Analysis of the Security and Reliability of Packet transmission in Wireless Mesh Networks (WMNs): A case study of Malicious Packet drop attack

By

Victor Oluwatobiloba Adeniji

Qualification: B. Sc (Honours) Computer Science

A dissertation submitted in fulfillment of the requirements of the Degree of Master of Science in Computer Science

---

Faculty of Science & Agriculture

University of Fort Hare

Together in Excellence

Department of Computer Science

Supervised by

Prof. K. Sibanda

Declaration

I hereby declare that *"**Analysis of the Security and Reliability of Packet transmission in Wireless Mesh Networks (WMNs): A case study of Malicious Packet drop attack**"* is my original work and it has not been submitted before for any degree or examination at any other university. All sources I have used, consulted or quoted are duly indicated and acknowledged herein.

……..

May 2018

Acknowledgement

All glory and adoration to God Almighty, for granting me this great privilege to successfully complete my Masters of Science (M.Sc.) degree programme in Computer Science in the Department of Computer Science of this great citadel of learning, University of Fort Hare (UFH), Alice, South Africa, and for the success of this research project.

Futher, the work presented in this study is based on the research undertaken within the Telkom Centre of Excellence (CoE) in ICT4D supported in part by Telkom SA, Coriant (Pty) Ltd, Saab Grintek Technologies and Khula Holdings. The opinions, findings and conclusions or recommendations expressed here are, however, that of the author and none of the above sponsors accepts any liability whatsoever in this regard.

Immensely, my heartfelt gratitude goes to my very wonderful, ever encouraging and very outstanding Supervisor, and the Head of Telkom CoE in the Department of Computer Science, Prof. K. Sibanda, for his day-to-day accommodation, constructive criticism, and utmost support in the course of this research project. Also, I would like to take hold of this rare and golden opportunity to appreciate the present Head of Department (HOD) of Computer Science department, UFH, Mr Dyakalashe, Prof M. Thinyane, the immediate former HOD of Computer Science department, UFH, Mr. M. S. Scott, Dr. Z. Shibeshi, Mr. S. Ngwena, Mrs Cebisa Manyonta, Miss Caroline Guragena and other staff members of the Department of Computer Science, UFH for their concerns in making this research project a success.

Moreover, my utmost acknowledgement goes to Govan Mbeki Research and Development Centre (GMRDC), UFH, Alice, South Africa, for their financial support during this M. Sc. degree programme.

From the depth of my heart, I preciously appreciate my ever loving, caring, and outstanding parents, Mr and Mrs J.O Adeniji for their time-to-time moral and financial support, and concerns ever before and during this M. Sc. degree program. My heartfelt appreciation also goes to my ever concerned and encouraging siblings, Tolu, Tayo, Tola, and Tope for always being there to encourage and motivate me all along in my academic pursuits.

I sincerely appreciate the family of Prof. and Prof. Mrs A. I. Okoh for their concerns throughout this M.Sc. degree programme. Again, I specially appreciate Mr Idowu Seriki for your words of encouragement in the course of this research project. Finally, my tremendous appreciation goes to my wonderful cousin, Olaide Ayoola, my very dear friends, Olutayo Falola, Olusayo Adesina and many others for their sincere concerns in the course of this research project.

Publications

Part of the research work presented in this dissertation has been published in the following paper:

Adeniji, V. O. and Sibanda, K., "Analysis of the effect of Malicious Packet drop attack on Packet transmission in Wireless Mesh Networks (WMNs)" In the Proceedings of ICTAS 2018: Information Communication Technology and Society Conference 2018, Durban, South African, March 8-9, 2018; and IEEE *Xplore*, May 31, 2018

Abstract

Wireless Mesh Networks (WMNs) are known for possessing good attributes such as low up-front cost, easy network maintenance, and reliable service coverage. This has largely made them to be adopted in various areas such as; school campus networks, community networking, pervasive healthcare, office and home automation, emergency rescue operations and ubiquitous wireless networks. The routing nodes are equipped with self-organized and self-configuring capabilities. The routing mechanisms of WMNs depend on the collaboration of all participating nodes for reliable network performance. However, it has been noted that most routing algorithms proposed for WMNs in the last few years are designed with the assumption that all the participating nodes will collaboratively be involved in relaying the data packets originated from a source to a multi-hop destination. Such design approach exposes WMNs to vulnerability such as malicious packet drop attack. Therefore, it is imperative to design and implement secure and reliable packet routing mechanisms to mitigate this type of attack. While there are works that have attempted to implement secure routing approach, the findings in this research unearthed that further research works are required to improve the existing secure routing in order to provide more secure and reliable packet transmission in WMNs, in the event of denial of service (DoS) attacks such black hole malicious pack drop attack. This study further presents an analysis of the impact of the black hole malicious packet drop attack with other influential factors in WMNs. In the study, NS-3 simulator was used with AODV as the routing protocol. The results show that the packet delivery ratio and throughput of WMN under attack decreases sharply as compared to WMN free from attack.

Keywords

Wireless Mesh Network; Malicious Packet drop Attack; Black hole Attack; Routing; AODV

Table of Contents

List of Tables

List of Figures

List of Acronyms

AP	Access Point
AODV	Ad Hoc On-Demand Distance Vector
BSS	Basic Service Set
DS	Distributed System
DoS	Denial of Service
DSS	Distributed Service
ESS	Extended Service Set
HWMP	Hybrid Wireless Mesh Protocol
IBSS	Independent Basic Service Set
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Point
MANET	Mobile Ad Hoc Network
MBSS	Mesh Basic Service Set
MSDU	MAC Service Data Unit
NIC	Network Interface Card
OLSR	Optimized Link State Routing Protocol
STA	Station
TG	Task Group

WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network

CHAPTER ONE

INTRODUCTION

This chapter is an introductory chapter that presents the background of the research domain, and the research problem statement. Also, the contents of the chapter include the research aim in Section 1.3, the research questions and objectives in Sections 1.4 and 1.5 respectively, the significance of the research work in Section 1.6. The chapter presents the arrangement of the research thesis in Section 1.7. The chapter is concluded with the summary of the chapter in Section 1.8.

1.1 Background

In the conventional IEEE 802.11 standard for Wireless Local Area Network (WLAN), there are two basic forms of wireless network mode. These include the Independent Basic Service Sets (IBSS) and infrastructure Basic Service Sets (BSS). In an IBSS, a simple form of wireless network is formed by connecting two or more wireless Stations (STAs) in a peer-to-peer network (Cisco Networking Academy, 2007). Also, an IBSS is usually an unplanned WLAN created to form an ad hoc network that can help to resolve immediate needs such as exchange of files and information between wireless devices (IEEE Computer Society, 2007). The infrastructure BSS is, however, a basic form of wireless network mode in which communication between STAs in a cell is controlled through an Access point (AP). Inherently, each device must obtain permission from the AP to communicate with each other as individual STAs cannot communicate directly with one another (Cisco Networking Academy, 2007).

Furthermore, in the architecture of the IEEE 802.11, coverage area can be expanded by connecting multiple BSSs to form an Extended Service Set (ESS) through a Distribution System (DS) (Cisco Networking Academy, 2007; IEEE Computer Society, 2007). The DS provides the DS service (DSS) for transporting MAC service data units (MSDUs) between APs; between APs and portals (a logical point to receive MSDUs from a non-802.11 LAN into the DS); and between stations within the same BSS that choose to involve in DSS (Wang and Lim, 2008).

Though, IBSS is an ad hoc network with the shortcoming of not providing the client support for Internet access, yet, it has the advantage of a self-configuring and self-healing. Again, in IEEE 802.11 ESS network architecture, the BSSs are connected to the DS through wired channels while Internet support for client is provided. "Thus, it is a good strategy to develop schemes to combine the advantages of ESS and IBSS" (Wang and Lim, 2008). As a result, IEEE 802.11s Wireless Mesh Network (WMN) draft standard defines an extension of 802.11 that facilitates frame forwarding in arbitrary multiple hops topology (Andreev and Boyko, 2010). The IEEE 802.11s defined wireless mesh frames forwarding operation over multiple radio hops are transparent to higher layer protocols such as IP. Sequel to this, mesh-capable stations can form a Mesh Basic Service Set (MBSS) by running a pair-wise peering protocol to create forwarding associations, and find paths through the network by running a routing protocol ('ns-3 Model Library', 2016).

A typical WMN architecture comprises of mesh clients, mesh routers and gateways. The Mesh routers within a WMN are mesh-capable nodes that relay packets to and from the Internet and also provide services to the mesh clients. The Mesh gate is a mesh entity that allows a MBSS to interconnect with a DS. Based on the ad hoc nature of the MBSS, routing nodes in WMNs are enhanced with dynamic self-organization and self-configuration algorithm. In respect to the WMN's self-organizing and self-configuring capabilities, WMN has the advantage of providing reliable, scalable and low upfront cost network. As a result, WMNs is now being adopted in various environments such as, school campus networks, community networking, pervasive healthcare, office and home automation, emergency rescue operations and ubiquitous wireless network (Saxena, Denko and Banerji, 2011).

Moreover, WMN is a type of Mobile Ad Hoc Network (MANET) that is aimed at providing wireless network services without relying on any infrastructure and it can be set up in an extreme mobile environment over an 802.11 WLAN. Again, WMN is an autonomous network that shares common characteristic with other distributed networks such as peer-to-peer networks and Wireless Sensors Network (WSN). Autonomous networks are decentralized, self-configuring and self-protecting with minimal administration that mostly requires policy-level management. Participating entities are all involved in the control of the network through collaborative communication (Jiang and Baras, 2006). Thus, there is an innate collaboration dependence

between network participating nodes to establish a secure and reliable communication channels within the network. The greatest difficulty in securing distributed networks is attributed to their distributed nature. Thus, in a distributed network such as WMNs, attacks including packet drop attack launched by any malicious mesh node in the network may impair the performance and reliability of the network if there is no secure routing mechanism implemented in the network.

1.2 Problem Statement

In a typical WMN, the discovery and maintenance of multi-hop paths depend on a routing or path selection protocol. Generally, routing protocols make use of path selection metric to choose a very optimal route from different possible paths (Andreev and Boyko, 2010; Sibeko *et al.*, 2016). As explained in (Akyildiz, Wang and Wang, 2005; Saxena, Denko and Banerji, 2011) WMNs are dynamically self-organized and self-configuring as Mesh nodes are equipped with an ad hoc capability to automatically initiate and maintain mesh connectivity with one another. The self-configured and self-adapted wireless multi-hop routing mechanisms of the WMN depends on the collaboration of all participating nodes for reliable network performance (Zhang *et al.*, 2008).

A number of routing algorithms have been proposed for WMNs, however most of these are designed with the assumption that all the participating nodes will cooperate in routing the data packets from the source to destination. This exposes WMNs to a vulnerability that can be exploited by both external and internal attacks. If a participating node is attacked in the network, the attacked node could compromise the routing mechanisms by generating incorrect routing information so as to be maliciously selective in forwarding the packets (gray hole attack), or dropping the entire packet (Black hole attack) to be relayed to destination node in the network. Consequently, the reliability and performance of the network may be degraded (Zhang *et al.*, 2008; Sarao and Garg, 2014).

1.3 Research Aim

The aim of the research is to analyse the security of packet transmission against malicious packet drop attack in WMNs.

1.4 Research Questions

- How can secure and reliable packet routing be established between source and destination nodes in WMNs?
- How optimal is the performance of the network in the absence of malicious packet drop attack?
- How impactful is packet drop attack on packet transmission in WMN?

1.5 Research Objectives

- To explore comprehensive approach that can provide secure and reliable packet routing capability in WMNs.
- To analyse the optimal performance of the network in the absence of malicious packet drop attack.
- To analyse the impact of packet drop attack on packet transmission in WMNs.

1.6 Significance of Study

A wireless mesh network as a type of distributed network is a self-configuring and self-healing network in which data is transmitted from the source to destination via multiple wireless hops (Sarao and Garg, 2014). The self-configured and self-adapted wireless multi-hop routing capability of the network is dependent on the collaboration of all routing nodes for secure and reliable packet transmission (Zhang *et al.*, 2008). On the other hand, WMNs are susceptible to attacks such as black hole malicious packet drop attack. This is attributed to the fact that most of the WMNs routing protocols are designed with the assumption that all the WMN participating routing nodes will harmoniously support the relay mechanism of the network. As explained in (Kolade *et al.*, 2017), black hole malicious attack can significantly impair the reliability and performance of the network under its attack. This is because the entire data packet routed from the source node to the destination node via the black hole node is maliciously dropped at the black hole node. The attack can either be exhibited by a single node or a number of malicious nodes. Sequel to the potential selfish and malicious behaviour, such as the black hole malicious behaviour of WMN's participating nodes, there is a need for continuous concerted efforts to

orchestrate secure and reliable routing capability in WMNs, and need to substantiate the need for secure and reliable routing in WMN through the analysis of the impact of this type of attack.

1.7 Arrangement of Dissertation

The dissertation contains seven chapters. The research project chapters are roughly categorised as follows:

Chapter One: This chapter presents the introductory part of the dissertation such as introduction, research background, problem statement, research aim and objectives, research significance and arrangement of dissertation.

Chapter Two: This chapter comprises of the basic concept of WMNs, IEEE 802.11s mesh standard and security issues in WMNs. This chapter also presents the discussion on the operation of AODV routing protocol adopted as the base routing protocol for the network configurations in the simulation experiments of this research. The chapter also presents the operations of the black hole attack on the routing mechanisms of the base routing protocol (AODV).

Chapter Three: This chapter presents the review of non-secure improved routing approaches and the existing solutions to mitigate the effect of malicious packet drop attack.

Chapter Four: This chapter presents the discussion on the research methodology and the network simulator adopted for the research simulation experiments.

Chapter Five: The fifth chapter presents the implementation and verification simulation of the malicious black hole packet drop attack in the NS-3 AODV routing protocol adopted in this research.

Chapter Six: The sixth chapter presents the research simulation experiments setup and comparative analysis of the simulation experiments results of the simulation scenarios.

Chapter Seven: It's the last chapter of the research project's thesis that presents the conclusion recommendations, and future work from the research work.

1.8 Summary

This chapter is an introductory chapter that has presented the background of the research domain, and the research problem statement. Also, the contents of the chapter include the research aim in Section 1.3, the research questions and objectives in Sections 1.4 and 1.5 respectively, the significance of the research work in Section 1.6. The chapter also presented the arrangement of the research thesis in Section 1.7. The basic concept of WMNs is presented in the next chapter.

CHAPTER TWO

WIRELESS MESH NETWORKS

In this chapter, the basic concept of wireless mesh networks is presented. Section 2.1 presents an overview on WMNs as one of the tools to resolve the issues of digital divide in developing societies. The section also identifies malicious packet drop attack as one of the vulnerabilities of WMNs. Section 2.2 discusses the architectures of WMNs. Section 2.3 presents an overview of the IEEE 802.11s standards for WMNs. The Section also discusses the basic protocols defined in the standard for peering and routing in WMNs. Section 2.4 captures the security issues in WMNs. Section 2.5 dwells on the routing mechanism of AODV routing protocol and the method adopted by black hole node to exploit ad hoc routing protocol such as AODV routing protocol.

2.1 Introduction

In recent years, Information and Communication Technology (ICT) has formed a vital aspect of international communities. The world at large has become a global village that is interconnected through the propagation of ICT. Hence, the current swift in the technological, social, political, and economical advancement is a key development that can be attributed to the rapid growth and prevalence of ICT (Yusuf, 2005; Buabeng-Andoh, 2012). As explained in (Yusuf, 2005), African nations and many other countries across the globe have identified the potentials embedded in ICT. This as a result, has brought about the development of national policies in various capacities to formulate frameworks that can enhance the integration of ICT in a numerous societal day-to-day activities. However, among many other factors (such as low basic access to ICT tools; low participation in the development of ICT equipment; and even low involvement in software development), low/inadequate Internet accessibility around most African nations also contributes to the factors that reflect the issue of digital divide between developed and developing nations (Yusuf, 2005; Oki, 2013). To reduce the digital divide especially, in the developing nations, a number of technologies have been developed with the intentions of overcoming challenges posed by lack of telecommunication infrastructure. The leading technology in this regard is the wireless networks which have taken various forms that include WMNs.

WMN is an emerging technology that is enriched/fortified with robust attributes that can facilitate the continuing efforts in bridging the existing digital divide in marginalized communities. The features of WMNs amongst many other valuable attributes include dynamic self-configuration and self-adaptation, cost-effectiveness and ease of maintenance (Pirzada and Portmann, 2007; Houaidia *et al.*, 2013; Tsado, Gamage and Lund, 2015). In recent years, WMN has been identified as a type of communication network that is suitable for realising more scalable, reliable, flexible and cost-efficient broadband wireless network to connect a magnitude of network users over wide terrain (Houaidia *et al.*, 2013; Zakaria *et al.*, 2013).

In their wide range of capabilities of providing ubiquitous interconnectivities (communication network) through their multi-hop connections, WMNs are capable of providing affordable and high speed broadband Internet infrastructure to the inhabitants of both rural and urban communities (Akyildiz, 2009; Houaidia *et al.*, 2013; Oki *et al.*, 2013; Tsado, Gamage and Lund, 2015). A wide variety of WMNs deployments have included school campus networks, community networking, pervasive healthcare, office and home automation, emergency rescue operations and ubiquitous wireless network (Akyildiz, Wang and Wang, 2005; Saxena, Denko and Banerji, 2011; Oki, 2013).

Furthermore, WMNs are autonomous networks that share common characteristic with other distributed networks such as peer-to-peer networks, WSNs and Mobile Ad hoc Networks (MANETs) (Jiang and Baras, 2006). Distributed networks compared with other types of network may occasionally experience an intermediate percentage of packet loss based on some known factors. As explained in (Edemacu, Euku and Ssekibuule, 2014; SAIRA AZIZ, 2016), packet dropping in autonomous networks can be classified into non-malicious and malicious packet drop. Non-malicious packet drop in WMN can be attributed to conditions such as traffic congestion, scalability, and link quality. On the contrary, malicious packet drop is an intentional dropping of data packet to be relayed by a node exhibiting malicious behaviour in the network. To launch malicious packet drop attack, a malicious node gets involved in the path selection from a source node to the destination node by exploiting the weakness in the path selection (routing) mechanism in the network.

Malicious packet drop attack can lead to a denial of service (DoS) that can leave a victim node isolated in the network and, thus degrade the reliability and performance of the network. Studies such as (Morote, 2011; Houaidia *et al.*, 2013) investigated the impact of traffic congestion, scalability, link quality on routing mechanisms in WMNs. This study presents an analysis of the impact of malicious packet drop attack on the real-time multi-hop routing capabilities of WMN in and explores possible solutions that can enhance secure and reliable packet transmission in the network. This is crucial to deploying a WMN that is secure against malicious data packet drop attack that can frustrate the efficiency of WMN in providing a ubiquitous network attributed with a secure and reliable service delivery. The architectures of WMNs are discussed in the following section.

2.2 WMN Architecture

Wireless Mesh Network (WMN) is a form of wireless ad hoc network in which the participating nodes are arranged into a multi-hop mesh topology. In the network setup, each mesh node is regarded as an entity that can relay packets along the path between a source node to the destination node (Zhang, Luo and Hu, 2007). In addition, a WMN can be described as a smart network that does not require the participating mesh nodes to be connected to a central switch using cables. The neighbouring mesh nodes communicate with one another by establishing a wireless connection through 802.11 links. Moreover, WMN is a type of communication network that can be easily set up as the nodes arrangement does not necessitate complex design and topology plan to obtain a reliable network performance. As a result, new mesh nodes can be deployed in the network to increase its scalability (Strix Systems Inc., 2005). The topology of a typical WMN can be either formed into a full mesh or partial mesh topology. In a full mesh topology, each network node is connected directly to every other node in the topology. On the other hand, in a partial mesh topology, some node are connected to every other node while other nodes are connected to only the nodes with which they most exchange data. An average WMN is composed of static wireless relay nodes that provide a distributed infrastructure for mobile client nodes over a partial mesh topology (Zhang, Luo and Hu, 2007).

Furthermore, the participating nodes in WMN include mesh routers and mesh clients. The mesh clients can function as hosts and also route information (Regan, Martin and Manickam, 2016).

Again, the traditional nodes such as desktops, laptops, PDA, smart phones and other devices integrated with a wireless Network Interface Card (NIC) can directly connect to mesh routers enabled with access point (AP) functionalities to access the available services provided on the Network. Also, devices without NICs can access the WMN via a channel such as Ethernet. As a result, WMN can provide ubiquitous Internet access to users. Existing non 802.11 networks and other wireless networks such as cellular, Wireless Sensor Network (WSN) and worldwide inter-operability for microwave access (WiMAX) can be integrated with WMNs through gateway/bridge enabled mesh routers. Thus, the users of an existing network can be provided with services of these networks through an integrated WMN (Akyildiz, Wang and Wang, 2005).

As explained earlier, WMNs consists of two basic network nodes, these are mesh routers and mesh clients. The mesh clients function as both hosts and also relay packets within the mesh network. Therefore, WMNs architecture can be generally classified into three types of architecture. These include: Client WMNs, Infrastructure/backbone WMNs and Hybrid WMNs. The three types of architectures are explained in the following subsections respectively.

2.2.1 Client WMNs

The WMN client architecture involves the peer-to-peer connection of client nodes/devices. In this network architecture, mesh routers are not required for the multi-hop routing mechanism within the network. Consequently, each mesh client node is enabled with both client and multi-hop routing functionalities. The routing capabilities of each mesh client enrich it with the ability to function as a relay node that helps in the multi-hop routing of a packet destined to a multi-hop destination node in the network. Compared to infrastructure WMNs, client nodes devices in client WMNs are usually equipped with one type of radio. Again, the responsibilities of mesh clients for routing and self-configuration activities in the network increase their requirements. Typical client mesh architecture is represented in Figure 2.1.

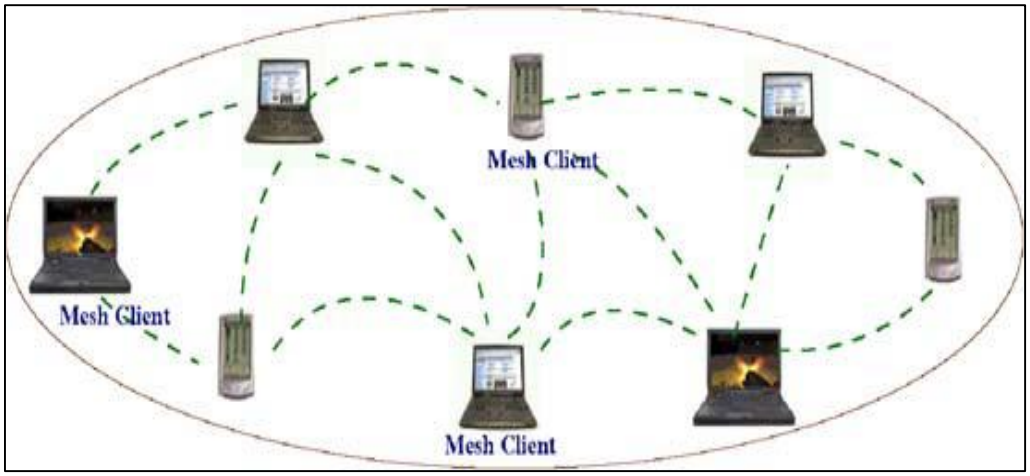


Figure 2. 1: Client WMNs (Akyildiz, Wang and Wang, 2005)

2.2.2 Infrastructure/Backbone WMNs

The infrastructure/backbone WMNs architecture usually includes mesh routers providing distributed infrastructure for mobile clients over a mesh topology. In this WMN architecture type, each mesh router is equipped with the ability to establish self-configuring and self-healing links with other neighbouring mesh routers. Also, various types of radio technologies can be used in addition to the IEEE 802.11 technologies in the network setup. Moreover, the infrastructure/backbone WMNs can be connected to the Internet via one or two mesh routers with gateway functionalities. In addition, conventional clients with Ethernet interface can be connected to mesh routers via Ethernet links. Again, through the gateway/bridge equipped mesh routers, infrastructure/backbone WMNs can be integrated with other existing wireless networks. Conventional clients with the same radio technologies as mesh routers can directly communicate with mesh routers while clients with different radio technologies are required to communicate with the base stations that are connected to mesh routers through Ethernet links. Infrastructure/Backbone WMNs are efficient in providing networks such as campus, community and neighbourhood networks. An example of an infrastructure/backbone is depicted in Figure 2.2.

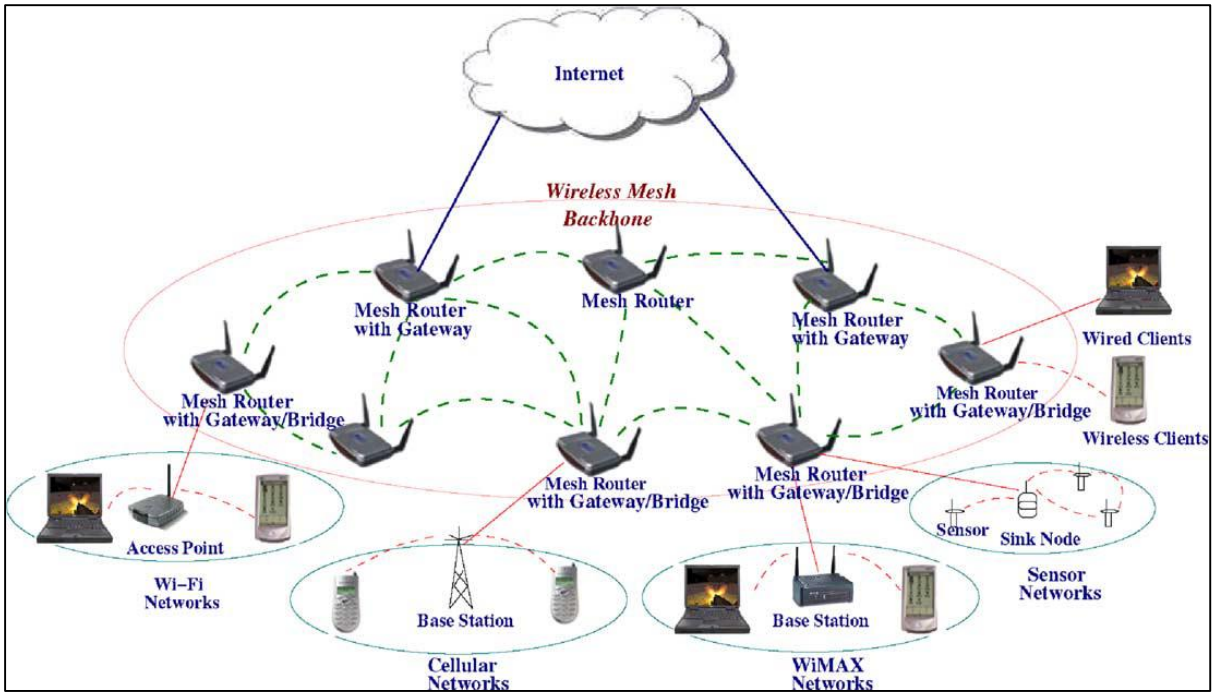


Figure 2. 2: Infrastructure/Backbone WMNs (Akyildiz, Wang and Wang, 2005)

2.2.3 Hybrid WMNs Architecture

The hybrid WMN architecture is formed with the integration of infrastructure and client mesh architectures. In this architecture, mesh clients can access the network through mesh routers and can also directly form mesh connections with other mesh clients. Thus, the backbone WMN can provide connectivity to other networks such as the Internet, Wi-Fi, WiMAX, cellular, and sensor networks while the routing capabilities of mesh clients can improve connectivity and coverage within the WMN (Akyildiz, Wang and Wang, 2005). The architecture of hybrid WMN is represented in Figure 2.3.

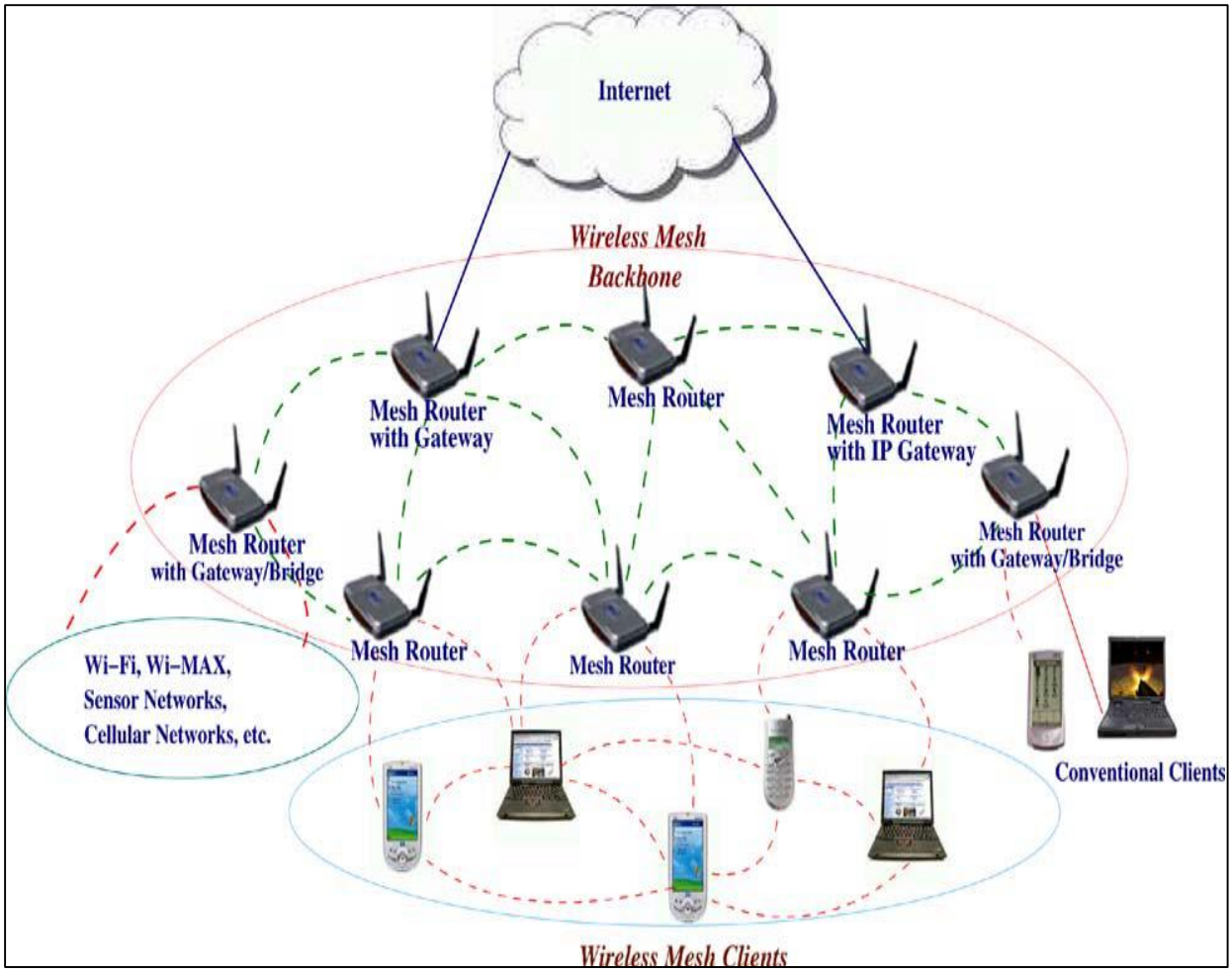


Figure 2. 3: Hybrid WMNs (Akyildiz, Wang and Wang, 2005)

2.3 IEEE 802.11s WMNs

The earlier developments of WMNs were industry based. As a result, there was no standard that consolidated the implementations of WMNs. There only existed proprietary deployments of WMNs from various companies. Thus, there were interoperability issues among different WMN technologies from various vendors. However, in the year 2004, the Institute of Electrical and Electronics Engineers (IEEE) initiated the 802.11s Task Group (TG) to form WMNs standard (Camp and Knightly, 2008; Tan *et al.*, 2013). The IEEE 802.11s standard facilitates MAC enhancements to support wireless mesh LAN infrastructure (IEEE, 2012). IEEE 802.11s WMNs architecture consists of nodes that can be classified into backbone mesh Stations (STAs) and

client STAs. The mesh STAs can be further be categorized as Mesh Points (MP), Mesh Access Points (MAPs) and Mesh Portals (MPPs) (Wang and Lim, 2008; Oki *et al.*, 2014).

The MP is simply a mesh STA that participates in the formation and operation of an MBSS to support the mesh services. The MAP is an MP enabled with AP functionalities. Hence, a MAP is also capable of providing services to client STAs. An MPP is an MP that serves as a logical point where MSDUs exit/enter mesh infrastructure to/from other networks. In other words, an MPP is a mesh STA that functions as a bridge or gateway between mesh network and external networks such as conventional IEEE 802.11 LAN, wired network or other non-IEEE 802.11 LAN. Client STAs are conventional wireless nodes that request the services provided by MBSS. A client STA cannot participate in the route discovery and relay mechanisms of the mesh infrastructure (Wang and Lim, 2008; Ndlela *et al.*, 2013). As explained in (IEEE, 2012), in the IEEE 802.11s mesh, the mesh facilities are exclusively available to existing member mesh STAs of an MBBS**.** Therefore, only the mesh discovery services are accessible by a mesh STA that is yet to connect to an MBSS. Moreover, mesh facility is simply referred to, as the collective unique enhancing factors that differentiate a mesh STA and non-mesh STA. A typical IEEE 802.11s WMN architecture is depicted in Figure 2.4

In addition, an individual mesh STA participating in IEEE 802.11s mesh infrastructure is regarded as a link layer router that is capable of collaborating with other neighbouring mesh STAs for frame forwarding and delivery. IEEE 802.11s mesh is supported by a number of dedicated protocols. The protocols include the mesh peering management (MPM) protocol, and mesh path selection and forwarding protocol which are the basic protocols in the IEEE 802.11s standard. The peering management protocol is the protocol that is responsible for the establishment of pairwise mesh link and maintenance between neighbouring mesh STAs while the routing protocol is the dedicated mesh protocol that initiates the multi-hop path discovery and maintenance within the IEEE 802.11s mesh infrastructure. The other defined 802.11s standard protocols include protocols such as inter-networking, security, power safe, channel assignment (Andreev and Boyko, 2010).

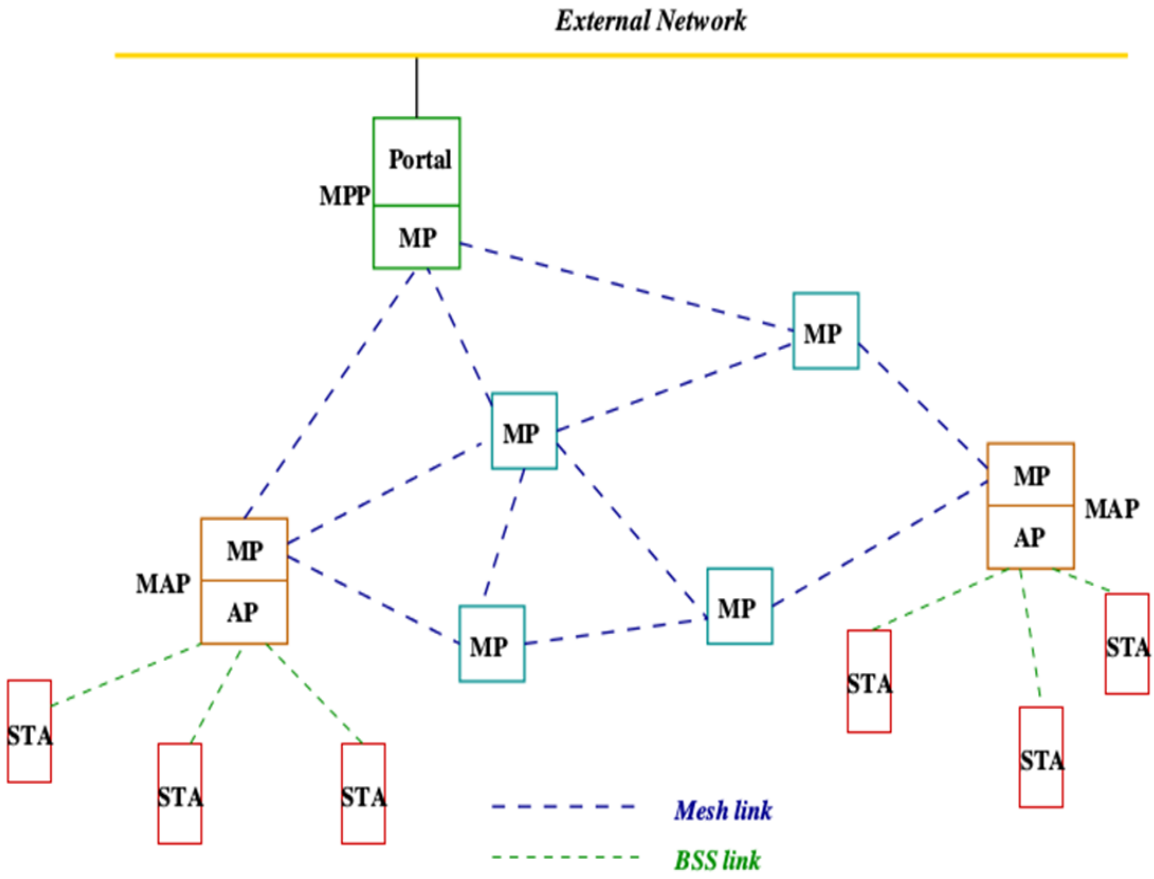


Figure 2. 4: IEEE 802.11s Mesh Architecture (Wang and Lim, 2008)

2.3.1 Mesh Peers formation and management

In the IEEE 802.11s standard, establishment and closure of mesh peering is facilitated by Mesh Peering Management protocol. A mesh STA is allowed to establish mesh peering with multiple neighbour mesh STAs. A mesh STA can also decide to open a new link and close links if failure of the established link is detected. However, until a successful mesh peer link is established between neighbour mesh STAs, a mesh STA is not capable of exchanging messages with its neighbour mesh STAs other than the mesh discovery frames required for initiating mesh peer with other neighbour mesh STAs. As defined in IEEE 802.11s standard, a mesh STA discovers an existing MBBS by performing active or passive scanning. The mesh discovery process of neighboring mesh STAs requires an individual mesh STA to periodically transmit Beacon (one-hop management frame) and respond with Probe Response frames whenever it receives a Probe

Request. The Beacon and the Probe Response frames usually contain the Mesh ID element that facilitates the identification of the mesh BSS (IEEE, 2012).

As mentioned earlier, beacons are strictly transmitted periodically. This is to enhance effective power management in mesh STAs. However, two beacons from two neighbor mesh STAs may collide endlessly. Thus, to help avoid collisions of beacons, the MPM Peer link handshake mechanism is initiated when mesh STAs receive beacons from unknown mesh STAs and decide to open link with the mesh STAs. In Figure 2.5, a successful mesh peer link establishment is depicted. In the figure representation, Peering Open one-hop management frame is transmitted in response to the beacon (which contains mesh elements) transmitted by a potential peer mesh STA. The mesh elements contained in the transmitted Peering Open frame are processed at the mesh STA that originates the beacon. In response to the Peering Open frame, the mesh STA initiating the peer connection sends a Peering Confirm management frame if it agrees with the elements of the received Peering Open frame. Through the handshake mechanism, the peer link is established between the two mesh STAs only after they have both successfully exchanged Peering Open requests and Peering Confirm replies. Thus, the peer links established between two mesh STAs can be guaranteed to be bidirectional (Andreev and Boyko, 2010; Morote, 2011).

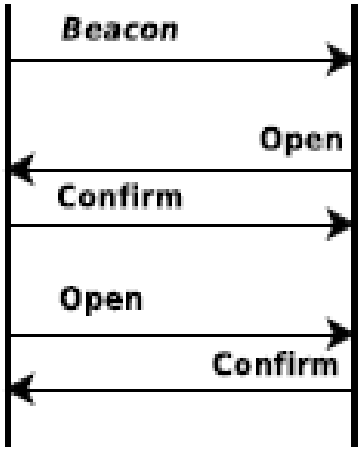


Figure 2. 5: IEEE 802.11s Peer Link Establishment (Andreev and Boyko, 2010)

2.3.2 Mesh path selection and forwarding protocol

As mentioned earlier in Section 2.3, the multi-hop wireless path discovery and maintenance within an MBBS is achieved through the mesh path selection protocol. The mesh path selection protocol adopts path selection metric in choosing an optimal path among alternative paths established from a source to the destination. In IEEE 802.11s standard for WMNs, the default path selection protocol defined for an MBSS is the hybrid wireless mesh protocol (HWMP) while the default path selection metric defined is airtime link metric (ATM). Alternatively, the standard accommodates the flexible implementations of path selections protocols and metrics other than the defined default mandatory path selection protocol and metric. In other words, the default and vendor specified path selection protocols and metrics can be implemented on a mesh STA. However, the standard specifies that only one path selection protocol and only one path selection metric shall be run by a mesh STA at a time (IEEE, 2012). The IEEE 802.11s default path selection protocol is explained in the following subsection.

2.3.2.1 Hybrid Wireless Mesh Protocol (HWMP)

The hybrid wireless mesh protocol (HWMP) is a mesh routing (path selection) protocol that integrates the flexible implementations of the reactive and proactive routing mechanisms of on-demand and tree based ad hoc routing protocols. Thus, HWMP supports two modes of operations, which include reactive and proactive modes. In HWMP, the reactive and proactive modes are inclusive depending on the configurations. The reactive and proactive modes can be adopted concurrently since the proactive mode is an extension of the on-demand mode. Thus, the combined routing mechanisms of the reactive and proactive elements of HWMP is believed to facilitate efficient path discovery and selection in a wide variety of mesh network deployments.

Moreover, HWMP implements MAC address-based path selection and link metric awareness, adopting a common set of protocol elements, generation and processing rules adapted from Ad Hoc On-Demand Distance Vector (AODV) (IETF RFC 3561 (Perkins, Belding-Royer and Das, 2003)). Also, the modes of operation defined in HWMP employ similar processing rules and primitives. The routing elements of HWMP include the PREQ (path request), PREP (path reply), PERR (path error), and RANN (root announcement). The path selection in HWMP is dependent on the metric cost of the links. A Metric field is defined in each HWMP element such as PREQ,

PREP, and RANN to relay the metric data between mesh STAs. To always avoid routing loop, path selection in HWMP adopts a sequence number (SN) mechanism used by mesh STAs to differentiate between newer path information and stale path information. Thus, each mesh STA is required to maintain its own HWMP SN which is transmitted to other mesh STAs in the HWMP elements. HWMP path discovery maintains a path to both source and destination (forward and reverse paths) at each intermediate station, therefore, each intermediate mesh STA knows both next and previous STAs known as precursor mesh STAs in the mesh path (Andreev and Boyko, 2010; Morote, 2011; IEEE, 2012).

