireless sensor networks," 2009. doi: 10.1109/ICICS.2009.5397594.
- [7] K. Eghonghon Ukhurebor, I. Odesanya, S. Soo Tyokighir, R. George Kerry, A. Samson Olayinka, and A. Oluwafemi Bobadoye, "Wireless Sensor Networks: Applications and Challenges," in *Wireless Sensor Networks - Design, Deployment and Applications*, IntechOpen, 2021. doi: 10.5772/intechopen.93660.
- [8] M. A. Matin and M. M. Islam, "Overview of Wireless Sensor Network," in *Wireless Sensor Networks - Technology and Protocols*, InTech, 2012. doi: 10.5772/49376.
- [9] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *2006 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006*, 2006, vol. 1, pp. 640–644. doi: 10.1109/CCNC.2006.1593102.

## REFERENCES

- [10] Y. Bo and X. Bin, "Detecting selective forwarding attacks in wireless sensor networks," in *20th International Parallel and Distributed Processing Symposium, IPDPS 2006*, 2006, vol. 2006. doi: 10.1109/IPDPS.2006.1639675.
- [11] Hung-Min Sun, Chien-Ming Chen, and Ying-Chu Hsiao, "An efficient countermeasure to the selective forwarding attack in wireless sensor networks," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, Oct. 2007, pp. 1–4. doi: 10.1109/TENCON.2007.4428866.
- [12] C. Cirstea, M. Cernaianu, and A. Gontean, "Packet loss analysis in wireless sensor networks routing protocols," in *2012 35th International Conference on Telecommunications and Signal Processing, TSP 2012 - Proceedings*, 2012, pp. 37–41. doi: 10.1109/TSP.2012.6256248.
- [13] N. Kothawade, A. Biradar, K. Kodmelwar, K. P. Tambe, and V. Deshpande, "Performance Analysis of Wireless Sensor Network by Varying Reporting Rate," *Indian Journal of Science and Technology*, vol. 9, no. 26, Jul. 2016, doi: 10.17485/ijst/2016/v9i26/91906.
- [14] Z. He and T. Voigt, "Precise packet loss pattern generation by intentional interference," 2011. doi: 10.1109/DCOSS.2011.5982225.
- [15] D. Huijuan, S. Xingming, W. Baowei, and C. Yuanfu, "Selective forwarding attack detection using watermark in WSNs," in *2009 Second ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM 2009*, 2009, vol. 3, pp. 109–113. doi: 10.1109/CCCM.2009.5268016.

## APPENDIX

### Appendix A – Network Description File (NED)

```

import inet.node.inet.INetworkNode;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.visualizer.integrated.IntegratedCanvasVisualizer;
import inet.physicallayer.unitdisk.UnitDiskRadioMedium;

network FYP2
{
    parameters:
        @display("bgb=1300,1500;bgg=100,1,grey95");
        @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

        @figure[recvPkText](type=indicatorText; pos=1000,50; anchor=w;
font=,18; textFormat="packets received: %g"; initialValue=0);

@statistic[packetReceived](source=baseStation.app[0].packetReceived;
record=figure(count); targetFigure=recvPkText);

    submodules:
        baseStation: <default("WirelessHost")> like INetworkNode {
            @display("p=50,750;i=misc/sensor2");
        }
        sourceNode: <default("SensorNode")> like INetworkNode {
            @display("p=1250,750;i=misc/sensor2");
        }
        node1: <default("SensorNode")> like INetworkNode {
            @display("p=950,750;i=misc/sensor2");
        }
        node2: <default("SensorNode")> like INetworkNode {
            @display("p=650,750;i=misc/sensor2");
        }
        node3: <default("SensorNode")> like INetworkNode {
            @display("p=350,750;i=misc/sensor2");
        }
        node4: <default("SensorNode")> like INetworkNode {
            @display("p=950,450;i=misc/sensor2");
        }
        node5: <default("SensorNode")> like INetworkNode {
            @display("p=650,450;i=misc/sensor2");
        }
        node6: <default("SensorNode")> like INetworkNode {
            @display("p=350,450;i=misc/sensor2");
        }
        node7: <default("SensorNode")> like INetworkNode {
            @display("p=950,1050;i=misc/sensor2");
        }
        node8: <default("SensorNode")> like INetworkNode {
            @display("p=650,1050;i=misc/sensor2");
        }
        node9: <default("SensorNode")> like INetworkNode {
            @display("p=350,1050;i=misc/sensor2");
        }
        node10: <default("SensorNode")> like INetworkNode {
            @display("p=800,150;i=misc/sensor2");
        }
        node11: <default("SensorNode")> like INetworkNode {