2.3.2.1.1 On-demand path selection mode

The on-demand routing mode is the HWMP's path selection mode in which the path discovery is initiated on-demand when a source mesh STA needs to forward its data packet to an unknown destination. Thus, to establish a forwarding path with the destination mesh STA, the source mesh STA broadcasts a PREQ management frame containing the destination address. Any mesh STA that receives the PREQ frame creates a route to the source, updates the metric field and forwards the PREQ. However, if a mesh STA receiving a PREQ has valid routing information to the destination or the mesh STA is the destination itself, it generates a PREP management frame. The condition that an intermediate mesh STA generates a PREP is determined by the PREQ tags. PREP is unicasted hop-by-hop to the source mesh STA. An optimal path among alternative PREPs is selected at the source mesh STA based on the path metric. As mentioned earlier in Subsection 2.3.2.1, SN mechanism is used to differentiate between current path information and stale path information. Furthermore, a PERR management frame is sent to all mesh STA that are known to have used broken links in their paths when a broken link is detected by the peering management protocol. The routing table thus maintains a list of precursors to keep track of the potential receivers of PERR. To establish a new path when an active route is discarded by PERR, a mesh STA initiates a new path discovery process as explained earlier (Andreev and Boyko, 2010). A typical on-demand route discovery process is depicted in Figure 2.6.

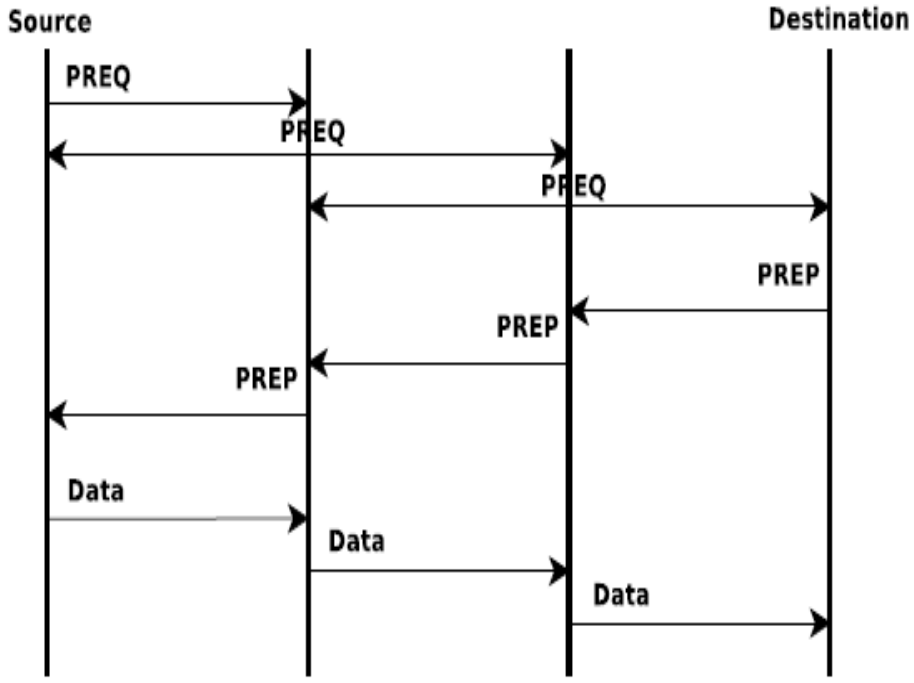


Figure 2. 6 HWMP on-demand route discovery (Andreev and Boyko, 2010)

2.3.2.1.2 Proactive tree path selection mode

As mentioned earlier, both on-demand and proactive routing modes (depending on the configurations) can be used concurrently since the proactive mode is an extension of the on-demand routing. The proactive tree path selection mode of HWMP involves the configuration of a mesh STA as a tree root mesh STA. To ensure that other mesh STAs are aware of the path to the root mesh STA, the root mesh STA periodically broadcasts proactive PREQ. Thereby, any mesh STA that receives a proactive PREQ generates or updates its forwarding path to the root mesh STA, updates the metric and hop count of the PREQ, records the metric and hop count to the root mesh STA and then relay the updated PREQ. The processing of the proactive PREQ is similar to that of the PREQ in the on-demand. Thus, a mesh STA updates its current path to the root mesh STA only on the condition that the PREQ contains a greater HWMP SN, or the HWMP SN is similar to the current path and the PREQ provides a better metric than the current path to the root mesh STA.

However, proactive PREP (depending on the subfield of the proactive PREP in the proactive PREQ received or for instance, if there is a need for a proactive PREQ recipient to transmit data to the root mesh STA) is required to establish forwarding path from the root mesh STA to proactive PREQ recipient mesh STA. Therefore, every mesh STA knows the path to the root mesh STA while the root mesh STA also knows a path to every mesh STA. If the path to a destination is unknown, the data to be forwarded are then forwarded to the root mesh STA to relay the data to the final destination. In addition, a root mesh STA can also adopt RANN element to periodically propagate its path information into the network but RANN does establish a path at any mesh STA receiving it. Therefore, each mesh STA that intends to create or update a path to the root mesh STA transmits a unicast PREQ to the root mesh STA through the mesh STA the RANN was received. Then the root mesh STA responds to the PREQ by sending a PREP (Andreev and Boyko, 2010; IEEE, 2012). The proactive mode using proactive PREQ is represented in Figure 2.7.

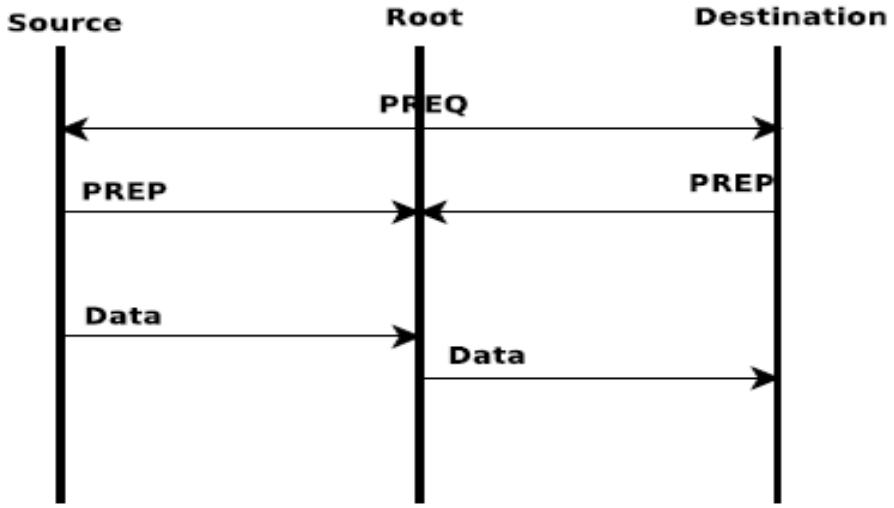


Figure 2. 7: HWMP tree-based route discovery using PREQ (Andreev and Boyko, 2010)

2.3.2.2 Airtime link metric (ALM)

The airtime link metric (ALM) is a default path metric that implements a radio-aware path selection metric to enhance a more realistic multi-hop path selection by path selection protocol. ALM reflects the approximate measure of the channel resources used for frame transmission

over a particular link. In other words, airtime estimates total medium access time needed for frame transmission and is designed for ease of implementation and interoperability. However, as specified in the IEEE 802.11s mesh profile extensibility framework, ALM can be overridden by other path selection metric. Its computation parameters/elements include an average number of retries, current data rate, some medium access overhead, such as frame headers, training sequences, acknowledgements, and some other features (Andreev and Boyko, 2010; IEEE, 2012).

2.4 Security Issues in WMNs

Generally, WMNs are vulnerable to various types of attacks at each layer of their protocol stacks. The attacks can be external or internal by malicious or selfish nodes. These attacks can range from attacks launched at the physical layer to application layer, which as a result the performance of the network can be degraded as these attacks can result to increased latency in packet delivery, increased packet drop and depleted throughput of the affected network. Also, the attacks can compromise the data confidentiality, data integrity, and availability and authentication mechanisms of the network (Sen, 2011, 2012, 2013). In the IEEE 802.11s WMN standard, the current security specifications are dedicated to establishing secure links between pair mesh STAs. Thus, mesh security association (MSA) services are specified in the 802.11s standard to establish a secure link between peering neighbor mesh STAs.

The defined IEEE 802.11s MSA is basically adapted from the security mechanisms of 802.11i. The MSA services extend the robust security network association (RSNA) framework of 802.11i to suit WMNs environment while it also inherit the authentication principles of 802.1X for initial authentication. Therefore, the defined IEEE 802.11s mesh link security protocols are responsible for the peer link authentication between neighbor mesh STAs. Also, the mesh link security protocols are dedicated to creating session keys between peer mesh STAs. The operations of the mesh authentication protocols include the establishment of a shared, common pairwise master key (PMK), and authentication of peer mesh STAs. As a result, the authenticated mesh peering exchange protocol is dependent on the existence of the PMK between the two mesh STAs to establish authenticated peering and obtain session keys (Wang and Lim, 2008; IEEE, 2012; Oki *et al.*, 2014).

As defined in IEEE 802.11s mesh standard, the security association key holders are of two types. These are namely; the mesh key distributor (MKD) and mesh authenticator (MA). A mesh STA can serve as an MA, MKD, or both MKD and MA. In the mesh network, a mesh STA that is not an MA or MKD assumes the role of a supplicant mesh STA. The role of a mesh STA as MA, MKD or both MKD and MA is determined by the 802.11s mesh standard specified role negotiation protocol. Also, a mesh key holder security association procedure is specified, since a secure association between an MA and MKD is required. Furthermore, a supplicant mesh STA that needs to establish secure links with other mesh STAs is required to first set up mesh key hierarchy. The mesh key hierarchy is established during the initial MSA authentication between the supplicant MP and an MKD through an MA. The mesh key hierarchy is subdivided into two branches. These are; key distribution branch and link security branch. The key distribution branch is responsible for the generation of security keys for key distribution, while link security branch handles the generation of keys for a secure link.

As specified in the IEEE 802.11s standard for mesh network, only one MKD shall exist in a WMN infrastructure while multiple MAs can be associated with it. Generally, the entire link authentication process between peer mesh STAs as defined by the IEEE 802.11s standard is dependent on the presence and availability of the MKD**.** However, there are occasions in which connections might be lost to the MKD because of the ad hoc configurations and transitory behavior of the wireless links between the mesh STAs in the WMN infrastructure. Furthermore, in a WLAN infrastructure, it is possible to assume that the connection between an authentication server (AS) and the authenticator is established over a trusted channel, whereas, in a wireless mesh network, the multi-hop wireless link between the MKD and MA lessens the guarantee that a trusted channel is established between the MA and MKD. A study such as the one presented in (Oki *et al.*, 2014) explored the performance of four energy-based leader selection algorithm (LSAs) from the domain of WSN to determine their suitability for MKD selection in WMNs.

Furthermore, the secure link specification provided in IEEE 802.11s mesh standard does not include secure path selection (routing) mechanisms within a mesh infrastructure. As part of the multi-hop packet routing nature of WMNs, routing messages are propagated from the source node to the destination nodes via multi-hop intermediate mesh nodes to establish a route for packet transmission between the source node and the destination node. As the case may be, the

routing messages are processed at the intermediate mesh nodes and forwarded towards the direction to the destination. The destination may be a forward destination node (the destination node of the path request message) or backward destination node (as in the case of path reply message originated from the destination node in response to the source node which is the originator of the path request packet). In a multi-hop routing environment such as WMNs, the trustful behavior of the intermediate mesh nodes cannot be completely guaranteed. Thus, the routing control messages are vulnerable to alteration at any malicious or selfish intermediate node along the path between the source and destination nodes. The most commonly manipulated elements of the routing packets may include the hop count and path metric requested or provided (Tan *et al.*, 2013; Peethambaran and Jayasudha, 2014).

Therefore, further attacks such as packet drop attack can be launched on a routed data packet in the network. These attacks may include attacks such as gray hole, black hole, wormhole attacks and Sybil. Coupled with the analysis of the approaches to provide secure and reliable packet transmission in WMNs, the impact of black hole malicious packet drop attack in the context of an infrastructure/backbone WMN is investigated in this research. The mentioned attacks are briefly discussed in the following subsections.

2.4.1 Black hole

A Black hole attack is a form of routing attack that results in DoS in WMNs. An adversary launches black hole attack by exploiting the route discovery mechanism of WMN routing protocols that adopt a similar routing approach related to that of on-demand routing protocols. Hence, the malicious node is always replying positively to a path request while it may not even have a valid route to the destination. Also, the attacker is always quick to reply to the path request message before other intermediate/destination nodes since it does not check its routing entries. Therefore, a greater amount of all the traffic transmitted around the neighborhood of the black hole node will be channeled towards the black hole node. Thus, the black hole node maliciously drops all the packets attracted to it, resulting in DoS. Black hole attack becomes more complex when multiple nodes connive to form a collaborative black hole attack, leading to complete disruption of the routing and packet forwarding operation of the network (Sen, 2011; Kayarkar, 2012; Peethambaran and Jayasudha, 2014).

2.4.2 Gray hole Atttack

A gray hole attack is a routing attack that adopts similar approach as the black hole attack in launching data packet dropping attack. However, the data packet attracted to the attacker are selectively dropped by the attacker. This makes the malicious data packet drop at the malicious node relatively unexposed. In other words, a gray hole attack does not result to complete DoS, and it may remain undetected over a long period of time. This is because the malicious packet dropping may be attributed to intermittent packet loss resulting from congestion in the network (Sen, 2011; Peethambaran and Jayasudha, 2014).

2.4.3 Worm hole Attack

A worm hole attack is a form of routing attack in which two or more malicious nodes connive to establish a tunnel using an efficient communication medium (wired connection or high-speed wireless connection). In the route discovery process of the routing protocols, the path request messages are forwarded between the malicious nodes through the established tunnel. Thus, the first path request message that arrives at the destination node is the one that is forwarded by the malicious nodes. As a result, the malicious nodes are included in the path between the source and the destination nodes. Consequently, the malicious nodes may subsequently drop the entire data packets, resulting in a complete DoS attack, or selective packet dropping in order to avoid detection (Sen, 2011; Kayarkar, 2012).

2.4.4 Sybil attack

Sybil attack is the form of attack where a malicious node creates multiple identities in the network, each appearing as a legitimate node. As explained in (Sen, 2011), Sybil attack was first discovered in distributed computing applications where the redundancy in the system was exploited by creating multiple identities and controlling considerable system resources. In the networking scenario, a number of services like packet forwarding, routing, and collaborative security mechanisms can be disrupted by the attacker using a Sybil attack. The attack affects the network layer of WMNs, which are supposed to take advantage of the path diversity in the network to increase the available bandwidth and reliability. If the malicious node creates multiple identities in the network, the legitimate nodes will assume these identities to be distinct

nodes and will add these identities in the list of distinct paths available to a particular destination. Thus, the malicious node that created the identities processes these packets after the packets have been forwarded to these fake nodes. Consequently, all the distinct routing paths will pass through the malicious node. The malicious node may therefore, launch any of the above-mentioned attacks. Even if no other attack is launched, the advantage of path diversity is diminished, resulting in degraded performance (Sen, 2011).

2.5 AODV Routing Protocol versus Black Hole Attack

The impact of a malicious packet drop attack that could be launched by a malicious node in WMNs was investigated using AODV routing protocol as the base routing protocol. Therefore, this section discusses the routing mechanism of AODV and the typical approach adopted by a black hole node to deceptively attract the routing path towards itself in WMN routing protocol such as AODV routing protocol.

2.5.1 Ad-hoc On-Demand Vector (AODV) Routing Protocol

As explained in (Kumar *et al.*, 2015), AODV is a type of dynamic reactive routing protocol in which routes are established based on the demand (upon request by source node). According to (Perkins, Belding-Royer and Das, 2003), the AODV routing protocol enables dynamic, self-starting, multi-hop routing between participating wireless ad hoc nodes that wish to establish and maintain ad hoc connections with each other. Its features include quick adaptation to dynamic link conditions, low processing and memory overhead, low network utilization, and unicast route discovery to destinations within the ad hoc network. Also, a unique feature of AODV is the use of a destination sequence number for each route entry. The destination sequence number is created by the destination and is included along with any route information it sends to the requesting node. The use of the destination sequence numbers ensures loop freedom. Hence, if a requesting node is required to choose between two routes to a destination, the node will select the route with the greatest sequence number.

In the AODV routing algorithm, the message types defined are Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs). The source node broadcasts a RREQ to find a route to the destination when a route to a new destination is needed. A route can be determined

when the RREQ reaches either the destination itself, or an intermediate node with a fresh enough route to the destination. A fresh enough route is a valid route entry for the destination whose associated sequence number is at least as great as that contained in the RREQ. The route is made available by unicasting a RREP back to the RREQ originator. Each node receiving the request caches a route back to the originator of the request, so that the RREP can be unicast from the destination along a path to the originator, or likewise from any intermediate node that is able to satisfy the request. Moreover, routing nodes are able to monitor the link status of the next hops in active routes. As a result, a RERR message is used to notify other nodes that the loss of a link has occurred when a link break in an active route is detected.

2.5.2 Black Hole Attack

As explained in (Kolade *et al.*, 2017), a malicious black hole attack can significantly impair the reliability and performance of the network under its attack. The attack can either be exhibited by a single node or a number of malicious nodes. In an ad-hoc routing protocol such as AODV, the malicious node launching black hole attack deceives the source node to have a fresh enough route to the destination node. This is achieved by the black hole node as it immediately responds to the source node RREQ broadcast packet with a unicast RREP message with the highest destination sequence number and lowest hop count value (1 to be precise). Since the routing protocol lacks a mechanism to establish trust among participating nodes, the originating node believes that the RREP is from an authentic intermediate node with a fresh route or the required destination node itself. Consequently, the source trusts that the destination is one hop neighbour of the black hole node and discards other received RREP packets. Based on the contained route information in the RREP from black hole node, the source then begins to send its intended data packets through the black hole node with the trust that the sent data packets will hop through to the destination. Conversely, instead of getting through to the destined node via the black hole node, the data packet dropped is maliciously by the black hole node.

As shown in Figure 2, node C is a malicious black hole node that aims to drop data packets from node S to D. In order to discover a path from node S to D, node S initially broadcasts RREQ packet to all its neighbours. As each neighbouring node receives the RREQ message, it continues to rebroadcast RREQ message until it reaches node D. The malicious node C, however, violates this rule and deceptively claims to node S that it has the shortest path to D by sending a fake

RREP packet to S. Hence, S assumes that the shortest path to node D is via C and begins to send data packets to D through C which is in turn dropped.

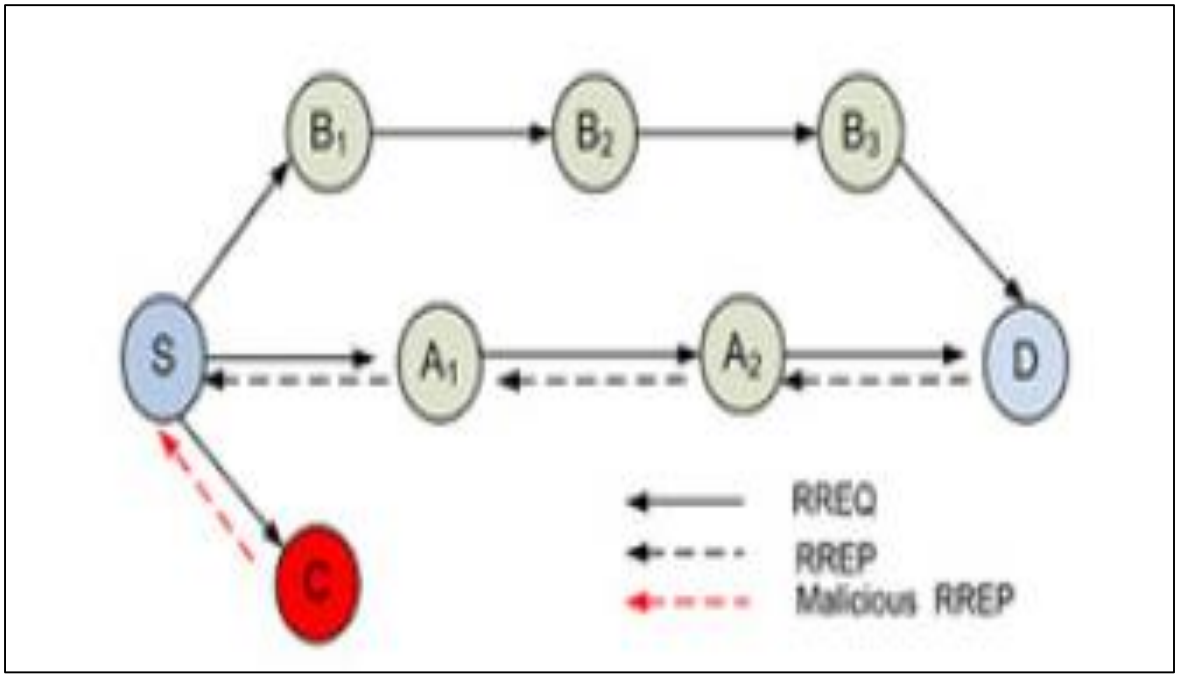


Figure 2. 8: Black Hole Attack in AODV (Edemacu, Euku and Ssekibuule, 2014)

2.6 Summary

This chapter presented the basic concept of wireless mesh networks. Section 2.1 presented an overview on WMNs as one of the tools to resolve the issues of digital divide in developing societies. The section also identified malicious packet drop attack as one of the vulnerabilities of WMNs. Section 2.2 discussed the architectures of WMNs. The architectures discussed include client, infrastructure/backbone and hybrid mesh architectures. Section 2.3 presented an overview of the IEEE 802.11s standards for WMNs. The section also discussed the basic protocols defined in the standard for peering and routing in WMNs. Section 2.4 captured the security issues in WMNs. In the section, the types of routing attacks which include black hole attack, gray hole, worm hole and sybili were discussed. Among the mentioned packet drop attacks, black hole attack was identified to pose a complete packet drop of data packets deceptively attracted to the attacker. Section 2.5 dwelt on the routing mechanism of AODV routing protocol and the method adopted by black hole node to exploit AODV routing protocol. The next chapter presents the survey/review of related non-secure and existing secure routing approach in WMNs.

CHAPTER THREE

EFFECTIVE AND SECURE ROUTING IN WIRELESS MESH NETWORKS

This chapter presents the survey/review of related non-secure and existing secure routing approach in WMNs. Section 3.1 presents an overview of the chapter. In Sections 3.2, related non-secure routing studies are presented. Section 3.3 discusses the existing secure routing methods. Section 3.4 presents the summary of the chapter.

3.1 Overview

As it was explained in Section 2.1, there are various factors that could result in packet drop beside malicious packet drop in WMNs. Among these non-malicious factors are the conditions which may include scalability, intra-flow and inter-flow interferences, congested links, and channel errors in the network. As with these factors, it is possible to attribute malicious packet drop due to the malicious behaviour of some participating nodes to some of these non-malicious conditions that may result in an intermediate packet drop in the WMNs network. Furthermore, security solutions have included improving on mesh peer link authentication of any new mesh node intending to participate in WMNs (Oki *et al.*, 2014), yet, peer link authentication cannot guarantee that an authenticated node would not behave selfishly or maliciously in compromising the routing mechanism of the routing protocol of the network for their own selfish or malicious intents. Moreover, efforts have been made on improving various aspects of the WMNs protocol stack to improve packet transmissions in WMNs. On the other hand, security of the routing mechanisms or protocols were not put into consideration.

In order to address the issue of packet drop due to malicious behaviour of any participating nodes in WMNs, there is a need to design and implement a routing protocol that can detect and mitigate this type of attack within the network to achieve efficient, reliable and secure service delivery in the network. On this basis, various works have been done in the recent years to improve the security of WMNs to address the issue of packet drop attack. However, none of the solutions provided have been fully guaranteed to be efficient enough to wholly mitigate the impact of malicious packet drop attack on packet transmission in WMNs. To develop a routing protocol

that is secure against malicious packet drop attack, various aspects of the network protocol stack have to be considered. For instance, it is possible to design a secure routing protocol that identifies/detects and isolates a participating node as a malicious node with respect to the low forwarding rate of the data packets that are forwarded by the node. On the other hand, some factors such as link quality, channel error, traffic congestion, and interference may otherwise contribute to the low forwarding rate of such a node in the network. Thus, a high false positive detection rate might be imminent if such a secure routing algorithm/protocol does not put the aforementioned non-malicious factors into consideration in its detection mechanisms. In order to develop a very comprehensive secure routing mechanisms against the potential malicious behaviour of the WMNs participating nodes, it is imperative to explore solutions that have examined/addressed the underlying non-malicious factors that could result in packet drop in WMNs and the existing secure routing solutions that have attempted to mitigate the potential malicious packet drop attack in WMNs. The survey of the non-secure and existing secure routing approach are presented in Sections 3.2 and 3.3 respectively.

3.2 Survey of Non-secure effective Packet Routing Approaches in WMNs

As explained in (Morote, 2011), conventional ad hoc routing protocols such as AODV operate at layer 3 of the network protocol stack to achieve their multi-hop routing. Wireless links are transient in nature and otherwise, less reliable compared to wired links. The 802.11 wireless standard does not specify the interfaces that the IP layer needs to derive link metrics from the media access control (MAC) layer. The ad-hoc routing protocol, AODV developed by the Internet Engineering Task Force's (IETF's) Mobile Ad-Hoc Networks (MANET) group is forced to rely on indirect measurements in sensing the radio environment. The acquired link metrics usually present a limited accuracy. On the other hand, the MAC layer has adequate knowledge of its radio neighborhood to make its measurements less outdated and more precise. Thus, the IEEE 802.11s specified HWMP as a MAC-based multi-hop routing protocol for wireless mesh networking. The HWMP layer 2 routing protocol is inspired by AODV. The routing metric used by HWMP is Air Time Link Metric (ALM) while the routing metric used by AODV is hop count metric.

Therefore, in (Morote 2011), an extensive evaluation of IEEE 802.11s Mesh Networking was presented. The aim of the study was to evaluate the IEEE 802.11s defined/specified HWMP MAC-based routing protocol against the IP layer based AODV routing protocol in a WMN environment. The method of experimentation adopted was simulation using NS-3 network simulator. The figures of merit measured were packet delivery fraction (PDF), average throughput, average end-to-end delay and average routing load ratio. As analyzed from the results of the simulation experiments of the study, the layer 2 HWMP routing protocol outperformed AODV in terms of end-to-end delay and routing load. The difference was said to present more increase when the number of nodes and connections were high. However, in the simulation experiments of the study, both HWMP and AODV maintained a similar results in terms of packet delivery ratio. On the other hand, the author reported that in some congestions scenario, the air time link metric performed slightly worse compared with normal hop count metric. The study, however, concludes that MAC layer showed advantages against the conventional layer 3 routing despite the difference from which the routing metric was responsible. The author maintained that the benefits would have high importance for real time applications.

Furthermore, to deploy a scalable WMN that can achieve reliable performance coupled with multi hop nature of packet transmissions in the network, it is important to take into cognisance the routing protocol to adopt. In (Zakaria *et al.*, 2013), the impact of routing protocols in WMNs was studied. The routing protocols evaluated were AODV, Optimized Link State Routing Protocol (OLSR) and HWMP. The evaluation was done in a simulated environment in a context of community-based WMNs with static mesh STAs. The simulation tool adopted for the evaluation was Qualnet 5.2. In the study, parameters measured against throughput and end-to-end delay were traffic loads, network sizes and number of sources for each evaluated routing scenario. The results of the findings based on the measured parameters enumerated that HWMP outperformed AODV and OLSR as it achieved the highest average throughput at lowest end-to-end delay as compared to AODV and OLSR.

As explained in (Mbougni and Ekabua, 2012), routing protocol are essential for achieving optimal quality of service in WMNs. The authors identified that the common hop count routing metric used by routing protocol for selecting best routing path is not efficient enough in

achieving optimal route selection for packet forwarding from the source to multi-hop destination in WMNs. Therefore,(Mbougni and Ekabua, 2012) proposed a novel routing metric to improve the Multi-hop Effective Bandwidth Routing Metric (MHEB) (Li, Cheng and Zhou, 2008). The proposed routing metric called *i*MHEB considers the link quality, and inter-flow and intra-flow interferences in the WMNs. The study further evaluated the effectiveness of the proposed routing metric against MHEB and Weighted Cumulative Expected Transmission Time (WCETT) (Draves, Padhye and Zill, 2004) routing metrics using AODV as the base routing protocol. The evaluation was simulation based using OPNET network simulator. The performance evaluation metrics used were throughput and routing overhead. The result of their findings depicted that the novel proposed routing metric *i*MHEB outperformed both MHEB and WCETT routing metrics in terms of throughput. On the other hand, in the representation of their findings, *i*MHEB experienced highest routing overhead as compared with MHEB and WCETT. The authors, however, concluded that *i*MHEB achieved link quality awareness, load balancing awareness, and inter-flow and intra-flow interferences awareness.

According to (Houaidia *et al.*, 2012, 2013), guaranteeing efficient data routing in the network generally requires the proper study of the impact of environmental factors and PHY/MAC attributes on higher layers, and adaptation of the design of the routing metric in order to effectively control the influenced parameters. As a result, (Houaidia *et al.*, 2013) presented the investigation of the efficiency of conventional routing strategies under lower layers to determine the relative impact of the choice of PHY/MAC/Routing protocols on the performance of the WMNs. The protocols considered in their study include three propagation models (FreeSpace, TwoRayGround and Shadowing), three different PHY/MAC protocols specified IEEE 802.11 standards (802.11b (Marks, 2002) 802.11s (IEEE, 2012) and 802.11n (Banerji and Singha Chowdhury, 2013)), and three routing protocols, AODV, OLSR (Clausen and Jacquet, 2003) and HWMP. Moreover, the method of their investigation was simulation-based using NS-2. The performance evaluation metrics adopted in the study include end-to-end delay, loss rate, normalized routing load and throughput. In the presentation of the authors' simulation experiments results, it was revealed that the propagation models have significant effect on the performance of the WMNs. Also, their findings showed that the throughput achieved by MIMO (Multi Input Multi Output) technology can generally enhance the performance of the network in terms of the throughput of data transmissions in the network. In all cases, their simulation

experiment result showed that the network obtained best performance in terms of end-to-end delay under the IEEE 802.11n protocol due to the low end-to-end delay achieved by IEEE 802.11n protocol. Moreover, the result of their experiments from the routing perspective indicated that the general performance of OLSR was higher especially when the network gets denser. The authors, however, suggested the need for further study of other topologies to validate their conclusion. The authors, further explained that proactive routing protocols could be less efficient in dynamic networks compared to reactive protocols especially in terms of overhead and routing load. The authors, based on the performance evaluation results of their study, argued that the scalability of HWMP is not guaranteed as it was affected by the network traffic size, and that the network size would have to be adjusted to maintain optimal performance in HWMP-based WMNs. Summarily, (Houaidia *et al.*, 2013) maintained that all parts and parameters of the network protocol stack must be considered to achieve optimal performance in WMNs.

Relative to the work presented in (Houaidia *et al.*, 2013), (Houaidia *et al.*, 2012) presented a measurement-based performance evaluation of the OLSR protocol. In this aspect of the work, three versions of OLSR were configured and evaluated to comparatively analyze the performance of Hop-Count, Expected Transmission Count (ETX) and Expected Transmission Time (ETT) routing metrics in a testbed environment. As shown from their performance evaluation results, OLSR-ETT outperforms OLSR-ETX and OLSR-Hop-Count in terms of packet loss, delay and load balancing. The authors stated that the results obtained are related to the considered topology and suggested further study of other topologies is needed to validate this conclusion. In a further study, (Houaidia *et al.*, 2012) proposed two novel link quality aware routing metrics as the improvement of the existing routing metrics. The first routing metric is a load-sensitive and additive metric that fairly distributes the traffic load between participating network node and concurrently/at the same time considers their occupancy and availability. The second routing metric is a concave metric based on residual link capacity estimation. According to the authors, the concave metric means that the total cost of a path is the minimum of the costs of individual links along the path. Also, the metric represents the precise amount of additional traffic the link can support. The two proposed metrics were said to be based on real traffic estimation and they are updated periodically using the control messages of OLSR. Further in their study, (Houaidia *et al.*, 2012) presented experiment-based performance evaluations of the proposed routing metrics. Based on the results obtained from the first set of the performed

experiments, the authors related that LOM and RLC outperformed ETX as they enhanced better performance of the nodes when they were engaged with large/considerable amount of data flows concurrently. The authors also argued, based on the obtained results that the Residual Link Capacity based routing decision is more reliable as it considers the bandwidth heterogeneity between links better.

However, the surveyed works based on effective routing approach in WMNs were focused on the improvements of packet routing in WMNs without considering the security aspect of the routing protocols. The work presented in (Morote, 2011) was based on comparative analysis of the IEEE 802.11s MAC based routing and AODV routing. The work inferred that HWMP MAC-based routing approach could provide more effective packet routing in WMNs compared to the traditional AODV routing. Also, (Zakaria *et al.*, 2013) investigated the impact of routing protocols on packet transmissions in WMNs, the study was able to conclude that HWMP could provide better routing approach in WMNs compared with the AODV and OLSR routing. Although, it was revealed in the studies presented in (Morote, 2011; Zakaria *et al.*, 2013) that HWMP MAC-based routing could provide more effective routing in WMNs, HWMP is a non-secure routing protocol that could not detect and mitigate the malicious packet drop attack. The work presented by (Mbougni and Ekabua, 2012) only focused on path selection metric that is aware of the link quality and inter-flow and intra flow interferences for optimal path selection in the WMNs. The secure routing approach was not considered in the proposed routing metric. In addition, (Houaidia *et al.*, 2012, 2013) was able to establish that all parts and parameters of the network protocol stack must be considered to achieve optimal routing performance in WMNs. The authors further presented a routing metric that could enhance effective packet forwarding in WMNs. The authors' investigation and proposed effective routing approach did not include implementing a secure routing approach that is capable of detecting and mitigating the malicious packet drop attack in WMNs.

On the other hand, it is important to identify that improving the routing capabilities for effective packet routing without considering the security of the routing protocols, would expose the network to vulnerabilities that could be exploited by malicious or selfish node in launching malicious attack such as packet drop attacks. To provide a more effective packet transmission that is secure and reliable in WMNs, it is therefore imperative to consider the security aspect of

the routing protocols. The surveyed work related to providing effective packet transmission in WMNs without secure routing would, however, be instrumental in designing and implementing comprehensive secure routing algorithm or improving on the existing secure routing approaches in WMNs. The next section presents the survey of existing secure packet routing approaches that can be adopted in WMNs while analysis of the impact of malicious packet drop attack presented in Chapter Six.

3.3 Survey of Existing Secure Packet Routing Approaches in WMNs

In this section, WMNs existing secure routing related studies are discussed. The study in (Saxena, Denko and Banerji, 2011), proposed a secure routing mechanism in the context of community based WMNs. In the study, the mesh routers are classified as managed and unmanaged mesh routers. According the authors, the managed mesh routers are assumed to be trustworthy since they are managed by the Internet Service Provider (ISP). On the other hand, the unmanaged mesh routers are mesh routers deployed by independent users to extend the mesh network. As related by the authors, the unmanaged mesh routers are believed to likely exhibit intentional packet drop behavior to increase their portion of the available network bandwidth. In order to address the potential selfish behavior of the unmanaged mesh routers, (Saxena, Denko and Banerji, 2011) proposed secure routing architecture adopts a distributed detection approach by organizing the mesh routers into a set of manageable clusters. In addition, the architecture defined two agents called monitoring agent and sink agent. The monitoring agents are hosted at the cluster head of each cluster and they are responsible for the local collection of the periodic reports generated by each mesh router about its neighbor mesh router in their cluster. The sink agent is hosted at the mesh gateway. Based on the report received about a mesh router from the monitoring agent hosted at its cluster head, the sink agent is responsible to decide if a mesh router is behaving selfishly or not. The authors maintained that delegating the local collection of the mesh router's behavior to the monitoring agent enhances less detection overhead in the secure routing mechanism.

Also, the proposed secure path selection architecture incorporates link quality measurement in its detection mechanism in order to avoid false positive detection rate. Moreover, the study presents simulation based experiments to evaluate the effectiveness of the proposed secure routing

protocol and architecture. The measured performance of metrics include packet delivery ratio, false positive detection rate and hop counts. According to their simulation experiment results, the authors argued that the proposed mechanism provide improvements against selfish behavior of selfish mesh routers compared to existing secure routing mechanism considered, D-SAFNAC (Santhanam *et al.*, 2006) and the unprotected mesh network. However, the hierarchical securing routing and architecture proposed in this study may become ineffective while the sink agent hosted at the mesh gateway is unreachable because of the transient nature of the wireless links between the nodes in the network. Considering the incorporation of autonomous/hybrid detection mechanism for the detection of malicious nodes would be well suited for the robustness of the proposed secure routing and architecture.

Another important work worth discussed is the study presented in (Sarao and Garg, 2014) in which the authors propose fuzzy logic based approaches to enhance the performance and reliability of WMNs in the presence of malicious and selfish nodes. In this approach, three variables: trust value, hop count value and route value were defined to find a secure and stable route from source to destination. According to the authors, the trust level used in selecting a reliable and secure route between the communicating nodes is not predefined. It is computed using the fuzzy logic trust evaluation technique modelled in the study using fuzzy logic toolbox in MATLab 7.0. In order to assign trust levels to the WMNs participating nodes in the proposed trust model, the fuzzy trust method keeps track of the information regarding the nodes' behaviour.

As indicated in the study, the trust value of each participating node *i* to its neighbor node *j* is computed considering the packet dropped by the node, packet forwarding to the wrong direction and regular packet delay. Therefore, every node adds its trust value and hop count value in RREQ packet at the time of discovery process in the network. Based on the route value, the path or route for data communication purposes is then decided at the destination. The route with greater value is selected to route data packets from source to destination. To investigate the effectiveness of the proposed enhanced secure routing protocol against conventional AODV routing protocol, the study also presented a simulation experiment using MATLAB 7.0 simulator. The metrics of performance evaluation adopted for the comparative analysis include packet delivery ratio, throughput and end-to-end delay. According to the results of the simulation

experiments, the authors (Sarao and Garg, 2014), argued that the proposed routing technique has significant performance and reliability enhancement in comparison with AODV.

In (H Hallani and Shahrestani, 2008; H. Hallani and Shahrestani, 2008; Hallani and Shahrestani, 2009) proposed a Fuzzy Trust Approach (FTA) that adopts the principles of fuzzy logic to establish trust relationships between wireless ad hoc participating nodes. In the proposed trust evaluation approach, the quantification of trust levels for a node is facilitated by collecting information about the history of the node's behaviour. The nodes' types of misbehaviour considered include random packet drop, packet forwarding to wrong destination, fabricated and falsified routing massages, and launching of replay attacks. According to the authors, the trust level that can be assigned to a node is not a crisp value due to various conditions that can affect the trustworthiness of the nodes. To initiate the trust level evaluation process, the combined information related to the four considered attacks were modelled by monitoring the neighbouring node using the fuzzy logic trust approach. Therefore, in the trust evaluation model, the trust level of a node is determined by keeping track of the percentage of packets dropped, the percentage of packets forwarded to the wrong destination, the number of replay attacks generated by the node, and the number of false routing messages produced by the node. The percentage of the misbehaviour are considered as fuzzy input variables characterized by Guassian membership functions, while the output variable of the model is represented as the trust level. According to the authors, the trust level evaluation was modelled using MATLAB.

Furthermore, the trust evaluation model is focused on the on-demand routing protocols in which route discovery is initiated when a source node needs to send data to a destination node. The base on-demand routing protocol adopted was AODV routing protocol. Based on the proposed FTA model, each route is assigned a trust level. The trust level assigned to a route is decided with respect to the node that has the lowest trust level in the route. Thus, the route with the highest trust level between the source and the destination nodes is comparably selected as the most reliable and secure route. Further, the studies present simulation experiments in OPNET Modeler to compare the effectiveness of the proposed secure routing protocol against AODV routing. The metrics of performance considered include throughput, round-trip delay and packet loss rate. The performance metrics were then combined into Overall Performance Index (OPI) to facilitate the comparison between AODV and the proposed FTA. Based on the comparison of the OPI values

obtained for both AODV and FTA enhanced on-demand routing protocol, the authors argued that the proposed FTA model improved the performance of the simulated networks in the presence of selfish and malicious nodes compared to AODV routing.

Moreover, the work presented in (Nie *et al.*, 2006) proposed a Fuzzy Logic Based Security-Level (FLSL) routing protocol. The proposed secure routing protocol is a security-level based distributed ad hoc routing protocol that operates as a source-initiated on-demand routing protocol. In the protocol, the Security-Level of a route is decided by the node which has the lowest Security-Level in that route, therefore, the node with the lowest Security-Level in a route with the highest Security-Level route has higher security level than the node with the lowest Security-Level in other routes. Thus, the route with the highest Security-Level is comparably most secure. The study further investigates effectiveness of the proposed secure routing protocol against conventional AODV routing protocol in a simulation based environment. The metrics of performance evaluation include average security-level of the route from source to the destination and packet delivery ratio. The authors, based on the results obtained from the simulation scenarios maintain that FLSL routing protocol is more secure and provides a higher packet delivery ratio compared to AODV routing protocol.

However, the work presented in (Nie *et al.*, 2006; H Hallani and Shahrestani, 2008; H. Hallani and Shahrestani, 2008; Hallani and Shahrestani, 2009; Sarao and Garg, 2014) did not incorporate non-malicious factor such as link quality measurement to differentiate between packet drop due to malicious attack and the packet drop due to the condition of the link in their detection mechanisms. Hence, high false positive detection rate is probable in the proposed secure routing mechanism against malicious packet drop attack. Considering the measurement of link quality would intensify robustness of the proposed secure routing methods.

(Marti *et al.*, 2000) proposed two techniques called watchdog and pathrater to provide a secure routing mechanism against packet drop attack in a multi-hop wireless ad hoc network environment using Dynamic Source Routing (DSR) (Johnson, Maltz and Josh, 1996) as the base routing protocol. In the secure routing approach, the watchdog maintains a buffer for the recently transmitted data packets and compares it with the overheard packets. The packets in the buffer are discarded on the basis that there is a match between the buffered and overheard packet. If a

packet has been otherwise kept in the buffer beyond the defined time limit, the node that is responsible for the relay of the packet earns an increment in its failure tally. Thus, a node is identified as a misbehaving node if it has exceeded the threshold defined for the failure tally, and an alert is sent to the source node about the misbehaving node.

On the other hand, the pathrater is adopted in the proposed secure routing mechanism to choose the most reliable route for data packet transmission based on the information gathered through the watchdog about a node, and information about the link quality. The path rather is run on every node which as a result, each node is able to maintain a rating for every one of its neighbor nodes. To compute the path metric along a path, the pathrater calculates/finds the average of the ratings of the nodes along the path. According to the authors, the metric gives a comparison of the overall reliability of different paths and enables pathrater to adopt the shortest length path algorithm when no reliability information has been collected. Therefore, the path with the highest metric is chosen among (or in a case that there are multiple paths) multiple possible paths to the same destination. The authors indicated that the path selection approach of the proposed secure routing method is different from that of the conventional DSR, which chooses the shortest path in the route cache. The authors further advocate/recommend that the routing approach must be adopt a source routing protocol as the base routing protocol since the pathrater depends on the knowledge of the exact path a packet has traversed.

Furthermore, (Marti *et al.*, 2000) present a simulation based analysis to investigate the effectiveness of the proposed secure routing protocol. The metrics of performance measurement adopted include throughput, overhead and the effect of wathchdog false positives on the network throughput. Based on the performance evaluation results, (Marti *et al.*, 2000) resolved that the two proposed techniques (watchdog and pathrater) can improve the network in an increased number of routing nodes while minimizing the effects of the misbehaving nodes. However, the authors identified that, while DSR with watchdog is advantageous at detecting the misbehavior at the forwarding level and link level, the weakness of watchdog may include its inability to detect a misbehaving node in the presence of conditions such as ambiguous collisions, receiver collisions, limited transmission power, collusion and partial dropping. Thus, the highlighted limitations of the proposed secure routing method need to be addressed to improve the proposed secure routing.

(Sbeiti *et al.*, 2011) proposed a novel hierarchical Position Aware Secure and Efficient Route discovery protocol (PASER) for wireless mesh networks. As part of the effort to make routing efficient and reliable in WMNs, the routing mechanism of PASER avoids the shortcoming of the proactive routing element of routing protocol such HWMP. As explained by (Sbeiti *et al.*, 2011), the proactive element is encouraged in WMNs to maintain a steady route to the gateway. On the other hand, proactive routing is resource consuming as it's always active when the route is not needed. Therefore, PASER was designed to adopt reactive route discovery approach in two ways. Firstly, the Mesh routers are assigned the responsibility of maintaining individual route to the gateway. Hence, the resources are used on demand while the benefit of the proactive element of the hybrid routing protocol is still adopted. Secondly, when it is necessary, the route requests are sent adopting unicast method instead of the traditional flooding approach used in the reactive route discovery technique.

According to (Sbeiti *et al.*, 2011), the novelty of the routing approach in terms of security is to provide a hybrid approach to secure the route discovery process in WMNs context. PASER secure routing mechanism involves a combination of asymmetric and lightweight symmetric cryptography to secure route discovery packets. Also, PASER includes/incorporates the exchange of geo-positioning to mitigate a large number of attacks and aids the management of the network. The cryptography-based security approach of PASER is based/dependent on a key distribution centre (KDC). Based on the hop-to-hop trusted relation, PASER aims to achieve node authentication, message authentication, message confidentiality and neighbor transmission authentication. Hence, PASER is a secure WMN protocol that is robust in providing security against attacks such as impersonation and malign attacks, replay and tempering attacks, man-in-the-middle attacks, black hole attacks, gray hole attacks, flooding attack and wormhole attacks. The efficiency of PASER was evaluated in (Sbeiti, M., Pojda, J., Wietfeld, 2012; Sbeiti, M. and Wietfeld, 2013; Alanazi *et al.*, 2016; Sbeiti *et al.*, 2016).

In (Sbeiti, M., Pojda, J., Wietfeld, 2012), PASER was investigated against reactive routing protocols, AODV and DYMO, and proactive routing protocols, BATMAN and OLSR. The evaluation of the routing protocols was done in the presence of impersonation, isolation and wormhole attacks. The performance evaluation was done both in an experimental testbed and simulation environment (using OMNET++ and INEMANET framework). The authors defined

impersonation attack as a type of malicious behaviour in which the attacker has the same IP-address as the gateway. Hence, the attacker would be able to reply to the route requests forwarded to the gateway in order to redirect data traffic of the requesting node to itself. The isolation attack was defined to be a malicious behaviour that could be launched by an attacker to isolate the gateway in such a way that all the network nodes would continue to use the attacker as the gateway. To achieve this, the attacker keeps broadcasting corrupt packets with very high sequence number. Consequently, the gateway packets with lower sequence numbers are discarded by the network nodes.

As explained by the authors, in the case of AODV and DYMO routing protocol which are reactive routing protocols, the gateway only replies to a route request. Thus, the impersonation is carried out by the attacker as it continuously floods route requests to the entire network with the address of the gateway, to search for unknown nodes. The nodes are, therefore, deceived into registering the route to the attacker as a valid gateway-route and all their packets are then transmitted to the attacker. Thus, the PDRs of both routing protocols were decreased to 0%. The wormhole was defined as a pair of attackers linked through a fast transmission path to forward route requests faster than the legitimate nodes. The authors, through the results of their findings, argued that PASER was able to achieve a better tradeoff between security and network performance. The findings of the study revealed that PASER achieved a comparable performance with the reactive and proactive routing protocols, while it was able to prevent the routing attacks in the network, compared to the other routing protocols evaluated in the study.

In (Sbeiti, M. and Wietfeld, 2013), PASER, in the case of wormhole attack, was evaluated against HWMP, BATMAN and OLSR in a simulation environment and an experimental testbed. The simulation experiment was done using INETMANET framework in OMNET++. The performance evaluation metric adopted for the performance analysis was packet delivery ratio. The results of the findings both in simulation and experimental testbed showed that PASER was able to protect the network against wormhole attack compared to HWMP, OLSR and BATMAN.

Furthermore, (Sbeiti *et al.*, 2016) analyzed the PASER secure routing approach in the context of an Unmanned Aerial Vehicles connected via WMN (UAV-WMN). In the study, the efficiency of PASER route discovery process, and its scalability with respect to network size and traffic load

was examined in a theoretical and simulation-based analysis. The evaluation was done using OMNET++ network simulator realistic mobility patterns of UAVs, and an experimentally derived channel model of UAV-WMN. The authors, through their findings argued that PASER, in UAV-WMN-assisted network and area exploration scenarios maintained a comparable performance with HWMP routing protocol integrated with the IEEE 802.11s security mechanisms In the investigated scenarios, the results of the study showed that PASER was efficient in mitigating more attacks than the secure routing mechanism of Authenticated Routing for Ad Hoc Networks (ARAN) (Sanzgiri *et al.*, 2005) and the specified security mechanisms of IEEE 802.11s/i.

In (Alanazi *et al.*, 2016), PASER was comparatively analyzed against AODV routing protocol in the context of static and mobile WMNs using OMNET++. In the study, the effect of three types of DoS attacks namely; hello flooding, selective forwarding and wormhole attacks was investigated in the configured static and mobile WMNs. The metrics of performance measurements were packet delivery ratio, throughput, and end-to-end delay of the network. In the results of the analysis, the PDR and throughput performance metrics were considerably affected by the attacks while AODV was adopted. On the contrary, the results showed that PASER, compared to AODV, was effective in preventing the evaluated networks from the launched attacks. The authors, however, maintained that PASER imposed a performance cost on the PDR and end-to-end delay because of the frequent usage of cryptography. (Alanazi *et al.*, 2016) further argued that the assumption that KDC exists for the issuing of key to nodes in PASER tends to limit the dynamic enrollment of nodes in WMNs since the existence of KDC is not usually guaranteed in WMNs.

(Kolade *et al.*, 2016) proposed a bait request algorithm for the mitigation of black hole packet drop attack using AODV as the base routing protocol. The proposed secure routing mechanism against black hole attack requires each participating node to detect and isolate the misbehaving node in its local transmission range. The operation of the mitigation approach was categorized into two phases. In the first approach, the intermediate node examines the received RREP packet to verify if the destination sequence number is maximum and the hop count is minimum. If after the check, the contained RREP's sequence number and hop count values are suspiciously maximum and minimum, the intermediate node would then buffer the received RREP and

initiate a local detection operation. Thus, the intermediate node creates a broadcast bait request (BRQ) packet to be forwarded to all its neighbour nodes. In the created BRQ, the destination address is set to the address of one of the intermediate node's recognized neighbour while the TTL (time to live) is set to 1 to control the transmission of the bait RREQ broadcast in the local transmission range of the intermediate node. To detect the malicious node, the intermediate node compares the sequence number contained in the bait reply (BRP) received from the suspected node and the destined node of the bait RREQ. If the destination-sequence-number of the RREP received from the suspected node is larger than destination-sequence-number received from the destination node, the suspected node is tagged as malicious. Thus, the initial buffered RREP received from the suspected malicious node is discarded. The suspected black hole node is therefore, added to the malicious list and alert packet is sent across the network.

As explained by (Kolade *et al.*, 2016), the second mechanism of the proposed secure routing approach was aimed at detecting and isolating black hole nodes by monitoring the forwarding behavior of all participating nodes in the network.. This is achieved as each participating node in the network is enabled to keep track of its neighbour node forwarding history. Therefore, two new parameters were added to the neighbour table of each node. The first parameter records the number count of packets forwarded by a node to its neighbour node. The second parameter is used to count the number of packets overheard by the node from its neighbour node. That is, the count of packets further forwarded by the neighbour node. Each time a data packet is forwarded from a node to its neighbour, it increments the forward count, fvcount for that neighbour in its neighbour table. As further explained, every participating node is expected to forward the packets that are not destined for it towards the packet destination. Therefore, after forwarding the data packet, the node overhears the transmission of the neighbour to verify that the data packet sent is correctly forwarded by the neighbour towards the destination node. Thus, the node increments the overhear count (ovcount) for the neighbour if the neighbour node had correctly forwarded the received data packet towards the destination.

Therefore, the trustworthiness of a node is rated based on its dropcount value. A node computes the dropcount value of its neighbour as the difference of packets forwarded to that neighbour and those forwarded by the neighbour. At each interval, each node accumulates the number of dropcount for each of its neighbour. The authors maintain that the dropcount for a node is

expected to be low for a genuine node in normal conditions, while on the contrary, it would be higher for a malicious node. At each interval, if the dropcount for a node exceeds the threshold, the node is considered as malicious. Therefore, an alert packet is sent to notify the network when a node is identified as such. As each node receives the notification, the node updates the address of the malicious node into to its malicious table. Subsequently, the routes along the malicious node are removed from the routing table. All future messages from malicious nodes are discarded and not processed. The proposed bait request secure routing methods is still at early stage of development which as a result, the effectiveness against black hole malicious activities have not be investigated. The authors, however, argued that the proposed secure routing approach is believed to improve PDR and throughput of the network in the presence of malicious black hole nodes.

However, the limitation of the proposed bait request secure routing algorithm lies in the exemption of the underlying non-malicious factor such as link quality measurement to differentiate between packet drop due to malicious attack and the packet drop due to the condition of the link in its detection mechanisms. The proposed secure routing mechanism against malicious packet drop attack is prone to high false positive detection rate. Incorporating the measurement of the link quality would provide more accurate detection rate in the proposed bait request secure algorithm methods.

3.4 Summary

In this chapter, the survey of related studies on non-secure improved routing approach and existing secure routing approach in WMNs was presented. Section 3.1 presented an overview of the chapter. In Sections 3.2, related non-secure improved routing studies were presented. Section 3.3 discussed the existing secure routing methods. In the survey, it was revealed that the malicious nodes launching packet drop attack can be identified and isolated adopting cryptography-based approach that depends on a dedicated centralized key distribution centre in the network. In addition, most of the existing secure routing mechanisms are based on trust evaluation of individual mesh node in the network. Thus, trustworthiness of mesh nodes is determined by their packet forwarding statistics. However, the existing solutions are liable to trade-off between security and quality of service (QOS) and issues such high false positive

detection rate. Though the review of the previous studies might not have been exhaustive, it has been revealed that there is a need for more concerted efforts to address routing attacks such as malicious packet drop attack. A major effective way to provide an effective, secure and reliable routing in WMNs is to consider the incorporation of the existing conventional improvement to packet routing in the literatures with any proposed secure routing approach in WMNs. The Next chapter is focused on the research methodology adopted in this research.

CHAPTER FOUR

RESEARCH METHODOLOGY

4.1 Introduction

This chapter presents the research methodology. In Section 4.2, various research methods are discussed, Section 4.3 presents the research process. Section 4.4 captures the research design and research techniques. In the section, possible network performance evaluation methods are discussed. Among possible network performance evaluation methods discussed, simulation model that was adopted as the research method is extensively explained. The explanation also dwells partly on the benefits of the simulation method. Section 4.5 presents the research methodology which largely describes the steps followed specifically for simulation experiments in this research. Moreover, the network simulator (NS-3) adopted for the simulation experiments in this research is discussed in Section 4.6. Section 4.6 concludes the chapter.

4.2 Research Method

As explained in (Kothari, 2004), research approach can be basically classified into qualitative approach and quantitative approach. The qualitative research approach usually encompasses subjective assessment of attitudes, opinions and behavior. Research conducted using qualitative approach usually represents a function of the researcher's insights and impressions. The results generated from qualitative research approach could either be non-quantitative or in form which are not subjected to extensive quantitative analysis. The commonly adopted techniques in qualitative research approach are focus group interviews, projective techniques and depth interviews.

On the other hand, quantitative research approach involves the generation of quantitative form of data that can be subjected to extensive quantitative analysis. Furthermore, the quantitative research approach can be categorized as inferential, experimental and simulation research approaches. In the inferential research approach, the research method involves the formulation of database from which the characteristics or relationships of the sample population can be

deduced. In other words, inferential research approach involves the survey of the sample population through the use of questionnaire or observation to explore the characteristics and speculate the related characteristics of the studied sample population.

The experimental research approach involves the manipulation of some variables to observe their effect on other variables. Also, experimental research is usually conducted to test a hypothesis or theory. Experimental research can either be performed in the laboratory or in the field. While experiments performed in the laboratory usually support the control of the environment so as to distinguish the phenomena under investigation, field experiments allow less environmental control. As a result, laboratory experiment are usually easy to replicate, compared to field experiments. Simulation approach entails the development of a simulated environment that permits the generation of relevant data and information. This type of research approach allows the observation of dynamic behavior of a system and/or its sub-system under monitored scenarios. The simulation approach can further be adopted in developing models that can be used to study future conditions and it is widely used in the computer and communication networks research (Kothari, 2004; Hofstee, 2011; Ayash, 2014). The analysis of the impact of malicious packet drop attack on packet transmission in this study was achieved by adopting simulation research method.

4.3 Research Process

According to (Kothari, 2004), the process of conducting research effectively necessitates a series of steps and the desired sequencing of the adopted steps. In this research, the research process included the definition of the research problem; development of the research questions; itemization of the research objectives in order to answer the research questions; research design (which comprises of the literature review and simulation research design techniques); analysis of the research findings; and research conclusion. The use of the literature as the research technique include the preliminary research conducted to investigate possible approaches to execute the second and third objectives of this study. The use of literatures include the review of concepts in relation to WMNs and their routing techniques. The literature review was also adopted to explore previous research findings with respect to the conventional and secure enhancement to packet transmission in WMNs. The simulation research approach is primary to the performance analysis

of the networks while under attack and free from attack and it includes simulation based experiments performed to achieve the performance analysis of the networks. The simulation research approach was used because, the performance analysis involves the manipulation of variables in the network model and observation of the resultant effect. The research process is represented in Figure 4.1.

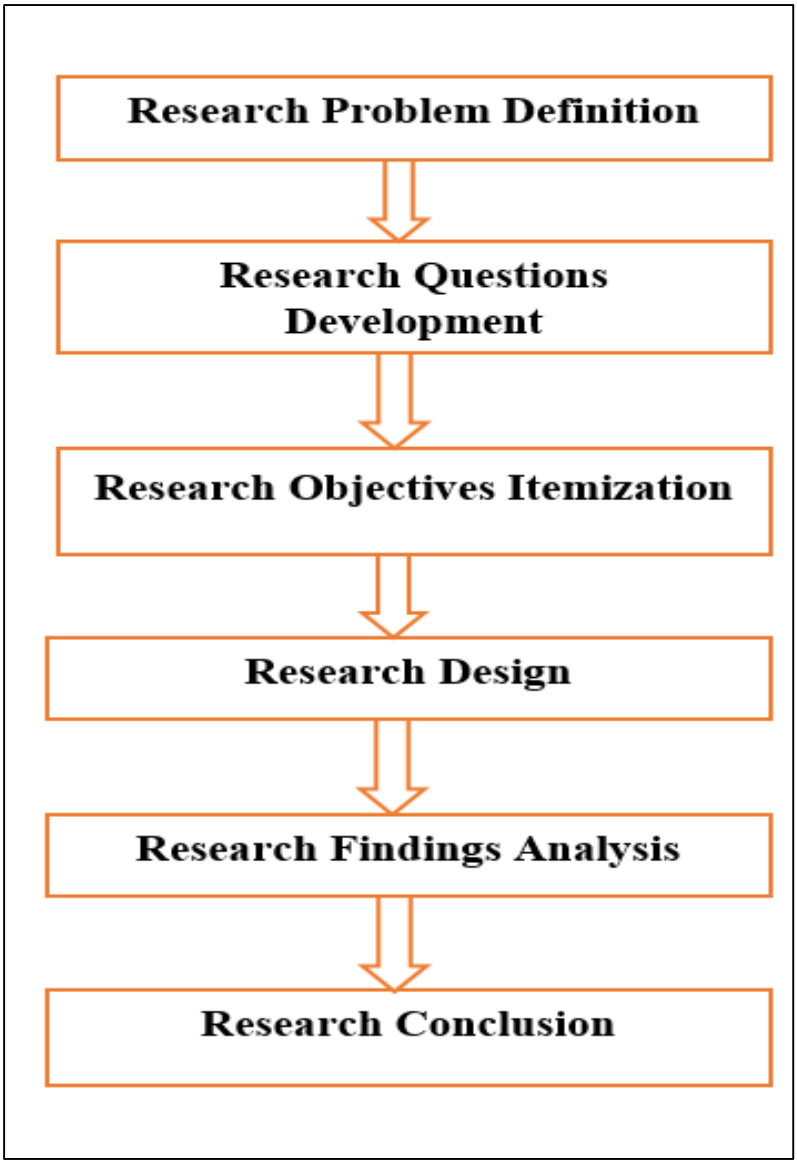


Figure 4. 1: Research Process

4.4 Research Design

This section presents possible research design techniques that can be adopted in the performance analysis of communication networks such as WMNs and further explanation of the simulation research approach primarily adopted in this research.

4.4.1 Overview of Possible Research Methods

As explained by (Hillston, 2001), performance evaluation involves the description, analysis and optimization of the dynamic behaviour of computer and communication systems. Again, performance evaluation can be widely utilized to extract reliable insight of network performance under real conditions (Anagnostopoulos and Nikolaidou, 2001). Other than communication networks, performance analysis is also paramount in studying any system that is costly to build and constrained in flexibility (Koksal, 2008; Taylor, Chick, *et al.*, 2013).

Moreover, performance evaluation is concerned with the systematic study of the flow of data, and control information, within and between components of a system. It is usually targeted at understanding the behaviour and exploration of the sensitive parts of a system in terms of performance. In addition, performance analysis might be focused on: specifying performance requirements, evaluating design alternatives, comparing two or more systems, determining the optimal value of a parameter (system tuning), finding the performance bottleneck (bottleneck identification), characterising the load on the system (workload characterisation), determining the number and sizes of components (capacity planning), and predicting the performance at future loads (forecasting) (Hillston, 2001).

Thus, the idea that a design should be subjected to some analysis to ensure that the resulting system will behave correctly is now widely accepted. The commonly used tools for the analysis purpose are models. Modelling is an applicable technique that is extensively adopted in providing solutions to many complex problems (Hillston, 2001; Taylor, Balci, *et al.*, 2013). The term model connotes a variety of meanings but it can also be termed as the abstract representation of a system in a form which allows predictions to be made about the behaviour of the system. A model can be constructed to represent some aspect of the dynamic behaviour of a system, and once it is constructed, such a model becomes a tool with which behaviour of the

system can be investigated (Balci, 2001; Hillston, 2001; Egea-Lopez and Vales-Alonso, 2005; Koksal, 2008; Ghaleb *et al.*, 2017).

According to (Owczarek and Zwierzykowski, 2014), the choice of evaluation model to study WMN routing protocols or routing metric can be made from different evaluation approach such as theoretical analysis, simulation models, emulation, virtualisation and real test-beds. These network evaluation models are explained in the following subsections.

4.4.1.1 Theoretical analysis

This is an evaluation method that involves using mathematical models to analyse network performance. Although theoretical analysis usually forms a vital part of the evaluation process, yet, formulating mathematical equations for performance evaluation becomes more complicated as the size of the state space of the network increases.

4.4.1.2 Simulations

Simulation is an approach in which special tools can be adopted to model a virtual environment to study the basic concept, detailed parameters and/or enhancement of a system, or to compare proposed solutions of a system. Through simulation, behaviour of distributed networks such as WMNs or WSNs can be analysed by varying some parameters as other parameters remain constant.

4.4.1.3 Emulation

Emulation is a hybrid environment that allows the adoption of the combination of both simulation and real systems for the investigation of network performance such as WMNs. In emulation, the choice of part to be simulated or made real can be made. Investigation results from emulation are more realistic as experimentation because of the integration of a real environment and simulation tools.

4.4.1.4 Virtualization

In virtualization, investigating network performance involves setting up a virtual environment where network hosts are run. The virtual environment is formed by installing virtual hosts on an

existing host rather than building new infrastructure involving multiple physical machines. Virtualization can include virtual hosts; virtual operating systems and all network equipment or that partially contains virtual hosts such as clients' hosts. Furthermore, virtualization can provide suitable tools for studying network protocols with multiple virtual hosts configured on a single physical host. As a result, the experiment can be run at a low cost.

4.4.1.5 Real test-beds

Evaluation process in test-beds is based on the implementation of prototype that is close to real-world system. If significant environmental influencing factors are well considered, ideas can be properly transformed to real-world system. Thus, presented results of analysis from test-beds are more realistic.

4.4.2 Simulation Model

Among the different evaluation methods mentioned in Section 4.4.1, simulation is the commonly adopted evaluation technique (Hillston, 2001). According to (Ghaleb *et al.*, 2017), simulation models have been broadly used as analysis tools in the computer network research. They are effective approach for analyzing algorithms and protocols at various phases of a project which may include design, development and implementation. (Gore *et al.*, 2015) explained that the predictions from simulations are now generally used in the mainstream of public policy and decision-making practices. Therefore, a simulation model could be described as a valid representation of some real or abstract systems (Nutaro and Zeigler, 2015).

According to (Egea-Lopez & Vales-Alonso, 2005, pp. 2), "simulation software commonly provides a framework to model and reproduce the behaviour of real systems". As further explained by (Perros, 2009), simulation techniques are easy to learn and are applicable to a wide range of problems. Simulation is usually a useful modelling tool when other models are not applicable. Also, simulation does not require building the entire real-life system under investigation. Rather, it involves simulating sub-systems which are related to the problems at hand, which involves modelling parts of the system at various levels of detail. Furthermore, simulation model is generally used to study real-life systems that do not currently exist. As a result, the performance of the system under study can be quantified for various values of its input

parameters. Thus the resulting quantified measures of performance can be useful in the managerial decision process.

4.4.2.1 Benefits of Simulation Models as Performance Analysis Tools

As explained in (Owczarek and Zwierzykowski, 2014), there are many advantages in using simulation tools for investigating network routing protocols operations, or any other related network performance. Among other benefits is the low cost of running simulations. According to (Kulgachev *et al.*, 2010), performance analysis with live equipment usually provides most reliable results, yet, the use of simulation tools for network performance evaluation is usually favourable in terms of saving cost of money and time, coupled with the capability of providing satisfactory precision on how a network would perform under particular settings or environments.

The use of a test-bed or prototype for investigating new concepts often requires redeveloping several modules or model in which a portion of the development can be financially or time costly (Al-Holou, Booth and Yaprak, 2000; Owczarek and Zwierzykowski, 2014). Adopting simulations allows utilizing existing models, thus, simulation helps in phasing out developing or redeveloping new models in the experimentation (Koksal, 2008; Owczarek and Zwierzykowski, 2014). Also, simulation has a benefit of maintenance simplicity. As it may be required, simulation can be run repeatedly to obtain realistic results. Moreover, simulation permits overseeing any phase of the investigation. Setting up a simulation environment for experimentation and results gathering is less demanding (Owczarek and Zwierzykowski, 2014).

4.5 Research Methodology

As mentioned earlier in Section 4.2, simulation research approach was adopted for achieving the primary performance analysis of the network in presence and absence of malicious nodes. The choice of adopting the simulation model was based on unavailability of real-life WMNs routing devices and the strength of the simulation technique discussed in Subsection 4.4.2. As explained in (Koksal, 2008), network simulation steps fundamentally include the development of a model such as the implementation of a protocol; creation of simulation scenario which may include network topology and traffic scenarios setup; and selection of statistics to be collected. Lastly,

the steps include the visualization and analysis of simulation results which may be executed after (or in some cases, during) simulation experiments.

Similarly, the steps followed in this research approach include the implementation of the routing protocols. The base routing protocol adopted was AODV routing protocol. AODV is one of the default wireless ad hoc routing protocols implemented in the network simulation tool (NS-3, which is discussed in Section 4.6) adopted in this research. However, by default, the NS-3 simulation tool does not include the model to simulate the potential black hole attack behavior of the participating WMNs nodes. Therefore, the protocol implementation phase included the implementation of a protocol that can exhibit malicious black hole attack in WMNs adopting AODV as the base routing protocol. In order to be sure that the protocol implemented was executed target functionalities, minimal simulation experiments were carried out on the protocol implemented. The black hole attack behavior implementation and the mini simulation experiments carried out are discussed in Chapter five.

Furthermore, after testing the protocol implementation, the actual simulation experiments phase include simulation experiments to evaluate the performance of WMNs while it is free from malicious black hole packet drop attack; and simulation experiments to evaluate the performance of WMNs while it's under malicious black hole packet drop attack. In the simulation experiments, the measured metrics of performance for each simulation scenario were collected for performance analysis as scheduled in the simulation implementation scripts of each scenario.

The last phase of the research method included the comparative analysis of WMNs performance based on the simulation experiments scenarios and the gathered metrics of performance. The simulation experiments setup, and results and discussions are presented in Chapter six. The simulation research method steps followed in this research are represented in Figure 4.2 below.

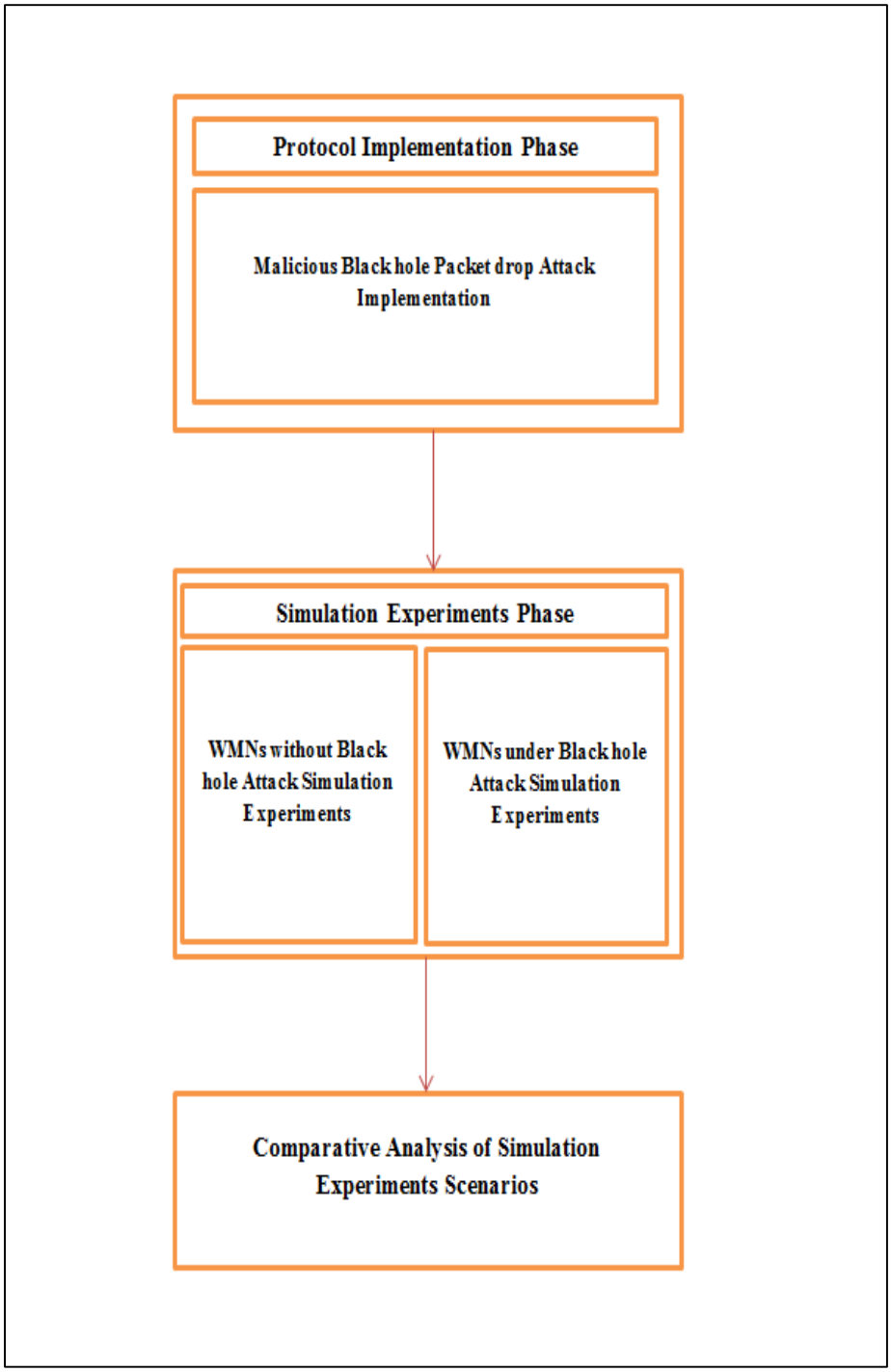


Figure 4. 2: Performance Evaluation Steps

4.6 Network Simulation Tool

In this research, the network simulation tool adopted for the simulation experiments was NS-3. The choice of simulation tool was based on the recommendation from the survey of network simulation tools for WMNs by (Owczarek and Zwierzykowski, 2014). In the survey, it was concluded that NS-3 was suitable for WMNs academic research. As explained in ('ns-3 Tutorial', 2016), NS-3 simulator is an open source (licensed under GNU GPLv2) discrete-event network simulator that is predominantly dedicated for research and educational purposes. It is a tool developed to provide an open and extensible network simulation environment for networking research and education. Again, NS-3 presents models and simulation engine to carry out simulation experiments on the operation of data packet networks. In addition, the available model set in NS-3 focuses on modelling how Internet protocols and networks work, while it can also be utilized to model non-Internet-based systems. Also, NS-3 includes a platform to investigate system operational behaviour in a highly controlled and reproducible environment. As a result, it is a useful network simulation tool that is suitable for complex system analyses or studies that are not feasible on real systems.

The distinguishing features of NS-3 over other simulation tools include its design as a set of libraries that can be combined with one another and other external software libraries. Moreover, Ns-3 is more modular in terms of user interface compared with other simulation platforms that provide users with a single, integrated graphical user interface environment in which all tasks are executed. Therefore, several external animators and data analysis and visualization tools (such as NetAnim and PyViz) can be used with NS-3. However, the primary user interface for NS-3 is based on command line and it is supported with C++ and/or Python software development tools. Furthermore, NS-3 is primarily Linux systems based, though it is supported on FreeBSD, Cygwin (for Windows), and native Windows Visual Studio support is in the development process. Thus, the version (ns-3.25) of the NS-3 simulator used for the investigation of WMN was being set up and configured on Ubuntu 14.04 LTS Operating System (OS).

4.6.1 NS-3 Organization

As explained in ('ns-3 Manual', 2017), the simulation core and models of NS-3 are implemented in C++. In addition, NS-3 is a network simulation tool that is built as a library that can be

statically or dynamically be integrated with a C++ main program that models the simulation topology and initiates the simulator. Also, NS-3 exports most of its API to Python which enables the import of Python programs to an NS-3 module almost the same way as the NS-3 library is linked by executables in C++. This is to say that users programs can be written in either the C++ or the Python programming language. The programming language adopted in NS-3 simulator in building the simulations topology in this research was C++.

Technically, NS-3 can be distributed as pre-built libraries for selected systems which give the hope that NS-3 could be distributed as pre-built libraries in the future, but currently, it is distributed as source code. Hence there is a need to install a software development environment on the target system/machine so as to initially build the libraries before user programs can be built. However, present distribution of NS-3 as a source code is useful to its users as NS-3 can be used by editing NS-3 source code itself (‘ns-3 Tutorial’, 2016). In this regard, the source code for NS-3 is mostly organized in the src directory (‘ns-3 Manual’, 2017).

Furthermore, NS-3 software is organized into separate modules. Each of the modules is built as a separate software library. Therefore, NS-3 simulation programs can individually link the modules (libraries) needed for the execution of the simulation experiments. Moreover, NS-3 models are abstract representations of real-world objects, protocols, devices, etc. As a result, an NS-3 module may consist of more than one model (for example, models for both TCP and UDP are contained in the internet module). However, NS-3 models do not generally span multiple software modules(‘ns-3 Model Library’, 2016). A representation of the NS-3 software organization is depicted in Figure 4.3 below.

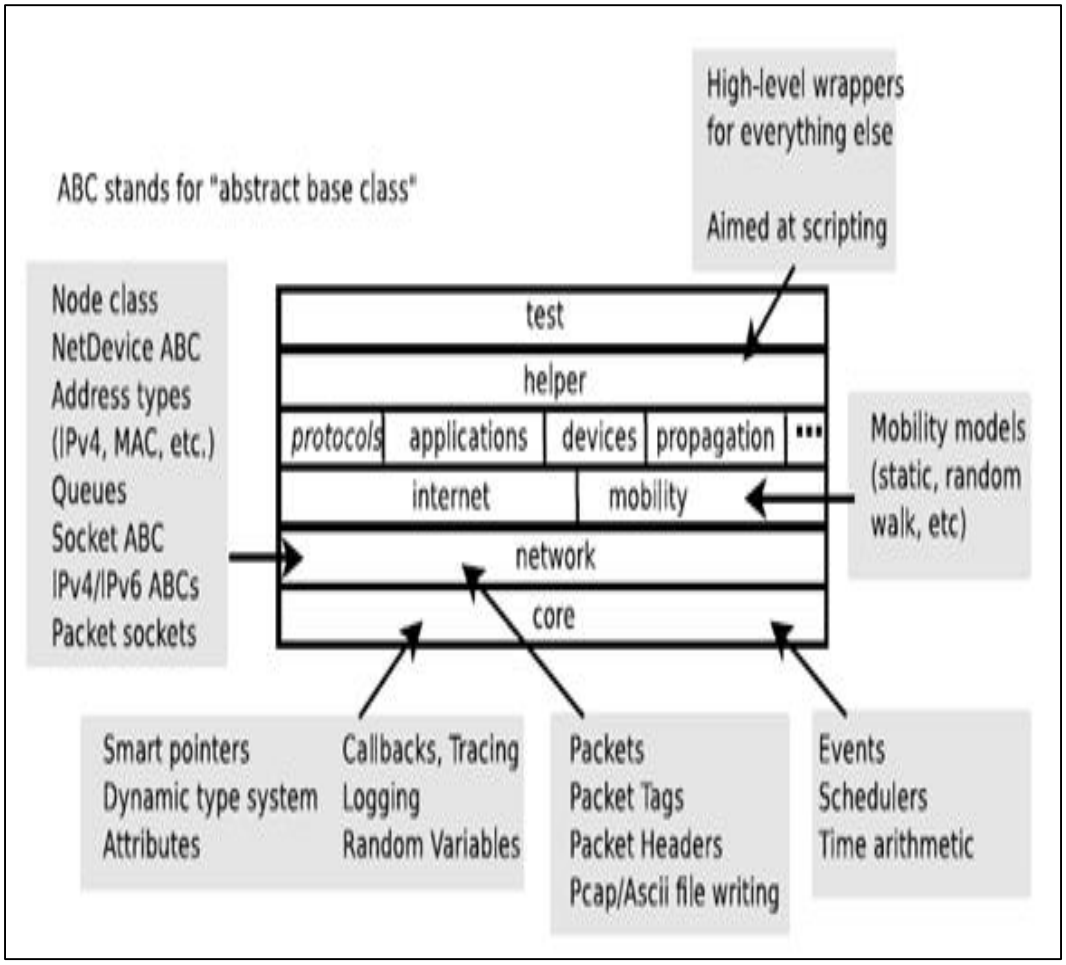


Figure 4. 3 Software organization of NS-3 (‘ns-3 Manual’, 2017)

Generally in the software stack represented in Figure 4.3, from the bottom upwards, modules are dependent on the modules below them. The simulation core is implemented in src/core directory. As important objects in a network simulator, packets models are implemented in src/network directory. Both core and network simulation modules are designed to comprise a generic simulation core that can be used across various types of networks other than the Internet-based networks. This is to say that the core and network modules of NS-3 are independent of specific network and device models. In addition, NS-3 programs may access the entire API directly or make use of helper API that provides convenient wrappers or encapsulation of low-level API calls.

4.6.2 Benefits of NS-3 for Networking Research

The available simulators have different features and distinct characteristics. An example is the NS-2 simulator, which is a very popular network simulator. NS-2 is complicated by a steep learning curve and requires advanced skills to perform meaningful and repeatable simulations. In addition, NS-2 has a number of restrictions on the energy model, packet formats and MAC protocols; and lacks an application model (Ghaleb *et al.*, 2017). NS-2 as the predecessor of NS-3 lacks some essential features such as interoperability, coupling between models, lack of memory management, debugging of split language objects or lack of realism (such as creation of packets). However, NS-3 provides more robust features as compared to NS-2. The main features include the new available high fidelity IEEE 802.11 MAC and PHY models coupled with real world design principle and concepts. Thus, Ns-3 is beneficial in conducting network research as it provides important tools to key aspects of Internet/networking research. As adapted from (Morote, 2011), among the benefits of Ns-3 network simulator are the followings highlighted:

- **Extensible software core**: NS-3 is written in C++ with optional Python interface and it's extensively documented API (doxygen).
- **Attention to realism**: Ns-3 models nodes are more like a real machine and support key interfaces such as sockets API and IP/device driver interface (in Linux).
- **Software integration**: Ns-3 conforms to standard input/output formats (pcap trace output, NS-2 mobility scripts, etc.) and adds support for running implementation code.
- **Support for virtualization and testbeds**: NS-3 develops two modes of integration with real systems. First, virtual machines can run on top of NS-3 devices and channels. Second, NS-3 stacks can run in emulation mode and emit/consume packets over real devices.
- **Flexible tracing and statistics**: NS-3 decouples trace sources from trace sinks making it possible to have customizable trace sinks.
- **Attribute system**: NS-3 controls all simulation parameters for static objects which make it possible to dump and read them all in configuration files.
- **New models**: includes a mix of new and ported models.

4.7 Simulation Experiment Flow Monitoring

As mentioned earlier in Section 4.5, the metrics of performance evaluation of each simulation scenario performed in this study were collected for the comparative analysis of the network while under malicious attack and without malicious attack. As it was also explained in ('ns-3 Tutorial', 2016), the main purpose of running simulation experiment is to generate results that can be further used to study the formulation of new idea, improved networking features, or behaviour of a system. This usually involves implementation of new idea and/or modification of some features of an existing system or a network; conducting simulation to observe the behaviour of the system under varying conditions; and collection of statistical features that represent the behaviour of the system under the varying conditions. According to ('ns-3 Manual', 2017), the feature that allows a researcher to generate performance evaluation metrics in NS-3 simulation environment is the tracing subsystem.

As explained in (Carneiro, Fortuna and Ricardo, 2009), tracing is an integral component of a simulator that can be adopted by users to explore/unearth important event running in the simulation and the conditions that influence them. Through tracing, researchers can obtain essential metrics from simulation experiments in order to comparatively quantify the effectiveness of a simulated model in relation to another model. In most simulators such as NS-2 which is the predecessor of NS-3, tracing usually involves generating a text file that describes a set of events with related time stamps and other related attributes where one event appears per line. Among other data that can be obtained from simulation tracing, the contents of the generated trace files usually include MAC layer transmission, reception, and queue packet drop.

Similarly in NS-3, the simulation events outputs can be generated into a text file which include minute subset of events detected/identified in NS-3. Also, the capture of simulation events in NS-3, especially packet transfer/receive events can be recorded as PCAP (Packet Capture) files. The NS-3 tracing subsystem is however, dependent on the NS-3 Callback and Attribute mechanisms. The tracing system is developed based on the tracing method that adopts independent tracing sources and tracing sinks coupled with a uniform method that links the sources to sinks. In NS-3, each possible trace source is associated with a unique object class which is identified by a name. Therefore, the user can define a C++ function or method to be

called when a particular (or a set of) trace source generates a new event while the data to be collected in the callback are decided by the user/programmer.

However, network monitoring in simulated environments usually poses some challenges. If the simulation models are not correctly/adequately/suitably designed and implemented, the generated results from the simulation experiment may contradict the expected behavior of the network protocols/models. Again, to gather performance metrics from simulation experiments, most simulators, especially NS-3 necessitates the researcher to undergo considerable amount of programming tasks. In other words, extracting simulation results needs the researcher to implement powerful lines of code, perhaps to be familiar with the use of various scripting languages. Hence, the researcher may be compelled to devote excessive amount of time and efforts to coding instead of putting more focus on the research while the quality of the simulation models may also be degraded.

4.7.1 FlowMonitor : NS-3 Network Monitoring Framework (Carneiro, Fortuna and Ricardo, 2009)

In order to automate most of the result gathering tasks in NS-3 in a more convenient and effective approach, a module called FlowMonitor is presented in (Carneiro, Fortuna and Ricardo, 2009). The network monitoring framework which is designed and dedicated for NS-3 flow monitoring is capable of extracting and store the network performance data in NS-3 simulation environment. The module is designed to be easily extended and to be efficient in the consumption of the memory and CPU (Central Processing Unit) resources. Also, FlowMonitor framework is designed with the aim that the generated simulation data can be simply processed to obtain the final results such as plots and comprehensive statistics. All the metrics of flow passing through the network that can be automatically collected by FlowMonitor among others may include bitrates, durations, delays, packet sizes and packet loss ratio. The simulation performance metrics are stored in an XML data storage file format and access to in-memory data structure is supported. Also, FlowMonitor architecture considers low memory and computation overhead to a reasonable extent.