```

```
        @display("p=500,150;i=misc/sensor2");
    }
    node12: <default("SensorNode")> like INetworkNode {
        @display("p=800,1350;i=misc/sensor2");
    }
    node13: <default("SensorNode")> like INetworkNode {
        @display("p=500,1350;i=misc/sensor2");
    }
    radioInterferer: <default("WirelessHost")> like INetworkNode {
        @display("p=50,850");
    }
    configurator: Ipv4NetworkConfigurator {
        @display("p=50,50");
    }
    visualizer: IntegratedCanvasVisualizer {
        @display("p=150,50");
    }
    radioMedium: UnitDiskRadioMedium {
        @display("p=250,50");
    }
}
```



**Appendix B – INI File**

```

[General]
network = FYP2
sim-time-limit = 20s

*.node*.ipv4.arp.typename = "GlobalArp"
*.sourceNode.ipv4.arp.typename = "GlobalArp"
*.baseStation.ipv4.arp.typename = "GlobalArp"
*.radioInterferer.ipv4.arp.typename = "GlobalArp"

*.sourceNode.numApps = 1
*.sourceNode.app[0].typename = "UdpBasicApp"
*.sourceNode.app[0].destAddresses = "baseStation"
*.sourceNode.app[0].destPort = 5000
*.sourceNode.app[0].messageLength = 500B
*.sourceNode.app[0].sendInterval = exponential(15ms)
*.sourceNode.app[0].packetName = "UDPData"

*.baseStation.numApps = 1
*.baseStation.app[0].typename = "UdpSink"
*.baseStation.app[0].localPort = 5000

*.node*.wlan[0].typename = "AckingWirelessInterface"
*.node*.wlan[0].radio.transmitter.communicationRange = 400m
*.node*.wlan[0].radio.receiver.ignoreInterference = true
*.node*.wlan[0].mac.headerLength = 23B

*.sourceNode.wlan[0].typename = "AckingWirelessInterface"
*.sourceNode.wlan[0].radio.transmitter.communicationRange = 400m
*.sourceNode.wlan[0].radio.receiver.ignoreInterference = true
*.sourceNode.wlan[0].mac.headerLength = 23B

*.baseStation.wlan[0].typename = "AckingWirelessInterface"
*.baseStation.wlan[0].radio.transmitter.communicationRange = 400m
*.baseStation.wlan[0].radio.receiver.ignoreInterference = true
*.baseStation.wlan[0].mac.headerLength = 23B

*.radioInterferer.wlan[0].typename = "AckingWirelessInterface"
*.radioInterferer.wlan[0].radio.transmitter.communicationRange = 150m
*.radioInterferer.wlan[0].radio.transmitter.interferenceRange = 150m

*.node*.**.bitrate = 1Mbps
*.sourceNode**.bitrate = 1Mbps
*.baseStation**.bitrate = 1Mbps
*.radioInterferer**.bitrate = 1Mbps

#-----
---
[Config Animations]
description = "Setting up some animations"

#Configurator
*.configurator.dumpAddresses = true
*.configurator.dumpTopology = true
*.configurator.dumpLinks = true
*.configurator.dumpRoutes = true

*.visualizer.dataLinkVisualizer.displayLinks = true