However, a single simulation will typically contain one FlowMonitorHelper instance, one FlowMonitor, one Ipv4FlowClassifer, and several Ipv4FlowProbes, one per Node. Probes

capture packets, then ask the classifier to assign identifiers to each packet, and report to the global FlowMonitor abstract flow events, which are finally used for statistical data gathering. The statistical results gathered from the simulation experiment performed in this research were extracted to text files through the simulation code and the data were processed in Microsoft Excel 2013 Spreadsheet.

4.7.1.1 FlowMonitor Flow Data Structure

As explained in (Carneiro, Fortuna and Ricardo, 2009), the major output of the flow monitoring process is the gathering of flow statistics. They are stored in memory data structures, and can be accessed through simple “getter" methods. The data structure of the flow statistics are categorized into two distinct data structures. These include *FlowMonitor::FlowStats* and *FlowProbe::FlowStats.* The *FlowMonitor::FlowStats* contains complete end-to-end flow statistics, while the *FlowProbe::FlowStats* contains only a small subset of statistics and from the point of view of each probe. The detailed description of the individual attributes in the flow data structures are presented in (Carneiro, Fortuna and Ricardo, 2009). The applicable attributes of the *FlowMonitor::Stats* that were adopted in the computation of the performance evaluation metrics in this research are the ones described in this subsection. The attributes are the folowings:

- **timeFirstTxPacket** Contains the absolute time when the first packet in the flow was transmitted. That is, the time when the flow transmission starts
- **timeLastTxPacket** Contains the absolute time when the last packet in the flow was transmitted. That is, the time when the flow transmission ends
- **timeFirstRxPacket** Contains the absolute time when the first packet in the flow was received by an end node. That is, the time when the flow reception starts;
- **timeLastRxPacket** Contains the absolute time when the last packet in the flow was received. The time when the flow reception ends
- **delaySum** Contains the sum of all end-to-end delays for all received packets of the flow
- **txBytes**, **txPackets** Total number of transmitted bytes and packets, respectively, for the flow

- **rxBytes**, **rxPackets** Total number of received bytes and packets, respectively, for the flow
- **lostPackets** Total number of packets that are assumed to be lost, i.e. those that were transmitted but have not been reportedly received or forwarded for a long time. By default, packets missing for a period of over 10 seconds are assumed to be lost, although this value can be easily configured in runtime.

4.8 Summary

The discussions in this chapter covered the research method. In Section 4.2, various research methods were discussed, Section 4.3 presented the research process. Section 4.4 captured the research design research technique. In the section, possible network performance evaluation methods were discussed. Among possible network performance evaluation methods discussed, simulation model that was adopted as the research method was extensively explained. The explanation also dwelled partly on the benefits of the simulation method. Section 4.5 presented the research methodology which largely described the steps followed specifically for simulation experiments in this research. Moreover, the network simulator (NS-3) adopted for the simulation experiments in this research was largely discussed in Section 4.6. The Next chapter presents the implementation process for the implementation of the black hole attack model in the NS-3 network simulator using NS-3 AODV as the base routing protocol.

CHAPTER FIVE

NS-3 AODV BLACK HOLE ATTACK IMPLEMENTATION AND VERIFICATION

5.1 Introduction

In the last chapter, the network simulator (NS-3) adopted for the simulation experiments in this research was discussed. This chapter discusses the implementation process of the black hole attack model in the NS-3 network simulator using NS-3 AODV as the base routing protocol. In addition, the chapter largely dwells on the mini simulation carried out to verify the effectiveness of the black hole model implementation using a simple wireless ad hoc network setup. The metrics measured in the simple implementation test simulation are also discussed in the chapter.

5.2 AODV Black Hole Attack Implementation

In this section, the implementation of the black hole attack behavior in NS-3 AODV model is discussed. As earlier explained in Section 2.5.2, malicious black hole attack can either be exhibited by a single node or a number of malicious nodes. In an ad-hoc routing protocol such as AODV routing protocol, the malicious node launching a black hole attack deceives the source node to have a fresh enough route to the destination node. This is achieved by the black hole node as it immediately responds to the source node RREQ broadcast packet with a unicast RREP message with the highest destination sequence number and lowest hop count value (1 to be precise). Since the routing protocol lacks a mechanism to establish trust among participating nodes, the originating node believes that the RREP is from an authentic intermediate node with a fresh enough route or the required destination node itself. Consequently, the source trusts that the destination is one hop neighbour of the black hole node and discards other received RREP packets. Based on the contained route information in the RREP from black hole node, the source then begins to send its intended data packets through the black hole node with the trust that the sent data packets will hop through to the destination. Conversely, the data packet instead of getting through to the destined node via the black hole node is maliciously dropped by the black hole node.

By default, NS-3 suite does not include a model to simulate the black hole attack behavior in NS-3 environment. However, efforts were made in (Satre and Tahiliani, 2014) to implement this functionality for the simulations of experiments that involve the analysis of black hole attack in NS-3. The base routing protocol adopted for the implementation was AODV routing protocol. Thus, in order to add black hole attack behavior to the AODV routing protocol used for the simulation experiments in NS-3 (version 3.25) simulator adopted in this research, the black hole attack behavior implemented for NS-3 AODV routing protocol in (Satre and Tahiliani, 2014) was adopted. The NS-3 black hole implementation code is provided as an open source patch in (Satre and Tahiliani, 2014). The patch was run on the default NS-3 AODV routing protocol and as a result, the NS-3 default AODV routing source files names remain the same after running the patch file. The steps (Satre and Tahiliani, 2014) followed for the application of the AODV black hole implementation patch on the NS-3 ADOV routing protocol default model adopted in this research are presented as the following:

1. Downloading of the Blackhole-ns-3.25.patch
2. Pasting the downloaded Blackhole-ns3 patch in the ns-allinone-3.25 directory.
3. Opening of the terminal in Ubuntu 14.04 LTS Operating System
4. Navigating to the ns-allinone-3.25 path on the terminal
5. Execution of the following command in ns-allinone-3.25 path in the terminal to apply the patch:

   patch -p1 < Blackhole-ns-3.25.patch
6. Navigating into the ns-allinone-3.25/ns-3.25 directory on the terminal to run the following command:

   ./waf

5.3 AODV Black Hole Attack Implementation Code Segment

This section seeks to show and explain the code segments that were added to the aodv-routing-protocol.h and aodv-routing-protocol.cc files after the application of the black hole implementation patch. However, it is important to relate that NS-3 is an open source simulation tool that can be used freely and modified for research purposes. The patch provided by (Satre and Tahiliani, 2014) to implement the black hole behaviour in NS-3 AODV routing protocol is

also a freely distributed patch that can be freely used by the research community. It is also important to note that NS-3 is a simulator that is continuously expanded by the open source community, thus, the implementation of black hole behaviour in NS-3 AODV is a tremendous contribution to NS-3 development which is credited to (Satre and Tahiliani, 2014) (*Contributed Code - Nsnam*, 2016). Hence, while it is needful to show the code implementation of the black hole behaviour, the researcher in this study does not claim the authorship of the code segments captured in this section. It is also important to indicate that the lines of code represented in each code segment were extracted for the purpose of this section and are not presented according to their exact line numbers in the AODV NS-3 corresponding *.cc* file after the black hole patch was applied.

In the NS-3 simulation environment source folder adopted in this research, the model for the AODV routing protocol is located in the …/ns-allinone-3.25/ns-3.25/src/aodv/model/. Among all the .h and.cc files located in the …aodv/model directory, the bulk/extensive and major implementation of AODV routing operations is written/executed in the aodv-routing-protocol.h and aodv-routing-protocol.cc files. Hence, the applied patch to implement the behaviour of black hole behaviour in NS-3 AODV routing protocol was effective in the aodv-routing-protocol.h and aodv-routing-protocol.cc files. The lines of C++ code contained in the aodv-routing-protocol.h and aodv-routing-protocol.cc files after applying the black hole patch are 281 and 1987 lines respectively. In the aodv-routing-protocol.h file, a variable *IsMalicious* of the type *bool* was declared to enable and disable the malicious black hole behaviour in the NS-3 AODV routing model. The malicious behaviour is enabled when *IsMalicious* is passed a *true* value while *false* value disables the malicious behaviour value in the user's simulation code.

As explained earlier in subsection 2.5.1, the AODV routing elements include RREQs, RREPs and RERRs. A source node that intends to send data packets to a destination node is required to first establish a forwarding route to the destination node by first sending out a RREQ as a broadcast message to all its neighbouring nodes till the RREQ message is propagated to the destination node. As defined in the AODV routing mechanism, a node responds to the RREQ message with a RREP message on one of the two following conditions:

1) The node is itself the destination

2) Or the node has an active route to the destination, the destination sequence number in the node's existing route table entry for the destination is valid and greater than or equal to the destination sequence number in the RREQ, and the "destination only" flag is not set.

Hence, in the aodv-routing-protocol.cc file, the functions to handle the actions of a node while it receives a RREQ are written in the *RecvRequest ()* method. In the aodv-routing-protocol.cc file, there exist among other declared methods, two methods defined to handle how the parameters for a RREP generation. The methods are *SendReply ()* and *SendReplyByIntermediateNode ().*The *SendReply ()* handles the first condition for the generation of RREP by a destination node itself while the *SendReplyByIntermediateNode ()* handles the condition that an intermediary node generates a RREP message on behalf of the destination node. Both *SendReply ()* and *SendReplyByIntermediateNode ()* are mutually exclusive called in the *RecvRequest ()* method as the condition/case may be for the generation of RREP.

As explained earlier in subsection 2.5.2, a malicious node launching a black hole packet drop attack is an intermediate node with the malicious intentions of dropping the data packet transferred from a source node to the destination node by deceitfully establishing a false route between them (the source and destination nodes). This is achieved when the black hole node generates a false RREP in response to the originator's RREQ message. Therefore, the operation of black hole attack behaviour is defined in the *SendReplyByIntermediateNode ()* method. To enhance the description of how the black hole node attracts the route to itself in the black hole implementation lines of code, the format for of a typical AODV routing RREP header is represented in figure 5.1.

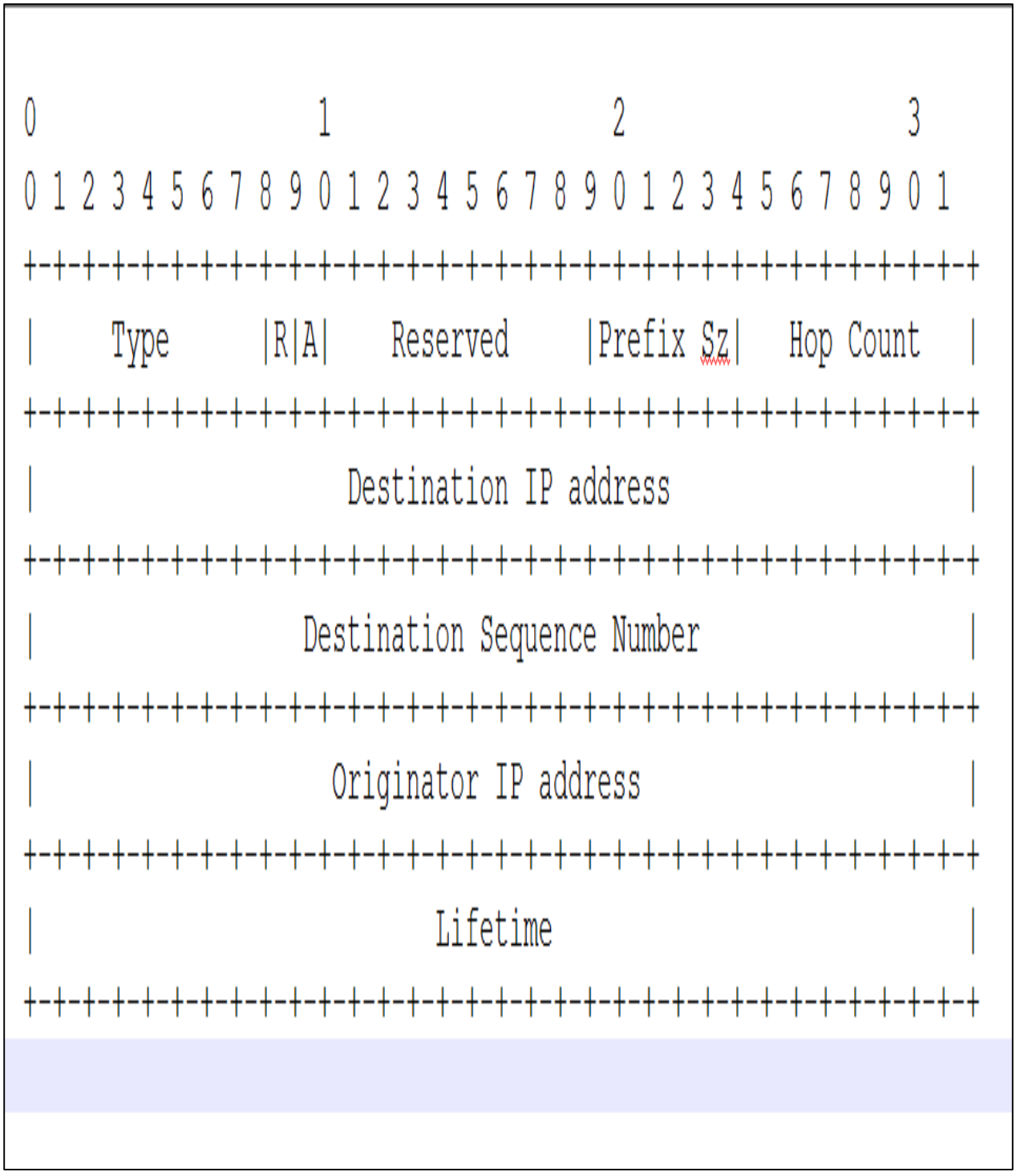


Figure 5. 1: AODV routing RREP header

As adapted from (Perkins, Belding-Royer and Das, 2003), the fields of the RREP format are explained in the table below.

Table 5. 1: RREP's field description

Type	2
R	Repair flag; used for multicast.
A	Acknowledgment required.
Reserved	Sent as 0; ignored on reception.
Prefix Size	If nonzero, the 5-bit Prefix Size specifies that the indicated next hop may be used for any nodes with the same routing prefix (as defined by the Prefix Size) as the requested destination.
Hop Count	The number of hops from the Originator IP Address to the Destination IP Address. For multicast route requests this indicates the number of hops to the multicast tree member sending the RREP.
Destination IP Address	The IP address of the destination for which a route is supplied.
Destination Sequence Number	The destination sequence number associated to the route.
Originator IP Address	The IP address of the node which originated the RREQ for which the route is supplied.
Lifetime	The time in milliseconds for which nodes receiving the RREP consider the route to be valid.

As explained and adapted from (Perkins, Belding-Royer and Das, 2003), whenever an intermediate node along the path between the RREQ originator (source node) and the destination generates a RREP message on behalf of the destination node, the intermediate node copies the current destination sequence number of the destination node in its routing table to the destination number field of the RREP message it generated. The intermediate node updates the forward route entry by placing the last hop node (from which it received the RREQ, as indicated by the source IP address field in the IP header) into the precursor list for the forward route entry. That is, the entry for the Destination IP Address. The intermediate node also updates its route table entry for the node originating the RREQ by placing the next hop towards the destination in the precursor list for the reverse route entry. That is, the entry for the Originator IP Address field of the RREQ message data. The intermediate node places its distance in hops from the destination (indicated by the hop count in the routing table) Count field in the RREP. The Lifetime field of the RREP is calculated by subtracting the current time from the expiration time in its route table entry.

As it appears in the figure depicting the code segment for *SendReplyByIntermediateNode ()* method, two of the arguments declared in the *SendReplyByIntermediateNode ()* method include *toDst* and *toOrigin* which are type *RoutingTableEntry* class of the NS-3 AODV model. The *toDst* is used by the intermediate node to create the forward routing table entry for the destination with respect to the RREP message generated by the intermediary node while the *toOrigin* is used to create a reverse routing table entry for the RREQ originator with respect to the RREP generated. Exhaustive interpretation of the NS-3 AODV routing model code implementation is beyond the scope of this research and this section. In addition, the focus of this section is to explain the lines of code that implement the approach adopted by a black hole node to contrive the establishment of a false route between a source node and a destination node. Therefore, further discussions in this section focus on the line of codes that implement how the black hole node manipulates the *toDst* to establish a false route between a source node and a destination node when the *SendReplyByIntermediateNode ()* method is called in the *RecvRequest ()* method.

In the *RecvRequest ()* method in the code segment 5.1, a false routing table entry for the destination was defined on the line number 16 with an instance *falseToDst* of the

RoutingTableEntry class in the block *If (IsMalicious) {…}* statement. In the *falseToDst,* the black hole is able to generate a very high destination sequence number in the routing table entry for the destination by passing the parameter *rreqHeader.GetDstSeqno()+100* to the destination sequence number field of the *falseToDst.* Also, the hop count field of the *falseToDst* that creates the false routing table entry of the black hole node is set to *1*. On line number 18 of the *RecvRequest ()* method code segment, *falseToDst* is passed as the parameter for the *toDst argument of the SendReplyByIntermediateNode ()* method.

```
1  void RoutingProtocol::RecvRequest (Ptr<Packet> p, Ipv4Address receiver, Ipv4Address src)
2  {
3      ...
4          ...
5              ...
6
7              /* Code added by Shalini Satre, Wireless Information Networking Group (WiNG), NITK Surathkal for simulating Blackhole Attack
8                * If node is malicious, it creates false routing table entry having sequence number much higher than
9                * that in RREQ message and hop count as 1.
10               * Malicious node itself sends the RREP message,
11               * so that the route will be established through malicious node.
12               */
13              if(IsMalicious)
14              {
15                Ptr<NetDevice> dev = m_ipv4->GetNetDevice (m_ipv4->GetInterfaceForAddress (receiver));
16                RoutingTableEntry falseToDst(dev,dst,true,rreqHeader.GetDstSeqno()+100,m_ipv4->GetAddress (m_ipv4->GetInterfaceForAddress(receiver),0),1,dst,ActiveRouteTimeout);
17
18                SendReplyByIntermediateNode (falseToDst, toOrigin, rreqHeader.GetGratiousRrep ());
19                 return;
20              }
21              /* Code for Blackhole Attack Simulation ends here */
22
23              ...
24                ...
25                    ...
26 }
```

Code Segment 5. 1: NS-3 AODV *RecvRequest ()* method

On the line number 4 of the *SendReplyByIntermediateNode ()* method code segment 5.2, the instance *rrepHeader* of the *RrepHeader* class defined and initialized implements an instance of RREP message according to the format of the RREP message represented in the figure above. As mentioned earlier in this section, an intermediate node places its distance in hops from the destination (indicated by the hop count in the routing table) Count field in its generated RREP message. In the block *If (IsMalicious) {…} s*tatement, the Hop Count field for the *rrepHeader* is

set to *1* with the expression *rrepHeader.SetHopCount (1)* on the line number 14 to fabricate the hop distance of the black hole node to the destination node.

In the AODV routing mechanism, a route with the lowest hop count and higher destination sequence number is chosen as the optimal route from the source to the destination node. As discussed earlier, a black hole node responds to the source node RREQ broadcast packet with a unicast RREP message with the highest destination sequence number and lowest hop count value of 1 to establish a false route between a source node and a destination node. Hence, the implementation of the black hole node's falsification of a route between a source and the destination by generating a false RREP message was implemented through the fabricated higher destination number in the *falseToDst* destination sequence number field and the fabricated lowest hop count of 1 set in the *rrepHeader* hop count field.

```
1  RoutingProtocol::SendReplyByIntermediateNode (RoutingTableEntry & toDst, RoutingTableEntry & toOrigin, bool gratRep)
2  {
3    NS_LOG_FUNCTION (this);
4    RrepHeader rrepHeader (/*prefix size=*/ 0, /*hops=*/ toDst.GetHop (), /*dst=*/ toDst.GetDestination (), /*dst seqno=*/ toDst.GetSeqNo (),
5                           /*origin=*/ toOrigin.GetDestination (), /*lifetime=*/ toDst.GetLifeTime ());
6  /* If the node we received a RREQ for is a neighbor we are
7   * probably facing a unidirectional link... Better request a RREP-ack
8   */
9
10 ///Attract node to set up path through malicious node
11
12 if(IsMalicious)                    //Shalini Satre
13 {
14   rrepHeader.SetHopCount(1);
15 }
16 ...
17
18 ...
19
20 ...
21 }
```

Code Segment 5. 2: NS-3 AODV *SendReplyByIntermediateNode ()* method

Furthermore, in the aodv-routing-protocol.cc, a method to handle the forwarding functions in the NS-3 AODV routing after a route has been established between a source and a destination is also

declared. In the method, the code implementation of the malicious packet drop behaviour is written as shown in the forward code segment shown 5.3. The *Forwarding ()* method is a type *bool* that returns a *true* value if there exit a valid route between a source and a destination node. On the other hand, a black hole functionality implemented to drop the packet maliciously in the *If (IsMalicious) {…}* statement terminates the forwarding method with a return *false* value whenever the *IsMalicious* is enabled in the user's simulation code. Hence the data packets traversing the forwarding route falsely established between the source and the destination by the black hole node are maliciously dropped.

```
1  bool
2  RoutingProtocol::Forwarding (Ptr<const Packet> p, const Ipv4Header & header,
3                               UnicastForwardCallback ucb, ErrorCallback ecb)
4  {
5  ...
6     ...
7        ...
8  /* Code added by Shalini Satre, Wireless Information Networking Group (WiNG), NITK Surathkal for simulating Blackhole Attack */
9   /* Check if the node is suppose to behave maliciously */
10       if(IsMalicious)
11        {
12           //When malicious node receives packet it drops the packet.
13            std :: cout <<"Launching Blackhole Attack! Packet dropped . . . \n";
14            return false;
15        }
16   /* Code for Blackhole attack simulation ends here */
17     ...
18        ...
19           ...
20 }
```

Code Segment 5. 3: *Forwarding ()* method

Thus, a small ad hoc network was set up to test that the applied black hole patch was executing the black hole behavior as expected. The implementation test simulation is explained in the next Section.

5.4 AODV Black hole Implementation Test

As it was earlier discussed in the previous section, NS-3 does not include a model to perform simulation experiments to analyze black hole attack. Thus, the implementation of the black hole attack in NS-3 AODV routing protocol adopting the patch provided in (Satre and Tahiliani, 2014) was discussed in the previous section. In this section, the simple simulation experiment to test the implemented black hole attack in NS-3 is discussed. The simple wireless ad hoc network set up for the implementation test and measured metrics are also discussed.

5.4.1 Simulation Experiment for Black hole Implementation Test and Measured Metrics

In this section, the simple wireless ad hoc network set up for the implementation test and measured metrics are discussed. The network setup contains 5 static wireless ad hoc nodes arranged into one dimensional grid topology as shown in Figure 5.2 you already have Figure 5.1 on page 4. In the simulation setup script, two types of nodes were defined, malicious and not_malicious. The node defined as malicious is Node 0 while Node 1, Node 2 Node 3 and Node 4 are the nodes defined as not_malicious. Therefore, in Figure 5.2 the nodes are named as Node 0, Node 1, Node 2, Node 3, and Node 4 from the left to the right of the grid. In the simulation environment, UDP was adopted as the transport protocol so as to accurately capture the simulation results. Thus, the source node can continuously send out UDP packets till the end of its connection with the destination node even if the data packets are maliciously dropped by the black hole node.

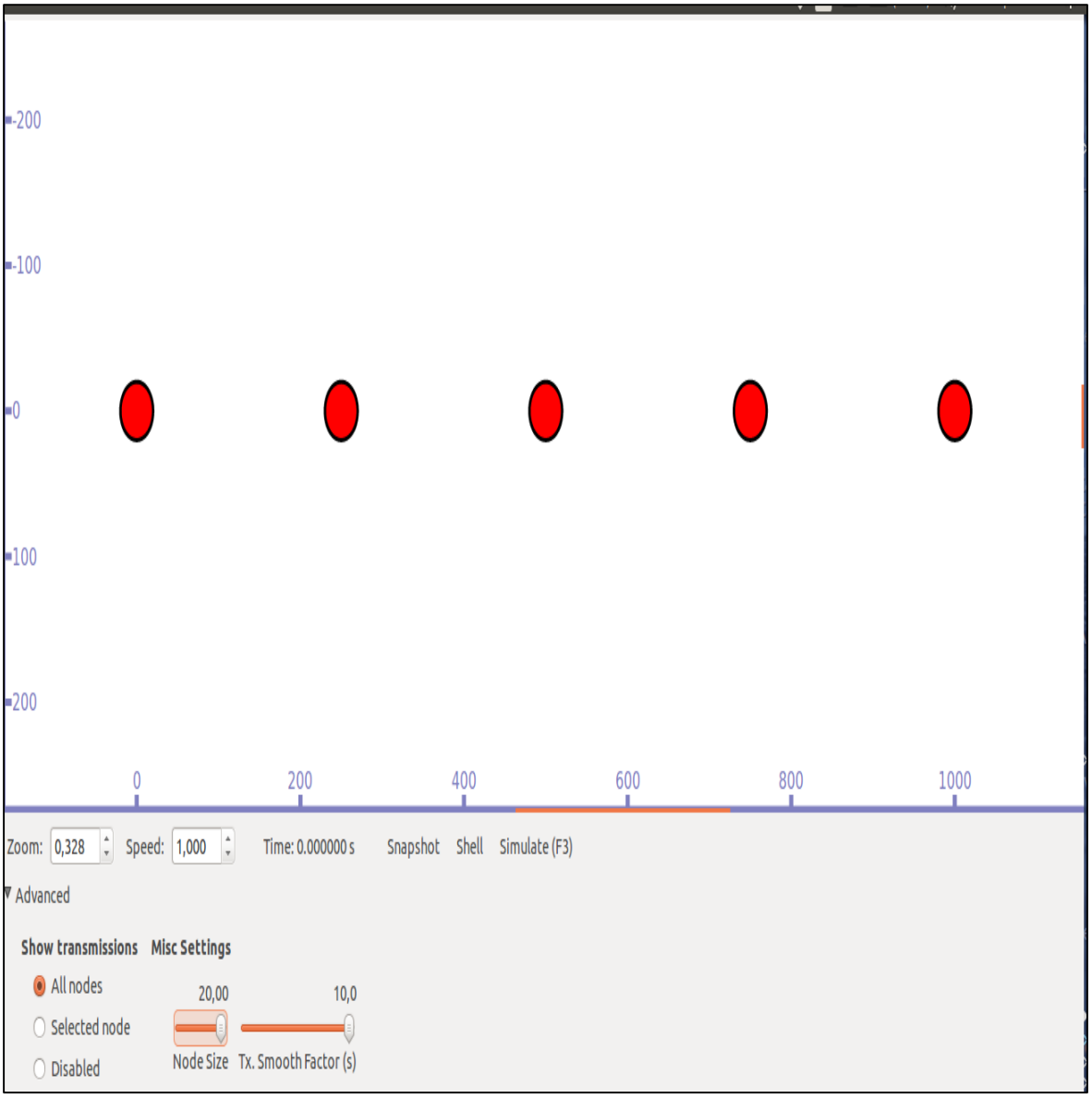


Figure 5. 2 Wireless Ad hoc network without wireless connection

In the small grid network created for the simulations, each node was in a transmission range of 250 meters to its immediate adjacent/neighbor node. As shown in Figure 5.3, there is a physical peer wireless link/connection established between each neighbor node. The peer wireless link is denoted with the green bidirectional link. The links were established at the start of the simulation. As a result, any source node intending to send data packet would be able to transmit its intended data packet to the destined node with the aid of the routing protocol (AODV) path selection mechanisms. Therefore, a UDP connection was established between Node 1 (source

node) and Node 4 (destination node) with an attached CBR (Constant Bit Rate) application that generates constant data packets transmission. The simulation was run for 50 seconds, and the CBR connection was scheduled to start at time 30 seconds. Total number of 10 CBR packets of size 1024 bytes long were transmitted at the data rate of 150 kbps from the transmission start time until the end of the simulation. The flow of the connection between source Node 1 and destination Node 4 during the simulation was printed. The sent and received packets were counted separately as the UDP connection was not lost during the simulation. The counting of the sent and received packets would not have been possible if TCP protocol was adopted as the transport protocol for the two scenarios evaluated. This is because of the fact that the node sending TCP packet would terminate the connection if it does not receive TCP ACK message after a point in time it has established the connection.

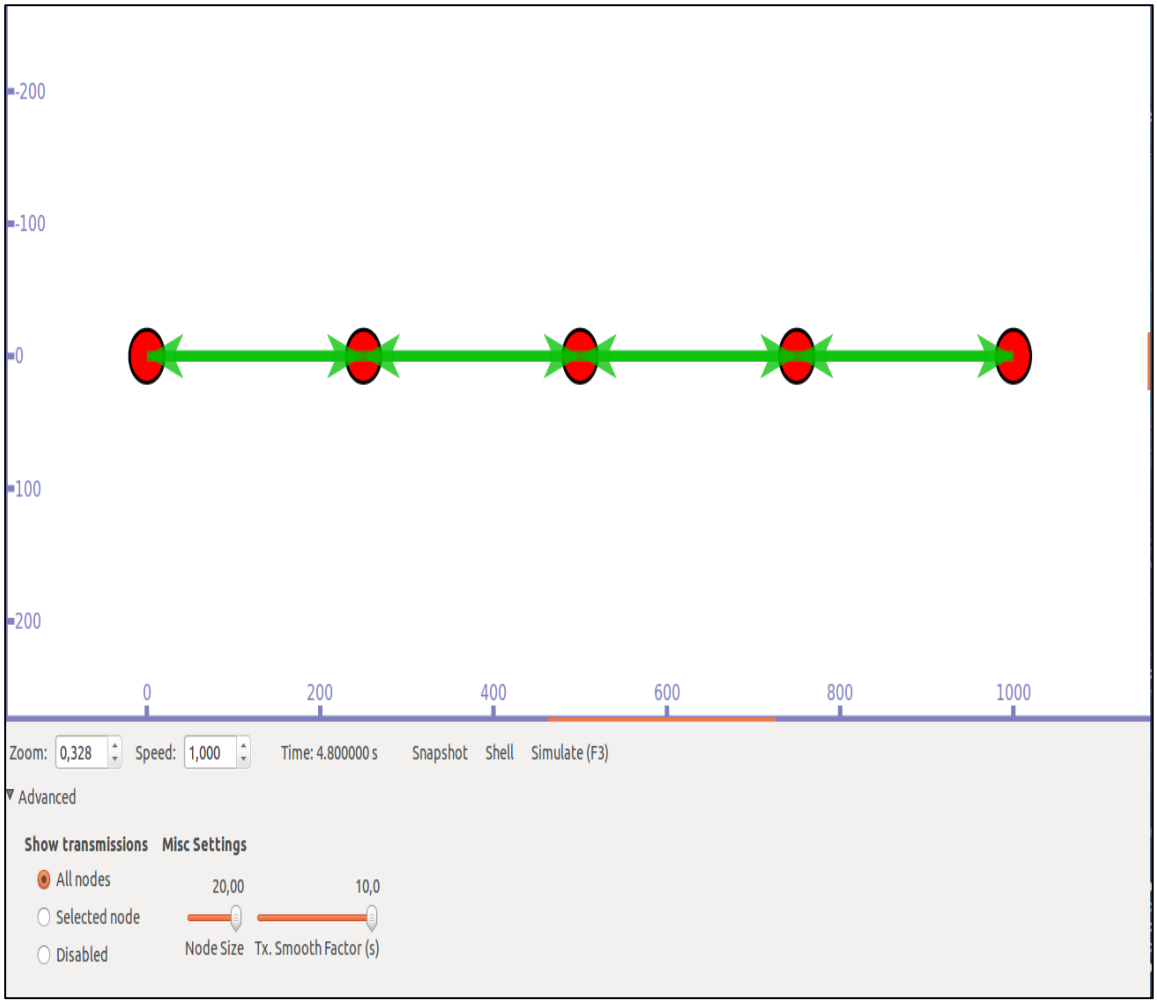


Figure 5. 3: Wireless Ad hoc network with wireless connections

5.4.1.1 Analysis of the Simulation Scenarios

In the first simulation scenario, the malicious packet drop behavior of Node 0 was not enabled; as a result, there was no black hole node present. All the participating nodes are in cooperation to help in forwarding the data packet destined for another node in the network. As Node 1 intends to start sending data packet to Node 4, it sends a RREQ broadcast to its immediate neighbor nodes, Node 0 and Node 2. Hence, the RREQ is propagated to the destination node, Node 4 from Node2 via Node 3. Then Node 4 sends a unicast RREP back to Node 1 along the back route through Node 3 and Node 2. After the fresh enough path was established from source Node 1 to the destination Node 4, Node 1 started sending out the data packets it intended to send to Node 4. As in Figure 5.4, it can be seen that the forward edge of the bidirectional links between Node 1, Node 2; Node 2, Node 3; Node3, Node 4 are broader compared to the forward edge of the link between Node 0, Node 1. The broader forward edge of the links between the former pairs of neighbor nodes indicates the forward path in which data packets are being correctly transmitted from source node, Node 1 to destination Node 4.

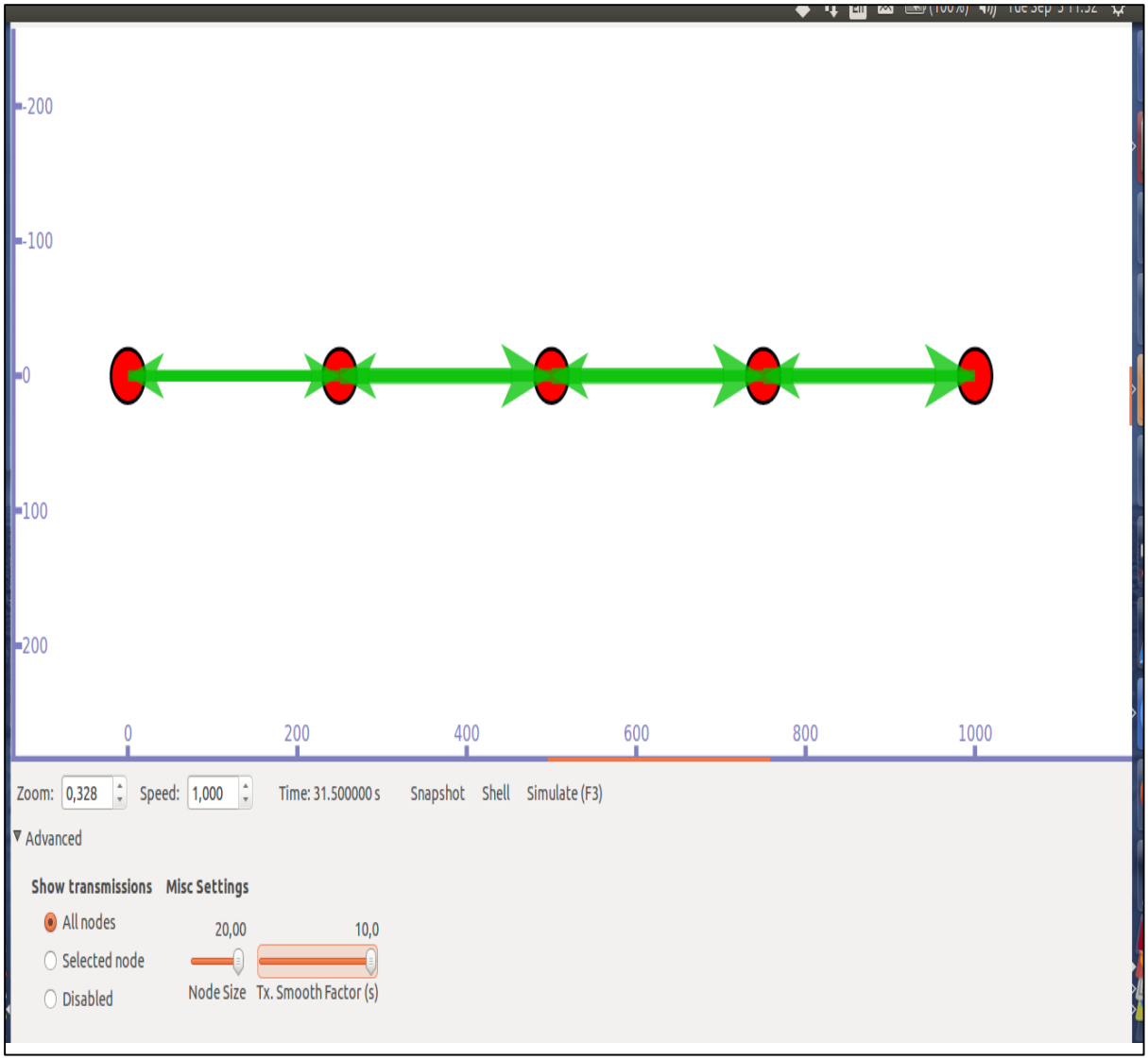


Figure 5. 4 Correct Transmission of Packet from Source Node 1 to Destination Node 4

As shown in Figure 5.5, the results of the simulation experiment metrics are represented. In the figure, the 10.1.2.2 represents the IP address for source node, Node 1 while 10.1.2.5 is the IP address of destination node, Node 4. Therefore, Flow 1 (10.1.2.2 -> 10.1.2.5) means the flow of data transmission between Node 1 and Node 4. Also, Tx means transmitted packet and Rx means received packet. As computed from the printed data flow file of the simulation, 10 data packets were transmitted by Node 1 to Node 4 and all the 10 data packets were successfully delivered to Node 4. This is because all the intermediary nodes are exhibiting normal routing behavior.

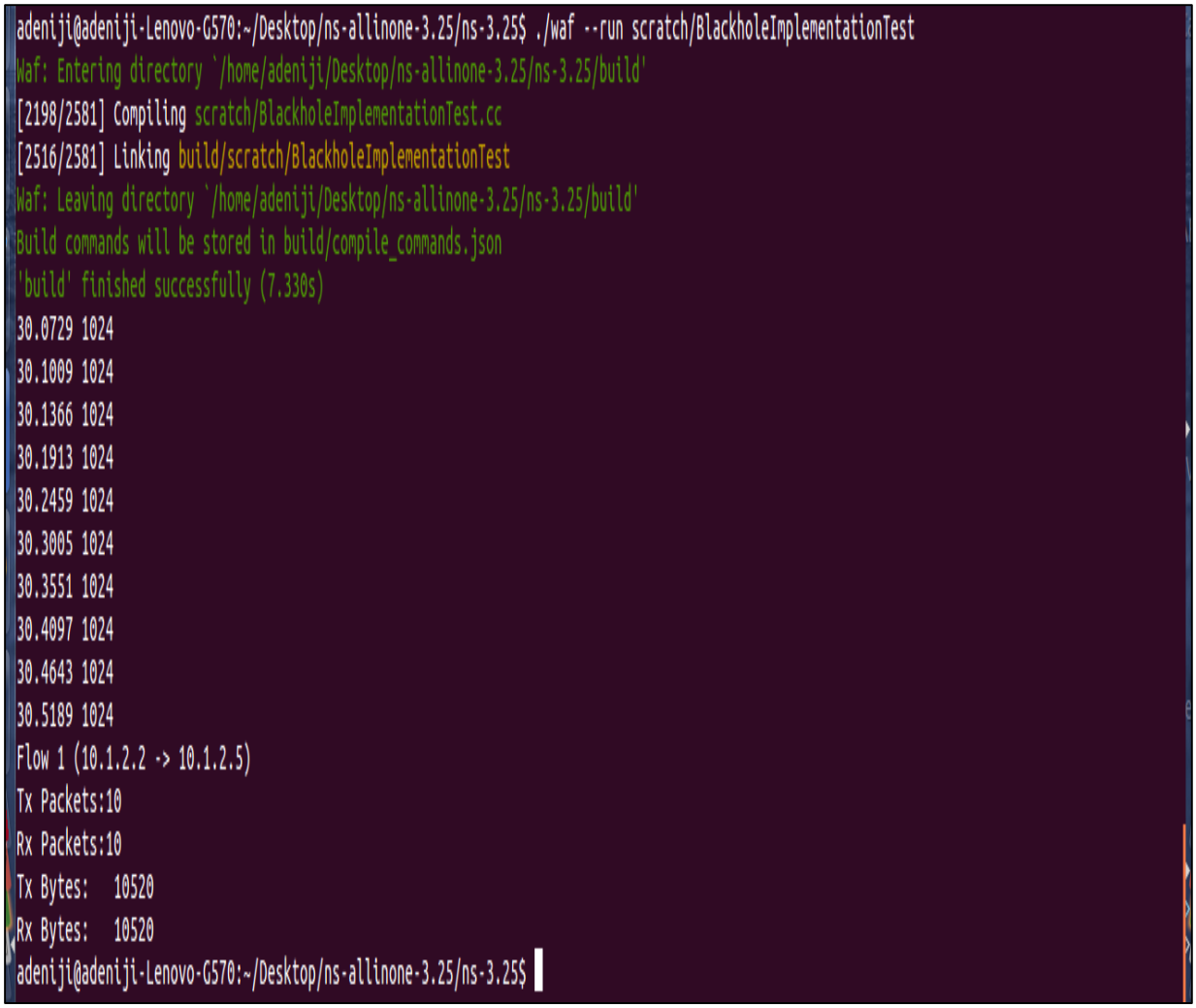


Figure 5. 5: Result of the Measured metric in Wireless ad hoc network without attack

In the second simulation scenario, Node 0 malicious black hole behavior was enabled in order to see the effect of the implementation of the black hole attack behavior in the AODV routing protocol. As in the first scenario, where Node 0 was not malicious, Node 1 intends to start sending data packet to Node 4. It therefore, sends a RREQ broadcast to its immediate neighboring nodes, Node 0 and Node 2. As Node 2 node receives the RREQ message, it continues to rebroadcast RREQ message until it reaches Node 4. The malicious Node 0, however, violates this rule and deceptively claims to Node 1 that it has the shortest path to Node 4 by immediately sending a fake RREP packet to Node 1. Consequently, Node 1 assumes that the shortest path to Node 4 is via Node 0 and discards other received RREP packets that might

later be sent from Node 4. Hence, Node 1 begins to send its intended data packets to Node 4 through Node 0 which were in turn dropped.

As in Figure 5.6, it can be seen that the backward edge of the bidirectional link between Node 0, Node 1 is broader compared to the forward edge of the bidirectional links between Node 0, Node 1, and forward and backward edges of the bidirectional link between Node 1, Node 2; Node 2, Node 3; Node3, Node 4. The broader backward edge of the link between the pair of Node 1 and Node 0 indicates a fake path was established between source Node 1 and destination Node 4 via Node 0 and that the data packet transmitted from Node 1 are transmitted to Node 4 along the path which included Node 0. Hence, as shown in Figure 6.5, the red arrow headed line pointing downwards indicates that the data packets transmitted from source node, Node 1 to destination Node 4 are continuously dropped at malicious black hole node, Node 0 which is along the fake route between Node 1 and Node 4.