```

## APPENDIX

```
*.visualizer.dataLinkVisualizer.packetFilter = "UDPData*"
*.visualizer.networkRouteVisualizer.displayRoutes = true
*.visualizer.networkRouteVisualizer.packetFilter = "UDPData*"
*.visualizer.interfaceTableVisualizer.displayInterfaceTables = true

#-----
---
[Config StaticRouting]
description = "Static Routing"
extends = Animations

*.node*.forwarding = true
*.sourceNode.forwarding = true
*.baseStation.forwarding = true

*.configurator.config = xml("<config><interface hosts='*'
address='10.0.0.x' netmask='255.255.255.0' /><autoroute
metric='errorRate' /></config>")
*.configurator.optimizeRoutes = false

*.node*.ipv4.routingTable.netmaskRoutes = ""
*.sourceNode.ipv4.routingTable.netmaskRoutes = ""
*.baseStation.ipv4.routingTable.netmaskRoutes = ""

#-----
---
[Config AODVRouting]
description = "AODV routing"
extends = Animations

*.configurator.optimizeRoutes = false
*.configurator.addStaticRoutes = false

*.node*.ipv4.routingTable.netmaskRoutes = ""
*.sourceNode.ipv4.routingTable.netmaskRoutes = ""
*.baseStation.ipv4.routingTable.netmaskRoutes = ""

*.sourceNode.typename = "AodvRouter"
*.baseStation.typename = "AodvRouter"
*.node*.typename = "AodvRouter"

*.visualizer.dataLinkVisualizer.packetFilter = "aodv* or UDPData*"
*.visualizer.routingTableVisualizer.displayRoutingTables = true
*.visualizer.routingTableVisualizer.destinationFilter = "baseStation"

[Config Interference]
description = "Adding Interference"
extends = AODVRouting

*.node*.wlan[0].radio.transmitter.interferenceRange = 0m
*.sourceNode.wlan[0].radio.transmitter.interferenceRange = 0m
*.baseStation.wlan[0].radio.transmitter.interferenceRange = 0m

*.baseStation.wlan[0].radio.receiver.ignoreInterference = false

*.radioInterferer.numApps = 1
*.radioInterferer.app[0].typename = "UdpBasicApp"
*.radioInterferer.app[0].destAddresses = "baseStation"
*.radioInterferer.app[0].destPort = 6000
```

## APPENDIX

```
*.radioInterferer.app[0].messageLength = 500B
*.radioInterferer.app[0].sendInterval = exponential(15ms)
*.radioInterferer.app[0].startTime = 1.5s
*.radioInterferer.wlan[0].radio.displayInterferenceRange = true

*.baseStation.numApps = 2
*.baseStation.app[1].typename = "UdpSink"
*.baseStation.app[1].localPort = 6000
```

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year:</b> 3, 3	<b>Study week no.:</b> 1 & 2
<b>Student Name &amp; ID:</b> Terry Teh 19ACB02136	
<b>Supervisor:</b> Ts Dr Vasaki a/p Ponnusamy	
<b>Project Title:</b> Analysis on the Cause of Packet Loss in Wireless Networks	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Check on previous FYP 1 Report
- Find out what needs to be improved and what needs to be done in FYP 2
- Find some new articles to read on and for references

## 2. WORK TO BE DONE

- Draft the layout of FYP 2
- Testing out the previous simulation model is still working
- Done some amendments to the simulation model

## 3. PROBLEMS ENCOUNTERED

- None

## 4. SELF EVALUATION OF THE PROGRESS

- Doing good on the first two weeks just researching some articles, coping well



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year:</b> 3, 3	<b>Study week no.:</b> 3 & 4
<b>Student Name &amp; ID:</b> Terry Teh 19ACB02136	
<b>Supervisor:</b> Ts Dr Vasaki a/p Ponnusamy	
<b>Project Title:</b> Analysis on the Cause of Packet Loss in Wireless Networks	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Drafted out the FYP 2 report
- Done Chapter 1 of FYP 2
- Add some new sub-chapters to Chapter 1

## 2. WORK TO BE DONE

- Testing out the previous simulation model is still working
- Done some amendments to the simulation model

## 3. PROBLEMS ENCOUNTERED

- None

## 4. SELF EVALUATION OF THE PROGRESS

- Coping well by finishing the Chapter 1



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year:</b> 3, 3	<b>Study week no.:</b> 5 & 6
<b>Student Name &amp; ID:</b> Terry Teh 19ACB02136	
<b>Supervisor:</b> Ts Dr Vasaki a/p Ponnusamy	
<b>Project Title:</b> Analysis on the Cause of Packet Loss in Wireless Networks	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Done Chapter 2 Literature Review
- Review all the chapter 2 contents
- Done testing the simulation model and is working fine

## 2. WORK TO BE DONE

- Done some amendments to the simulation model
- Complete Chapter 3
- Debate whether to change the methodology on FYP 1

## 3. PROBLEMS ENCOUNTERED

- The simulation model is way hard to implement than expected

## 4. SELF EVALUATION OF THE PROGRESS

- Satisfied with the progress completed as expected, just had some issues with the simulation model