As shown in Figure 5.7, the results of the simulation experiment metrics are represented. As in first scenario, the 10.1.2.2 represents the IP address for source node, Node 1 while 10.1.2.5 is the IP address of destination node, Node 4. Therefore, Flow 1 (10.1.2.2 -> 10.1.2.5) means the flow of data transmission between Node 1 and Node 4. Also, Tx means transmitted packet and Rx means received packet. As computed from the printed data flow file of the simulation, 10 data packets were transmitted by Node 1 to Node 4 but all the 10 data packets were lost. This was attributed to the malicious packet drop behavior of Node 0 within the network compared to first scenario where Node 0 was not malicious.

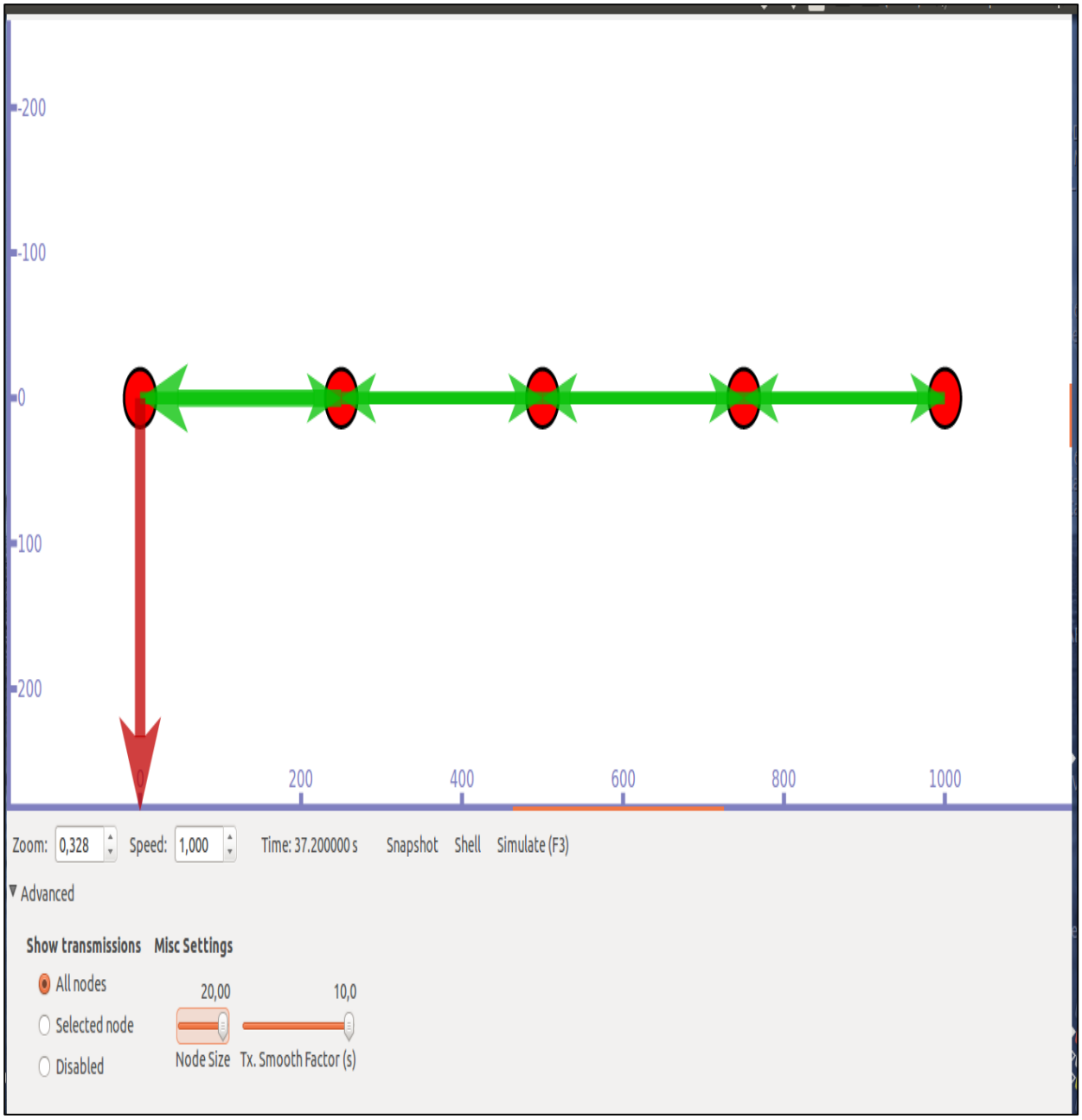


Figure 5. 6: Wrong packet forwarding and malicious Packet drop in Wireless ad hoc network under attack

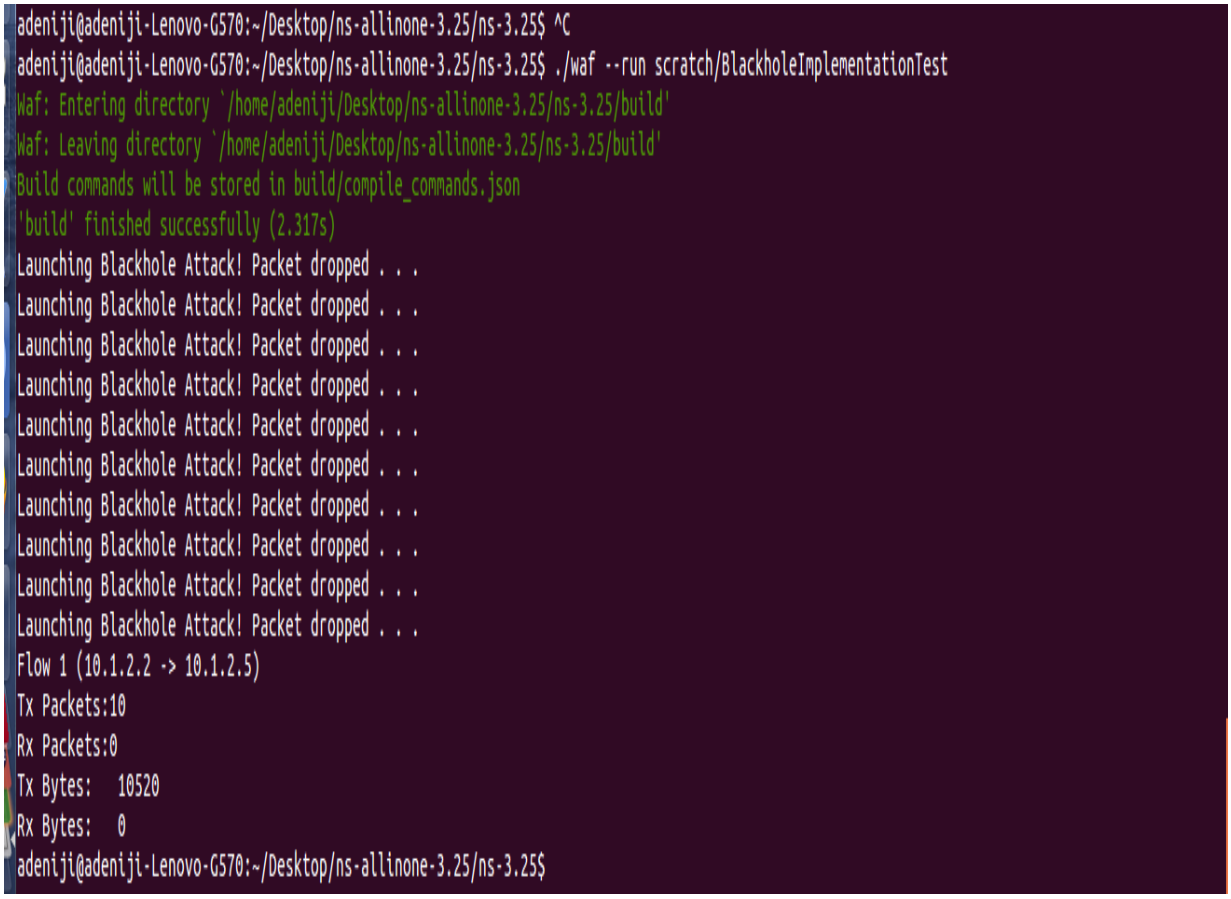


Figure 5. 7: Result of the Measured metric in Wireless ad hoc network without attack

The comparative analysis of the simulation results of the two scenarios experimented for the small grid network setup attested that the black hole attack behavior has been successfully implemented in the default AODV routing protocol. Hence, the security and reliability of packet transmission in WMNs under malicious packet drop attack was simulated and analyzed using AODV routing protocol as the routing protocol. The network simulation environment set up is as explained in Section 6.2.

5.5 Summary

The discussions in this chapter was able to present the implementation of the black hole attack model in the NS-3 network simulator using NS-3 AODV as the base routing protocol. In addition, the chapter largely dwelled on the mini simulation carried out to verify the effectiveness of the black hole model implementation using a simple wireless ad hoc network

setup. Also, the metrics measured in the simple implementation test simulation were discussed in the chapter. In the next chapter, the actual simulation experiments performed to analyze the impact of black hole attack on packet transmissions in WMNs are presented.

CHAPTER SIX

SIMULATION EXPERIMENTS

6.1 Introduction

The discussions in the previous chapter presented the implementation of the black hole attack model in the NS-3 network simulator using NS-3 AODV as the base routing protocol. Moreover, the chapter largely dwelt on the mini simulation carried out to verify the effectiveness of the black hole model implementation using a simple wireless ad hoc network setup.

In this chapter, the simulation experiments performed to analyze the impact of black hole attack on packet transmissions in WMNs are presented. The chapter will first present the experimental setup of the simulation scenarios for the WMNs evaluations. The experimental setup includes the environment setup, the routing protocol configurations, and the network topologies and simulation scenarios.

Secondly, the chapter discusses the method used for the performance evaluations of the WMNs infrastructure simulated scenarios. The measured performance evaluation metrics will include Packet Delivery Ratio (PDR), average throughput and average end-to-to end delay. Finally, the chapter presents and discusses the results of the simulation experiments adopting the measured performance evaluation metrics.

6.2 Experimental Setup

In this research, the choice of the routing protocol was AODV routing protocol as mentioned earlier. This was because, HWMP is a routing protocol that was inspired by AODV and there are many similarities between the HWMP reactive mode and AODV routing mechanisms. In addition, the proactive element of HWMP is an extension of its reactive element. Based on the extensive work done in (Morote, 2011), the simulation experiments setup for the performance evaluation of WMNs with and without malicious packet drop attack in this study was supported

by the experimental study in (Morote, 2011). The discussions around the simulation environment are presented in subsection 6.2.1.

6.2.1 Environment Setup

As explained in (Houaidia *et al.*, 2013), all parts and parameters of the network protocol stack must be considered to achieve optimal performance in WMNs. This section seeks to dwell on the environment setup and factors considered to obtain optimal network performance from the perspective of a typical wireless network environment.

Furthermore, it was mentioned in (Borgo *et al.*, 2004) that in recent years, IEEE 802.11 protocols have evolved as the de-facto standards for wireless networks. The uniqueness of 802.11 includes its medium access control (MAC) which supports all its physical (PHY) specifications. In the standard, the primary access control follows the distributed coordination function (DCF) access methods which supports two medium access mechanisms (Marks, 2002). These include basic access and collision avoidance (Borgo *et al.*, 2004). In essence, the basic access and collision avoidance is based on carrier sense multiple access with collision avoidance (CSMA/CA) (Islam *et al.*, 2010). The choice of one of the two mechanisms is important in addressing issue of interfering transmissions in the network (Borgo *et al.*, 2004).

As explained in (Wang *et al.*, 2010; Morote, 2011), clear channel assessment (CCA) is a mechanism to sense the activities in the medium so as to detect if the medium is busy or idle. The features of CCA include carrier sensing (CS) and energy detection (ED) capabilities. The Carrier Sense (CS) element is classified into a physical CS and virtual CS. The physical CS is a direct measurement of the strength of the received signal of a valid 802.11 symbol and it is provided by the PHY. Thus, the medium is considered busy at a value that is above the defined threshold.

On the other hand, the virtual CS is provided by the MAC and it's also termed as the network allocation vector (NAV). The NAV functionally notifies a station the next time the medium would be available and it is maintained by the session duration values that is included in all frames. A NAV value may be an indication for a station not to transmit, even if the physical CS

indicates that the medium is not busy. It is updated if the duration value is greater than the current NAV value each time a valid 802.11 frame is not addressed to the receiving station.

Conversely, the energy detection procedure is used to examine the activity of the medium by measuring the total energy a station receives irrespective of the validity of the 802.11 signal. The medium is considered busy if the received energy is above a defined threshold (Wang *et al.*, 2010; Morote, 2011).

As explained in (Borgo *et al.*, 2004; Wang *et al.*, 2010), a station, before initiating a transmission, is required to sense the channel using either ED or CS (or both) to check the channel's status since another station may be busy transmitting on the medium. Thus, a node, in a Distributed coordination function InterFrame Space (DIFS) time interval only transmits its packet if the channel is sensed idle. Otherwise, the node delays its transmission. As the channel becomes idle for a DIFS interval, the node will generate a random backoff delay uniformly chosen in a contention window (CW), that is, [0, W], where W is the size of the CW.

The backoff timer decreases by one as long as the channel is sensed idle for a backoff time slot. The backoff counter will be frozen when a transmission is detected on the channel, and resumed when the channel is sensed idle again for a DIFS interval. When the backoff timer counts down to zero, the node transmits a packet. Immediately after receiving a packet correctly, the destination node waits for a Short InterFrame Spacing (SIFS) interval and then sends an acknowledgement (ACK) back to the source node. If the source node receives the ACK, the size of CW remains the same value; otherwise, it doubles.

However, carrier sensing as a means of sensing the availability of the channel becomes less effective in the event of the issue of hidden terminal station that may occur while the receiver is within the range of several transmitters that are unable to hear each other. Consequently, 802.11 includes a request-to-send/clear-to-send (RTS/CTS) protocol (the collision avoidance scheme of the CSMA/CA) which, in essence, notifies other users that the medium is expected to be busy and therefore provides a virtual carrier sense. A station may, therefore, defer its transmission in response to either its physical or its virtual carrier sense (Marks, 2002).

The virtual carrier sensing of RTS/CTS mechanisms employs RTS/CTS packets exchange for channel reservation. The sender at first transmits a RTS frame to its receiver. The receiver sends a CTS frame in response. All other entities receiving a RTS, CTS or both mark the channel as busy by updating their NAV with prescribed duration of the talk time defined in sender's RTS and/or receiver's CTS. After the reservation, the sender transmits the DATA frame and receives an ACK (Islam *et al.*, 2010).

As RTS/CTS mechanism is more effective to address issue of the hidden terminal in a wireless network with a moderate number of hidden terminals, it, however, generates some control traffic overhead cost (Marks, 2002; Borgo *et al.*, 2004). Based on this, the adoption of RTS/CTS mechanism is discourage in a WMN environment (Morote, 2011; Oki, 2013). Bearing this in mind, the RTS/CTS mechanisms were disabled in all the simulation scenarios in this study using a threshold of 2500 bytes as in the case of (Morote, 2011).

Moreover, the physical model adopted in all the simulation environments was IEEE 802.11a. 802.11a adopts Orthogonal Frequency-Division Multiplexing (OFDM) modulation while its signal is transmitted at 5 Ghz band. The transmission rate/bit rate of 802.11a module include 6 Mbps, 9 Mbps, 12 Mbps, 18 Mbps, 24 Mbps, 36 Mbps, 48 Mbps and 54 Mbps The OFDM modulation scheme of 802.11a is an efficient coding technique that splits the radio signal into several sub signals before they reach/arrive a receiver. A feature that is advantageous to achieve reduced interference between signals (Borgo *et al.*, 2004).

It is, however, important to highlight that each bit rate requires a different signal to noise and interference ratio (SNIR) to ensure a fixed bit error rate (BER). Thus, at an instance of transmission power, the maximum distance for correct reception decreases as the transmission rate increases. In other words, it can be simply said that transmission at lower bit rate are more reliable and are able to reach/arrive longer distance than transmission at higher bit rate (Lundgren, Nordströ and Tschudin, 2002; Borgo *et al.*, 2004). This is evident in (Morote, 2011) where a comparison test was presented between 802.11a's 6 Mbps and 12 Mbps bit rates. In the test, 6 Mbps performed better than 12 Mbps bit rate in terms of packet delivery fraction (PDF). Therefore the bit rate adopted for the 802.11a protocol used for all the simulation scenarios in this study was set to 6 Mbps bit rate. Furthermore, in all the simulation experiment, a log-

distance path loss propagation model was used in predicting the loss a signal might encounter in vast network areas over distance.

6.2.2 Routing Protocol Configurations

In order to investigate the impact of the presence of malicious black hole nodes in WMNs, there was a need to setup the network in such a way that it provided its optimal performance while all the participating nodes were exhibiting normal routing behavior. Thus, the real performance of the network could be evaluated based on the routing mechanism/capability of the adopted base routing protocol and other influential factors which may include interference, link quality, PHY and MAC protocols, network size, hop count and network traffic load. As discussed earlier in Section 3.2, the study in (Morote, 2011) evaluated the performance of IEEE 802.11s HWMP layer 2 routing protocol against layer 3 AODV routing using NS-3 simulator. The study revealed that HWMP outperformed AODV in terms of end-to-end delay and routing load. On the other hand, it maintained that the experiments presented similar results in terms of PDF performance for both HWMP and AODV. The study further revealed that the reactive routing approach of HWMP is suitable for internal network traffic while it argued that the proactive tree routing approach is suitable for external network traffic in WMNs. Internal network traffic was assumed and initiated in all the simulation experiment scenarios performed in this research while the reactive routing approach of AODV routing was adopted. The setup for all the simulation scenarios performed in this study was built on the setup model in (Morote, 2011) as mentioned earlier in Section 6.2. This was based on the sophisticated approach adopted in the study.

However, it is important to highlight that the study in (Morote, 2011) used an older release of NS-3 which was earlier than release ns-3.15. The version of NS-3 simulator adopted in this study was NS-3 release 3.25 (*ns-3.25 « ns-3,* 2016) which was released in March, 2016 and it was the latest release as at when this research was commenced. In each release, new features are always added which as a result, some modules or models might have been rewritten, restructured, upgraded or moved to another directory in the distributed source folder for the new/latest release. This might also mean that the path to some header files/helper API might have changed. All these factors may lead to changes in the approach to call a method or inherit the instance of a class of a particular module or model in the user's simulation code compared to an earlier

release. An example of such changes is the NS-3 random variable wrapper. In ns-3.3 and earlier, NS-3 simulations adopted a different wrapper class called *ns3::RandomVariable*. As of ns-3.15, the class has been replaced by *ns3::RandomVariableStream*. Though the underlying pseudo-random number generator is still maintained ('ns-3 Manual', 2017).

Taking cognizance of these, a series of simulations were performed prior to the core simulation experiments performed in this study to evaluate the current effectiveness of the simulation model setup in (Morote, 2011). This was to enhance the findings obtained from the intended simulation experiments though minimal connection flows were adopted for the core simulation experiments performed. Hence, after various rigorous tests, the configuration parameters of the environment setup and routing protocols are as shown in the Table 6.1. Some of the routing protocol default configuration parameters defined in (Perkins, Belding-Royer and Das, 2003) and implemented in (*ns-3.25 « ns-3*, 2016) AODV model were changed as done in (Morote, 2011) to provide optimal routing capability for the various network scenarios evaluated.

Table 6. 1: Environment and Routing Protocol Configuration Parameters

Parameters	Values
PHY/MAC	802.11a
Transmission gain	1 dB
Reception gain	1 dB
Transmission power level	1
Maximum available transmission level	18 dbm
Maximum available reception level	18 dbm
Reception noise Figure	7 dB
Transmission rate	6 Mbps
Propagation model	Log-distance path loss
Routing protocol	AODV
Hello interval	3 s
Route Request retry	5
Active Route timeout	100 s
Allowed hello loss	20
Destination only	Enable

6.2.3 Network Topology and Scenarios

In the simulation experiments, mesh nodes of each network size were distributed over grid topologies. The distribution of the mesh nodes into grid topologies was based on its suitability and common adoption for mesh networking compared to random distribution of mesh nodes (Robinson and Knightly, 2007; Morote, 2011; Alotaibi and Mukherjee, 2012; Benyamina, Hafid and Gendreau, 2012; Saadi, Bouchaib and Haqiq, 2014). The network grid size include a 4x4 (16 nodes), 6x6 (36 nodes), 8x8 (64 nodes), 10x10 (100 nodes) to 12x12 (144 nodes). In each grid topology, all the participating nodes were configured as static backbone/infrastructure mesh routers that could relay packets on behalf of other mesh routers. The distance between each neighboring node was 170 meters. This was to ensure that each participating node could only establish mesh connection with its immediate neighbors and adopt the routing mechanism of the routing protocol to establish forwarding path to distant/remote destinations. Moreover, in the simulation setup scripts (a model simulation script for the evaluated mesh grids is presented in appendix A), two types of nodes were defined, "malicious" and "not_malicious". This was to easily enable or disable the malicious behavior of the node(s) defined as the malicious node when the network was intended to be free or not free of malicious node(s) attacks in the simulation scenarios.

The numbers of malicious nodes present while each network grid was under attack include one and two black hole nodes. The experiments was intended to explicitly include 1 and 2 black hole nodes in mutually exclusive experiments for each network grid simulated. This was to see the effect of the attack with increased number of black hole nodes in each network grid simulated. The locations of black hole nodes were varied within each network grid under both 1 and 2 black hole nodes independently. The performance metrics when each network grid was under 1 malicious node attack were computed based on the average of varying location of 1 node at a time for each grid and the procedure was repeated under the events of the presence of 2 black hole nodes once at a time in each network grid simulated. It is however, important to highlight that the varying location of 1 black hole node at a time was done over 2 independent variations in each network grid. The events of 2 black hole nodes at a time comprise 2 mutually inclusive black hole nodes that also include each of the previously used single black hole nodes. The

events of varying the locations of 2 black hole nodes were done on 2 independent variations in each network grid simulated.

The topologies for each WMN grid (4x4 (16 nodes), 6x6 (36 nodes), 8x8 (64 nodes), 10x10 (100 nodes) to 12x12 (144 nodes) grid respectively) evaluated are represented in appendix B, C, D, E, and F respectively. In each topology representation, mesh nodes are red nodes while the physical peer wireless links/connections between neighbour nodes are represented with green bidirectional links as shown in the figures. In each evaluated WMN grid, the logical arrangements of the nodes start from the top row to the bottom row of the grid. The node distribution could be best explained as a two-dimensional matrix. That is, the nodes are arranged in a row by column order from the first top row of each grid. The position index of the nodes arrangement starts with 0 (as in the case of an array index) from the left to the right on the first row. This means that the position index of the first node is 0 in the order of row and column. Therefore, each node is named according to its position index. Thus, in each mesh grid, the first node with position index 0 is named as Node 0. For instance, in the 4x4 (16 nodes) mesh grid, the first node is Node 0 while the sixteenth node is Node 15.

As in the case of black hole implementation test simulation experiments presented in Section 5.4 of the previous chapter, UDP was adopted as the transport protocol in the simulation experiments so as to accurately capture the simulation results. Thus, the source node could continuously send out UDP packets till the end of its connection with the destination node, even if they were maliciously dropped by the black hole node during the simulation. In the simulation experiments for each mesh grid, some nodes were set as source and destination nodes to establish connections with one another to transfer and receive UDP Constant Bit Rate (CBR) packets to/from one another. In each of the scenarios evaluated, the malicious nodes were exempted as a source and/or destination nodes. As a result, the defined malicious node could only serve as an intermediate node for the relay of packets in the network simulations scenarios of the WMNs with and without black hole malicious packet drop attack. This was to really have a fair comparison of WMNs performance while under attack and without malicious black hole attack.

In all the mesh grid evaluated, 16 CBR connection flows were established between randomly generated source and destination nodes. It is, however, important to note that the 16 CBR

connection flows were implemented distinctively in all the simulation experiments scenarios, both under the conditions that each network grid was free from black hole attacks and under the conditions that each mesh grid was under malicious black hole attack. It is also important to highlight that 16 CBR connections were implemented in the simulation scenarios in each mesh grid to ensure close relativity in the values of the performance evaluation metrics (see Section 6.3) adopted in this study. This is explained further and better under the discussions of the simulation experiment results in Section 6.4.

Furthermore, NS-3 supports a number of random variable objects from the base class *RandomVariableStream*. The objects are derived from *ns3::Object* and they are handled by smart pointers. Though NS-3 simulations use a fixed seed and run number by default, simulations can still be configured to produce deterministic or random results. The seed and run values are stored in two *ns3::GlobalValue* instances: g_rngSeed and g_rngRun respectively. In essence, if there is any random generation of input parameters in a simulation and it is also configured to use a fixed, deterministic seed with the same run number, the simulation will always generate the same output each time it is run. To compute simulation statistics on a large number of independent runs, a user can run a simulation as a sequence of independent trials. To achieve this, either the global seed can be changed, then the simulation is rerun, or the substream state of the RNG (a long sequence of pseudo- random number generator (PRNG)) can be advanced, which is also termed as incrementing the run number. However, there is no guarantee that the streams produced by two random seeds will not overlap. The more sophisticated statistical technique to configure multiple independent repeated experiments is to use a fixed seed and to increase the run number. Through this technique, a maximum of $2.3*10^{15}$ independent replications can be implemented using the substreams (‘ns-3 Manual’, 2017).

In this study, NS-3 default fixed seed and default run number were adopted. The repeated simulation using the default seed and default run number was based on the exemption of the black hole nodes (malicious behavior disabled) as the source and/or destination nodes in the networks without attack. The repeated simulations were done in two independent scenarios while two different pairs of the black hole nodes were disabled and exempted as the CBR source and destination nodes. The procedures for the repeated scenarios using the default seed and the default run number for the networks under one and two nodes black hole were discussed earlier

in this section. The black hole nodes were also exempted as the source node and/or destination nodes.

In addition, the packet size was set to 1024 bytes at different data transmission rates of 100 kbps, 200 kbps, 300 kbps, 400 kbps and 500 kbps in mutually exclusive scenarios for each evaluated mesh grid. In other words, the experimentation was done setting the packet size to 1024 bytes at 100kbps, 200kbps, 300 kbps, 400 kbps and 500 kbps data rates in different independent scenarios for both mesh grids free of malicious attack and mesh grids under attack. This was to also investigate the optimal performance of the simulated networks grids under increasing traffic load with respect to the number of CBR connections and the network configuration settings even, when the network was free from packet drop attack. In each scenario, the simulations were run for the duration of 1000s. As in (Morote, 2011), the connection flows established between the randomly generated source and destination follows a uniform arrival distribution. The duration of an individual flow was set using an exponential variable with a mean value of 100 seconds. Therefore, various connection flows can run concurrently during the entire simulation period.

It is however, important to highlight that the CBR application adopted in all the simulation scenarios was generated using *ns3::OnOffApplication* class instantiated through the helper class *ns3::OnOffHelper* in the simulation code. As adapted and explained in (*ns-3: ns3::OnOffApplication Class Reference*, 2017), *ns3::OnOffApplication* generates traffic to a single destination in accordance to an On/Off pattern. After the method *Application::StartApplication* is called, the "On" and "Off" states alternate. The duration of each of these states is determined with the *onTime* and the *offTime* random variables. During the "Off" state, no traffic is generated. During the "On" state, CBR traffic is generated. The CBR traffic is characterized by the specified "data rate" and "packet size".

As noted in (*ns-3: ns3::OnOffApplication Class Reference*, 2017), when an application is started, the first packet transmission occurs after a delay equal to the ratio of packet size and bit rate (i.e. packet size/bit rate). Also, it was noted that when an application transits into an off state in between packet transmissions, the remaining time until when the next transmission would have occurred is cached and used when the application starts up again. In a given example where the

packet size was set to 1000 bits and bit rate was set to 500 bits/sec, it was explained that if the application was started at time 3 seconds, the first packet transmission would be scheduled for time 5 seconds (i.e. 3 + (1000/500) = 5) and subsequent transmissions at 2 second intervals. Furthermore, it was explained that if the application was rather stopped at time 4 seconds, and restarted at time 5.5 seconds, the first packet would be transmitted at time 6.5 seconds. This is to say that when the application was stopped at 4 seconds, there was only 1 second left before the first packet would be transmitted based on the defined transmission schedule principle. Thus, at the pause time, the remaining information was cached and used to schedule the next transmission upon the continuation of the packet transmission at time 5.5 seconds.

However, bearing in mind that diverse packet sizes/payloads could be generated and varied across the network, it is important to highlight that the choices of the packet size and the applied data rates mentioned earlier were the ones used. This was based on the perspective that with the used packet size and the varied transmission rates, lower to higher traffic load were generated and were able to enhance the overall context of the intended simulation experiments. In order words, the number of packets generated in all the simulation experiments with the packet size of 1024 bytes over the data rates of 100, 200, 300, 400, and 500 kbps in mutually exclusive scenarios delineates that the higher the data rate, the higher the number of packets generated per second with respect to the packet size and the NS-3 OnOff CBR application transmission schedule principle explained earlier. Hence, considering the entire network configuration settings, the increasing data rates were able to expose the effect of transiting from lower to increasing traffic load on the network performance while there was no black hole node active in each mesh grid simulated.

The details of the common parameters used for the simulation experiments topology and scenarios are represented in Table 6.2.

Table 6. 2: Topology and Scenario parameters

Parameters	Value
Simulation time	1000 seconds
Number of Nodes	16, 36, 64, 100, 144
Number of Black hole nodes	1, 2
Node Mobility	Static
Nodes Distribution	Grid
Traffic Type	CBR
Packet Size	1024 bytes
Data Rate	100 kbps, 200 kbps, 300 kbps, 400 kbps, 500 kbps
No of CBR Connections	16 flows

6.3 Performance Evaluation Metrics

This section presents the evaluation metrics adopted for performance evaluation in this study. To enhance the discussions of the simulation experiment results (adopting the metrics) in Section 6.4, efforts are made in this section to explicitly show the derivation of each performance evaluation metric as used in the simulation experiment codes. In all the performance evaluation metrics, computations were done with the summation (∑) of the average metrics gathered in each individual connection flow of every simulated scenario, both in the mesh grids that were free from attack and the mesh grids under attack. Hence, the term 'individual' as it may appear or may be used in each performance evaluation metric's derivation indicates that the computation of each metric is based on the summation of the average metric computed from each (individual) connection flow in each mesh grid simulated. The followings are the measured parameters of the performance measurement procedure.

6.3.1 Packet Delivery Ratio (PDR)

The PDR is the ratio of data packets received by destination node over the total data packets transferred by the source node. A lower percentage usually indicates a larger amount of packets drop in the network due to link failures or network traffic congestion. The following expressions were used to compute PDR:

Given that,

$$Individual\ Connection\ Flow\ Packet\ received = \boldsymbol{rxPackets}$$

$$\boldsymbol{Packet\ received} = Individual\ Connection\ Flow\ Packet\ recieved$$

Also, given that,

$$Individual\ Connection\ Flow\ Packet\ transfered = \boldsymbol{txPackets}$$

$$\boldsymbol{Packet\ transfered} = Individual\ Connection\ Flow\ Packet\ transfered$$

Therefore:

$$\boldsymbol{PDR = \frac{\sum Packet\ received}{\sum Packet\ transfered} \times 100}$$

Note: txPackets and rxPackets are *FlowMonitor::Stats* attributes described in Subsection 4.7.1.1

6.3.2 Average Throughput

The throughput is the number of bits successfully delivered per second at each destination node. It is, however, important to note that the data rate is influential on the generated throughput values. That is, the higher the data rate, the higher the throughput measured in kbps. Hence, it is expected that for instance, when the data rate is maintained at 200 kbps in each network grid, the throughput value, by default, should be higher than the obtained throughput value when the data rate is maintained at 100 kbps. This is further explained under the discussions of the throughput results in Subsection 6.2.2. However, the throughput was calculated using the following expressions:

Given that,

$$\text{Individual Connection Packet reception time difference} = \mathbf{timeLastRxPacket - timeFirstTxPacket}$$

$$Flow's\ Packet\ reception\ duration = \text{Individual Connection Packet reception time difference}$$

Let $\text{RxTDiff}_i = Flow's\ Packet\ reception\ duration$

Where i = individual connection flow

$$\boldsymbol{Individual\ Connection\ Throughput} = (\boldsymbol{rxBytes} * bytes) * \left(\frac{8 * bits * kb}{bytes * 1024 * bits * RxTDiff_i * 10^{-9}s}\right)$$

Where kb = kilobits, s = seconds

$$\boldsymbol{Flow's\ Throughput} = Individual\ Connection\ Throughput$$

Therefore,

$$\boldsymbol{Throughput} = \sum \boldsymbol{Flow's\ Throughput}$$

Note: **timeLastRxPacket**, **timeLastTxPacket** and **rxBytes** are *FlowMonitor::Stats* attributes described in Subsection 4.7.1.1

6.3.3 Average End-to-End Delay

This is the time taken by a packet to arrive at the destination from the source. In essence, this time includes buffering, queuing, retransmission and propagation delays. The following expressions were used for its computation:

Given that,

$$Individual\ Connection\ Flow\ Packet\ received = \boldsymbol{rxPackets}$$

$$\boldsymbol{Packet\ Received} = Individual\ Connection\ Flow\ Packet\ received$$

Also, given that,

$$Individual\ Connection\ Delay = \boldsymbol{delaySum}$$

$$\boldsymbol{Flow's\ delay} = Individual\ Connection\ Delay$$

$$\boldsymbol{Delay} = \frac{\sum \boldsymbol{Flow's\ delay}}{\sum \boldsymbol{Packet\ Recieved}}$$

Note: **rxPackets** and **delaySum** are *FlowMonitor::Stats* attributes described in Subsection 4.7.1.1

6.4 Simulation Experiment Results and Discussions

In this section, the results of the performance evaluation of the simulation scenarios are presented. As mentioned earlier in Subsection 6.2.3, the simulated scenarios of the network were evaluated over a range of increasing network size and 16 CBR connections for both WMN free of malicious black hole attack and the other under malicious black hole attack. The results of the measured performance metrics are discussed in the following Subsections.

6.4.1 Packet Delivery Ratio (PDR)

This subsection discusses the results of the packet delivery ratio obtained from the simulation scenarios for both WMN free of malicious black hole attacks and WMN infrastructure under malicious black hole attack. The results of the packet delivery ratio were computed using the functions presented in subsection 6.3.1. The average percentage of the overall packet sent and successfully received at the intended destination nodes were measured to determine how the multi-hop and self-healing path selection and forwarding capabilities of WMN could efficiently hop through in the absence and presence of malicious packet drop attack of black hole nodes. The results of the measured packet delivery ratio are represented in the Figures 6.1, 6.2, 6.3, 6.4 and 6.5.

Generally, it can be observed from the graph and table representations of the PDR that the networks that were free from attacks experienced a very optimal to a very fair percentage of packet delivery ratio as the data rate increases. In all the scenarios where the data rate was maintained at 100 kbps (the lowest data rate), the values of the PDRs obtained across all the network sizes peaked at PDR values that range from 98.9% to 99.7%. On the other hand, while the data rate was maintained at 500 kbps (the highest data rate), the values of the PDRs obtained across all the network sizes have decreased to values that range from 72.1% to 87.2% which can still be said to be a fair percentage in terms of packet delivery compared to the values obtain while the networks were under attacks. In a more clarified term, while looking at the PDR values of individual mesh grid size under each data transmission rate (i.e. 100, 200, 300, 400 and 500 kbps) PDR graph and table representations, the PDR values continuously presents slight decrease in value. This effect could be attributed to the fact that the higher the data rate, the higher the traffic load that would be generated along the routing channels between the data packet source

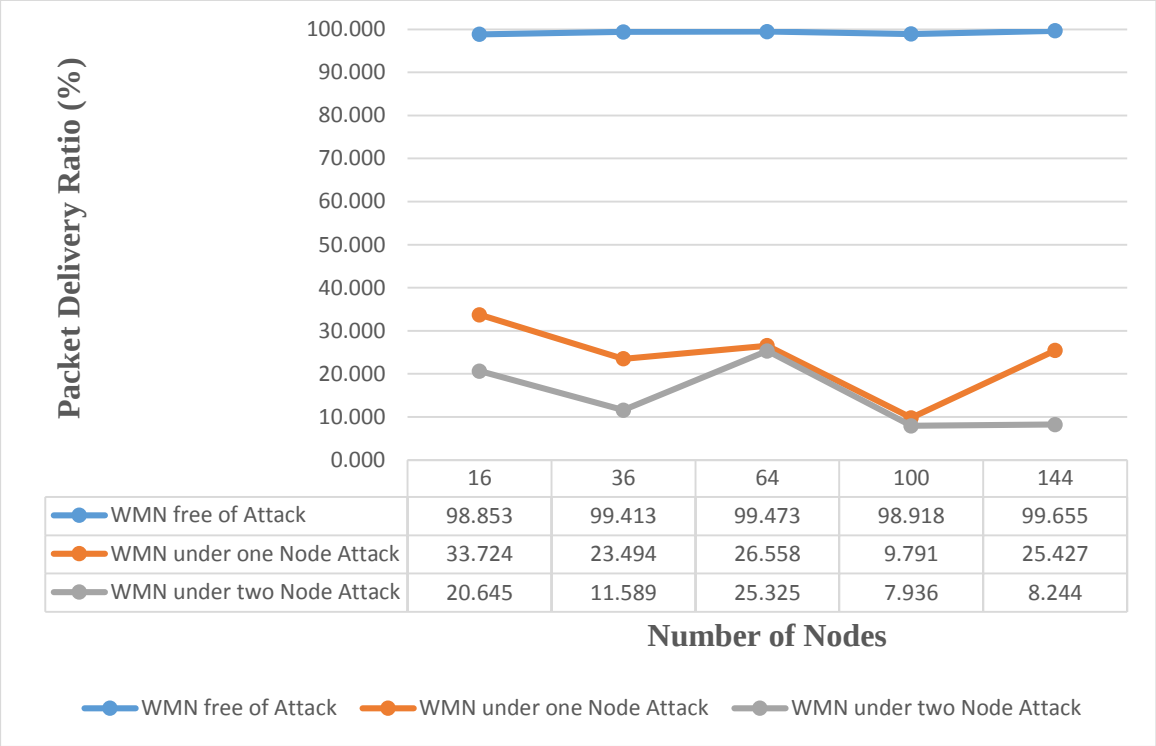
and destination nodes. Therefore, as the traffic load increases, the networks become more congested which may increase collisions in the networks. Consequently, more data packets were dropped due to the reduced link quality that might have resulted from the increased frame error rates caused by the collisions along the routing channels.

Furthermore, in the PDR graph and table representations of the 100 kbps data rate while the network grids were free from attacks, the PDRs values while looking across the PDR trend from 16 node size to 144 node size mesh grids approximately maintains a very consistent trend. On the other hand, from 200 kbps to 500 kbps PDR graph and table representations, the trend of the PDRs across the network sizes somewhat appear unstable. This behavior could be attributed to the fact that both data packet source and destination nodes were randomly generated in each network grid. In essence, this means that the path built by the routing protocol from a randomly generated source to a randomly generated multi-hop destination would be quite dynamic/different as the case may be in each mesh grid size. While some links might maintain a very good quality, some links may otherwise maintain lesser quality in some grids though the same number of CBR connection flows (16 flows) were implemented in all the mesh grid sizes. Invariably, if the source and destination nodes were strategically selected (for example, if the source and destination nodes were selected from the diagonals nodes) in each mesh grid, the paths built between the source and destination node would still be dynamic with respect to the multi-hop nature of the mesh routing. Nevertheless, under the earlier identified data rates (200-500 kbps) PDR graphs while the networks were free from attack, the point worth stressing is the effect of the increasing data transfer rate on the performance of the networks in terms of the PDR as the case may be in each mesh grid size.

However, as shown in the graphs and tables representations of the PDR results in Figures 6.1, 6.2, 6.3, 6.4 and 6.5, the PDR of the network under attacks were significantly degraded. In all cases, the WMNs grids under attack experienced a massive low packet delivery ratio compared to the PDR achieved in the WMNs grids without malicious packet drop attack. As represented in the Figures 6.1-6.5, the considerable low percentage of packet delivered while the network was under both one and two black hole attacks depicts that a large amount of packet sent from the source nodes to the destination nodes were intercepted and dropped by the black hole nodes. Also, as it was observed during the experiments, the massive amount of packet drop in the

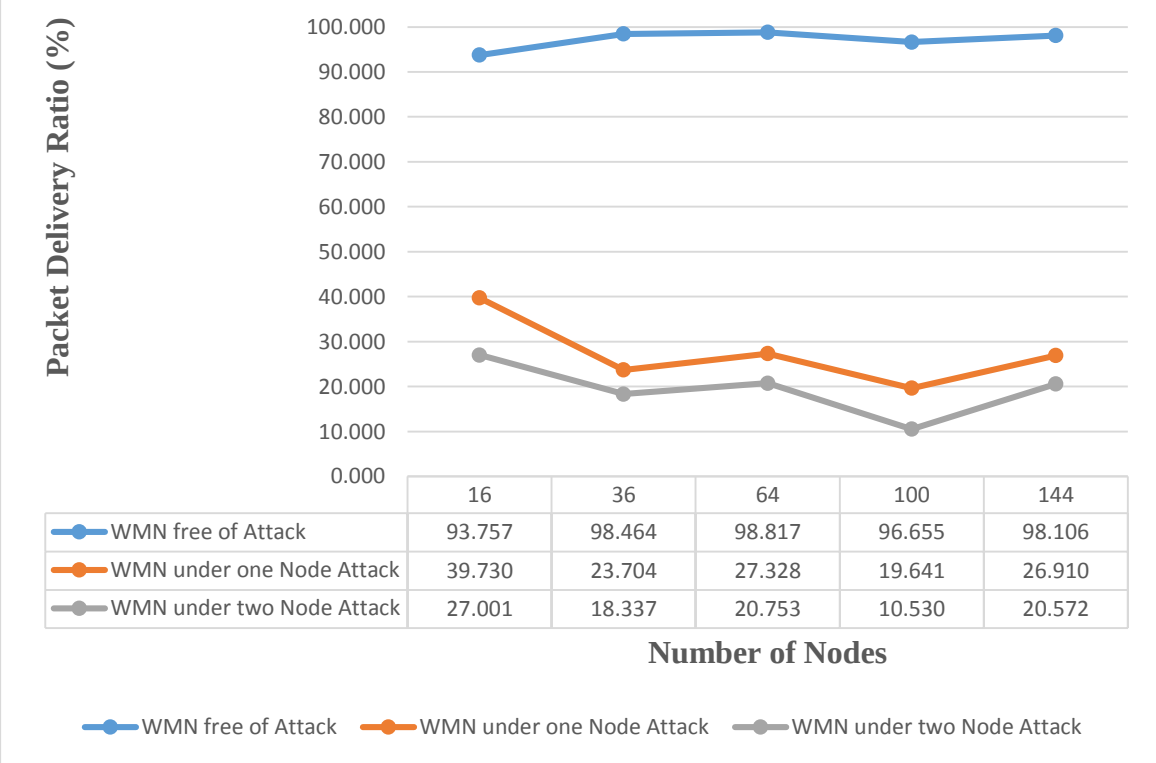
WMNs under attack could also be attributed to more collisions experienced along the routes towards the black hole nodes. This was because the data packet forwarding routes were mostly attracted/channelled towards the black hole node through its fake route reply to most of the route request sent by the source nodes. Hence, more interference were caused in the network even when the black hole node was unable to intercept the data packet transmitted from some source nodes to their intended destination nodes. Besides, the trend in the values of the PDR across the simulated network grids may appear more erratic under one and two black holes in all the PDR graph and table representations, this behaviour was expected since the conditions in each network grid would be dynamic from the other as the case may be.

Comparatively, as in the cases of the fair effect of transmission rate from 200 kbps to 500 kbps on the PDR results (as earlier discussed) of the WMNs without malicious packet drop attack, the effect of the malicious packet drop attack on the packet delivery ratio of the WMNs grids under the malicious attack was massive. Another point worth discussing here is the effect of the presence of more black hole nodes on the packet delivery ratio in the network grids. As it can be observed from the PDR graph and table representations, the values of the PDRs values while two black hole nodes were present in the networks present more decreasing values compared to the PDR values under one black hole node attack. This is to say that the presence of more black hole node may result to absolute denial of service in the network. Taking for instance, in the 300 kbps PDR graph and table representation in Figure 6.3, the PDR under two malicious black hole nodes went as low as 5.72% in a 144 node mesh grid. This means that the packet drop in the network had increased with 86.7% in the presence of two black hole nodes compared to while the network was free from attack. Therefore, it can be said that the values of the PDR in the network might even go as low as 0% while the network is under more black hole attacks. Thus, the comparative analyses of the results obtained in terms of packet delivery ratio from all the evaluation scenarios show that the packet transmission is neither secure nor reliable if there is at least a single black hole node present in the network. The presence of more black hole nodes in the network would intensively impair the network performance in terms of packet transmission. Hence the network performance is degraded and would loose its reliability under such malicious attack.



	16	36	64	100	144
WMN free of Attack	98.853	99.413	99.473	98.918	99.655
WMN under one Node Attack	33.724	23.494	26.558	9.791	25.427
WMN under two Node Attack	20.645	11.589	25.325	7.936	8.244

Figure 6. 1: PDR at 100 kbps data rate



	16	36	64	100	144
WMN free of Attack	93.757	98.464	98.817	96.655	98.106
WMN under one Node Attack	39.730	23.704	27.328	19.641	26.910
WMN under two Node Attack	27.001	18.337	20.753	10.530	20.572

Figure 6. 2: PDR at 200 kbps data rate

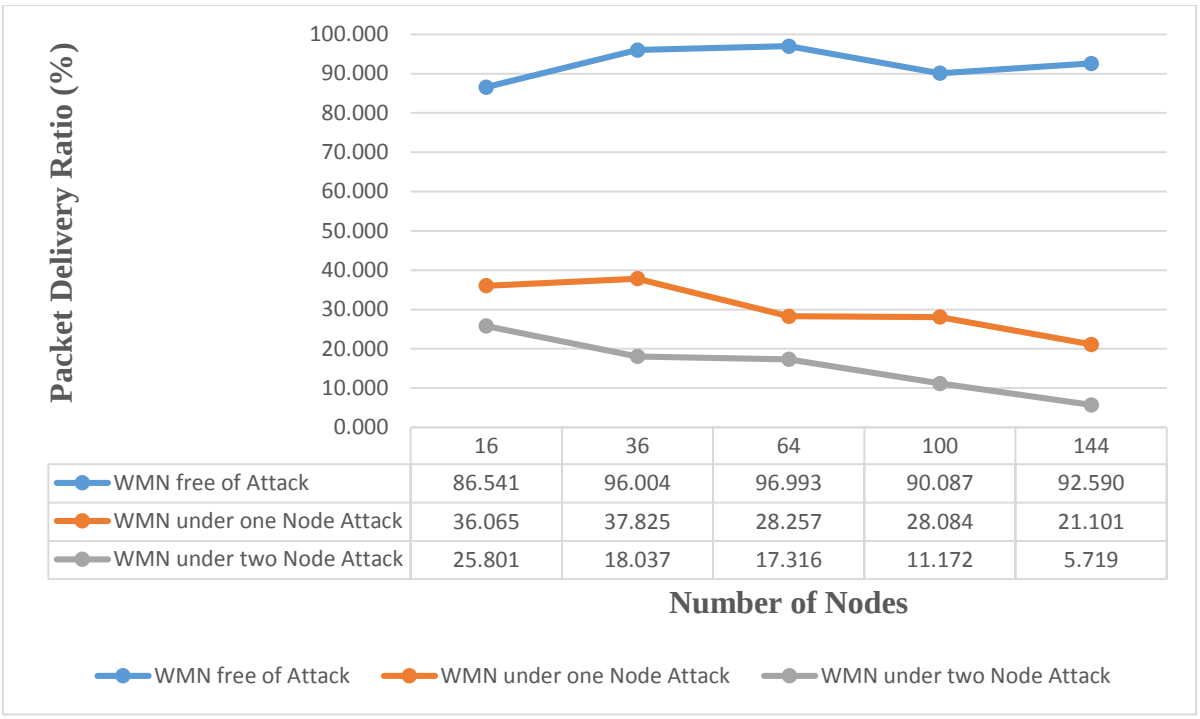


Figure 6. 3 PDR at 300 kbps data rate

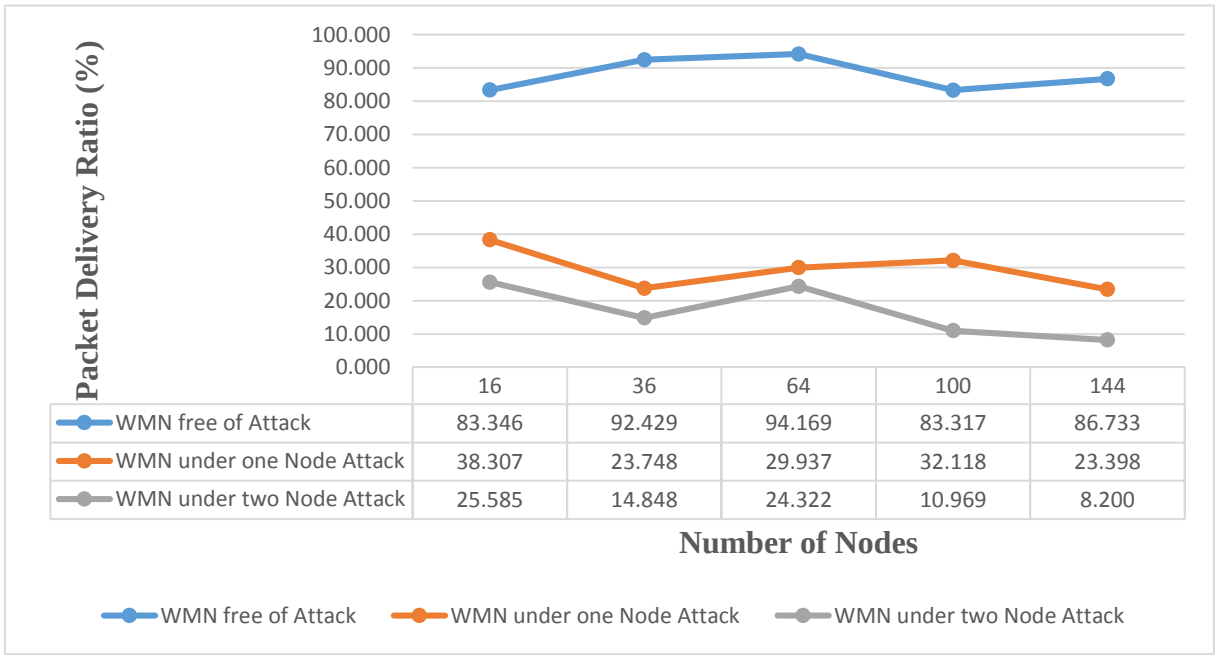


Figure 6. 4: PDR at 400 kbps data rate

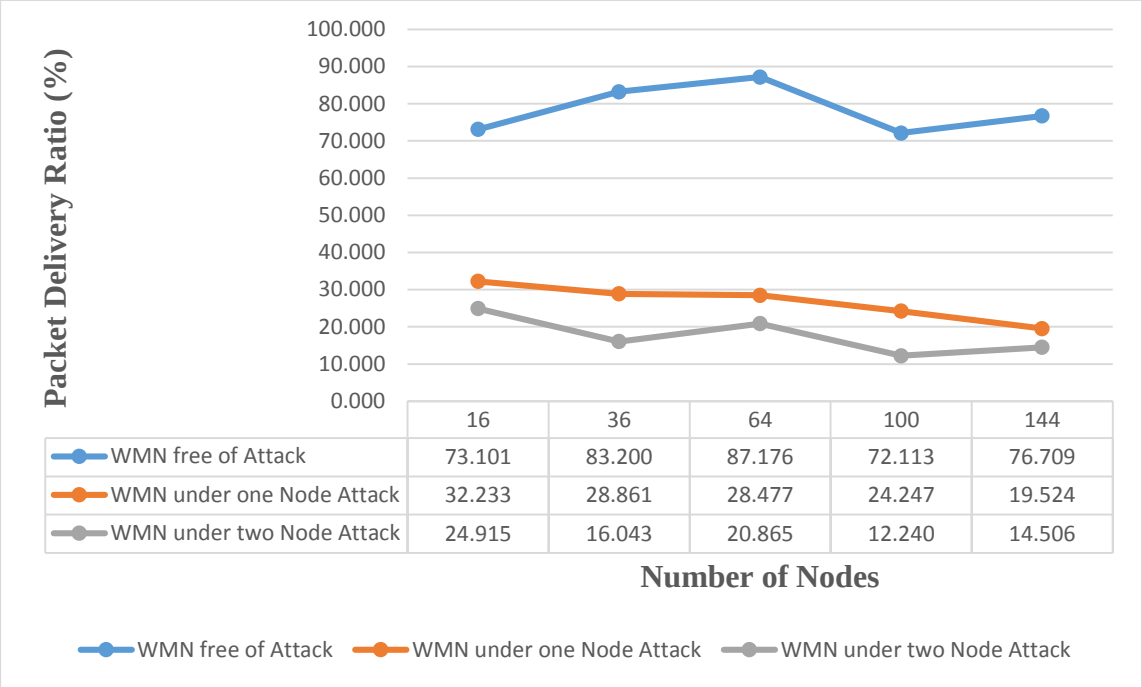


Figure 6. 5: PDR at 500 kbps data rate

6.4.2 Average Throughput

This section explains the results of the average throughput obtained from the simulation scenarios for both WMN free of malicious black hole attacks and WMN infrastructure under malicious black hole attack. The results of the average throughput were computed using the functions presented in subsection 6.3.2. Taking into consideration of the routing mechanism of the adopted AODV routing protocol, the average throughput of the overall packet bits successfully received at the intended destination nodes were measured to determine the effect of malicious packet drop attack on the average throughput of the multi-hop and self-healing path selection and forwarding capabilities of WMN infrastructure. The results of the measured average throughput are represented in Figures 6.6, 6.7, 6.8, 6.9 and 6.10.

In a broader domain of communication/computer networks, the PDR and throughput are usually proportional. That is, the higher the PDR, the higher the throughput and vice versa. Also, the lower the PDR, the lower the throughput and vice versa. Therefore, in a real sense, it could be said unequivocally that the network grids while they were under the conditions that they were free from attack present performance that range from optimal to fair performance with respect to

the throughput values and their corresponding PDR values as the data rate keeps increasing from 100 kbps to 500 kbps. However, while the principle of the proportionality of PDR and throughput values still applies in this study, there is a point worth discussing with respect to the throughput values and the corresponding PDR values in the performed experiments, especially, while the networks were free from attacks.

As it can be observed in the throughput figures of 100 kbps to 500 kbps data rate, the throughput values, as the case may be in each mesh grid size, keeps increasing considerably under each higher data rate (for instance, transiting from a lower data rate of 100 kbps to a higher data rate of 200 kbps and so on). Whereas, the PDR values would have slightly depreciated in the corresponding PDR figure compared to the trend in the PDR figure of the preceding lower data rate. This might be perceived as anomalies but as in the context of the experimental setup in this study, the considerable increment in the throughput values of a higher data rate while the PDR values would have somewhat depreciated is not anomalous. It can be explained from a standpoint that the higher the data rate, the higher the expected throughput value with respect to the total number of CBR connection flows.

In a more clarified term, the expected average throughput value in a connection flow, by default, is expected to be, at least, the same as the data rate value itself or at maximum, a value greater than the data transfer rate but lesser than the data rate multiplied (*) by 1.2 (Morote, 2011) if the achieved PDR is 100% in that connection flow. For instance, if the data rate is maintained at 100 kbps and the PDR is 100% in a connection flow, the expected throughput is at least 100 kbps or at maximum, value greater than 100 kbps but lesser than 120 kbps (100 kbps * 1.2). In addition, if the PDR is 100% while the data rate is maintained at 200 kbps in a connection flow and the PDR is 100% in the connection flow, the average throughput is at least expected to be 200 kbps or at maximum, values between 200 kbps and values lesser than 240 kbps (200 kbps * 1.2). Based on this, the point mentioned in Subsection 6.3.2 that the higher the data rate the higher the throughput values can be justified.

Moreover, it was discussed earlier in Section 6.3 that the computation of each performance metric was based on the summation of the average performance metric of each individual connection flow. Bearing in mind that the number of connection flows implemented in all

scenarios was 16 CBR connection flows, the total throughput expected in any of the simulated mesh grid size was 16 multiply by the instance of the data rate or 16 multiply by the maximum value lesser than the data rate multiply by 1.2 if 100 % PDR is achieved in all the 16 CBR connection flows. For instance, the expected throughput, by default, in each simulated network grid while the data rate was maintained at 100 kbps and assuming a 100% PDR was achieved in all the 16 CBR connection flows was 1600 kbps (i.e. 100 kbps*16) or 1904 kbps (i.e. (100 kbps * 1.2) – 1 = 119) * 16)). Also, the expected throughput, by default, in each simulated network grids while the data rate was maintained at 200 kbps and assuming a 100% PDR is achieved in all the 16 CBR connection flows was 3200 kbps (i.e. 200 kbps*16) or 3824 kbps (i.e. (200 kbps * 1.2) – 1 = 239) * 16)). The illustrated instances apply to the other data rates. Hence, the considerable incremental values of the throughput while transiting from a lower data rate to a higher data rate was expected based on these illustrations.

In the same vein, it was discussed in Subsection 6.2.3 that 16 CBR connection flows were implemented for close relativity in the values of the performance metrics. This is practically applicable under the throughput and its corresponding PDR value. As it was just foregrounded, at any data rate instance, the total throughput is dependent on the number of flows in a network size. If therefore, at an instance of data rate, different CBR connection flows were implemented in two different mesh grids, the expected throughput will differ unequivocally since the total throughput is computed with respect to the total number of CBR connection flows implemented in the network grids. Based on this, if the two different network sizes for instance, achieve 100% PDR at the same data rate but different number of flows, literally, the throughput will be different across the trend of the corresponding throughput graph representation. In such a condition, the values of the PDR and the corresponding throughput might appear ambiguous/unclear while in actual sense, it is normal. Hence, to keep the throughput values across the simulated network sizes with the corresponding PDR values closely relative taking the effect of each data rate on the packet transmission into consideration, 16 CBR connections flows were implemented in all the networks sizes. The 16 CBR connection flows were implemented to suite the random generation of the source and destination nodes in the least network size (16 node mesh grid).

Having discussed these, it is also important to discuss the slight instability in the trend of the throughput figures for 200 -500 kbps data transfer rate while the network grids were free from attacks. The unstable trend could be related to the point highlighted earlier that the values of the PDR and throughput are usually proportional. Hence, the trend in the throughput values could be likened to the discussion around the slight up and down in the trend of the PDR values across the simulated networks in Subsection 6.4.1.

On another note, it could be observed from the throughput figure of each data rate that the throughput values of each network grid while they were under attacks have considerably depleted compared to the corresponding throughput under conditions that the network grids were free from attacks. In addition, from a broader domain perspective of communication/computer networks, it was mentioned earlier in this subsection that the lower the PDR, the lower the throughput. On this basis, the considerable decrease in the throughput values could be largely related to the considerable decrease in the corresponding PDR due to the considerable amount of packet drop experienced in the network grids while they were under black hole attacks. Furthermore, in the throughput figure of each data rate, it can be observed that the throughput values across the trend of the simulated network grids may appear erratic at some point. This behaviour was expected since the same behaviour occurred under the corresponding PDR. The erratic trend can also be likened to the discussion in Subsection 6.4.1 that it was by no doubt that different behaviour would emerge in each network under the black hole attacks due to the random generation of both source and destination nodes, and the dynamic nature of the path selection in the network. The effect of the black hole attack is, however, conspicuously massive on the performance of the network irrespective of the random selection of source and destination node, mesh grid sizes, and the data transmission rate.

However, there is another point worth discussing with regards to the throughput while the networks were under attack. For instance, while the networks were under two black hole attacks in Figure (6.8), the throughput values for the 36 node grid and 100 node grid are 1165.08 kbps and 1131.935 kbps respectively whereas the corresponding PDR values are 18.037% and 11.172% respectively. Inherently, since the same number of CBR connections were implemented in all the mesh grid sizes, and based on the respective corresponding PDR values of the two mesh grids, it was expected that the difference in the throughput of the 36 node grid and 100

node grid would be greater than it appears. On the hand, the throughput margin appears so close. To investigate this, the Flow Monitor XML trace file for the simulation experiment of the 100 node mesh grid while under the two black hole attacks was explored.

Bearing in mind that the overall computation of each metric was based on the summation of the average metric of the individual connection flows, a particular flow was traced out to have a very low PDR based on the total number of the packets transmitted in that flow while the throughput was so high. While it is nontrivial to append the entire XML data in this thesis, a screenshot of the Flow is presented in Appendix G. Hence, if the expressions for the computation of the individual PDR and throughput as defined in Subsections 6.3.1 and 6.3.2 are applied to the relevant parameters in the represented flow Identifier (flowId), the PDR is 0.730 % against total number of 4928 packets transferred and 36 packets received in the flow while the throughput is 288.831kbps. This may be perceived to be ambiguous but in another sense, it can be said to be the effect of more interference introduced into the network by the black hole node as earlier discussed in Subsection 6.4.1.

In other words, this means that while the black hole node was unable to intercept and drop the packets in this connection flow, the packet transmission flow was interfered by the effect of the resultant interference caused by the black hole node in the network. Thus, the resulting throughput of 288.831 kbps computed with the ratio of the 36 packets ((converted to kilobits (kb) with rxBytes = 37872 bytes * (8 * (bits/bytes)) * (kb / (bits*1024)) = 295.87 kb)) received out of the 4928 packets transferred against the duration of 1.025 s (i.e timeLastRxPacket - timeFirstTxPacket = 775055641868 $*10^{-9}$ s - 774031254099 * 10^{-9} s = 1.025 s) of the total packet received while the resulting PDR of 0.730% computed with the ratio of the total packet received against total transferred packets in the flow (i.e. PDR= (rxPackets / txPackets) *100% = (36/4928) * 100 % = 0.730%) means that there might be eventuality that under this condition, a flow presents a very low PDR but a very high throughput.

In essence, the connotation of the disparity between the PDR and throughput is that the few number of packets received at the destination reached the destination node within the short duration stated above while the source node kept sending the remaining packet till the end of the allotted transmission period for the flow but the packets were never received at the destination

node. Therefore, the PDR computation with the ratio of the packets received against the total packet transmitted in the entire connection flow duration and the computation of the throughput based on the total packets received and the short duration of the their reception at the destination node would result in a very low PDR but a very high throughput. Consequently, the entire PDR and throughput computation based on the average metric of the individual connection flows would be affected and, hence, present a lower PDR against higher throughput. This might even be more conspicuous if there are more connection flows with this type of disparity in the scenario. This discussion could help enhance the understanding of any other throughput and corresponding PDR with such disparity under malicious conditions in the throughput and PDR figures. The discussion of the delay results is presented in the next subsection.

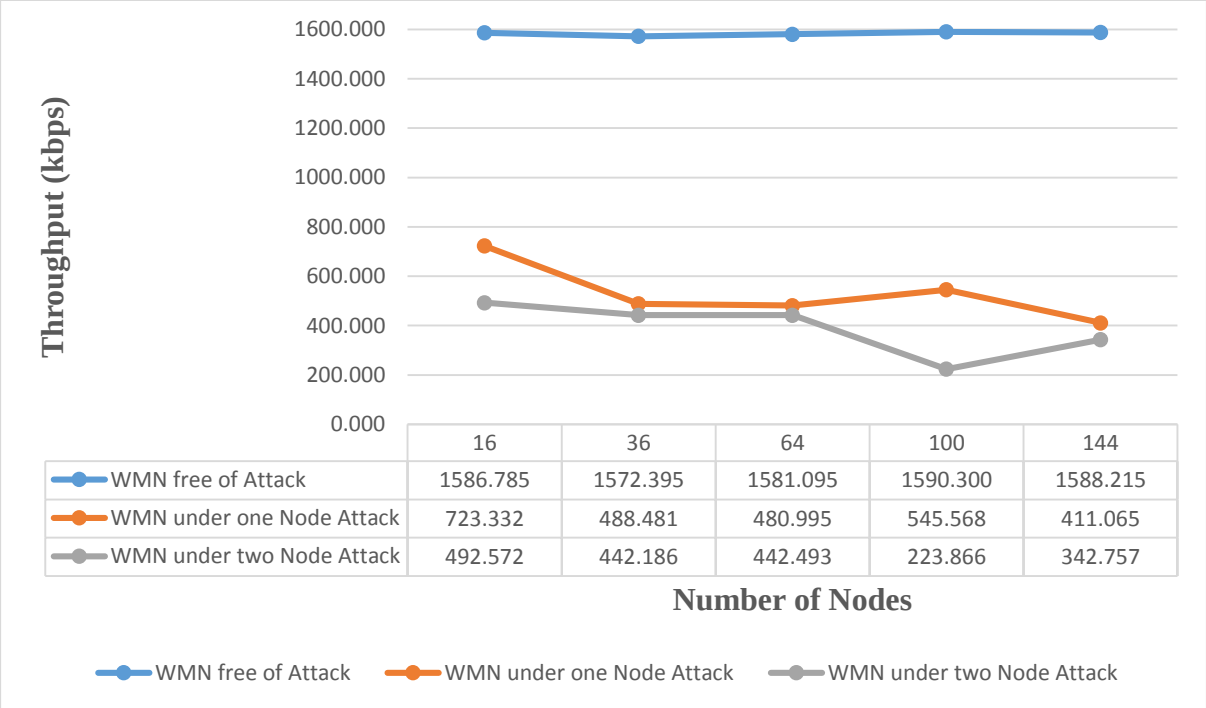


Figure 6. 6: Average throughput at 100 kbps data rate

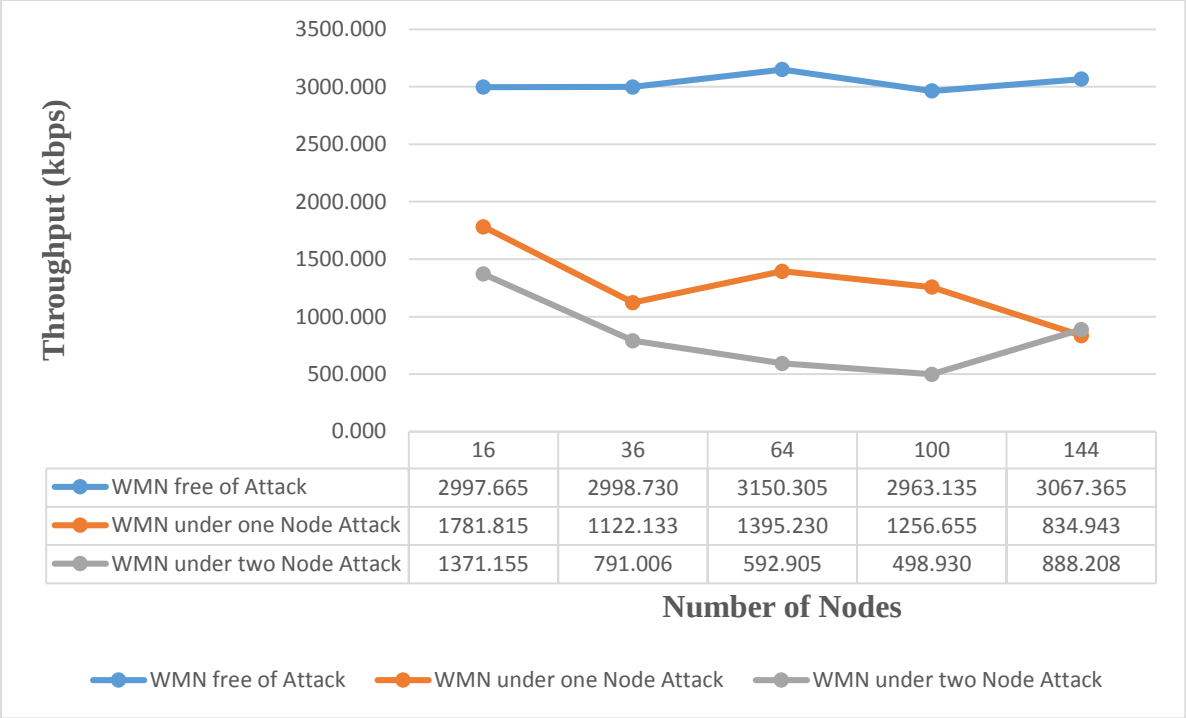


Figure 6. 7: Average throughput at 200 kbps data rate

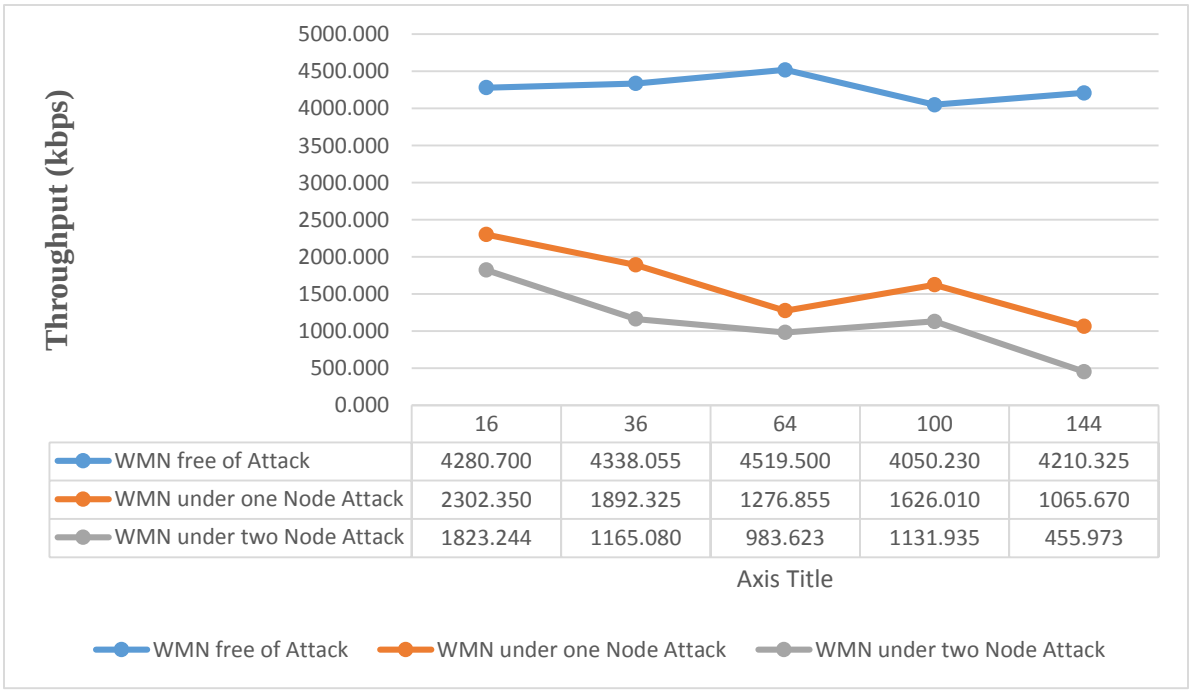


Figure 6. 8: Average throughput at 300 kbps data rate

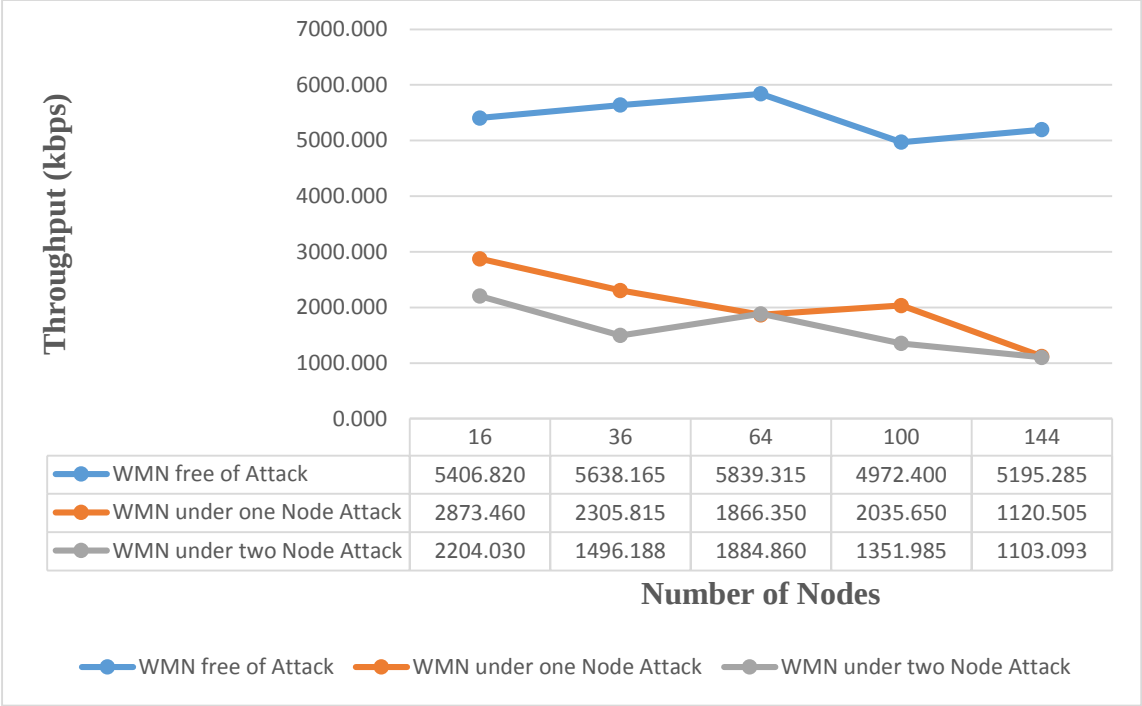


Figure 6. 9: Average throughput at 400 kbps data rate

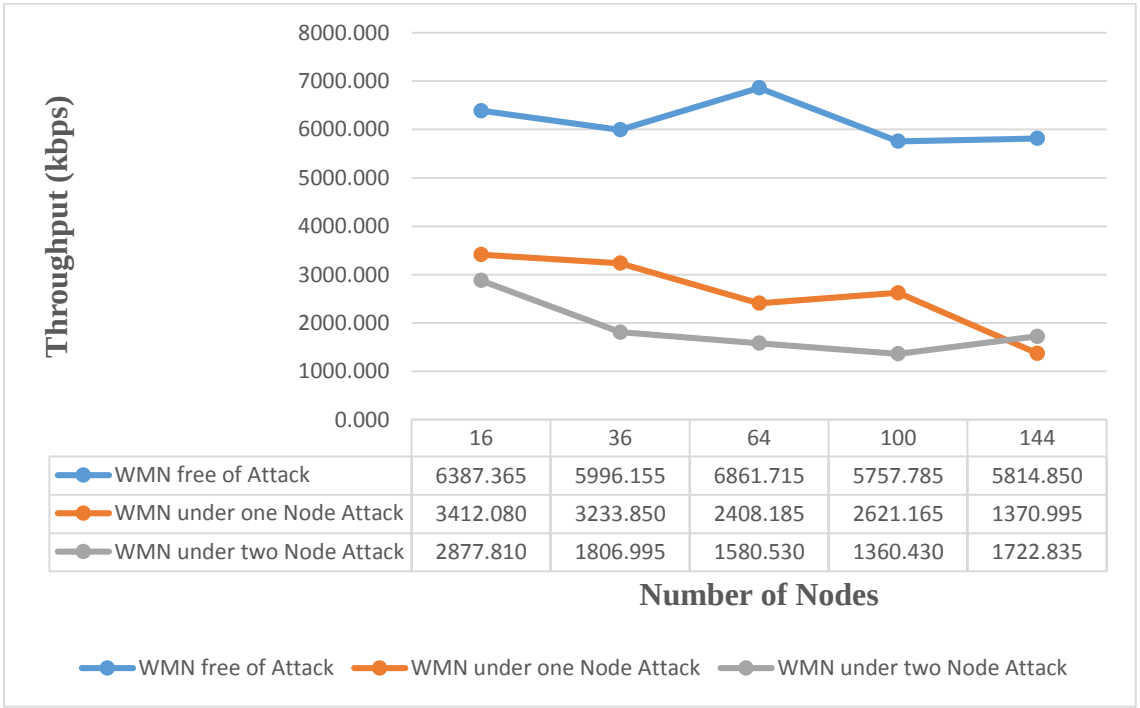


Figure 6. 10: Average throughput at 500 kbps data rate

6.4.3 Average End-to-End Delay

This section explains the results of the average end-to-end delay measured from the simulation scenarios for both WMN free of malicious black hole attacks and WMN infrastructure under malicious black hole attack. The results of the average end-to-end delay computation were achieved adopting the functions defined in Subsection 6.3.3. The average end-to-end delay of the overall packet sent and successfully received at the intended destination nodes were computed to explore the average end-to-end delay that can be achieved by the WMN infrastructure grids evaluated in the absence and presence of malicious packet drop attack of the black hole nodes. The results of the computed average end-to-end delay are represented in Figures 6.11, 6.12, 6.13, 6.14 and 6.15.

In a broader domain of communication/computer networks, the delay is usually inversely proportional to the throughput. That is, the lower the delay, the higher the throughput. Also, the higher the delay, the lower the throughput. In addition, a lower delay indicates a good performance of the network. While the efficiency of AODV routing in terms of delay compared to other WMN routing protocols could not be argued in this study, studies such as (Morote, 2011; Zakaria *et al.*, 2013) maintained that AODV is less efficient in terms of end-to-end delay compared to HWMP and OLSR. However, in the context of the simulation scenarios in this study, it can be said that the results of the delay and throughput also display the common inverse proportionality, especially in the networks without attacks. When the delay is lower, the throughput is higher, when delay is higher, the throughput is lower. The relationship between the delay and throughput could also help to expound the unstable trends in the graph representation of the delay results in relation to the throughput results.

Furthermore, it can be said that the 16 node mesh grid experienced the highest delay in all cases, especially, under the non-malicious conditions. In real sense, the delay experienced in this mesh grid could be interpreted that the network was largely loaded with the implemented CBR traffic. In other words, from the perspective that 16 CBR connection flows were established between randomly generated sources and destinations in this network grid as other network grids, it can be said that the network size was loaded at a very high capacity compared to the other mesh grid sizes simulated. Therefore, the collision rate in this network would be higher and thus, result in

higher delay of the received packets since congestion might not mean that all the packets trapped in traffic congestion would be lost in collision. Packets jammed in congestions are likely to be queued up and later get delivered.

Again, while it was said that the 16 mesh grid maintains the highest delay in all non-malicious cases, its throughput on the other hand turned out to be greater than some of the other network grids with higher PDR and lower delay in some cases. For instance, the throughput of the 16 node mesh grid is higher than the throughput of the 36 and 144 node mesh grids as represented in the 500 kbps data rate throughput table in Figure 6.10. Its PDR is approximately 10% and 3% lesser than the PDR of these mesh grids respectively while they also maintain lower delay compared to the 16 node mesh grid. The higher throughput might seem anomalous but it could rather be explained that the throughput of the 16 node mesh grid was compensated with the throughput of the connection flows with better throughput based on the throughput computation functions in Subsection 6.3.3. Hence, higher throughput of these few connections would result in higher throughput while the higher delay in the congested flows would generally affect the overall delay.

Though the computation of the evaluation metrics was based on the average of two independent simulations as discussed in Subsection 6.2.3, one of the independent simulation experiments of the 16 node mesh grid under 500 kbps data rate was explored to give a clear understanding of this discussion. The PDR, throughput and delay of each connection flow in the explored simulation are represented in Figures 6.16, 6.17 and 6.18 respectively. In Figure 6.16, it could be observed that flows like flows 2, 5, 8 and 12 achieved 76.4%, 74.9%, 40.3%, and 18% PDR with corresponding throughput of 383 kbps, 376 kbps, 202 kbps and 116 kbps respectively while they maintain higher delay especially, flow 12 with the highest delay. On the other hand, throughput of flows such as flows 1, 13, 14, 15 and 16 were able to compensate for the entire throughput of this scenario despite the higher end-to-end delay of flows 2, 5, 8, and 12. Hence, it can be substantiated that despite the highest delay influenced by the smaller size of the16 node mesh grid and data rate, the throughput would be comparable or even higher than some of the mesh grids with lower delay and optimal PDR.

Moreover, in the networks without attacks, the 100 node grid also presents a higher delay with the corresponding PDR and throughput. While it might be argued that on the contrary the

network size is large enough compared to the 16 node mesh grid that was explained to be affected by its smaller size under a very high traffic load, it could be otherwise be explained that the delivery of packets was delayed by the number of hops the data packet traversed to reach the intended destination nodes. This could be related to the discussion of the effect of the number of hops on data packet transmission in WMNs in (Sibeko *et al.*, 2016). Though the study adopted OLSR as the routing protocol in the context of a Gateway-based traffic scenario in WMNs, it was explained that each mesh hop adds to the delay of the packet forwarded through it. Thus, the higher the number of hops the data packets traverse, the higher the delay the data packet could experience before reaching the destination.

On another note, it can be observed that the delay values under one and two black hole attacks are inversely proportional to their corresponding throughput values. When the delay is lower, the throughput is higher, when delay is higher, the throughput is lower. This could also help to contextualize the unstable trend in the graph representation of the delay results relative to the throughput results. However, it can be further explained that the delay are lower both under one and two black hole attacks compared to the delay of the network grids without attacks in most cases, though there are some exceptions in the delay graph representations most especially, 200 to 500 kbps data rates. The exceptional cases are later explained while the lower delay could still be said not to be favourable to the packet delivery in the networks. The lower end-to-end delay could be attributed to the massive malicious drop of data packets experienced in the attacked WMNs grids. Average end-to-end delay was computed based on the number of successfully delivered data packets from each connection flow. Hence, the average end-to-end delay would be insignificant as larger percentage of the data packet hoping through the black hole node route were maliciously dropped at the black hole node.

That said, the exceptional higher delay experienced under attacked networks (under both one and two black hole attacks) compared to the networks without attack could be explained relative to the discussion in Subsection 6.4.1 that the presence of the malicious nodes also increased the interference in the network. Thus, the eventuality of the higher delay of the data packets that were eventually received at the intended destination nodes in the attacked networks. It is, however, important to highlight that this does not contravene the discussion under the throughput result that few packet were delivered in a short duration while the interference caused by the

presence of black hole node seized the delivery of the remaining packet transmitted in the said connection flow. The case happened to be other way round that the data packets that were eventually delivered were excessively delayed along the interfered routes. This could be as well interpreted as part of the abnormalities that could be caused by the presence of black hole nodes in the network. The summary of the discussion of the simulation experiment is presented in the nest section.