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year:</b> 3, 3	<b>Study week no.:</b> 7 & 8
<b>Student Name &amp; ID:</b> Terry Teh 19ACB02136	
<b>Supervisor:</b> Ts Dr Vasaki a/p Ponnusamy	
<b>Project Title:</b> Analysis on the Cause of Packet Loss in Wireless Networks	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed Chapter 3 methodology
- Decided not to change the methodology that was proposed on FYP 1
- Finalize the network simulation model

## 2. WORK TO BE DONE

- Done some research on the OMNeT++ network simulator
- Find out the source code library of the INET framework to check whether it can be modified

## 3. PROBLEMS ENCOUNTERED

- Difficult to modify the source code on the INET framework
- Need to do some research on the source code library itself

## 4. SELF EVALUATION OF THE PROGRESS

- Some accidents had been encountered; the source code library is not easy to be modified as expected



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year:</b> 3, 3	<b>Study week no.:</b> 9 & 10
<b>Student Name &amp; ID:</b> Terry Teh 19ACB02136	
<b>Supervisor:</b> Ts Dr Vasaki a/p Ponnusamy	
<b>Project Title:</b> Analysis on the Cause of Packet Loss in Wireless Networks	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Done chapter 4.1, chapter 4.2 for the software setup and system configuration
- Continue researching the INET framework documentation

## 2. WORK TO BE DONE

- Figure out the solution for network simulation implementation
- Chapter 4.3 and Chapter 4.4
- Chapter 5 System Evaluation

## 3. PROBLEMS ENCOUNTERED

- Still, the same issue from the last two weeks, need to find a way to modify the INET framework source code

## 4. SELF EVALUATION OF THE PROGRESS

- Not satisfied with the progress, still can't solve the problem of the source code



Supervisor's signature



Student's signature



# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year:</b> 3, 3	<b>Study week no.:</b> 11 & 12
<b>Student Name &amp; ID:</b> Terry Teh 19ACB02136	
<b>Supervisor:</b> Ts Dr Vasaki a/p Ponnusamy	
<b>Project Title:</b> Analysis on the Cause of Packet Loss in Wireless Networks	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed the whole report
- Submitted for Turnitin check
- Network simulation had been implemented, however, it is not a satisfactory result
- Completed weekly report

## 2. WORK TO BE DONE

- Continue implementing the code for the simulation model
- Prepare slides for the presentation
- Final checking on the report before submission

## 3. PROBLEMS ENCOUNTERED

- The network simulator still cannot be fixed to modify the source code library to produce the intended result
- The objective of this project might not be achieved

## 4. SELF EVALUATION OF THE PROGRESS

- Satisfied with the progress at least it can complete the report on time
- However, the simulation model is not satisfied as the result is not intended and it might not achieve the objective



Supervisor's signature



Student's signature

POSTER

# ANALYSIS ON THE CAUSE OF PACKET LOSS IN WIRELESS NETWORKS

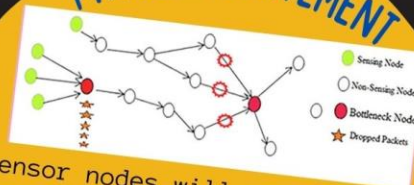
Terry Teh

## INTRODUCTION



- Wireless Sensor Networks are emerging nowadays in various applications
- Due to the nature of wireless technologies, Wireless Sensor Networks are susceptible to attacks

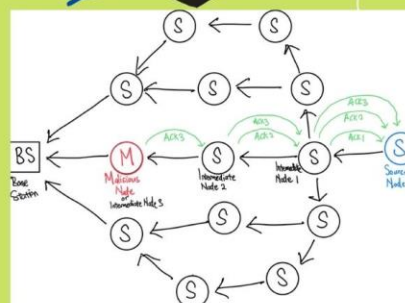
## PROBLEM STATEMENT



- Sensor nodes will sense some sensitive information to the Base Station for decision making
- Reason for packet loss need to be identified to have quick response  
E.g. Malicious Discard Packet Loss & Network Issue Packet Loss

## METHODOLOGY

- Base Station will detect there is a packet loss
- Query source node whether it has receive all three ACK packets
- If yes, conclude as network issue, otherwise malicious discard.