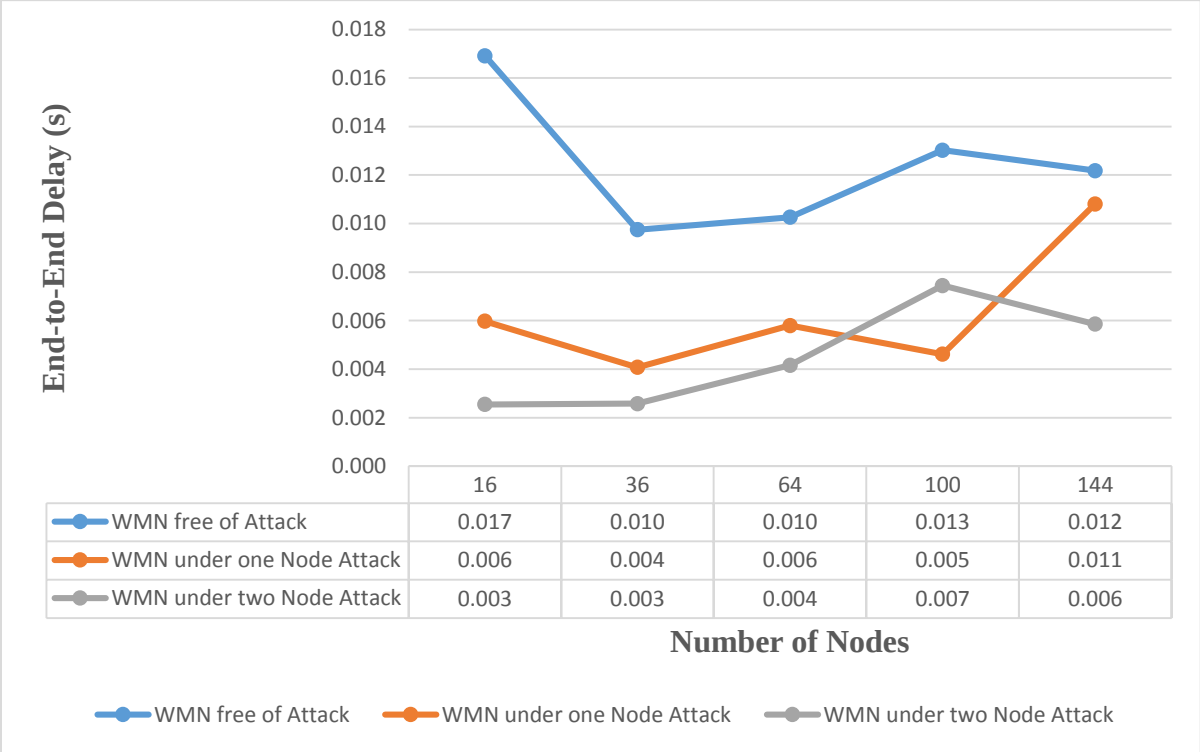


Figure 6. 11: Average end-to-end delay at 100 kbps data rate

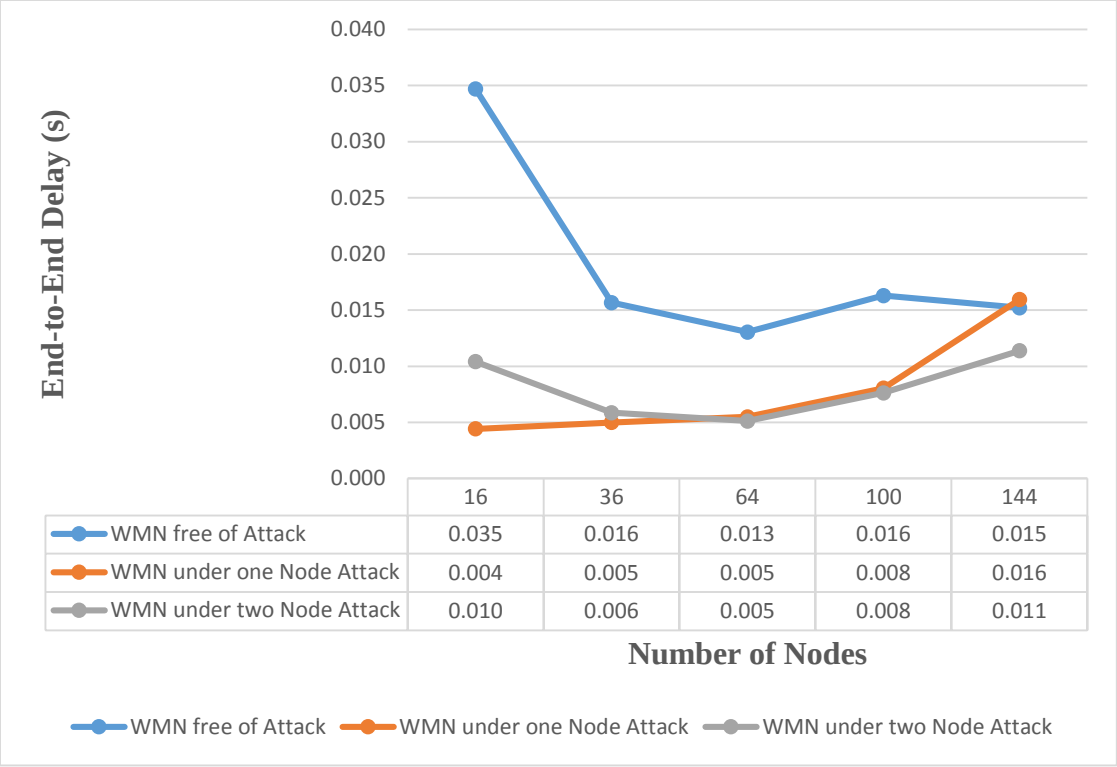


Figure 6. 12: Average end-to-end delay at 200 kbps data rate

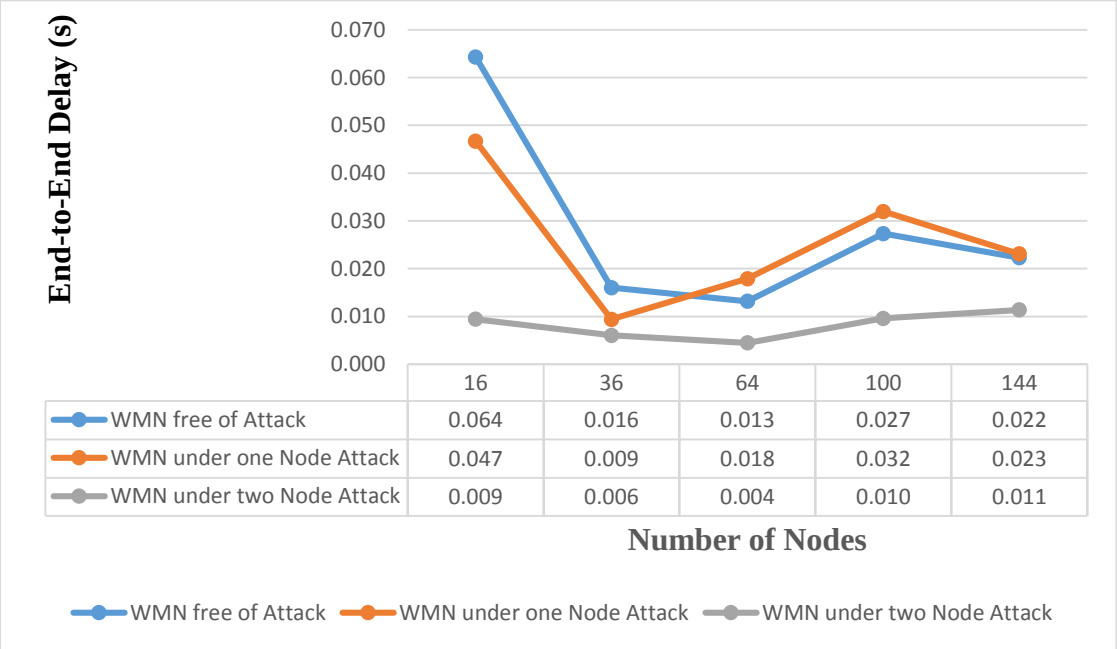


Figure 6. 13: Average end-to-end delay at 300 kbps data rate

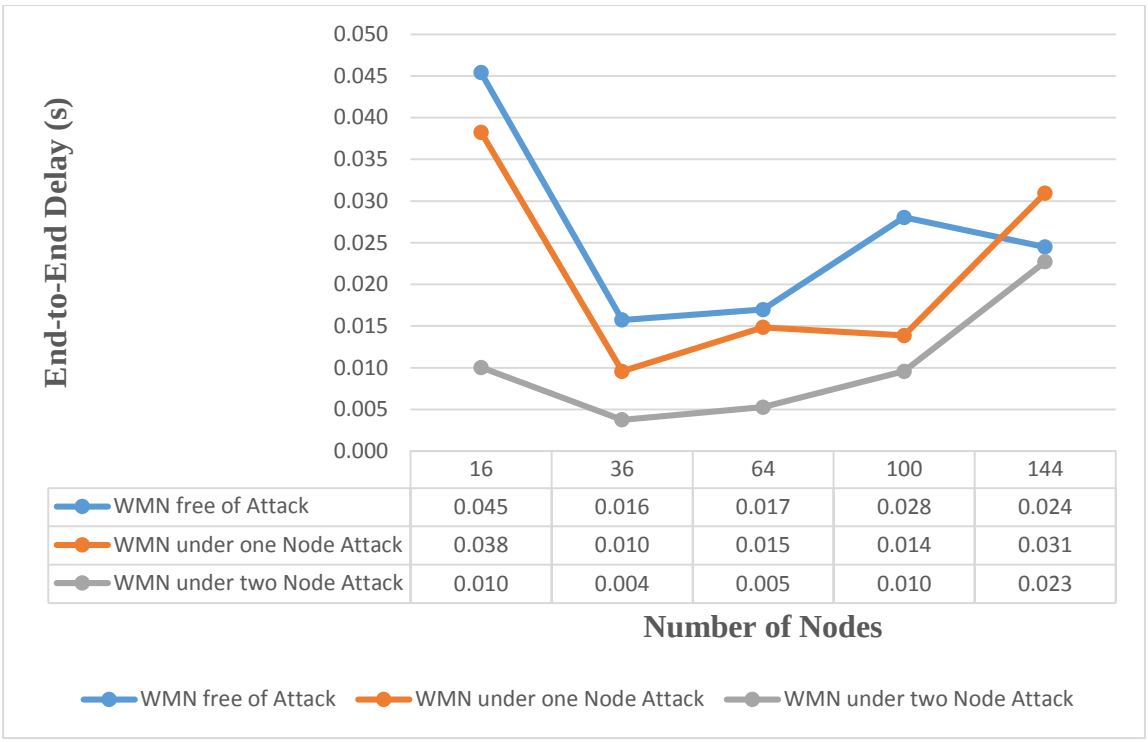


Figure 6. 14: Average end-to-end delay at 400 kbps data rate

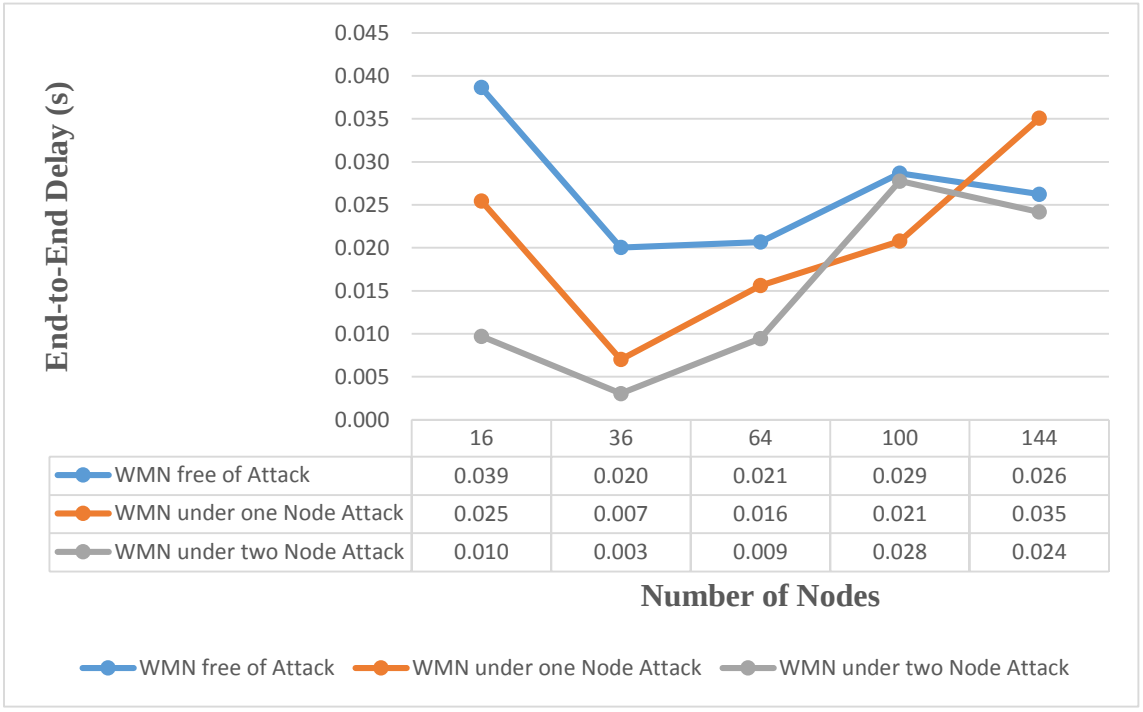


Figure 6. 15: Average end-to-end delay at 500 kbps data rate

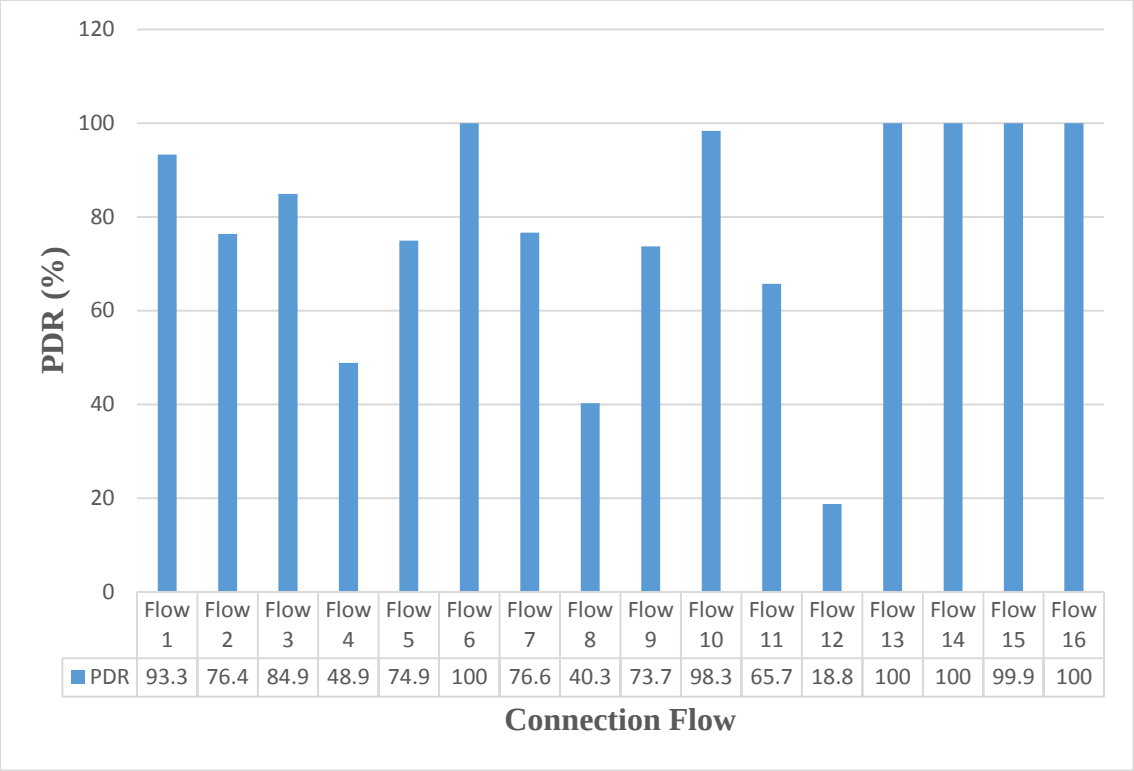


Figure 6. 16: 16 node mesh grid PDR

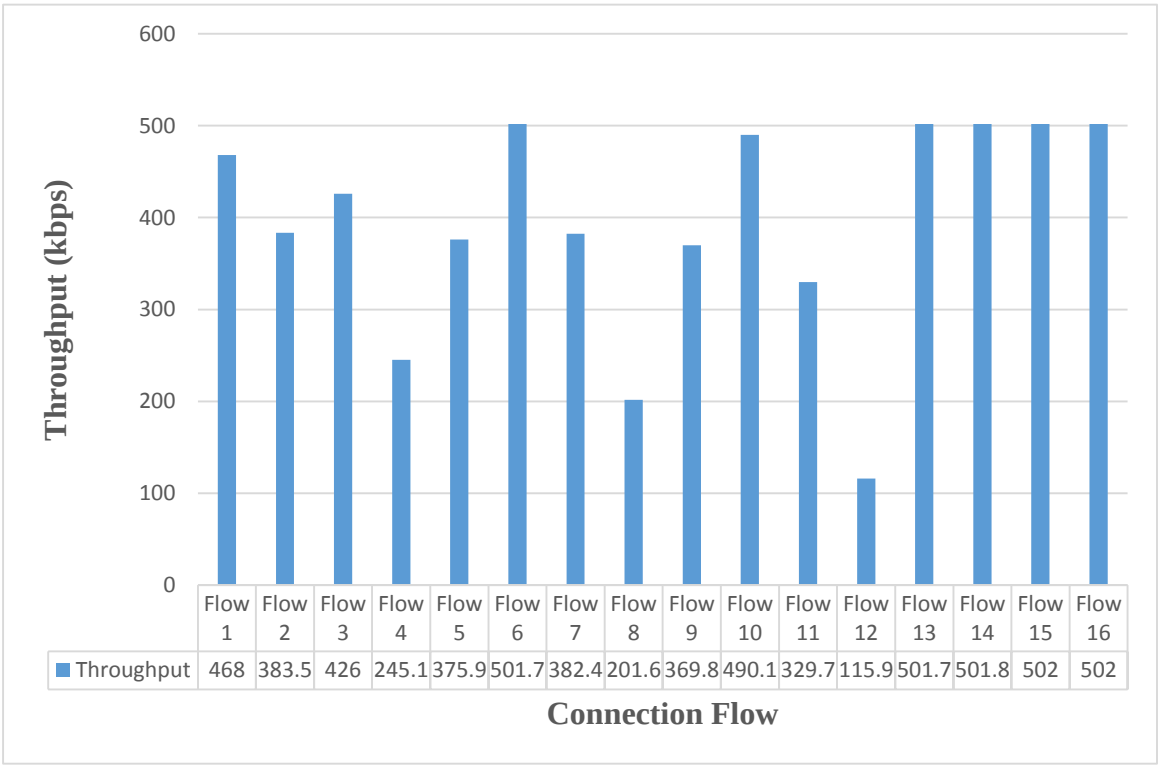


Figure 6. 17: 16 node mesh grid Throughput

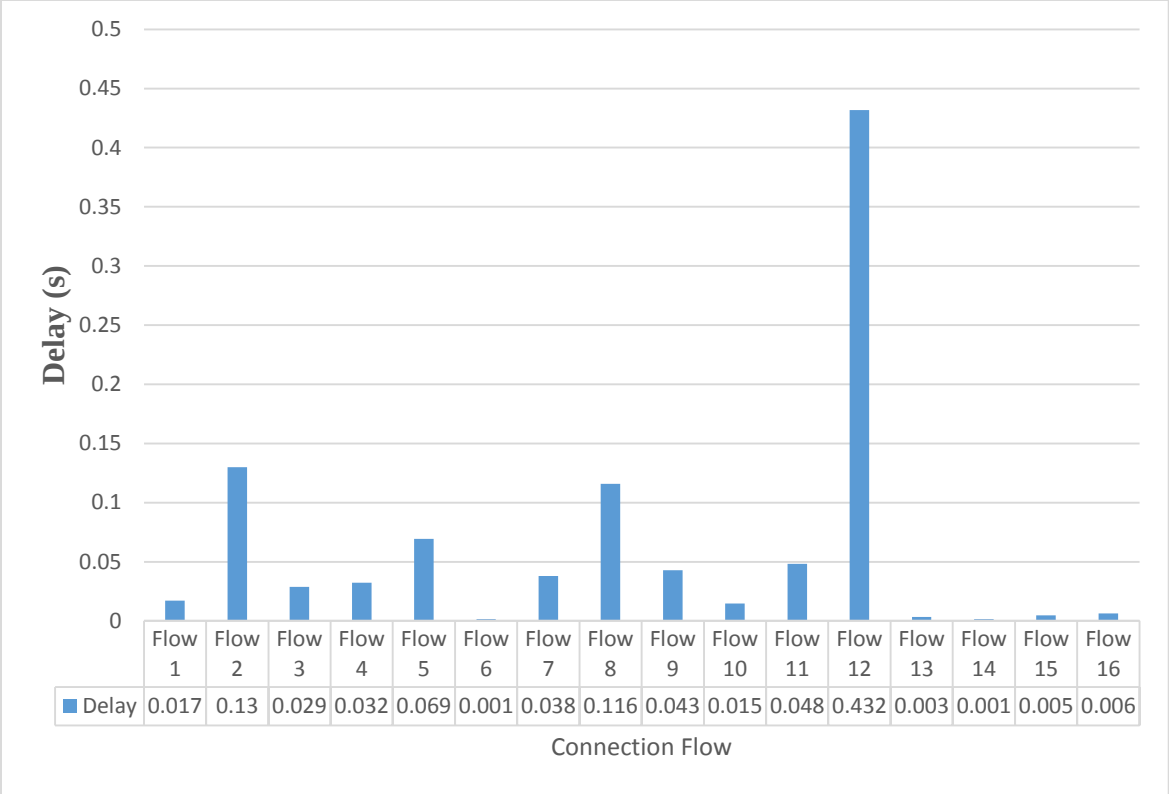


Figure 6. 18: 16 node mesh grid Delay

6.5 Summary

In this chapter, the simulation experiments performed to analyze the impact of black hole attack on packet transmissions in WMNs were presented. The chapter first presented the experimental setup of the simulation scenarios for the WMNs evaluations. The experimental setup discussed included the environment setup, the routing protocol configurations, and the network topologies and simulation scenarios. Secondly, the chapter discussed the method used for the performance evaluations of the WMNs infrastructure simulated scenarios. The measured performance evaluation metrics included PDR, average throughput and average end-to-to end delay. Finally, the chapter presented and discussed the results of the simulation experiments adopting measured performance evaluation metrics. The results of the measured performance evaluation metrics, revealed that the network was largely affected by the attack of malicious packet drop behavior of the black hole nodes. The results showed that the packet delivery ratio and the throughput of the WMN under attack dropped sharply compared to the WMNs that were free of attacks. On the

other hand, the average end-to-end delay of packets hoping through the routes that were free from black hole attack turned out to be higher compared to attacked networks while in some exceptional cases, delay was higher under the attacked networks. In the next chapter, the conclusions from this research and the future direction are presented.

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

The work presented in this study was the analysis of the security and reliability of Packet transmission in Wireless Mesh Networks (WMNs). The work sought to explore comprehensive secure and routing capability against attack such as packet drop attack in WMN and analyse the impact of malicious packet drop attack such as black hole malicious packet drop attack on packet transmission in WMNs. The routing protocol adopted for the route discovery and data packet routing was AODV. In the study, the analysis was done on the performance of the network in the absence and presence of malicious black hole attack in a simulated environment using NS-3 network simulator. This concluding chapter aims to substantiate the research findings in line with the itemized research objectives to answer the posed research questions. The chapter is concluded by presenting the future direction for further research on providing a secure and reliable routing approach in WMNs**.**

7.1 Summary

WMNs are multi-hop wireless network with self-healing and self-adaptation capabilities. The multi-hop packet routing in WMNs is dependent on the collaboration of the mesh participating nodes. Packet transmitted from a source node to a multi-hop destination node is transmitted through the intermediate nodes between the source and destination node. Therefore, it is possible for a malicious or selfish node to exploit this mechanism as a vulnerability to launch malicious packet drop attack. The malicious packet drop attack can lead to the denial of service or isolation of the victim node in the network. Thus, the overall network performance and reliability of the network would be degraded under such malicious attack. Based on this, the work presented in this dissertation was designed to answer the three research questions posed in chapter one of this research dissertation. The followings are the research questions:

1) How can secure and reliable packet routing be established between source and destination nodes in WMNs?

2) How optimal is the performance of the network in the absence of malicious packet drop attack?
3) How impactful is packet drop attack on packet transmission in WMN?

In order to answer the above research questions, the following research objectives were set:

1) To explore comprehensive approach that can provide secure and reliable packet routing capability in WMNs.
2) To analyse the optimal performance of the network in the absence of malicious packet drop attack.
3) To analyse the impact of packet drop attack on packet transmission in WMNs.

In Chapter two, the basic concepts of WMN was discussed. In Section 2.1 of the chapter, it was revealed that WMNs in the recent years have come into play as part of the technologies to provide ubiquitous communication infrastructure, through their self-configured, self-adapted and cost-effective capabilities, to reduce the challenges of digital divide in the developing nations. The chapter further presented the architectures of WMNs which include Client, Infrastructure/backbone and Hybrid WMNs. As explained in the chapter, the clients WMNs comprise of wireless client devices connected peer-to-peer to collaborate to relay data on behalf of each other. In the infrastructure/backbone WMNs, the mesh comprise mesh routers and mesh access points with routing capabilities and mesh gateways that can connect the network with an external network such as Internet. The Hybrid WMNs is the interconnection of both Client WMNs and Infrastructure/backbone WMNs.

Section 2.3 further presented the IEEE 802.11s standard for WMNs. The section discussed the IEEE 802.11s WMNs base architecture which is based on Infrastructure/backbone WMNs. The knowledge of the IEEE 802.11s architecture fostered the building of the topologies used for the experiments performed to achieve the second and third set research objectives. Moreover, the basic IEEE 802.11s protocols such as peering management and HWMP routing protocols WMNs were discussed. The discussion unraveled that the HWMP is a routing protocol that combines the reactive on-demand and proactive tree routing inspired by the common AODV ad hoc routing protocol. The related works, especially the works of (Andreev and Boyko, 2010; Morote, 2011;

IEEE, 2012) reviewed gave a whole insight into the routing mechanisms of IEEE 802.11s HWMP. The choice of using AODV routing protocol for the experimentation was based on the understanding that the HWMP is an extension of AODV routing at the MAC layer.

Furthermore, the security issues in WMNs were discussed in Section 2.4. The section uncovered the vulnerability of routing control messages to alteration at any malicious or selfish intermediate node along the path between the source and destination nodes. It was further highlighted in the section that the most commonly manipulated elements of the routing packets may include the hop count and path metric requested or provided. To have further understanding of the secure routing challenges in WMNs, related works that gave the whole insight were reviewed, especially the works of (Wang and Lim, 2008; Sen, 2011, 2012, 2013; IEEE, 2012; Tan *et al.*, 2013; Peethambaran and Jayasudha, 2014). The aim of reviewing those works was to gather information for the design of the performed experiments. The identified routing attacks that could lead to packet drop attack include Sybil, worm-hole, gray hole and black hole attacks. Among the mentioned packet drop attacks, black hole attack was identified to pose a complete packet drop of data packets deceptively attracted to the adversary. As a result, the type of packet drop attack investigated in this research experiment was black hole malicious packet drop attack. The overview of the basic routing mechanism of AODV routing and the deceptive approach adopted by black hole node in AODV to launch malicious packet drop attack was presented in the Section 2.5 of chapter two.

In Chapter three, the surveys of the previous attempts to improve packet routing in WMNs and existing secure routing approaches in the literature were presented. The review of literatures presented in this chapter helped to address the first objective set in this research. In the survey, it was discovered that more efforts are needed to fortify the existing secure routing methods in the literature. The brief of the research finding through the first objective is presented in section 7.2.1.

In chapter four, the research methodology that guided this research was presented. In Section 4.3, the step-by-step process to achieve the set research objectives was discussed. In section 4.3, possible research design techniques that can be adopted for the performance analysis of communication networks such as WMNs were discussed. The section further dwelled largely on

the simulation research approach primarily adopted for the performance analysis performed to achieve the second and third objectives of this research. Section 4.5 presented the research method which largely described the steps followed specifically for simulation experiments in this research. Moreover, the network simulator (NS-3) adopted for the simulation experiments in this research was largely discussed in Section 4.6. The choice of choosing NS-3 was based on the recommendation made in (Owczarek and Zwierzykowski, 2014).

In Chapter five, the implementation of the black hole attack in NS-3 AODV routing protocol adopting the patch provided in (Satre and Tahiliani, 2014) was discussed. The implementation of the of the black hole behavior helped to address the third objective of the research. Simple simulation experiment to test the implemented black hole attack in NS-3 was presented in the chapter. The simulation experiment was done using a simple wireless ad hoc network set up. Based on the results of the simple simulation experiment, it was confirmed that the black hole malicious behavior was successfully added to the NS-3 AODV routing protocol.

Therefore, in Chapter six, the simulation experiments performed to achieve the second and third itemized objectives were discussed. The discussion in this chapter basically presented the evaluation of the performance of the networks while they were under malicious packet drop attack and free from malicious packet drop attack. The simulation experiments were done over a range of increasing network size and traffic load for both WMN free of malicious black hole attack and the other under malicious black hole attack. The performance evaluation metrics accounted for, in the simulation experiments scenarios include; packet delivery ratio, average throughput and average end-to-end delay. The brief of the research findings with regards to the second and third research objectives aimed at answering the second and third research questions are discussed in subsection 7.2.2.

7.2 Research Findings versus Research Objectives

The research findings with regards to the research objectives drawn to answer the research question are discussed in this section.

7.2.1 Objective One

The first objective which was set to explore comprehensive approach that can provide secure and reliable packet routing capability in WMNs was presented in Chapter three. Based on the literature review of related works on the existing secure routing approach against malicious or selfish packet drop attack in WMNs ad hoc routing protocols, it was gathered that the malicious packet drop attack can be mitigated by establishing a secure route between the source node and destination nodes using cryptography-based approach that depends on a dedicated centralized key distribution centre in the network. Moreover, most of the existing secure routing mechanisms are based on trust reputation ratings of individual mesh node in the network. In such secure routing methods, mesh nodes are rated to be trustworthy based on their packet forwarding history.

7.2.2 Objectives Two and Three

The second and third objectives of the work presented in this study to analyze the optimal performance of the network in the absence of malicious packet drop attack, and analyze the impact of packet drop attack on packet transmission in WMNs, were presented in Chapter six of this dissertation. The results revealed that the packet delivery ratio of the WMN under attack dropped sharply compared to the WMNs that were free of attack. Also, the throughput was also affected as the average throughput of the attacked infrastructure/backbone WMNs were considerably depleted compared to the WMNs that was free of attack. On the other hand; the average end-to-end delay of packets through a route that was free from black hole attack turned to be higher while in some exceptional cases, delay was higher under the attacked networks. Based on the results of the performance analysis conducted through the simulation experiments, it can be argued that the multi-hop routing and self-healing nature of WMNs would be largely disrupted in the presence of malicious packet drop attack such as black hole packet drop attack investigated in this study. Hence, the performance of the network in providing reliable and secure packet transmission would be significantly degraded in the event of such a malicious behaviour of WMNs participating nodes. Therefore, it can be concluded that it is imperative to consider the security aspect of the packet routing/transmission in WMNs as against the common

assumption that all the WMNs participating nodes would be cooperative in supporting the multi-hop routing and self-healing in the WMNs environment.

7.3 Recommendations

Further findings in this research have revealed that there are various aspects that still need to be considered to provide secure and reliable routing mechanisms in WMNs. The existing secure routing in the literature have their own limitations in mitigating the effect of malicious packet drop attack in WMNs. WMNs are distributed in nature, connection between key distribution centre and the participating mesh nodes are not always guaranteed due to transient nature of wireless links between nodes in WMNs. Security approach considering the ad hoc nature of connections between participating nodes in WMNs would provide more secure and reliable routes for packet transmission in WMNs.

In addition, the findings showed that most of the security approaches based on trust reputation of the neighbour nodes did not consider some of the underlying non-malicious factors such as link quality, packet drop due to congestion in rating the trust reputation of a participating node. Thus, high false positive detection rate is imminent in such trust reputation based secure routing against malicious packet drop attack. Considering the integration of the existing secure routing approach with the works that have conventionally attempted to improve the routing capability of WMNs without considering its security as part of their improvements would provide more accurate trust evaluation of participating mesh nodes in WMNs. The conclusion from this research is presented in the next section.

7.4 Conclusion

WMNs in the recent years have surfaced as part of the technologies to reduce the challenges of digital divide in the developing nations as they are able provide ubiquitous communication infrastructure, through their self-configured, self-adapted and cost-effective capabilities. The impact of black hole malicious packet drop attack in the context of infrastructure/backbone WMNs was investigated in this research. The analyses showed that the impact of the attack was massive on the performance with respect to the measured performance metrics such as the PDR

and throughput of the evaluated network scenarios hence, secure routing is imperative to deploy a secure and reliable WMNs which are still able to provide a secure, robust and cost-effective communication service in the event of such malicious attack. Various works have been done to provide secure routing in WMNs, the findings in this research further unearthed that further research works are required to improve the existing secure routing approaches in order to provide more secure and reliable services in WMNs, in the event of DoS attacks such black hole malicious pack drop attack. The future direction is discussed in the next section.

7.5 Future Work

It is important to highlight that in this study, the analysis of the impact of malicious packet drop attack only focused on black hole attacks. The future work would include the analysis of the collective effect of black hole, gray hole and worm-hole attacks on WMNs using larger network size. The limitations of the existing secure routing methods in WMNs are future directions for more advanced research in providing secure and reliable packet transmission routes in WMNs. Therefore, the future work would also include a robust implementation of an enhanced secure and reliable routing protocol that is capable of mitigating the collective activities of the aforementioned attacks that could lead to malicious packet drop attacks in WMNs.

REFERENCES

Akyildiz, I. F. (2009) *Wireless Mesh Networks*.

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

Al-Holou, N., Booth, K. K. and Yaprak, E. (2000) 'Using computer network simulation tools as supplements to computernetwork curriculum', *30th Annual Frontiers in Education Conference. Building on A Century of Progress in Engineering Education. Conference Proceedings (IEEE Cat. No.00CH37135)*, 2(February 2000), pp. 13–16. doi: 10.1109/FIE.2000.896644.

Alanazi, S., Saleem, K., Al-Muhtadi, J. and Derhab, A. (2016) 'Analysis of Denial of Service Impact on Data Routing in Mobile eHealth Wireless Mesh Network', *Mobile Information Systems*, 2016, pp. 1–19. doi: 10.1155/2016/4853924.

Alotaibi, E. and Mukherjee, B. (2012) 'A survey on routing algorithms for wireless Ad-Hoc and mesh networks', *Computer Networks*. Elsevier B.V., 56(2), pp. 940–965. doi: 10.1016/j.comnet.2011.10.011.

Anagnostopoulos, D. and Nikolaidou, M. (2001) 'AN OBJECT-ORIENTED MODELLING APPROACH FOR DYNAMIC COMPUTER NETWORK SIMULATION', *International Journal of Modeling and Simulation*, 21(4), pp. 249–257.

Andreev, K. and Boyko, P. (2010) 'IEEE 802.11s Mesh Networking NS-3 Model', *NS-3 Workshop*, p. 8. doi: 10.1109/JPROC.2007.909930.

Ayash, M. (2014) 'Research Methodologies in Computer Science and Information Systems', *Computer Science*, pp. 1–4.

Balci, O. (2001) 'A methodology for certification of modeling and simulation applications', *ACM Transactions on Modeling and Computer Simulation*, 11(4), pp. 352–377. doi: 10.1145/508366.508369.

Banerji, S. and Singha Chowdhury, R. (2013) 'On IEEE 802.11: Wireless Lan Technology', *International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol. 3, No.4, August 2013*, 3(4), p. 64. doi: 10.5121/ijmnct.2013.3405.

Benyamina, D., Hafid, A. and Gendreau, M. (2012) 'Wireless mesh networks design - A survey', *IEEE Communications Surveys and Tutorials*, 14(2), pp. 299–310. doi: 10.1109/SURV.2011.042711.00007.

Borgo, M., Zanella, A., Bisaglia, P. and Merlin, S. (2004) 'Analysis of the hidden terminal effect in multi-rate IEEE 802 . 11b networks', *Simulation*, pp. 12–15.

Buabeng-Andoh, C. (2012) 'Factors influencing teachers ' adoption and integration of information and communication technology into teaching: A review of the literature', *International Journal of Education and Development using Information and Communication Technology*, 8(1), pp. 136–155.

Camp, J. D. and Knightly, E. W. (2008) 'The IEEE 802.11s extended service set mesh networking standard', *IEEE Communications Magazine*, 46(8), pp. 120–126. doi: 10.1109/MCOM.2008.4597114.

Carneiro, G., Fortuna, P. and Ricardo, M. (2009) 'FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3)', *Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools*, 3(September 2015), p. 10. doi: 10.4108/ICST.VALUETOOLS2009.7493.

Cisco Networking Academy (2007) *CCNA Discovery 4.0 Networking for Homes and Small Busineses*. Cisco Systems, Inc.

Clausen, T. and Jacquet, P. (2003) 'RFC 3626 - Optimized Link State Routing Protocol …OLSR—'.

Contributed Code - Nsnam (2016). Available at: https://www.nsnam.org/wiki/Contributed_Code (Accessed: 14 June 2016).

Draves, R., Padhye, J. and Zill, B. (2004) 'Routing in multi-radio, multi-hop wireless mesh networks', in *Proceedings of the 10th annual international conference on Mobile computing and networking - MobiCom '04*, pp. 114–128. doi: 10.1145/1023720.1023732.

Edemacu, K., Euku, M. and Ssekibuule, R. (2014) 'Packet Drop Attack Detection Techniques in Wireless Ad hoc Networks: A Review', 6(5), pp. 75–86. doi: 10.5121/ijnsa.2014.6506.

Egea-Lopez, E. and Vales-Alonso, J. (2005) 'Simulation tools for wireless sensor networks', in *Summer Simulation Multiconference - SPECTS 2005*, pp. 2–9. doi: 10.1109/WAINA.2011.59.

Ghaleb, M., Felemban, E., Subramaniam, S., Sheikh, A. A. and Qaisar, S. Bin (2017) 'A Performance Simulation Tool for the Analysis of Data Gathering in Both Terrestrial and Underwater Sensor Networks', *IEEE Access*, 5(March), pp. 4190–4208. doi: 10.1109/ACCESS.2017.2684539.

Gore, R., Reynolds Jr., P. F., Kamensky, D., Diallo, S. and Padilla, J. (2015) 'Statistical Debugging for Simulations', *ACM Transactions on Modeling and Computer Simulation*, 25(3), pp. 1–26. doi: 10.1145/2699722.

Hallani, H. and Shahrestani, S. A. (2008) 'Enhancing the Reliability of Ad-hoc Networks through Fuzzy Trust Evaluation', in *International Conference on APPLIED COMPUTER SCIENCE (ACS'08)*, pp. 93–98.

Hallani, H. and Shahrestani, S. A. (2008) 'Fuzzy Trust Approach for Wireless Ad-hoc Networks', *Communications of the IBIM*, 1, pp. 212–218.

Hallani, H. and Shahrestani, S. A. (2009) 'Mitigation of the effects of selfish and malicious nodes in Ad-hoc networks', *WSEAS Transactions on Computers*, 8(2), pp. 205–221.

Hillston, J. (2001) 'Modelling and Simulation'. Edinburgh.

Hofstee, E. (2011) *Constructing Good Dissertation*. Sandton: EPE.

Houaidia, C., Idoudi, H., Bossche, A. Van Den, Val, T., Saidane, L. A., Nationale, E. and Informatique, D. (2013) 'Impact of IEEE 802 . 11 PHY / MAC Strategies on Routing Performance in Wireless Mesh Networks', in *IEEE International Symposium on Frontiers of Information Systems and Network Applications - WAINA, Barcelona, Spain.*, pp. 803–808. doi: 10.1109/WAINA.2013.2.

Houaidia, C., Idoudi, H., Van, A., Bossche, D., Val, T. and Saidane, L. A. (2012) 'Towards an Optimized Traffic-Aware Routing in Wireless Mesh Networks', *International Journal of Space-Based and Situated Computing*, x(x).

IEEE (2012) *IEEE Standard for Information technology--Telecommunications and information*

exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*. doi: 10.1109/IEEESTD.2012.6178212.

IEEE Computer Society (2007) *IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements*.

Islam, M. S., Ashikur, R., Rifat Ahsan, S. M., Hassan, N. and Rahman, A. (2010) 'Exploiting packet distribution for tuning RTS Threshold in IEEE 802.11'. doi: 10.1109/BSC.2010.5472955.

Jiang, T. and Baras, J. J. S. (2006) 'Trust evaluation in anarchy: A case study on autonomous networks', *Proceedings - IEEE INFOCOM*, pp. 1–12.

Johnson, D. B., Maltz, D. A. and Josh, B. (1996) 'Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks', in Imielinski, T. and Hank, K. (eds) *Mobile Computing*. Kluwer Academic Publishers, pp. 153–181. doi: 10.1007/BF01193336.

Kayarkar, H. (2012) 'A Survey on Security Issues in Ad Hoc Routing Protocols and their Mitigation Techniques', *International Journal of Advanced Networking & Applications*, 3(5), pp. 1–14.

Koksal, M. M. (2008) 'A Survey of Network Simulators Supporting Wireless Networks', p. 11.

Kolade, A. T., Zuhairi, M. F., Dao, H. and Khan, S. (2016) 'Bait Request Algorithm to Mitigate Black Hole Attacks in Mobile Ad Hoc Networks', *IJCSNS International Journal of Computer Science and Network Security*, 16(5).

Kolade, A. T., Zuhairi, M. F., Yafi, E. and Zheng, C. L. (2017) 'Performance analysis of black hole attack in MANET', in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication - IMCOM '17*, pp. 1–7. doi: 10.1145/3022227.3022228.

Kothari, C. R. (2004) *Research Methodology: Methods & Techniques, New Age International (P) Ltd*. doi: 10.1017/CBO9781107415324.004.

Kulgachev, V., Student, M. S., Drive, N., Heights, H., Jasani, H. and Ph, D. (2010) '802 . 11 Networks Performance Evaluation Using OPNET', pp. 149–152.

Kumar, S., Singh, D., Assistant, R., Chandra, S. and Professor, D. (2015) 'Analysis and Implementation of AODV Routing Protocol against Black Hole Attack in MANET', *International Journal of Computer Applications*, 124(1), pp. 975–8887.

Li, H., Cheng, Y. and Zhou, C. (2008) 'Multi-hop effective bandwidth based routing in multi-radio wireless mesh networks', in *GLOBECOM - IEEE Global Telecommunications Conference*, pp. 5292–5296. doi: 10.1109/GLOCOM.2008.ECP.1011.

Lundgren, H., Nordströ, E. and Tschudin, C. (2002) 'Coping with communication gray zones in IEEE 802.11b based ad hoc networks', *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia - WOWMOM '02*, pp. 49–55. doi: 10.1145/570796.570799.

Marks, R. B. (2002) 'ADVANCES IN WIRELESS NETWORKING STANDARDS', *Pacific Telecommunications Review*, 4, pp. 28–35.

Marti, S., Giuli, T. J., Lai, K. and Baker, M. (2000) 'Mitigating routing misbehavior in mobile ad hoc networks', in *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pp. 255–265. doi: 10.1145/345910.345955.

Mbougni, M. and Ekabua, O. O. (2012) 'Towards Improved Multi-Hop Effective Bandwidth Routing Metric in Wireless Mesh Networks', in *SATNAC 2012*, pp. 1–6.

Morote, M. E. (2011) *IEEE 802 . 11s Mesh Networking Evaluation under NS-3*.

Ndlela, N. Z. ., Mudali, P., Mutanga, M. B. and Adigun, M. O. (2013) 'An Evaluation of the Mesh Access Points Placement Schemes for Rural Wireless Mesh Networks', in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2013*, pp. 325–330.

Nie, J., Wen, J., Luo, J., He, X. and Zhou, Z. (2006) 'An adaptive fuzzy logic based secure routing protocol in mobile ad hoc networks', *Fuzzy Sets and Systems*, 157(12), pp. 1704–1712. doi: 10.1016/j.fss.2005.12.007.

ns-3: ns3::OnOffApplication Class Reference (2017). Available at: https://www.nsnam.org/doxygen/classns3_1_1_on_off_application.html#details (Accessed: 12

November 2017).

ns-3.25 « ns-3 (2016). Available at: https://www.nsnam.org/ns-3-25/ (Accessed: 6 April 2016).

'ns-3 Manual' (2017).

'ns-3 Model Library' (2016) *ns-3 project*, p. 151. doi: 10.1145/570758.570772.

'ns-3 Tutorial' (2016) *ns-3 project*.

Nutaro, J. J. and Zeigler, B. P. (2015) 'Towards A Probabilistic Interpretation of Validity for Simulation Models', in *DEVS '15 Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, pp. 197–204.

Oki, O. ., Mudali, P., Oladosu, J. B. and Adigun, M. O. (2013) 'Comparison of routing protocols performance using wireless mesh network simulation and testbed', *2013 International Conference on Adaptive Science and Technology*, (November 2013), pp. 1–7. doi: 10.1109/ICASTech.2013.6707507.

Oki, O. A. (2013) 'Evaluating the Effect of Quality of Service Mechanisms in Power-Constrained Wireless Mesh Networks'.

Oki, O., Mudali, P., Zulu, N. and Adigun, M. (2014) 'Comparison of Energy-based Leader Selection Algorithms in Wireless Mesh Networks', in *Southern Africa Telecommunication Newtworks and Applications Conference (SATNAC) 2014*, pp. 247–252.

Owczarek, P. and Zwierzykowski, P. (2014) 'Review of simulators for wireless mesh networks', *Journal of Telecommunications and Information Technology*, 2014(3), pp. 82–89.

Peethambaran, P. and Jayasudha, J. S. (2014) 'Survey of Manet Misbehaviour Detection Approaches', 6(3), pp. 19–29. doi: 10.5121/ijnsa.2014.6302.

Perkins, C., Belding-Royer, E. and Das, S. (2003) 'Ad hoc On-Demand Distance Vector (AODV) Routing', *Network Working Group Request for Comments: 3561*. doi: 10.1074/jbc.R109.041087.

Perros, H. (2009) *Computer Simulation Techniques : The de nitive introduction !*

Pirzada, A. and Portmann, M. (2007) 'High performance AODV routing protocol for hybrid

wireless mesh networks', *Mobile and Ubiquitous Systems:*

Regan, R., Martin, J. and Manickam, L. (2016) 'A Survey on Wireless Mesh Networks and its Security Issues', *International Journal of Security and Its Applications*, 10(3), pp. 405–418. doi: 10.14257/ijsia.2016.10.3.35.

Robinson, J. and Knightly, E. W. (2007) 'A Performance Study of Deployment Factors in Wireless Mesh Networks', in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*. IEEE, pp. 2054–2062. doi: 10.1109/INFCOM.2007.238.

Saadi, Y., Bouchaib, N. and Haqiq, A. (2014) 'CBF evaluation through simulation', in. doi: 10.1109/NGNS.2014.6990250.

SAIRA AZIZ, R. S. & V. D. (2016) 'Packet Dropping Attack Detection Techniques in Manets: a Review', *International Journal of Computer Science and Engineering (IJCSE)*, 5(3), pp. 1–6.

Santhanam, L., Nandiraju, N., Yoo, Y. and Agrawal, D. P. (2006) 'Distributed Self-policing Architecture for Fostering Node Cooperation in Wireless Mesh Networks', in *IFIP TC6/WG6.8*, pp. 147–158. doi: 10.1007/11872153_13.

Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B. N., Shields, C. and Belding-Royer, E. M. (2005) 'Authenticated routing for ad hoc networks', *IEEE Journal on Selected Areas in Communications*, 23(3), pp. 598–610. doi: 10.1109/JSAC.2004.842547.

Sarao, P. and Garg, S. (2014) 'A Secure and Trusted Routing Scheme', *International Journal of Computer Science and Mobile Computing*, 3(3), pp. 488–500.

Satre, S. and Tahiliani, M. P. (2014) *Mohit P. Tahiliani: [ns-3] Blackhole Attack Simulation in ns-3, Mohit P. Tahiliani: [ns-3] Blackhole Attack Simulation in ns-3*. Available at: http://mohittahiliani.blogspot.co.za/2014/12/ns-3-blackhole-attack-simulation-in-ns-3.html (Accessed: 7 April 2017).

Saxena, N., Denko, M. and Banerji, D. (2011) 'A hierarchical architecture for detecting selfish behaviour in community wireless mesh networks', *Computer Communications*. Elsevier B.V., 34(4), pp. 548–555. doi: 10.1016/j.comcom.2010.04.040.

Sbeiti, M., Pojda, J., Wietfeld, C. (2012) 'Performance Evaluation of PASER - an Efficient

Secure Route Discovery Approach for Wireless Mesh Networks', in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications - PIMRC, Sydney, Australia*.

Sbeiti, M. and Wietfeld, C. (2013) 'On the Implementation Code of the Secure Mesh Routing Protocol PASER in OMNeT++: The Big Picture', in *International Workshop on OMNeT++, Cannes, France*.

Sbeiti, M., Goddemeier, N., Behnke, D. and Wietfeld, C. (2016) 'PASER: Secure and Efficient Routing Approach for Airborne Mesh Networks', *IEEE Transactions on Wireless Communications*, 15(3), pp. 1950–1964. doi: 10.1109/TWC.2015.2497257.

Sbeiti, M., Wolff, A., Wietfeld, C. and Technology, I. (2011) 'PASER: Position Aware Secure and Efficient Route Discovery Protocol for Wireless Mesh Networks', in *SECURWARE 2011 : The Fifth International Conference on Emerging Security Information, Systems and Technologies* •, pp. 63–70.

Sen, J. (2011) 'Secure Routing for Wireless Mesh Networks', in *eprint arXiv:1102.1226*, pp. 237–280.

Sen, J. (2012) 'Secure and Privacy-Preserving Authentication Protocols for Wireless Mesh Networks', *Book: 'Applied Cryptography and Network Security'. ed: Jaydip Sen. INTECH Publishers, Croatia.*, pp. 3–34. doi: 10.5772/39176.

Sen, J. (2013) 'Security and Privacy Issues in Wireless Mesh Networks: A Survey', *Book: 'Wireless Networks and Security- Issues, Challenges and Research Trends'. eds: Shafiullah Khan et al. Springer, London.* , pp. 189–272. doi: 10.1007/978-3-642-36169-2_7.

Sibeko, N., Mudali, P., Oki, O., Bethel, M. and Matthew, A. (2016) 'The effect of distance on QOS in a Gateway-based traffic scenario for Wireless Mesh Networks', in *South African Telecommunication Networks and Applications Conference (SATNAC)*.

Strix Systems Inc. (2005) 'Solving the Wireless Mesh Multi-Hop Dilemma', *Network Witout Wires*.

Tan, W. K. ., Lee, S.-G. ., Lam, J. H. . and Yoo, S.-M. . (2013) 'A Security analysis of the 802.11s wireless mesh network routing protocol and its secure routing protocols', *Sensors*

(Switzerland), 13(9), pp. 11553–11585. doi: 10.3390/s130911553.

Taylor, S. J. E., Balci, O., Cai, W., Loper, M. L., Nicol, D. M. and Riley, G. (2013) 'Grand challenges in modeling and simulation: Expanding Our Horizon', in *Proceedings of SIGSIM-PADS'13*. ACM, pp. 409–414. doi: 10.1117/12.474899.

Taylor, S. J. E., Chick, S. E., Macal, C. M., Brailsford, S., L'Ecuyer, P. and Nelson, B. L. (2013) 'Modeling and simulation grand challenges: An OR/MS perspective', in *Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013*, pp. 1269–1282. doi: 10.1109/WSC.2013.6721514.

Tsado, Y., Gamage, K. and Lund, D. (2015) 'Performance Evaluation of Wireless Mesh Network Routing Protocols for Smart Grid AMI Networks'.

Wang, X. and Lim, A. O. (2008) 'IEEE 802.11s wireless mesh networks: Framework and challenges', *Ad Hoc Networks*, 6(6), pp. 970–984. doi: 10.1016/j.adhoc.2007.09.003.

Wang, X., Yuan, W., Linnartz, J.-P. P. M. G. and Niemegeers, I. G. M. M. (2010) 'Experimental validation of a coexistence model of IEEE 802.15.4 and IEEE 802.11b/g networks', *International Journal of Distributed Sensor Networks*. Hindawi Publishing Corporation, 2010. doi: 10.1155/2010/581081.

Yusuf, M. O. O. (2005) 'Information and Communication Technology and Education: Analysing the Nigerian National Policy for Information Technology', *International Education Journal*, 6(3), pp. 316–321.

Zakaria, A., Mohamad, H., Ramli, N. and Ismail, M. (2013) 'Performance Evaluation of Routing Protocols inWireless Mesh Network', in *ICACT2013*, pp. 1111–1115.

Zhang, W., Wang, Z., Das, S. K. and Hassan, M. (2008) 'Security Issues in Wireless Mesh Networks', *Wireless Mesh Networks. Springer US.*, pp. 309–330.

Zhang, Y., Luo, J. and Hu, H. (2007) 'Wireless mesh networking: architectures, protocols and standards', *Security Management*, pp. 8–13.

APPENDICES

Appendix A (*Adapted from (Morote, 2011) as the basis of the performed simulation experiments*)

```
#include "ns3/core-module.h"
#include "ns3/aodv-module.h"
#include "ns3/internet-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/flow-monitor.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/ipv4-flow-classifier.h"
#include "ns3/random-variable-stream.h"
#include <iostream>
#include  <sstream>
#include  <fstream>
#include "ns3/ptr.h"
#include <map>
#include <cmath>
#include <vector>


using namespace ns3;

class MeshTest
{
public:
// Init test
MeshTest ();
// Configure test from command line arguments
void Configure (int argc, char ** argv);
```

```
// Run test
int Run ();
private:
int
m_xSize; //x size of the grid
int
m_ySize; //y size of the grid

int maliciousNodeA;
int maliciousNodeB;
double
m_step; //separation between nodes
double
m_totalTime;
uint16_t m_packetSize;
bool
m_pcap;
std::string m_txrate;
double
m_txrate_dob;

// List of network nodes
NodeContainer nodes;
NodeContainer not_malicious;
NodeContainer malicious;

// List of all wifi devices
NetDeviceContainer wifiDevices;

//Addresses of interfaces:
Ipv4InterfaceContainer interfaces;

WifiHelper wifi_aodv;
```

```
private:
// Create nodes and setup their mobility
void CreateNodes ();
// Install internet m_stack on nodes
void InstallInternetStack ();
// Install applications randomly
void InstallApplicationRandom ();
void Report ();
};

MeshTest::MeshTest () :
m_xSize (4),
m_ySize (4),
maliciousNodeA(1),
maliciousNodeB (14),
m_step (170),
m_totalTime (1000),
m_packetSize (1024),
m_pcap (false),
m_txrate ("500kbps"),
m_txrate_dob (500) //needed in kbps for the trace file
{
}

void
MeshTest::Configure (int argc, char *argv[])
{
CommandLine cmd;
cmd.AddValue ("m_xSize", "m_xSize", m_xSize);
cmd.AddValue ("m_ySize", "m_ySize", m_ySize);
cmd.AddValue ("m_txrate", "m_txrate", m_txrate);
cmd.AddValue ("m_txrate_dob", "m_txrate_dob", m_txrate_dob);
cmd.Parse (argc, argv);
```

```
}
void MeshTest::CreateNodes ()
{

double m_txpower = 18.0; // dbm

// Create the nodes
nodes.Create (m_xSize*m_ySize);
//Define MaliciousNodes
int maliciousNodes[] = {maliciousNodeA, maliciousNodeB};
int nodeSwitch=0;
int blackholeNode=0;

for(int p=0;p<m_xSize*m_ySize;p++)
{
for(int q=0;q<2; q++)
{
   if(p==maliciousNodes[q])
{
            nodeSwitch=1;

blackholeNode=p;
}

}

if(nodeSwitch==1)
{
   malicious.Add(nodes.Get(blackholeNode));
   nodeSwitch=0;
}else{

  not_malicious.Add(nodes.Get(p));
```

```
  nodeSwitch=0;
}

}

// Configure YansWifiChannel
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
wifiPhy.Set("EnergyDetectionThreshold", DoubleValue (-89.0) );
wifiPhy.Set("CcaMode1Threshold", DoubleValue (-62.0) );
wifiPhy.Set("TxGain", DoubleValue (1.0) );
wifiPhy.Set("RxGain", DoubleValue (1.0) );
wifiPhy.Set("TxPowerLevels", UintegerValue (1) );
wifiPhy.Set("TxPowerEnd", DoubleValue (m_txpower) );
wifiPhy.Set("TxPowerStart", DoubleValue (m_txpower) );
wifiPhy.Set("RxNoiseFigure", DoubleValue (7.0) );
YansWifiChannelHelper WifiChannel;
WifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
/*WifiChannel.AddPropagationLoss ("ns3::TwoRayGroundPropagationLossModel",
                                                    "SystemLoss", DoubleValue(1),
                                                    "HeightAboveZ", DoubleValue(1.5));*/
WifiChannel.AddPropagationLoss ("ns3::LogDistancePropagationLossModel","Exponent",StringValue ("2.7"));
wifiPhy.SetChannel (WifiChannel.Create ());
//wifi_aodv.SetStandard (WIFI_PHY_STANDARD_80211b);
wifi_aodv.SetStandard (WIFI_PHY_STANDARD_80211a);
//wifi_aodv.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode", StringValue ("DsssRate1Mbps"), "RtsCtsThreshold", UintegerValue (2500));
wifi_aodv.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode", StringValue ("OfdmRate6Mbps"), "RtsCtsThreshold", UintegerValue (2500));

// Install protocols and return container
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default();
wifiMac.SetType ("ns3::AdhocWifiMac");
wifiDevices = wifi_aodv.Install (wifiPhy, wifiMac, nodes);
```

```
// Arranging mesh nodes in a grid topology
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                               "MinX", DoubleValue (0.0),
                               "MinY", DoubleValue (0.0),
                               "DeltaX", DoubleValue (m_step),
                               "DeltaY", DoubleValue (m_step),
                               "GridWidth", UintegerValue (m_xSize),
                               "LayoutType", StringValue ("RowFirst"));
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodes);
}

void MeshTest::InstallInternetStack ()
{
//configure AODV
AodvHelper aodv;
AodvHelper malicious_aodv;

aodv.Set ("AllowedHelloLoss", UintegerValue (20));
aodv.Set ("HelloInterval", TimeValue (Seconds (3)));
aodv.Set ("RreqRetries", UintegerValue (5));
aodv.Set ("ActiveRouteTimeout", TimeValue (Seconds (100)));
aodv.Set ("DestinationOnly", BooleanValue (true));

malicious_aodv.Set ("AllowedHelloLoss", UintegerValue (20));
malicious_aodv.Set ("HelloInterval", TimeValue (Seconds (3)));
malicious_aodv.Set ("RreqRetries", UintegerValue (5));
malicious_aodv.Set ("ActiveRouteTimeout", TimeValue (Seconds (100)));
malicious_aodv.Set ("DestinationOnly", BooleanValue (true));

//Install the internet protocol stack on all nodes
InternetStackHelper internetStack;
```

```
internetStack.SetRoutingHelper (aodv);
internetStack.Install (not_malicious);

malicious_aodv.Set("IsMalicious",BooleanValue(false)); // putting *false* instead of *true* would disable the malicious behavior of the node
internetStack.SetRoutingHelper (malicious_aodv);
internetStack.Install (malicious);

//Assign IP addresses to the devices interfaces (m_nIfaces)
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
interfaces = address.Assign (wifiDevices);
}

void MeshTest::InstallApplicationRandom ()
{
        // Create connections
        int m_nconn = 16;

        int m_source, m_dest, m_dest_port;
        double start_time, stop_time, duration;

        // Set the parameters of the onoff application
        Config::SetDefault ("ns3::OnOffApplication::PacketSize",UintegerValue (m_packetSize));
        Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue (m_txrate));
        ApplicationContainer apps [m_nconn];


        Ptr<UniformRandomVariable> rand_nodes = CreateObject<UniformRandomVariable> ();
        Ptr<UniformRandomVariable> rand_port = CreateObject<UniformRandomVariable> ();

        // 50 seconds for transitori are left at the beginning.
```

```
Ptr<UniformRandomVariable> a = CreateObject<UniformRandomVariable> ();
for (int i = 0; i < m_nconn; i++){
start_time = a->GetValue (50,m_totalTime-20);

Ptr<ExponentialRandomVariable> b = CreateObject<ExponentialRandomVariable> ();
   b->SetAttribute ("Mean", DoubleValue (100));
duration = b->GetValue()+1;



//If the exponential variable added to the start time gives a value that
//is greater than the maximum permitted, it has to be forcefully stopped at 15 seconds
//for proper computation of the statistics of each flow
if ( (start_time + duration) > (m_totalTime - 15)){
stop_time = m_totalTime-15;
}else{
stop_time = start_time + duration;
}

// Set random variables of the destination (server) and destination port.
m_dest =rand_nodes->GetInteger (0,m_ySize*m_xSize-1) ;
m_dest_port = rand_port->GetInteger (4900,49100);

          //the malicious nodes are exempted as destination nodes
while(m_dest==maliciousNodeA || m_dest== maliciousNodeB){
m_dest =rand_nodes->GetInteger (0,m_ySize*m_xSize-1);
}
// Set random variables of the source (client)

m_source = rand_nodes->GetInteger (0,m_ySize*m_xSize-1);

// Client and server can not be the same node.
// and the malicious nodes are exempted as source nodes
while (m_source == m_dest || m_source==maliciousNodeA || m_source== maliciousNodeB){
```

```
                m_source =rand_nodes->GetInteger (0,m_ySize*m_xSize-1);

                }

        OnOffHelper onoff ("ns3::UdpSocketFactory", Address (InetSocketAddress(interfaces.GetAddress (m_dest), m_dest_port)));
        onoff.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"));
        onoff.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0.0]"));
        apps[i] = onoff.Install (nodes.Get(m_source));
        apps[i].Start (Seconds (start_time));
        apps[i].Stop (Seconds (stop_time));

        // Create a packet sink to receive the packets
        PacketSinkHelper sink ("ns3::UdpSocketFactory",InetSocketAddress(interfaces.GetAddress (m_dest),m_dest_port  ));
        apps[i] = sink.Install (nodes.Get (m_dest));
        apps[i].Start (Seconds (1.0));

                }

}

int MeshTest::Run ()
{
CreateNodes ();
InstallInternetStack ();
InstallApplicationRandom ();

// Install FlowMonitor on all nodes
FlowMonitorHelper flowmon;
Ptr<FlowMonitor> monitor = flowmon.InstallAll();
//m_timeStart=clock();
```

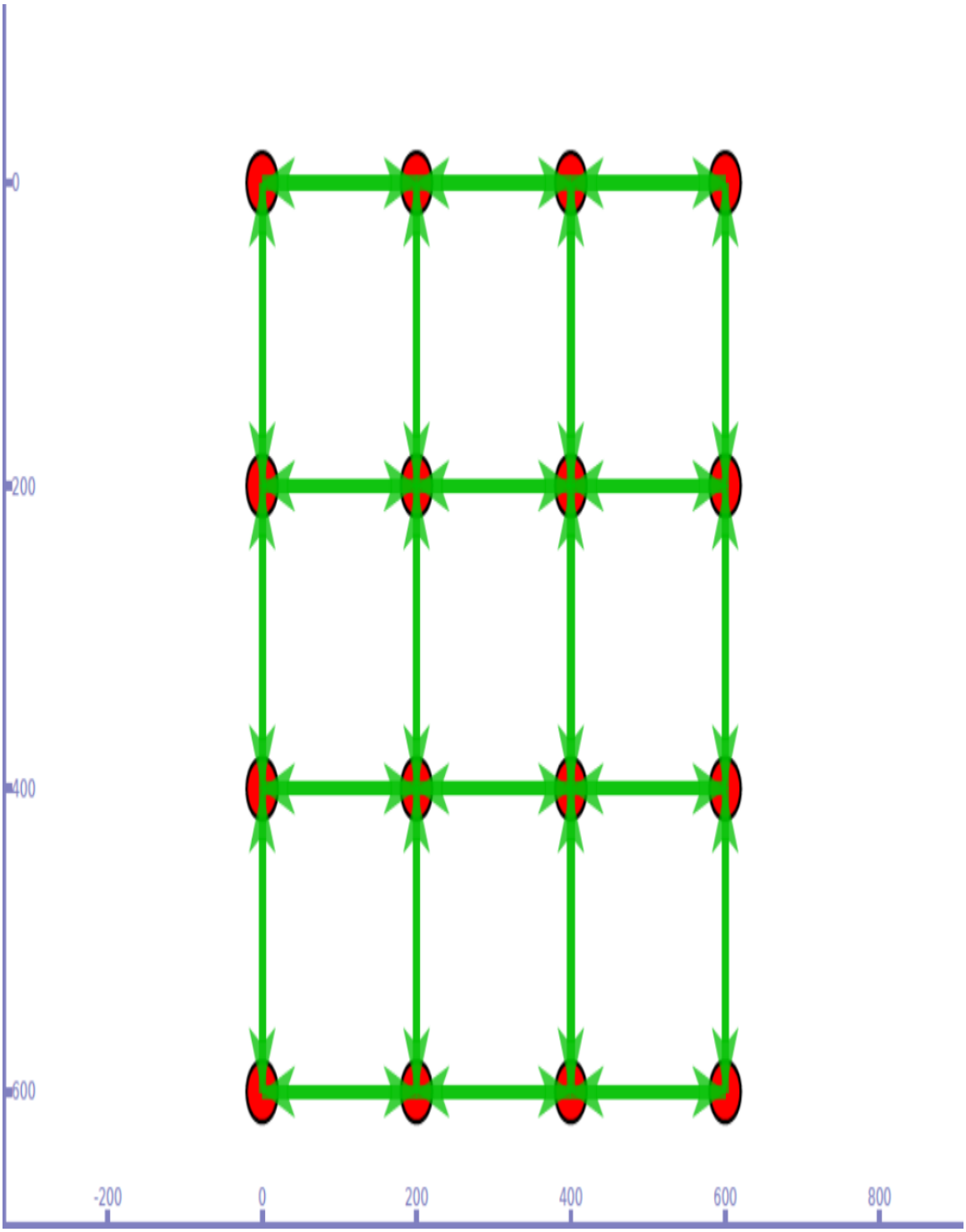
```
Simulator::Stop (Seconds (m_totalTime));
Simulator::Run ();
// Define variables to calculate the metrics
int k=0;
int totaltxPackets = 0;
int totalrxPackets = 0;

double totaldelay = 0;
double totalrxbitrate = 0;

double difftxTime =0;
double diffrx =0;
double pdr_value, rxbitrate_value, txbitrate_value, delay_value;
double pdf_total, rxbitrate_total, delay_total;

//Print per flow statistics
monitor->CheckForLostPackets ();
Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier>(flowmon.GetClassifier ());
std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats ();

for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator i = stats.begin (); i != stats.end (); ++i)
{
Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (i->first);
difftxTime = i->second.timeLastTxPacket.GetSeconds() - i->second.timeFirstTxPacket.GetSeconds();
diffrx = i->second.timeLastRxPacket.GetSeconds() - i->second.timeFirstTxPacket.GetSeconds();
pdr_value = (double) i->second.rxPackets / (double) i->second.txPackets * 100;
txbitrate_value = (double) i->second.txBytes * 8 / 1024 / difftxTime;

if (i->second.rxPackets != 0){
rxbitrate_value = (double) i->second.rxBytes* 8 / 1024 / diffrx;
delay_value = (double) i->second.delaySum.GetSeconds() / (double) i->second.rxPackets;
}
else{
```

```
rxbitrate_value = 0;
delay_value = 0;
}

// We are only interested in the metrics of the data flows
// AODV implementation create other flows with routing information at low bitrates,
// so a margin is defined to ensure only the connections data flows are extracted.

if ( (!t.destinationAddress.IsSubnetDirectedBroadcast("255.255.255.0")) && (txbitrate_value > m_txrate_dob/1.2) && (rxbitrate_value < m_txrate_dob*1.2)&& t.destinationPort!=654 )
{
k++;
std::cout << "\nFlow " << k << " (" << t.sourceAddress << " -> " << t.destinationAddress << ")\n";
std::cout << "Tx Packets:" << i->second.txPackets << "\n";
std::cout << "Rx Packets:" << i->second.rxPackets << "\n";
std::cout << "Lost Packets:" << i->second.lostPackets << "\n";
std::cout << "Dropped Packets:" << i->second.packetsDropped.size() << "\n";
std::cout << "PDR: " << pdr_value << " %\n";
std::cout << "Average delay:" << delay_value << "s\n";
std::cout << "Rx bitrate: " << rxbitrate_value << " kbps\n";
std::cout << "Tx bitrate: " << txbitrate_value << " kbps\n\n";


// Acumulate for average statistics
totaltxPackets += i->second.txPackets;
totalrxPackets += i->second.rxPackets;
totaldelay += i->second.delaySum.GetSeconds();
totalrxbitrate += rxbitrate_value;
}

}

//Average all nodes statistics
if (totaltxPackets != 0){
```

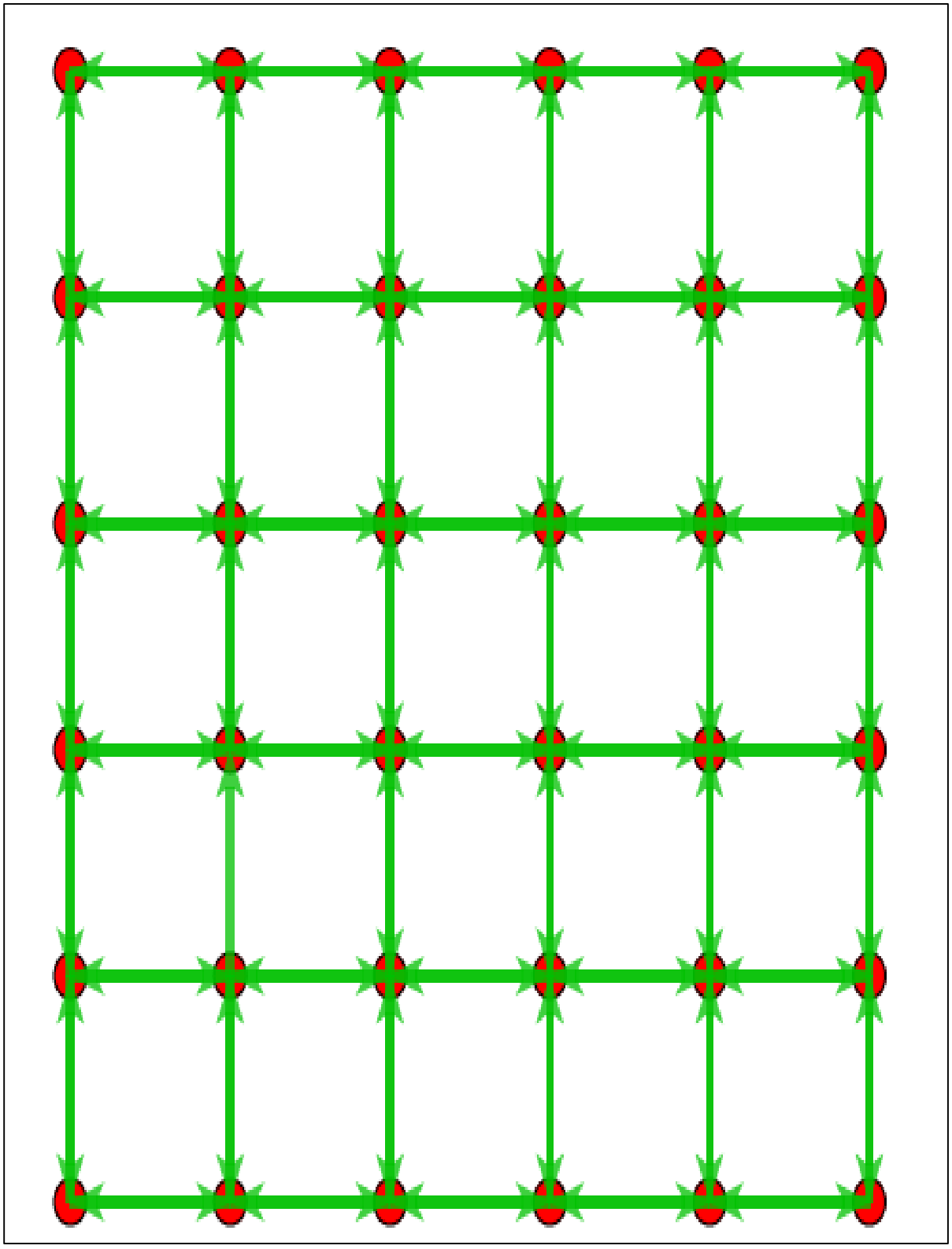
```
pdf_total = (double) totalrxPackets / (double) totaltxPackets * 100;

}
else{
pdf_total = 0;

}
if (totalrxPackets != 0){
rxbitrate_total = totalrxbitrate;

delay_total = (double) totaldelay / (double) totalrxPackets;

}
else{
rxbitrate_total = 0;
delay_total = 0;

}

// Print all nodes statistics

std::cout <<"\nTotal PDF: " << pdf_total << " %\n";
std::cout <<"Total Rx bitrate: " << rxbitrate_total << " kbps\n";
std::cout <<"Total Delay: " << delay_total << " s\n";

// Print all nodes statistics in files
std::ostringstream os;
os << "MeshAttackTest_PDF.txt";
std::ofstream of (os.str().c_str(), std::ios::out | std::ios::app);
of << pdf_total <<"\n";//"@"<<m_txrate<<""<<"kbps\n"
std::ostringstream os2;
os2 << "MeshAttackTest_Delay.txt";
std::ofstream of2 (os2.str().c_str(), std::ios::out | std::ios::app);
```

```
of2 << delay_total << "\n";
std::ostringstream os3;
os3 << "MeshAttackTest_Throu.txt";
std::ofstream of3 (os3.str().c_str(), std::ios::out | std::ios::app);
of3 << rxbitrate_total << "\n";

of.close (); of2.close (); of3.close ();
Simulator::Destroy ();

monitor->SerializeToXmlFile("MeshAttackTest.flowmon", true, true);

return 0;
}


int main (int argc, char *argv[])
{
MeshTest t;
t.Configure (argc, argv);
return t.Run();
}
```

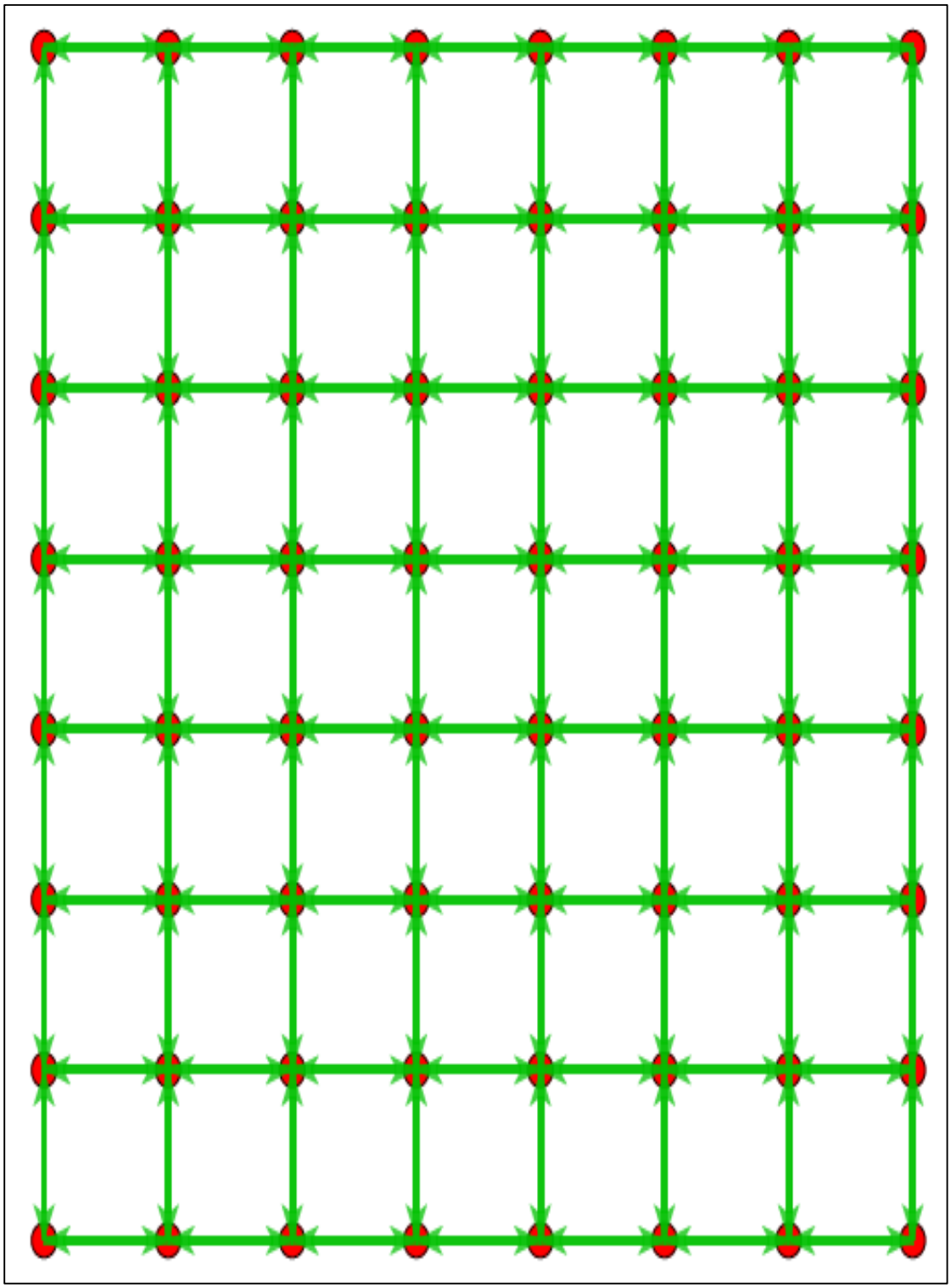
Appendix B



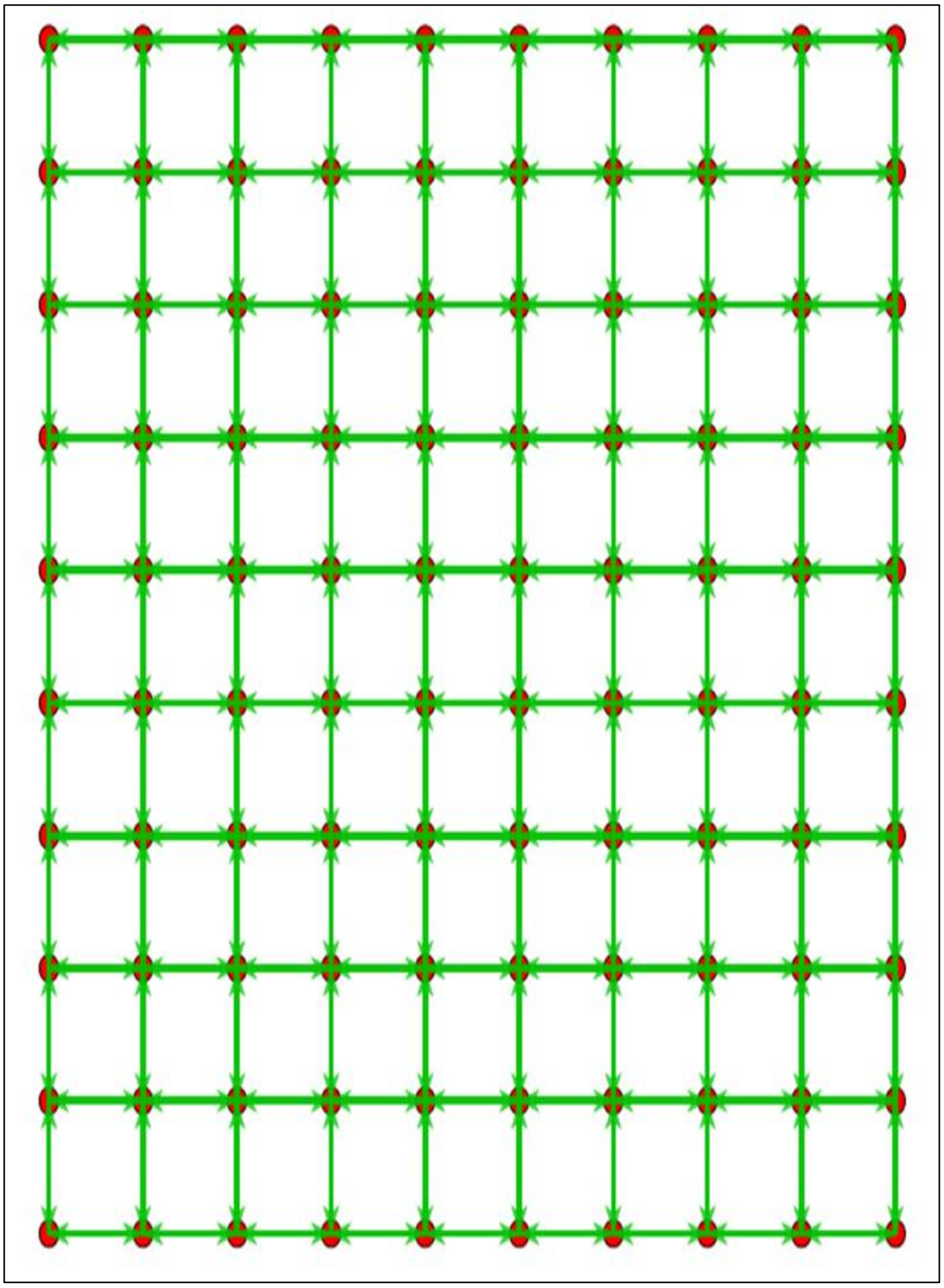
Appendix C



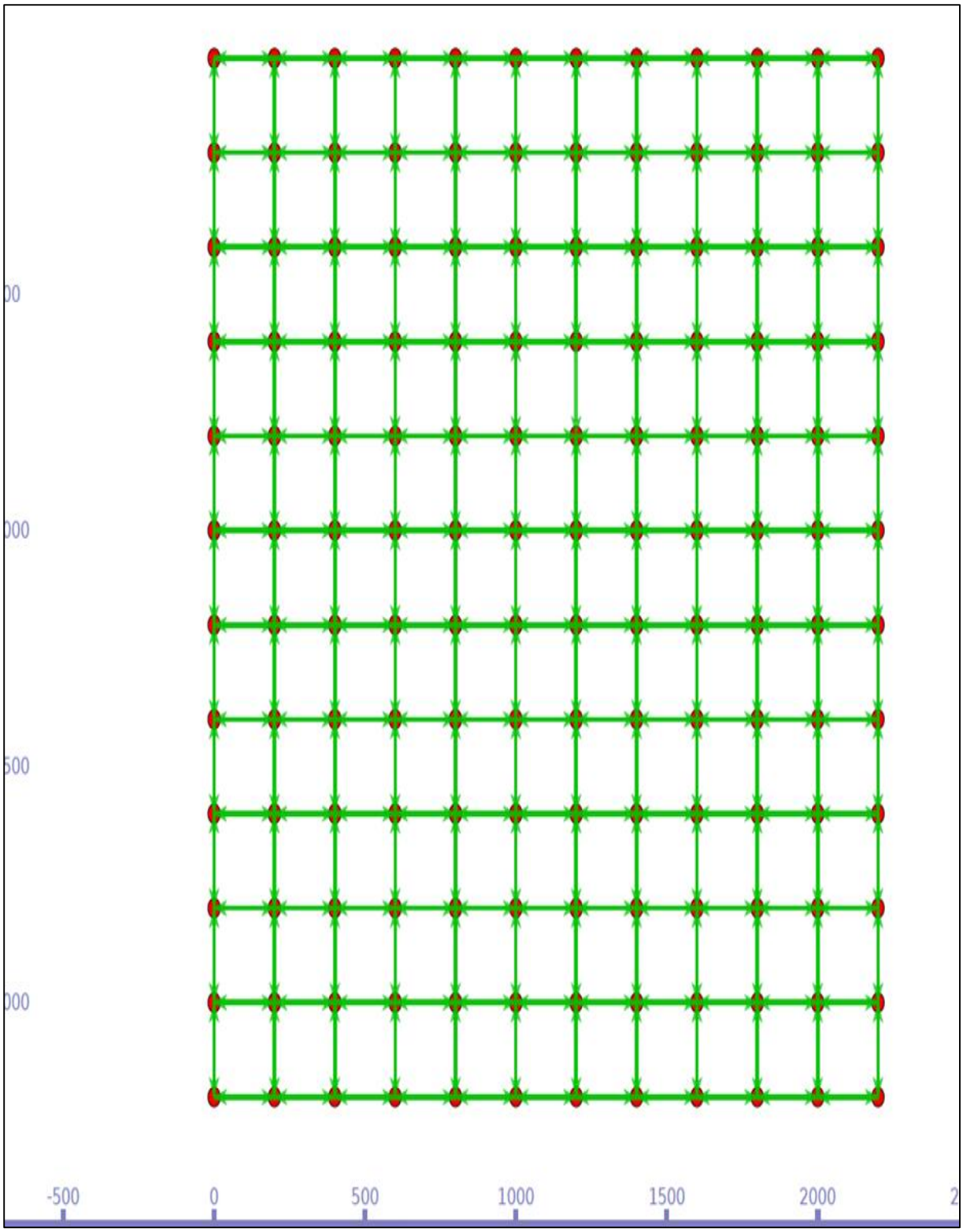
Appendix D



Appendix E



Appendix F



Appendix G

```
<Flow flowId="395" timeFirstTxPacket="+774031254099.0ns" timeFirstRxPacket="+774165780134.0ns"
timeLastTxPacket="+908571200752.0ns" timeLastRxPacket="+775055641868.0ns" delaySum="+790096908.0ns" jitterSum="+120484932.0ns"
lastDelay="+14041103.0ns"txBytes="5184256" rxBytes="37872"
txPackets="4928" rxPackets="36" lostPackets="4892" timesForwarded="291">
      <packetsDropped reasonCode="0" number="4758" />
      <bytesDropped reasonCode="0" bytes="5005416" />
      <delayHistogram nBins="135" >
        <bin index="14" start="0.014" width="0.001" count="33" />
        <bin index="83" start="0.083" width="0.001" count="1" />
        <bin index="108" start="0.108" width="0.001" count="1" />
        <bin index="134" start="0.134" width="0.001" count="1" />
      </delayHistogram>
      <jitterHistogram nBins="70" >
        <bin index="0" start="0" width="0.001" count="32" />
        <bin index="25" start="0.025" width="0.001" count="2" />
        <bin index="69" start="0.069" width="0.001" count="1" />
      </jitterHistogram>
      <packetSizeHistogram nBins="53" >
        <bin index="52" start="1040" width="20" count="36" />
      </packetSizeHistogram>
      <flowInterruptionsHistogram nBins="0" >
      </flowInterruptionsHistogram>
    </Flow>
```