# PLAGIARISM CHECK RESULT

4/16/22, 11:09 AM

Tumtun

### Turnitin Originality Report

Processed on: 16-Apr-2022 10:58 +08  
 ID: 1811825763  
 Word Count: 14955  
 Submitted: 1

Similarity Index	Similarity by Source
17%	Internet Sources: 12% Publications: 13% Student Papers: N/A

**ANALYSIS ON THE CAUSE OF PACKET LOSS IN WIRELESS NETWORKS** By Terry Teh

---

1% match (Internet from 24-Dec-2021)  
<https://turkjphiotherrehabil.org/pub/pdf/321/32-1-2992.pdf>

1% match (Internet from 16-Jul-2021)  
<http://docplayer.net/13751069-Appling-intrusion-detection-systems-to-wireless-sensor-networks.html>

1% match (Internet from 18-Jun-2017)  
<http://www.i-scholar.in/index.php/indjst/article/download/135170/123462>

1% match (publications)  
[Cosmin Cirstea, Mihail Cernaianu, Aurel Gontean. "Packet loss analysis in wireless sensor networks routing protocols", 2012 35th International Conference on Telecommunications and Signal Processing \(TSP\), 2012](#)

1% match (publications)  
[Hung-Min Sun, Chien-Ming Chen, Ying-Chu Hsiao. "An efficient countermeasure to the selective forwarding attack in wireless sensor networks", TENCON 2007 - 2007 IEEE Region 10 Conference, 2007](#)

1% match ( )  
[null Bo Yu, null Bin Xiao. "Detecting selective forwarding attacks in wireless sensor networks", "Institute of Electrical and Electronics Engineers \(IEEE\)", 2006](#)

< 1% match (Internet from 12-Oct-2018)  
<https://docplayer.net/54314605-Wireless-sensor-networks-technology-and-protocols-edited-by-mohammad-a-matin.html>

< 1% match (Internet from 07-Nov-2021)  
[https://docshare.tips/the-state-of-the-art-in-intrusion-prevention-and-detection\\_58b19e05b6d87fea778b45f5.html](https://docshare.tips/the-state-of-the-art-in-intrusion-prevention-and-detection_58b19e05b6d87fea778b45f5.html)

< 1% match (publications)  
[Young-jun Oh, Kang-whan Lee. "Energy-efficient and reliable routing protocol for dynamic-property-based clustering mobile ad hoc networks", International Journal of Distributed Sensor Networks, 2017](#)

< 1% match (Internet from 18-Jul-2021)  
[https://fict.utar.edu.my/documents/FYP/FYP2\\_template/FYP2\\_Report\\_Template\\_CS.docx](https://fict.utar.edu.my/documents/FYP/FYP2_template/FYP2_Report_Template_CS.docx)

< 1% match (Internet from 17-Nov-2020)  
<https://www.bilibili.com/read/cv5546059/>

< 1% match (Internet from 11-Dec-2020)  
<https://www.intechopen.com/books/wireless-sensor-networks-technology-and-protocols/overview-of-wireless-sensor-network>

< 1% match (Internet from 09-Dec-2020)  
<https://bobbewegt.com/akilli-telefonlar/wi-fi-kanallari/802-11-a-b-g-n-acwdos17754b-e->

https://www.tumtun.com/newreport\_printview.asp?eq=1&eb=1&esm=0&oid=1811825763&sid=0&n=0&m=2&svr=31&r=51.47759200126871&lang=en... 1/24

<b>Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

<b>Full Name(s) of Candidate(s)</b>	TERRY TEH
<b>ID Number(s)</b>	19ACB02136
<b>Programme / Course</b>	CN
<b>Title of Final Year Project</b>	ANALYSIS ON THE CAUSE OF PACKET LOSS IN WIRELESS NETWORKS

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)</b>
<b>Overall similarity index: <u>12</u> %</b>  <b>Similarity by source</b> Internet Sources: <u>12</u> % Publications: <u>13</u> % Student Papers: <u>N/A</u> %	Checked and verified
<b>Number of individual sources listed of more than 3% similarity: <u>0</u></b>	Checked and verified
<b>Parameters of originality required, and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

Signature of Supervisor

Signature of Co-Supervisor

Name: Vasaki Ponnusamy

Name: \_\_\_\_\_

Date: 20 April 2022

Date: \_\_\_\_\_

## FYP 2 CHECKLIST



### UNIVERSITI TUNKU ABDUL RAHMAN

#### FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

#### CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB02136
Student Name	TERRY TEH
Supervisor Name	Ts Dr Vasaki a/p Ponnusamy

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
-	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
-	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

\*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 21.04.2022